



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA
EMPRESA CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.”

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS**

AUTORES:

ESCURRA COLQUIS, MAURIHÑO ENRIQUE (0000-0002-4766-2226)

ZAMORA NOREÑA, LUIS FERNANDO (0000-0002-9025-0497)

ASESOR:

Mg. PETRLIK AZABACHE, IVÁN CARLO (0000-0002-1202-2143)

LÍNEA DE INVESTIGACIÓN:

SISTEMA DE INFORMACIÓN Y COMUNICACIONES

LIMA – PERÚ

2019

DEDICATORIA

Dedicatoria de Ecurra Colquis, Maurihño Enrique

Dedico esta investigación a mis hermanos,
porque siempre están conmigo.

Dedicatoria de Zamora Noreña, Luis Fernando

Dedico esta investigación a mis
padres por el apoyo.

AGRADECIMIENTO

Agradecimiento de Ecurra Colquis, Maurihño Enrique

Un agradecimiento especial a mi asesor

Iván Carlo Petrlik Azabache por

Asesorarnos.

Agradecimiento de Zamora Noreña, Luis Fernando

Un agradecimiento muy especial a mis padres

por apoyarme en todo momento.

PRESENTACIÓN

Señores miembros del Jurado:

Yo ESCURRA COLQIS MAURIHÑO ENRIQUE, identificado con DNI N° 73537707.

Yo ZAMORA NOREÑA LUIS FERNANDO, identificado con DNI N° 76473553.

En cumplimiento de las normas establecidas en el Reglamento de Grados y Títulos de la Universidad César Vallejo para obtener el Título en Ingeniería de Sistemas, ante ustedes presentamos nuestra tesis: “Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.”.

Esta tesis se divide siete capítulos:

Capítulo 1: Planteamiento del problema, formulación del problema, objetivos, hipótesis, justificaciones, antecedentes y marco teórico.

Capítulo 2: Marco metodológico, población, muestra, tipo de diseño, técnicas e instrumentos de recolección de datos y métodos de análisis.

Capítulo 3: Resultados.

Capítulo 4: Discusiones.

Capítulo 5: Conclusiones.

Capítulo 6: Recomendaciones.

Capítulo 7: Referencias bibliográficas.

Señores miembros del jurado espero que nuestra investigación sea valorada y obtenga su aprobación.

Los autores

ÍNDICE

DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
PRESENTACIÓN.....	vii
ÍNDICE.....	viii
ÍNDICE DE FIGURAS.....	x
INDICE DE TABLAS.....	xii
RESUMEN.....	xiii
ABSTRACT.....	xiv
I. INTRODUCCIÓN.....	15
1.1. Realidad problemática.....	16
1.2. Trabajos Previos.....	18
1.3. Teorías relacionadas al tema.....	20
1.4. Formulación del problema.....	26
1.5. Justificación del estudio.....	26
1.6. Hipótesis.....	28
1.7. Objetivo.....	28
II.MÉTODO.....	29
2.1. Diseño de investigación.....	30
2.2. Variables y Operacionalización.....	31
2.3. Población y muestra.....	33
Muestra:.....	
2.3. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.....	34
2.4. Métodos de análisis de datos.....	37
2.5. Aspectos éticos.....	
III.RESULTADOS.....	39
3.1. Análisis descriptivo.....	40
3.2. Análisis inferencial.....	42

3.3. Prueba de Hipótesis	45
IV.DISCUSIÓN	51
V.CONSLUSIÓN	53
VI.RECOMENDACIONES	55
VII.REFERENCIAS BIBLIOGRÁFICAS	57
ANEXOS.....	63
ANEXO 1: Matriz de consistencia	64
ANEXO 2: Ficha técnica. Instrumento de recolección de datos	65
ANEXO 3: Instrumento de investigación.....	66
ANEXO 4: Base de datos Experimental.....	70
ANEXO 5: Resultados de la Confiabilidad del instrumento	71
ANEXO 6: Validación del instrumento.....	77
ANEXO 7: Carta de aprobación de la empresa	86
ANEXO 8: Entrevista en la empresa Construcciones & Acabados Zamora S.A.C. ...	88
ANEXO 9: Desarrollo de la metodología para la Aplicación móvil	90
ANEXO 10: Tabla T-Student	464

ÍNDICE DE FIGURAS

FIGURA 1. ÍNDICE DE AUSENTISMO.....	17
FIGURA 2. ÍNDICE DE ROTACIÓN.....	17
<i>FIGURA 3. FÓRMULA DEL INDICADOR ÍNDICE DE AUSENTISMO</i>	21
<i>FIGURA 4. FÓRMULA DEL INDICADOR ÍNDICE DE ROTACIÓN</i>	21
FIGURA 5. FLUJO DE SCRUM PARA UN SPRINT.....	24
<i>FIGURA 6. FICHA TÍPICA DE HISTORIA DE USUARIO 1</i>	25
FIGURA 7. DISEÑO DE LA INVESTIGACIÓN.....	30
FIGURA 8. COEFICIENTE DE CONFIABILIDAD.....	35
<i>FIGURA 9. CONFIABILIDAD PARA EL ÍNDICE DE AUSENTISMO</i>	36
FIGURA 10. CONFIABILIDAD PARA EL ÍNDICE DE ROTACIÓN.....	36
FIGURA 11. DISTRIBUCIÓN Z.....	38
FIGURA 12. DISTRIBUCIÓN T-STUDENT	38
FIGURA 13. ÍNDICE DE AUSENTISMO ANTES Y DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL	40
FIGURA 14. ÍNDICE DE ROTACIÓN ANTES Y DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	41
FIGURA 15. PRUEBA DE NORMALIDAD DEL ÍNDICE DE AUSENTISMO ANTES DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	42
FIGURA 16. PRUEBA DE NORMALIDAD DEL ÍNDICE DE AUSENTISMO DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	43
FIGURA 17. PRUEBA DE NORMALIDAD DEL ÍNDICE DE ROTACIÓN ANTES DE IMPLEMENTAR LA APLICACIÓN MÓVIL	44
FIGURA 18. PRUEBA DE NORMALIDAD DEL ÍNDICE DE ROTACIÓN DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	44
FIGURA 19. ÍNDICE DE AUSENTISMO ANTES DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	45
FIGURA 20. ÍNDICE DE AUSENTISMO DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	46
FIGURA 21. ÍNDICE DE AUSENTISMO – COMPARATIVA GENERAL.....	46
FIGURA 22. PRUEBA T-STUDENT DEL ÍNDICE DE AUSENTISMO	47
FIGURA 23. ÍNDICE DE ROTACIÓN ANTES DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	48

FIGURA 24. ÍNDICE DE ROTACIÓN DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL.....	48
FIGURA 25. ÍNDICE DE ROTACIÓN – COMPARATIVA GENERAL.....	49
FIGURA 26. PRUEBA T-STUDENT DEL ÍNDICE DE ROTACIÓN	50

ÍNDICE DE TABLAS

TABLA 1. SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA LA APLICACIÓN MÓVIL	23
TABLA 2. FASES Y PROCESOS DE SCRUM	24
TABLA 3. OPERACIONALIZACIÓN DE VARIABLES	31
TABLA 4. INDICADORES DEL CONTROL DE PERSONAL	32
TABLA 5. POBLACIÓN	33
TABLA 6. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	34
TABLA 7. VALIDEZ PARA EL INDICADOR ÍNDICE DE AUSENTISMO	35
TABLA 8. VALIDEZ PARA EL INDICADOR ÍNDICE DE ROTACIÓN	35
TABLA 9. ESTADÍSTICO DESCRIPTIVO ANTES Y DESPUÉS DE LA APLICACIÓN MÓVIL – ÍNDICE DE AUSENTISMO	40
TABLA 10. ESTADÍSTICO DESCRIPTIVO ANTES Y DESPUÉS DE LA APLICACIÓN MÓVIL – ÍNDICE DE ROTACIÓN	41
TABLA 11. PRUEBAS DE NORMALIDAD – ÍNDICE DE AUSENTISMO	42
TABLA 12. PRUEBAS DE NORMALIDAD – ÍNDICE DE ROTACIÓN	43
TABLA 13. PRUEBA DE T-STUDENT PARA EL ÍNDICE DE AUSENTISMO EN EL CONTROL DE PERSONAL ANTES Y DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL	47
TABLA 14. PRUEBA DE T-STUDENT PARA EL ÍNDICE DE ROTACIÓN EN EL CONTROL DE PERSONAL ANTES Y DESPUÉS DE IMPLEMENTAR LA APLICACIÓN MÓVIL	49

RESUMEN

Esta tesis abarca el desarrollo de una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., ya el escenario previo al desarrollo de la app móvil tenía varios inconvenientes respecto al índice de ausentismo e índice de rotación.

El objetivo fue determinar la influencia de una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C. Se utiliza el marco de trabajo Scrum para el desarrollo de la aplicación móvil. El tipo de investigación es aplicada, el diseño de la investigación es pre experimental y el enfoque es cuantitativo.

La población para el indicador índice de ausentismo se decretó a 400 obreros generados por las ausencias de estos, estratificados por días durante un mes. Por consiguiente, quedó definido en 25 fichas de registro la población para este indicador. La población para el indicador índice de rotación se decretó a 400 obreros generados por la desvinculación de estos, estratificados por días durante un mes. Por consiguiente, quedó definido en 25 fichas de registro la población para este indicador. El tamaño de la muestra estuvo conformado por 196 obreros, estratificados por días durante un mes en 25 fichas de registro. El muestreo es el probabilístico aleatorio simple. La técnica de recolección de datos fue el fichaje y el instrumento la ficha de registros, los cuales fueron validados por expertos.

Finalmente, la app móvil permitió disminuir el índice de ausentismo de 64% a 39%, del mismo modo, se disminuyó el índice de rotación de 56% a 38%. Los resultados indicados previamente, admitieron llegar a la conclusión que la aplicación móvil mejoró el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

Palabras claves: APLICACIÓN MÓVIL, CONTROL DE PERSONAL, SCRUM

ABSTRACT

This thesis covers the development of a mobile application for the control of personnel in the company Construcciones & Acabados Zamora S.A.C., and the scenario prior to the development of the mobile app had several drawbacks regarding the absenteeism rate and turnover rate.

The objective was to determine the influence of a mobile application for personnel control in the company Construcciones & Acabados Zamora S.A.C. The Scrum framework is used for the development of the mobile application. The type of research is applied, the research design is pre-experimental and the approach is quantitative.

The population for the absenteeism index indicator was decreed to 400 workers generated by their absences, stratified by days for a month. Consequently, the population for this indicator was defined in 25 record sheets. The population for the index of rotation index was decreed to 400 workers generated by the separation of these, stratified by days for a month. Consequently, the population for this indicator was defined in 25 record sheets. The sample size consisted of 196 workers, stratified by days for a month on 25 registration forms. Sampling is the simple random probabilistic. The data collection technique was the signing and the instrument the record sheet, which were validated by experts.

Finally, the mobile app allowed to reduce the absenteeism rate from 64% to 39%, in the same way, the turnover rate was reduced from 56% to 38%. The results indicated previously, admitted to conclude that the mobile application improved personnel control in the company Construcciones & Acabados Zamora S.A.C.

Keywords: MOBILE APPLICATION, PERSONAL CONTROL, SCRUM

I. INTRODUCCIÓN

1.1. Realidad problemática

El problema principal, según la entrevista que se realizó al Gerente General Javier Antonio Zamora (Ver Anexo 8), nos menciona que: cuando se inician los proyectos de construcción civil, estos se ejecutan en lugares diferentes (zonas de campo), y genera una gran desventaja a la empresa, por el cual es difícil llevar un control de lo que hace y no hace el personal de la empresa a larga distancia como: supervisión de entradas y salidas, faltas, remuneraciones, asignación de obra, vacaciones, así mismo, los trabajadores de campo no pueden tener información relacionada a sus labores ya que no existe un acercamiento para los obreros con el área de recursos humanos, actualmente cuenta con 400 obreros de campo. Los registros de entrada y salida del personal se hace manualmente cada día y son de fácil modificación por parte de los mismos trabajadores de obra, retraso o tráfico en la hora de entrada para marcar las asistencias, los trabajadores no entran a la hora correspondiente, salen a la hora que quieren, algunos ni asisten a trabajar, tampoco no se puede saber con certeza si un obrero se merece un aumento cuando lo solicite, porque la empresa no tiene información relevante para ver la productividad que tiene cada trabajador y esto produce un desequilibrio, muchas veces los mismos empleados se encubren entre ellos mismos para así no ser perjudicados y la empresa no se da cuenta y tome medidas en el asunto. De la misma manera la empresa dio cuenta de la gran cantidad de dinero que pierde para controlar las asistencias de los trabajadores utilizando el control de asistencia biométrico o cualquier otro dispositivo para el control de personal, este dispositivo biométrico ha sido de gran ayuda para controlar al personal en oficina pero no ha sido eficiente tanto en costo y tiempo para los trabajadores de campo, ya que por cada obra se debe contratar a un especialista para las instalaciones de dicho dispositivo, el costo sube más aún cuando las obras son en provincia, se debe proporcionar viáticos, viajes para que el especialista pueda trasladarse, pero lo más irrentable es que estas instalaciones son temporales, y se debe repetir el mismo proceso por cada punto de obra. Por otro lado, no se puede dar soporte al sistema cuando deja de funcionar, porque no existe un personal de sistema en los centros de obra. Además, los obreros no tienen la facilidad de comunicarse con el área de recursos humanos, cuando necesitan hacer consulta sobre sus faltas, tardanzas, vacaciones, justificaciones o algún inconveniente con sus compensaciones, deben de ir a la sede central y dejar sus actividades un día de semana, por la única razón es que la jornada es de lunes a viernes, esto genera pérdidas para la empresa porque en algunas ocasiones el trabajador se toma más de medio día, o hasta un día

(**indicador índice de ausentismo**), por el cual no permite a la organización luchar en su rubro y genera molestias o disconformidad para los trabajadores si es que no se le da una sustentación de lo sucedido, esto ha provocado cierta desvinculación en ellos (**indicador índice de rotación**), y para la empresa genera inestabilidad, si no se sabe la cantidad de obreros desvinculados no se podrá tomar las medidas correspondiente para que así la empresa pueda tomar estrategias de negocio.

© Fuente: Elaboración propia

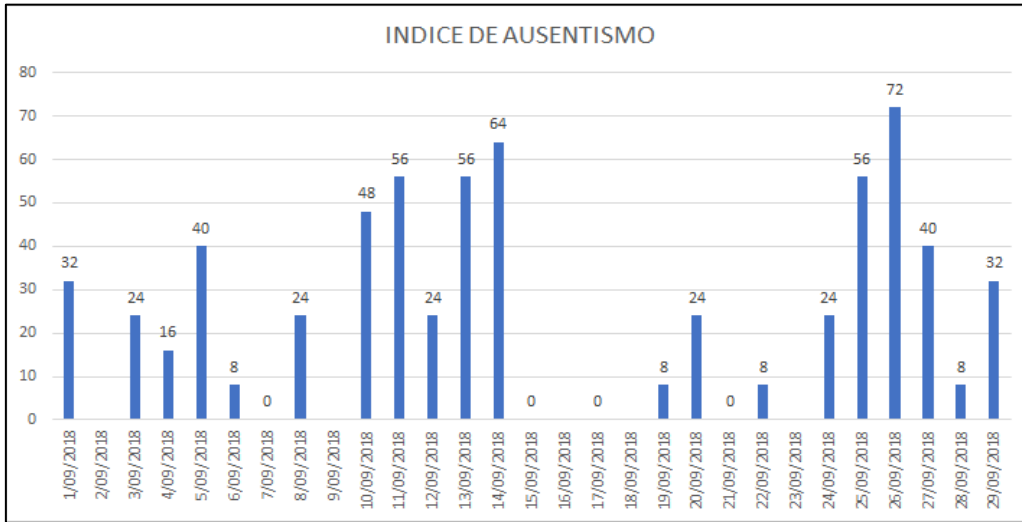


Figura 1. Índice de ausentismo

© Fuente: Elaboración propia

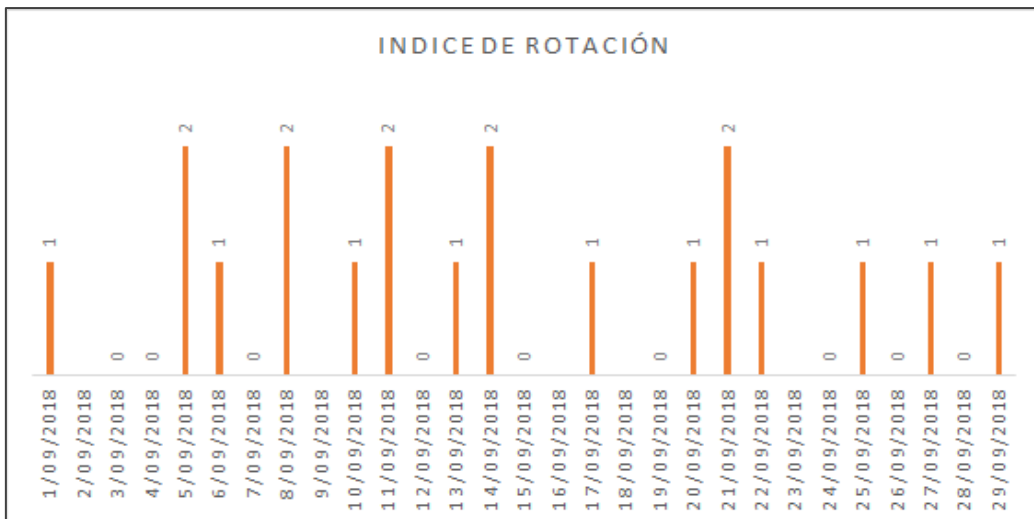


Figura 2. Índice de Rotación

1.2. Trabajos Previos

Antecedentes Internacionales:

En el año 2018, Carlos Plúas, en su tesis titulada “Desarrollo de aplicación móvil usando geolocalización para la gestión de control de asistencia del personal técnico en obra, de la empresa Hardsofnet CIA.LTDA”, desarrollada en la Universidad de Guayaquil (Guayaquil - Ecuador), dio solución al problema de exceso de presupuesto en las instalaciones de aparatos biométricos para el control de asistencia en trabajadores de campo. El objetivo fue la creación de una aplicación móvil usando geolocalización, para la gestión de control de asistencia del personal técnico en obra. La justificación fue que se pudo agilizar los procesos cotidianos de oficina del personal Administrativo, ya que en un mundo moderno con avances tecnológicos las Instituciones utilizan cada vez más aplicaciones móviles que se ajustan a sus necesidades. Se realizó un tipo de investigación mixta, de carácter descriptivo y exploratorio, es descriptiva porque su propósito está enmarcado en analizar e interpretar el proceso de registro de asistencia del personal técnico residente en obra, de tal forma que mostrará cual es la problemática que se está generando y exploratoria porque averiguara y se reconocerá cuáles son las necesidades para la creación del prototipo en estudio, debido a que la propuesta del proyecto a efectuarse jamás ha sido estudiada con el propósito de mejorar los procesos de control de asistencia a personal técnico residente en obra y facilitar el método de elaboración de planillas de pagos y reportes de las incidencias del personal técnico residente en la obra las que estén asignados. Las técnicas utilizadas fueron de observación, encuestas y entrevistas. La metodología con la que se desarrolló la aplicación móvil fue XP (Programación Extrema). La población estuvo dividida en la administración y personal técnico de la empresa Hardsofnet Cía. Se puede deducir como indicadores de esta investigación al índice de ausentismo y al índice de rotación, el primero son los empleados de obras que faltan en sus labores cotidianas y el segundo son muchas veces por las desvinculaciones y/o despidos de los mismos empleados por alguna razón. Se elaboraron siete preguntas en el cual fueron contestadas por los empleados de la empresa, la primera pregunta fue para qué sirve el control de personal o interno en el cual los empleados respondieron que es para vigilar a los trabajadores en el cual hace un 24.14%, para controlar la asistencia un 17.24%, y para apoyar en la consecución de los objetivos de la institución un 58.62%, la segunda pregunta fue que si los métodos actuales para el control de personal son los más óptimos en el cual el 75.86% fue positivo y un 24.14% fue negativo, la tercera pregunta fue que si es importante implementar el sistema en el cual un 93.10% fue positivo

y un 6.9% fue negativo, la cuarta pregunta fue que al utilizar el sistema permite monitorear al empleado en el cual un 93.10% fue positivo y un 6.9% fue negativo, la quinta pregunta fue que cual es el sistema operativo que utilizan más y por descarte es Android con un 65.51%, la sexta pregunta fue que como considera a Android en el cual respondieron con 58.62% un excelente sistema, con 41.38% un buen sistema y con 0% fueron regular y mala, y para la séptima pregunta que fue que se debe de capacitar a los usuarios en el cual todos respondieron que si haciendo un 100%. Los resultados mostraron optimizar el proceso de control de asistencia del personal técnico en obra, permitiendo obtener un control de forma organizada facilitando la información y obteniéndose de manera rápida y oportuna. Las conclusiones indican que se logró desarrollar una app para dispositivos con sistema operativo Android, la cual reconoce y valida rostros de personas en tiempo real. Además, el usuario logra obtener información en base a la misma, como la ubicación donde realizó los fichajes de marcación de asistencia mediante imagen georreferenciada, así mismo también permitió generar un reporte de marcaciones generales de todos los usuarios que admitan la oportuna y confiable disponibilidad de la información cuando sea requerida; facilitando de tal forma el proceso de generación de planillas de pago y la ubicación de los empleados por medio del histórico que contiene el sistema. De la presente tesis se puede tomar como referencia a los supuestos indicadores índice de ausentismo e índice de rotación. Tomamos estos supuestos indicadores para llevar a cabo un alcance más específico para el desarrollo de la aplicación móvil en nuestra investigación.

Antecedentes Nacionales:

En el año 2017, Alipio Paredes y Wagner Paredes, en su tesis titulada “Sistema de localización basado en dispositivos móviles para el control y monitoreo del personal en el campamento de la empresa minera VanessaSac en el primer trimestre 2016”, desarrollada en la Universidad Nacional del Altiplano (Puno - Perú). El objetivo fue determinar la influencia del sistema de localización en el control y monitoreo del personal en el campamento de la empresa minera VANESSASAC en el primer trimestre del 2016. Trató el problema de la falta de sincronización de actividades que se realizan durante la jornada laboral diaria y el tiempo dedicado a las labores de documentación de obtención de orden de trabajo. El tipo de estudio fue aplicada porque su principal objetivo es resolver problemas, con un diseño experimental de tipo pre - experimental. La población fue de 83 personas que laboran en la empresa minera y el muestreo fue no probabilístico, intencional. Las técnicas

utilizadas fue la recopilación de datos y la observación, el instrumento fue una ficha de observación que fue validado por medio de juicio de expertos. El lenguaje de programación se utilizó Java con el entorno Android Studio y Php para los servicios, como motor de base de datos se utilizó MySQL. Para los resultados se utilizó una encuesta de satisfacción antes y después del proceso de investigación a 83 trabajadores de la empresa que laboraron entre los meses abril, mayo y junio del 2017, con el consolidado de la encuesta se llegó a la conclusión de que el 81% de los encuestados se encuentra satisfecho con el uso del sistema para el control y monitoreo y así mismo se comprobó la hipótesis de la influencia satisfactoria de la aplicación del sistema de localización en el control y monitoreo del personal. De la presente tesis se tomó como referencia el diseño de investigación pre - experimental ya que habrá resultados antes de implementar la app y cuando la app está ya en producción en la empresa.

1.3. Teorías relacionadas al tema

A. Control de personal

Según Chiavenato (2009), manifiesta que el control de personal es seguir, acompañar, orientar y mantener la conducta del personal en una empresa. El control de personal debe cumplir todas las actividades que realmente deben y necesitan controlarse y es muy importante para que así se pueda cumplir todos los objetivos y metas que tiene una determinada organización (p.504).

Fases del control de personal:

Según Chiavenato (2009), indica que las fases del control de personal son las siguientes:

- 1. Registro de personal:** Manifiesta que el registro de personal es la fase más importante de control de personal, porque se obtienen los datos de las marcaciones personales de cada uno de los empleados para así tener un control de ellos desde que empiezan a laborar hasta que se van de la organización.
- 2. Registro de secciones y puestos:** Esta fase consiste en recopilar los datos de los empleados de cada área y cada puesto, roles y funciones dentro de la organización.
- 3. Registro de beneficios:** Esta fase consiste en todos los beneficios que tienen los empleados de una organización como, por ejemplo, la gratificación, vacaciones y justificaciones.

4. Registro de remuneraciones: Esta fase hace mención a los datos sobre los salarios e incentivos salariales que van de acuerdo a la frecuencia o tarjetas de registro de asistencia del personal de una empresa (p.511).

Indicador: Índice de ausentismo

© Fuente: Chiavenato
(2009)

$$IA = \frac{TP/HP}{TP/HL}$$

Figura 3. Fórmula del indicador índice de ausentismo

Indicador: Índice de rotación

© Fuente: Chiavenato
(2009)

$$IR = \frac{NES}{NEA}$$

Figura 4. Fórmula del indicador índice de rotación

B. Aplicación Móvil

Según Gardner y Davis (2014), definen que es un software, generalmente proyectado para que funcione en Smartphones, que permite que el usuario interactúe con las funcionalidades que tiene la app (p.20).

Aplicación Híbrida:

Según Pimienta (2014), define que se desarrollan para diversos sistemas operativos lo que se conoce como multiplataforma, para ello se usan marcos de trabajos, por ejemplo, Ionic (párr.9).

En la presente tesis se utilizará una aplicación híbrida, ya que el desarrollo será programado con el framework Ionic para que se pueda usar y aprovechar al máximo toda la tecnología necesaria de este marco de trabajo ya que en la actualidad es la más usada por los desarrolladores en programación híbrida y en la empresa Construcciones & Acabados Zamora S.A.C, los dispositivos en cada punto de obra pueden ser iOS o Android.

Metodología de desarrollo de software para la aplicación móvil

Para esta investigación se tomó tres metodologías de desarrollo de software para primero compararlas y luego seleccionar una mediante juicio de expertos y hablar de ella. Previamente se define cada una.

Metodología RUP:

Según Pabón (2018), manifiestan que es una metodología de tipo tradicional para desarrollar software, que este a su vez es el conjunto de todas las tareas que son muy necesarias para cambiar requerimientos de un usuario en un producto de software, además es una metodología que utiliza el lenguaje unificado de modelado, dicho lenguaje ayuda a construir, documentar el software y se utilizan varios diagramas, por ejemplo, casos de usos (párr.1).

SCRUM:

Según SCRUMstudy (2017), manifiesta que es un framework que se basa en la entrega constante de productos para satisfacer al cliente ya que es interactivo e incremental (p.41).

Programación Extrema (XP):

Según Izquierdo (2014), manifiesta que es una metodología ágil para proyectos determinados en una organización. [...] (párr.1).

Metodología de software para la aplicación móvil seleccionada

Tabla 1. Selección de la metodología de desarrollo de software para la aplicación móvil

Experto	Grado	Puntaje de la Metodología			Metodología seleccionada
		SCRUM	XP	RUP	
Ing. Gálvez Tapia, Orleans	Magister	40	24	34	SCRUM
Ing. Chumpe Agosto, Juan	Magister	37	24	32	SCRUM
Ing. Vargas Human, Jhonatan	Magister	40	28	26	SCRUM
TOTAL		117	76	92	SCRUM

Fuente: Elaboración propia

Aquí en la tabla uno se ve que el framework Scrum obtuvo unos 117 puntos. Por lo tanto, se utilizará este marco de trabajo para la app móvil a implementar.

Metodología seleccionada para el desarrollo de la aplicación móvil

SCRUM:

Según SCRUMstudy (2017), manifiesta que es un framework muy rápido, eficaz, iterativo, adaptable y muy eficaz para un proyecto determinado en empresas (p.2).

© Fuente: SCRUMstudy (2017)

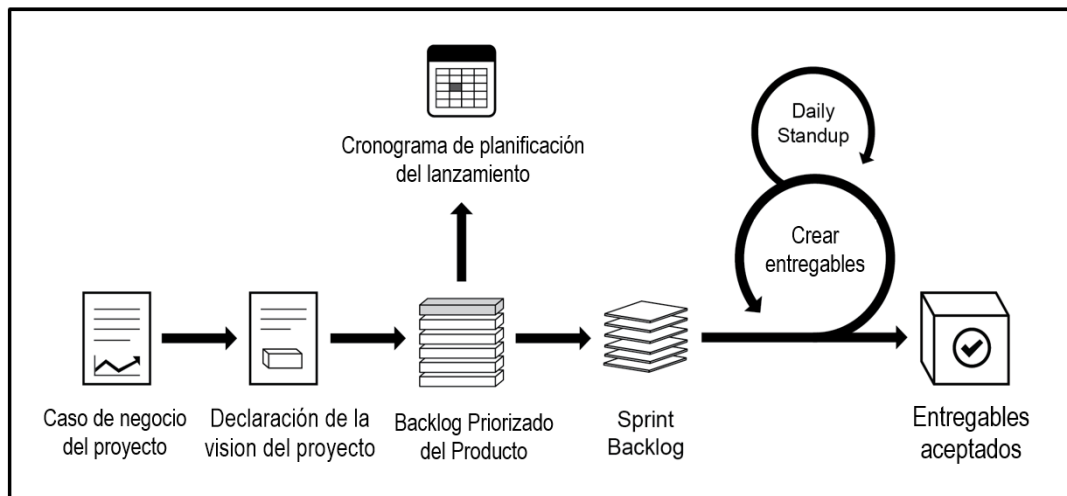


Figura 5. Flujo de Scrum para un sprint

Fases y Procesos de Scrum

Tabla 2. Fases y Procesos de Scrum

Fases	Procesos fundamentales de Scrum
Inicio	<ol style="list-style-type: none"> 1. Crear la visión del proyecto 2. Identificar al Scrum Master y Stakeholder(s) 3. Formar Equipos Scrum 4. Desarrollar épicas 5. Crear el Backlog Priorizado del Producto 6. Realizar la planificación de lanzamiento
Planificación y estimación	<ol style="list-style-type: none"> 7. Crear historias de usuario 8. Estimar historias de usuario 9. Comprometer historias de usuario 10. Identificar tareas 11. Estimas tareas 12. Crear el Sprint Backlog

Implementación	13. Crear entregables 14. Realizar Daily Standup 15. Refinar el Backlog Priorizado del Producto
Revisión y retrospectiva	16. Demostrar y validar el sprint 17. Retrospectiva del sprint
Lanzamiento	18. Enviar entregables 19. Retrospectiva del proyecto

Fuente: SCRUMstudy (2017)

Historias de Usuario

Según Menzinsky, López y Palacio (2018), manifiestan que se aplican en la mayoría de los métodos ágiles, siendo así una herramienta muy importante también en Scrum (p.10).

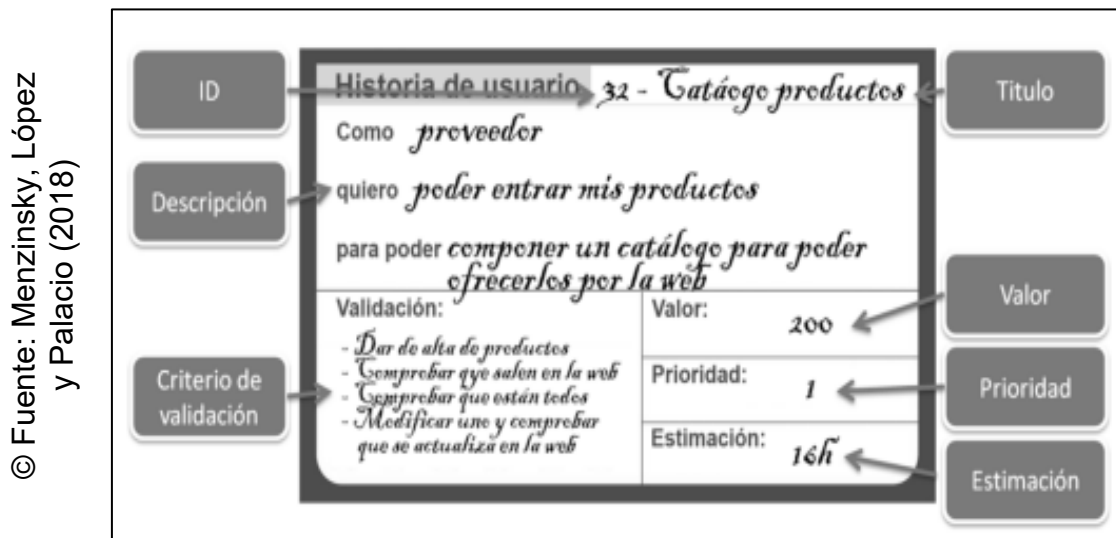


Figura 6. Ficha típica de Historia de Usuario 1

1.4. Formulación del problema

Problema General:

PG: ¿De qué manera influye una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.?

Problemas Específicos:

PE1: ¿De qué manera una aplicación móvil influye en el índice de ausentismo en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.?

PE2: ¿De qué manera una aplicación móvil influye en el índice de rotación en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.?

1.5. Justificación del estudio

Justificación Económica:

Se justifica económicamente, porque le permitirá a la empresa Construcciones & Acabados Zamora S.A.C., el ahorro de recursos económicos, ya que no utiliza un sistema biométrico o cualquier otro sistema para el control de personal por que las obras que realiza dicha empresa son de un lugar a otro, entonces es muy complicado instalar un sistema para cada punto de obra, y contratar a alguien que lo instale, lo configure y le de soporte, ya que todo esto implica tiempo y costo en instalar el sistema cada vez que el personal se encuentren de una obra en otra, pero ya con la app implementada, no se tendrá que hacer ninguna instalación, y así sincronizará las asistencias del personal, fácil y rápido con tan solo usar la aplicación móvil. Por cada punto de obra es un aproximado de S/.6,500.00 para tener un sistema instalado y con soporte. Ahora imaginemos que a la empresa le salga 10 construcciones, sería muy elevado todo este proceso de instalación, por ende, la empresa pierde económicamente más de lo que va ganar. Entonces con la app móvil implementada la empresa se ahorra S/.60,500.00 y como está en auge de crecimiento, no le vendría bien gastar en estos dispositivos ni mucho menos instalarlos y darle soporte cada vez que se requiera. Por eso se justifica económicamente que no se utilizará un sistema biométrico o cualquier otro tipo de sistema para el control de personal sino una aplicación móvil. Al implementar esta aplicación móvil no hay costos muy grandes porque se realizará en Open Source, lo que significa que es un ahorro de dinero en licencias para la programación de la

app, a la vez toda la parte tecnológica para el desarrollo de la aplicación será bajo este tipo de licencia.

Justificación Tecnológica:

Esta investigación propone la gran importancia tecnológica que soporta la app móvil implementada para el control de personal, logrando de esta manera automatizar este proceso y así aumentar la eficacia, el tiempo promedio de registro, y esperar los mejores resultados de esta investigación. En lo actual la tecnología ha permitido desarrollar este tipo de aplicación móvil (todo gratis) porque va hacer programado en Open Source con el framework Ionic ya que será híbrido, las tecnologías a usar son HTML, CSS, lenguaje de programación JavaScript, se desarrollará con la arquitectura MVC para un mejor orden, la base de datos se usará MySQL, como servidor Apache Xampp y el lenguaje PHP para el desarrollo del servicio web de la aplicación móvil en el cual es REST en formato JSON con el Framework Laravel, ya que la aplicación móvil consumirá estos servicios REST para el buen funcionamiento. Además, se usará el repositorio Bitbucket y la herramienta GitKraken para todo el control del código fuente de la app. Al tener la aplicación móvil ya desarrollada e implementada, el usuario contará con un smartphone para que pueda gozar de las funcionalidades de la aplicación que gracias a la tecnología actual se pudo implementar en la empresa Construcciones & Acabados Zamora S.A.C. y que será competitiva con otras empresas de su mismo sector. Además, se generará el APK por cada dispositivo de la obra de la empresa, y esto ayudará a que los usuarios interactúen con el funcionamiento adecuado de la app móvil.

Justificación Operacional:

Esta investigación es operativamente justificable ya que el personal de la compañía cuenta con conocimientos de interactuar con diversas aplicaciones porque en la actualidad el 99% de los empleados entre 20 - 35 años de edad que están en obras tienen un smartphone en su poder y no solo ellos, sino también el administrador de cada obra y el área de recursos humanos que velará por el reporte completo del control de personal. A todos estos empleados les resultará fácil de interactuar con la aplicación móvil de la empresa ya que tiene una interfaz agradable, de fácil manejo, comprensión, el uso de las funcionalidades es entendible, y además habrá una capacitación para el uso adecuado de la app, con lo cual los obreros

pueden cumplir con todo su trabajo, ocasionando un buen clima laboral entre ellos y la obra que está en marcha acabarla satisfactoriamente.

Justificación Institucional:

La presente investigación es justificable institucionalmente, porque al tener a la app móvil implementada ayuda de manera eficiente y óptima no solo el proceso de control de personal sino también a sus demás procesos, así misma mejora en gran escala la imagen y prestigio de la empresa Construcciones & Acabados Zamora S.A.C. Además, la aplicación fortalece en gran escala a cumplir todos los objetivos de la organización, en el cual es “Dar los mejores servicios de construcción civil de calidad a las necesidades de los clientes”.

1.6. Hipótesis

Hipótesis General:

HG: La aplicación móvil mejora el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

Hipótesis Específicas:

H1: La aplicación móvil disminuye el índice de ausentismo en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

H2: La aplicación móvil disminuye el índice de rotación en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

1.7. Objetivo

Objetivo General:

OG: Determinar la influencia de una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

Objetivos Específicos:

O1: Determinar la influencia de una aplicación móvil en el índice de ausentismo en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.

O2: Determinar la influencia de una aplicación móvil en el índice de rotación en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.

II.MÉTODO

2.1. Diseño de investigación

Método de Investigación

Aca es hipotético-deductivo.

Tipo de estudio:

Aca es aplicada y experimental.

Diseño de estudio:

Es pre – experimental.

© Fuente: Hernández,
Fernández y Baptista
(2014)

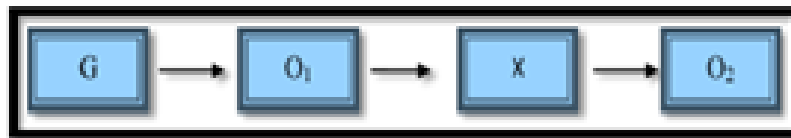


Figura 7. Diseño de la investigación

Nivel de estudio:

Aca es explicativa.

2.2. Variables y Operacionalización

Tabla 3. Operacionalización de variables

Tipo	Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicador	Escala de medición
Variable independiente	Aplicación móvil	Según Gardner y Davis (2014), definen que es un software, generalmente proyectado para que funcione en Smartphones, que permite que el usuario interactúe con las funcionalidades que tiene la app (p.20).	Sistema de información en tecnología móvil que ayudará a optimizar y mejorar el control de personal en la empresa Construcciones & Acabados Zamora S.A.C. Con la app se va a tener un control completo de todo el personal de la organización.			
Variable dependiente	Control de personal	Chiavenato (2009), manifiesta que el control de personal es seguir, acompañar, orientar y mantener la conducta del personal en una empresa. El control de personal debe cumplir todas las actividades que realmente deben y necesitan controlarse y es muy importante para que así se pueda cumplir todos los objetivos y metas que tiene una determinada organización (p.504).	Serie que se lleva a cabo en la organización Construcciones & Acabados Zamora S.A.C. y que tiene como finalidad conocer todas las incidencias sobre la asistencia del personal que se dan.	Registro de remuneraciones	Índice de ausentismo	Razón
				Registro de personal	Índice de rotación	Razón

Fuente: Elaboración propia

Tabla 4. Indicadores del control de personal

Indicador	Descripción	Técnica	Instrumento	Unidad de medida	Fórmula
Índice de ausentismo	Chiavenato (2009), menciona que el índice de ausentismo son todas las faltas y/o retrasos que se pierde cuando los empleados no se presentan a laborar (p.89-90).	Fichaje	Ficha de Registro	Porcentaje	$IA = \frac{TP/HP}{TP/HL}$ <p>Donde: TP = Total de personas HP = Horas perdidas HL = Horas laboradas</p>
Índice de rotación	Chiavenato (2009), manifiesta que el índice de rotación es el resultado de los empleados que son desvinculados de la empresa ya sea por renuncia o despidos (p.91).	Fichaje	Ficha de Registro	Porcentaje	$IR = \frac{NES}{NEA}$ <p>Donde: NES = Número de empleados separados NEA = Número de empleados asistidos</p>

Fuente: Elaboración propia

2.3. Población y muestra

Población:

Tabla 5. *Población*

Población	Indicador
25 fichas de registro (por día)	Índice de ausentismo
	Índice de rotación

Fuente: Elaboración propia

Aquí en la tabla cinco se deduce que:

Índice de ausentismo

Se decretó a 400 obreros que fueron ocasionados por el ausentismo de los obreros, estratificados por días en el mes de octubre de 2018. Por consiguiente, quedó definido en 25 fichas de registro la población.

Índice de rotación

Se decretó a 400 obreros que fueron ocasionados por la rotación de obreros, estratificados por días en el mes de octubre de 2018. Por consiguiente, quedó definido en 25 fichas de registro la población.

Muestra:

$$m = \frac{\frac{z^2 \cdot p(1-p)}{e^2}}{1 + \left(\frac{z^2 \cdot p(1-p)}{e^2 N}\right)}$$

Aplicando la fórmula:

$$m = \frac{\frac{1.96^2 \times 0.5(1 - 0.5)}{0.05^2}}{1 + \left(\frac{1.96^2 \times 0.5(1 - 0.5)}{0.05^2 \times 400} \right)}$$

$$m = \frac{\frac{0.9604}{0.0025}}{1 + \left(\frac{0.9604}{1} \right)}$$

$$m = \frac{384.16}{1.9604}$$

$$m = 196 \text{ obreros}$$

Muestreo:

Aquí se usa el aleatorio simple.

2.3. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Técnica Fichaje e Instrumento Ficha de Registro

Tabla 6. *Técnicas e instrumentos de recolección de datos*

Variable	Dimensión	Indicador	Técnicas	Instrumentos
Variable Dependiente	Registro de remuneraciones	Índice de ausentismo	Fichaje	Ficha de registro (Anexo 5)
Control de personal	Registro de personal	Índice de rotación	Fichaje	Ficha de registro (Anexo 5)

Fuente: Elaboración propia

Validez

Tabla 7. Validez para el indicador índice de ausentismo

N°	Experto	Grado Académico	Puntaje
1	Ing. Gálvez Tapia, Orleans	Magister	80%
2	Ing. Chumpe Agosto, Juan	Magister	71%
3	Ing. Vargas Huamán, Jhonatan	Magister	85%
Total			79%

Fuente: Elaboración propia

Tabla 8. Validez para el indicador índice de rotación

N°	Experto	Grado Académico	Puntaje
1	Ing. Gálvez Tapia, Orleans	Magister	80%
2	Ing. Chumpe Agosto, Juan	Magister	71%
3	Ing. Vargas Huamán, Jhonatan	Magister	85%
Total			79%

Fuente: Elaboración propia

Método

© Fuente: Hernández,
Fernández y Baptista (2014)

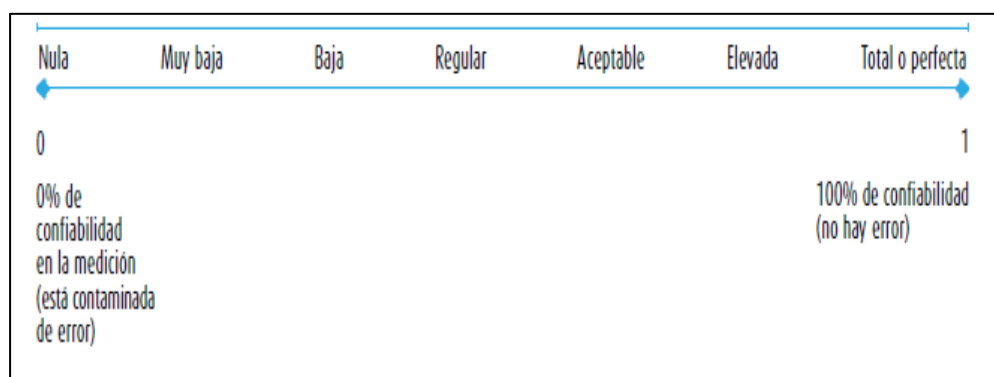


Figura 8. Coeficiente de confiabilidad

© Fuente: Elaboración propia

		IA_RETEST	IA_TEST
IA_RETEST	Correlación de Pearson	1	,732**
	Sig. (bilateral)		,000
	N	25	25
IA_TEST	Correlación de Pearson	,732**	1
	Sig. (bilateral)	,000	
	N	25	25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Figura 9. Confiabilidad para el índice de ausentismo

© Fuente: Elaboración propia

		IR_RETEST	IR_TEST
IR_RETEST	Correlación de Pearson	1	,713**
	Sig. (bilateral)		,000
	N	25	25
IR_TEST	Correlación de Pearson	,713**	1
	Sig. (bilateral)	,000	
	N	25	25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Figura 10. Confiabilidad para el índice de rotación

2.4. Métodos de análisis de datos

Aquí el método de análisis de datos es cuantitativo.

Pruebas de Normalidad

Hipótesis de Investigación 1

a. Hipótesis Especifico 1 (HE1)

La aplicación móvil disminuye el índice de ausentismo en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

Hipótesis de investigación 2

a. Hipótesis Especifico 2 (HE2)

La aplicación móvil disminuye el índice de rotación en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.

Nivel de significancia

Estadística de prueba

$$t = \frac{\overline{x_1 - x_2}}{\sqrt{\frac{S_1^2}{N_1} + \frac{S_2^2}{N_2}}}$$

Cálculo de la media

$$n = \sum_{i=1}^n X_i$$

Cálculo de la varianza

$$x = \frac{\sum_{i=1}^n (X_i - \bar{x})^2}{n - 1}$$

Distribución normal Z

© Fuente: Hernández,
Fernández y Baptista (2014)

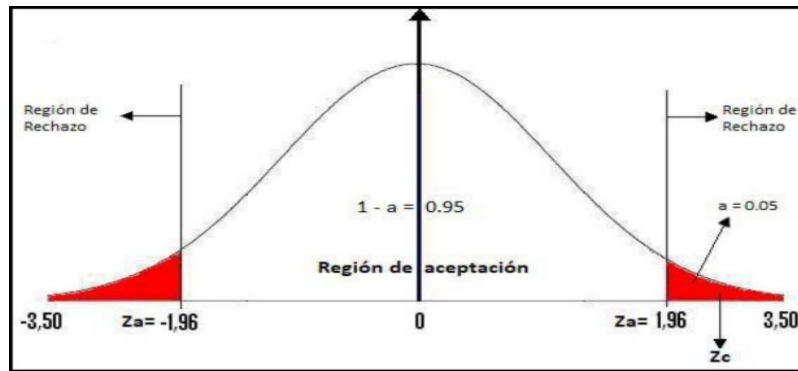


Figura 11. Distribución Z

Distribución T-Student

© Fuente: Hernández,
Fernández y Baptista (2014)

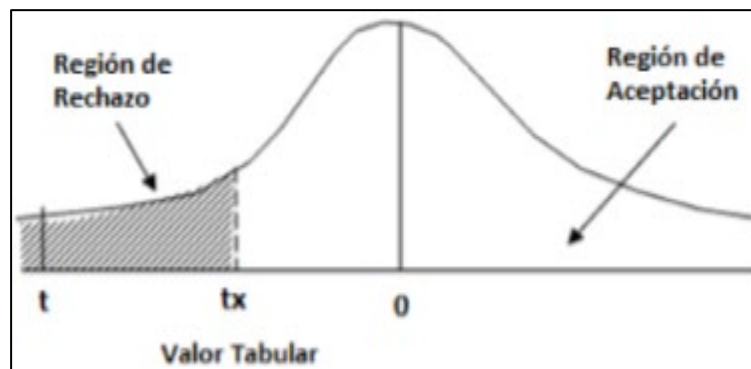


Figura 12. Distribución T-Student

2.5. Aspectos éticos

Aquí los investigadores van a guardar la confidencialidad de la data de la organización Construcciones & Acabados Zamora S.A.C.

En esta investigación toda la información necesaria fue recogida de la empresa, pues todo está sustentado en los instrumentos de los indicadores índice de ausentismo e índice de rotación. Además, los investigadores siguieron los lineamientos de la Universidad Privada César Vallejo.

III.RESULTADOS

Resultados

3.1. Análisis descriptivo

Indicador: Índice de ausentismo

Tabla 9. Estadístico descriptivo antes y después de la aplicación móvil – Índice de ausentismo

Estadísticos descriptivos

	N	Mínimo	Máximo	Media	Desviación estándar
IA_PRETEST	25	38,46	84,67	64,2150	23,1052
IA_POSTEST	25	18,75	59,84	39,2950	20,5457

Fuente: Elaboración propia

© Fuente: Elaboración propia

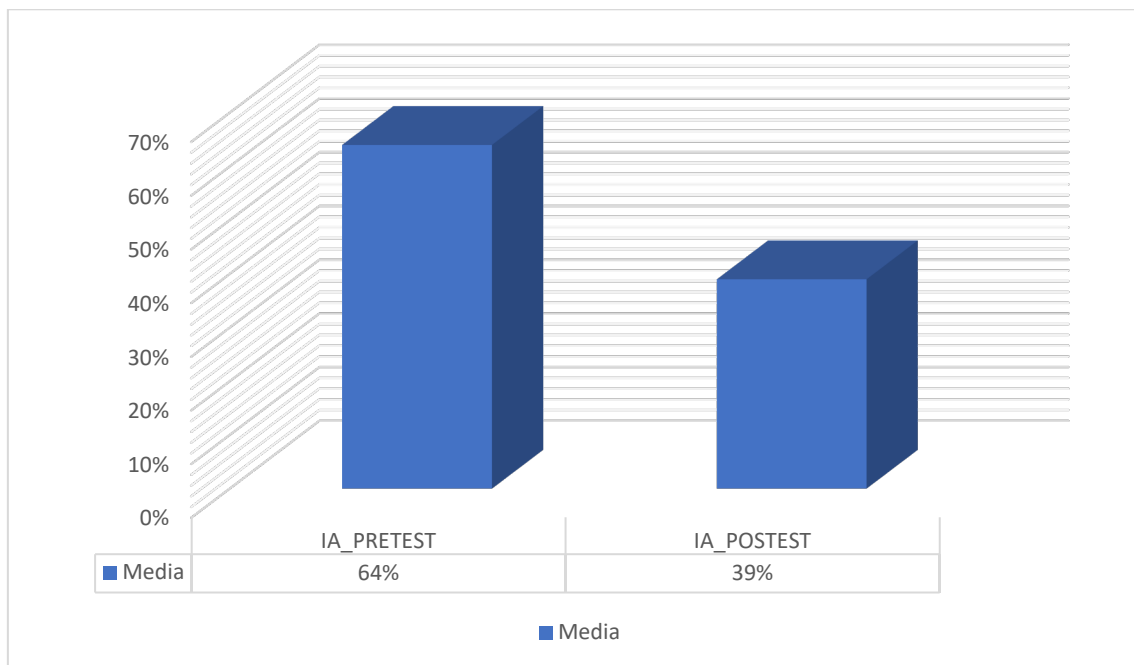


Figura 13. Índice de ausentismo antes y después de implementar la aplicación móvil

Indicador: Índice de rotación

Tabla 10. Estadístico descriptivo antes y después de la aplicación móvil – Índice de rotación

Estadísticos descriptivos

	N	Mínimo	Máximo	Media	Desviación estándar
IR_PRETEST	25	32,46	79,95	56,2051	23,745
IR_POSTEST	25	17,21	78,44	38.3254	30,615

Fuente: Elaboración propia

© Fuente: Elaboración propia

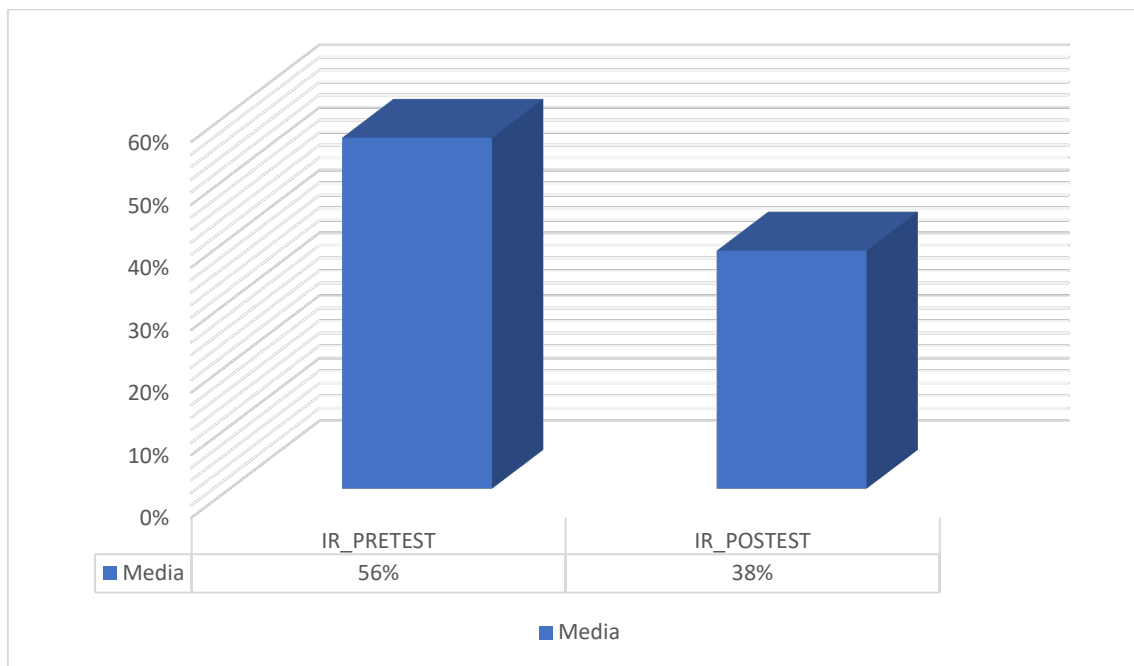


Figura 14. Índice de rotación antes y después de implementar la aplicación móvil

3.2. Análisis inferencial

Prueba de Normalidad

Indicador: Índice de ausentismo

Tabla 11. Pruebas de normalidad – Índice de ausentismo

	Pruebas de normalidad		
	Shapiro-Wilk		
	Estadístico	gl	Sig.
IA_PRETEST	0,861	25	0,123
IA_POSTEST	0,834	25	0,131

a. Corrección de significación de Lilliefors

Fuente: Elaboración propia

© Fuente: Elaboración propia

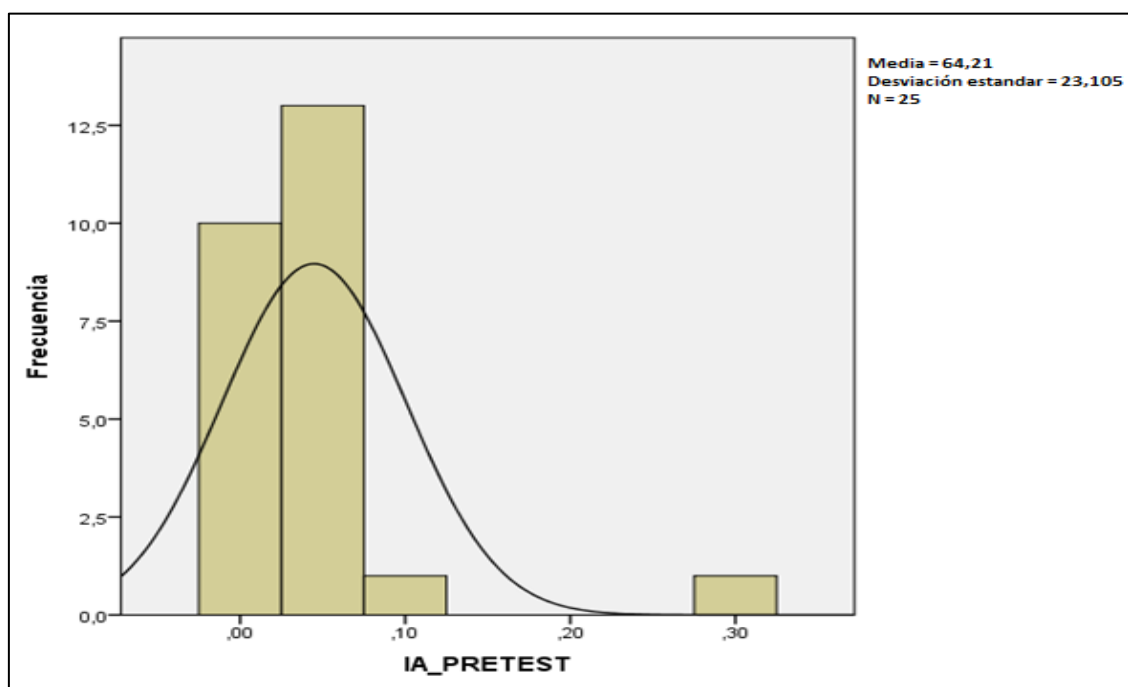


Figura 15. Prueba de normalidad del índice de ausentismo antes de implementar la aplicación móvil

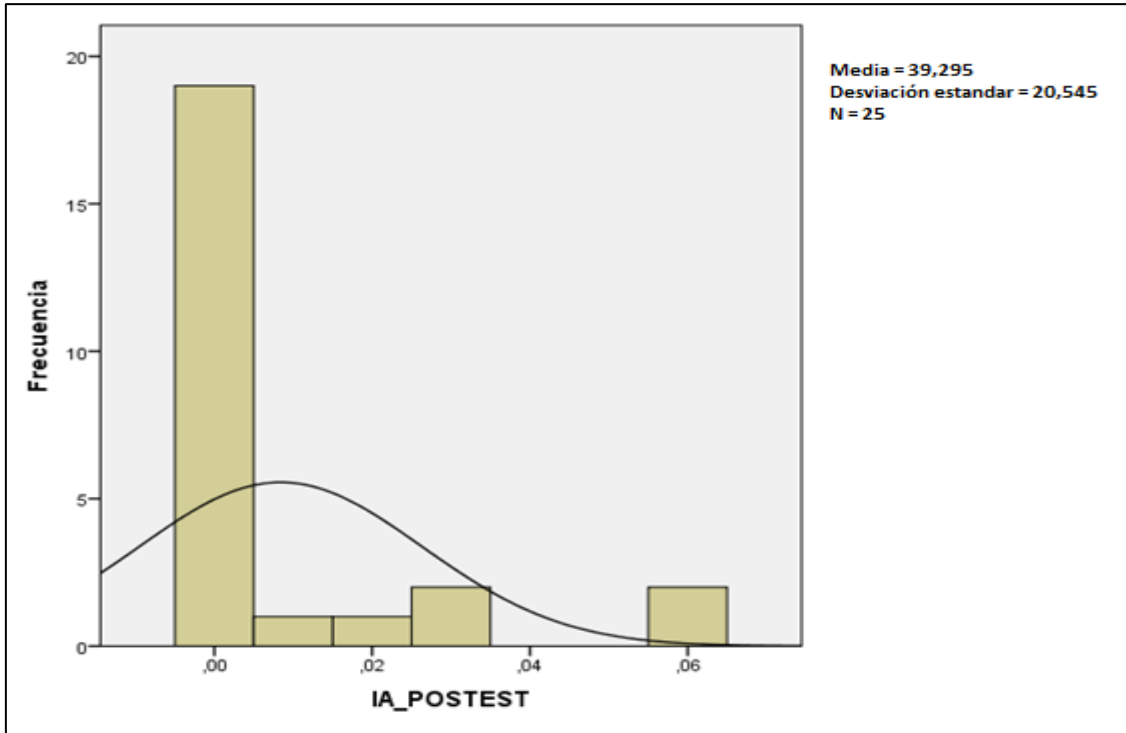


Figura 16. Prueba de normalidad del índice de ausentismo después de implementar la aplicación móvil

Indicador: Índice de rotación

Tabla 12. Pruebas de normalidad – Índice de rotación

Pruebas de normalidad			
	Shapiro-Wilk		
	Estadístico	gl	Sig.
IR_PRETEST	0,634	25	0,104
IR_POSTEST	0,612	25	0,101

a. Corrección de significación de Lilliefors

Fuente: Elaboración propia

© Fuente: Elaboración propia

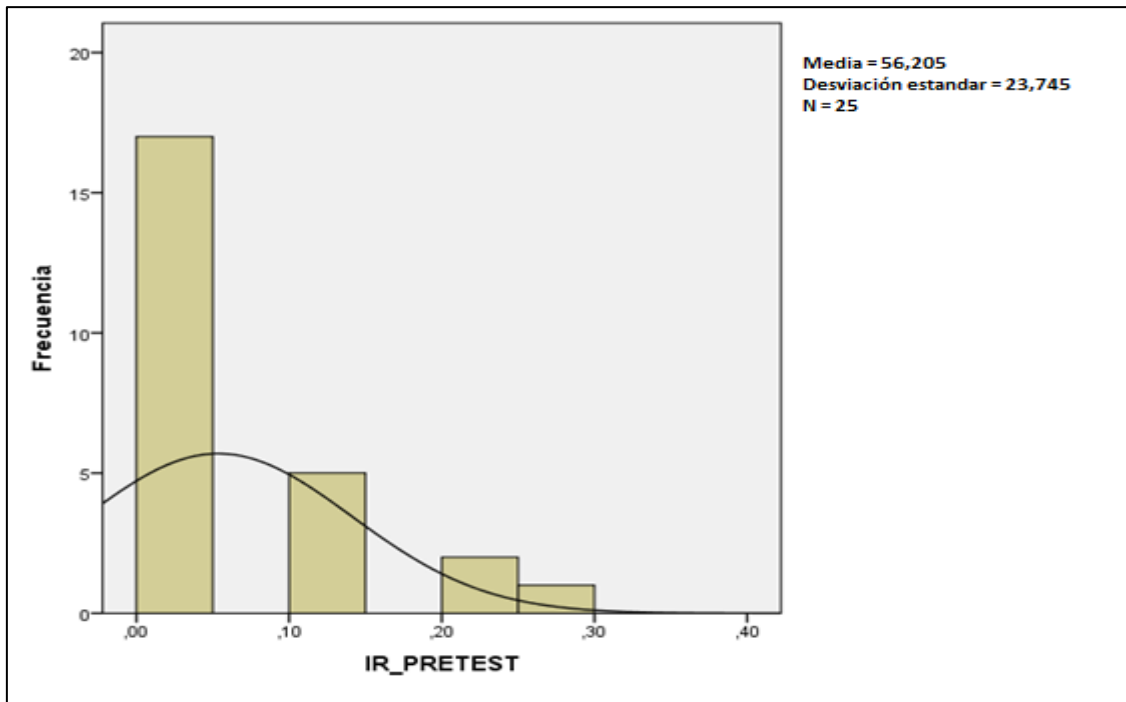


Figura 17. Prueba de normalidad del índice de rotación antes de implementar la aplicación móvil

© Fuente: Elaboración propia

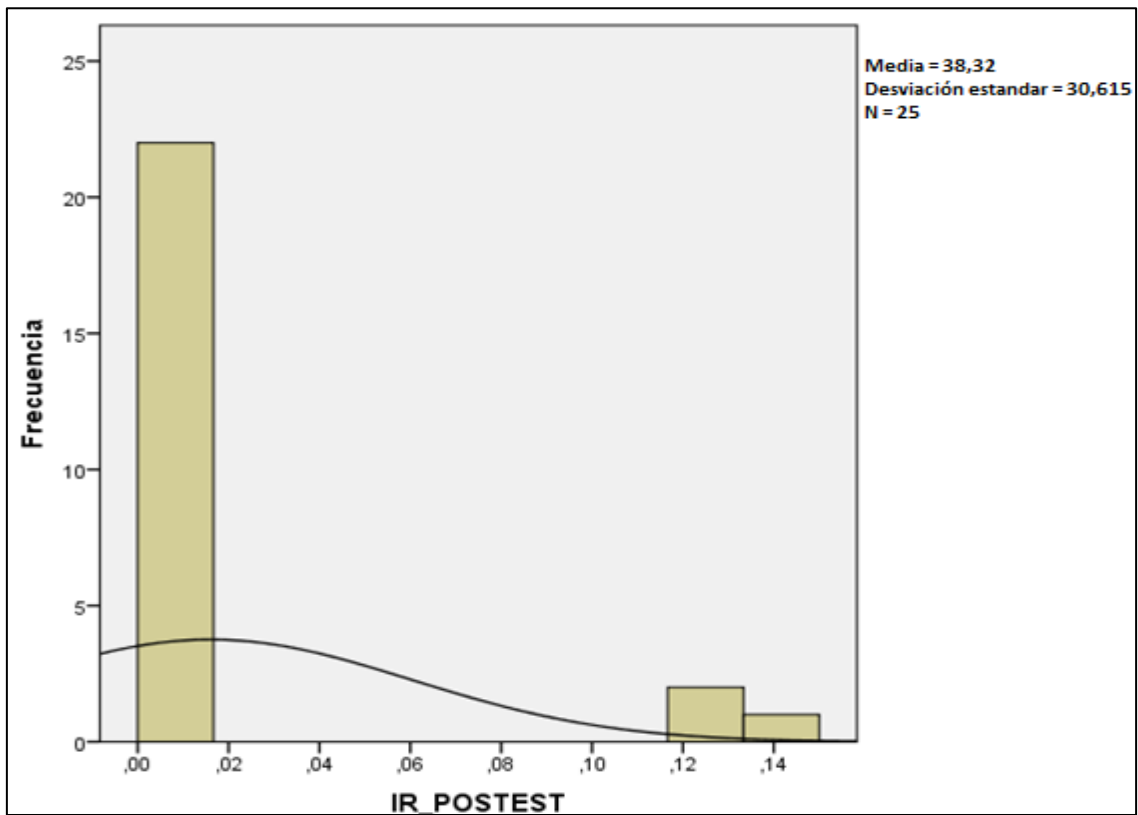


Figura 18. Prueba de normalidad del índice de rotación después de implementar la aplicación móvil

3.3. Prueba de Hipótesis

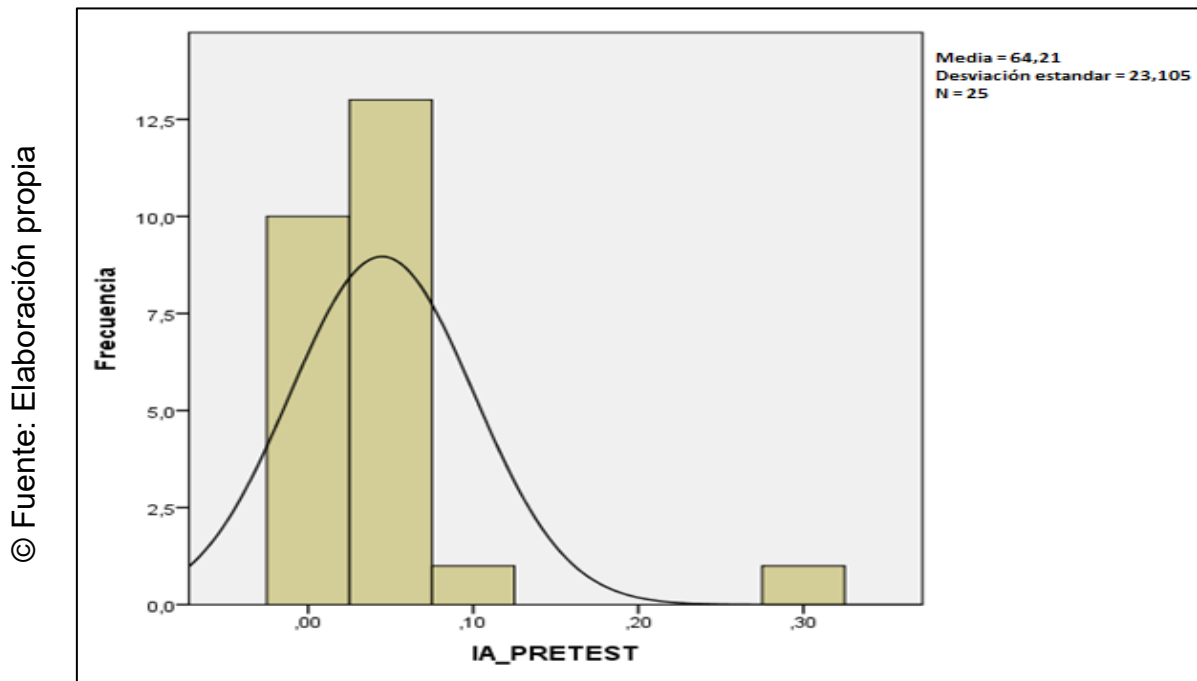


Figura 19. Índice de ausentismo antes de implementar la aplicación móvil

© Fuente: Elaboración propia

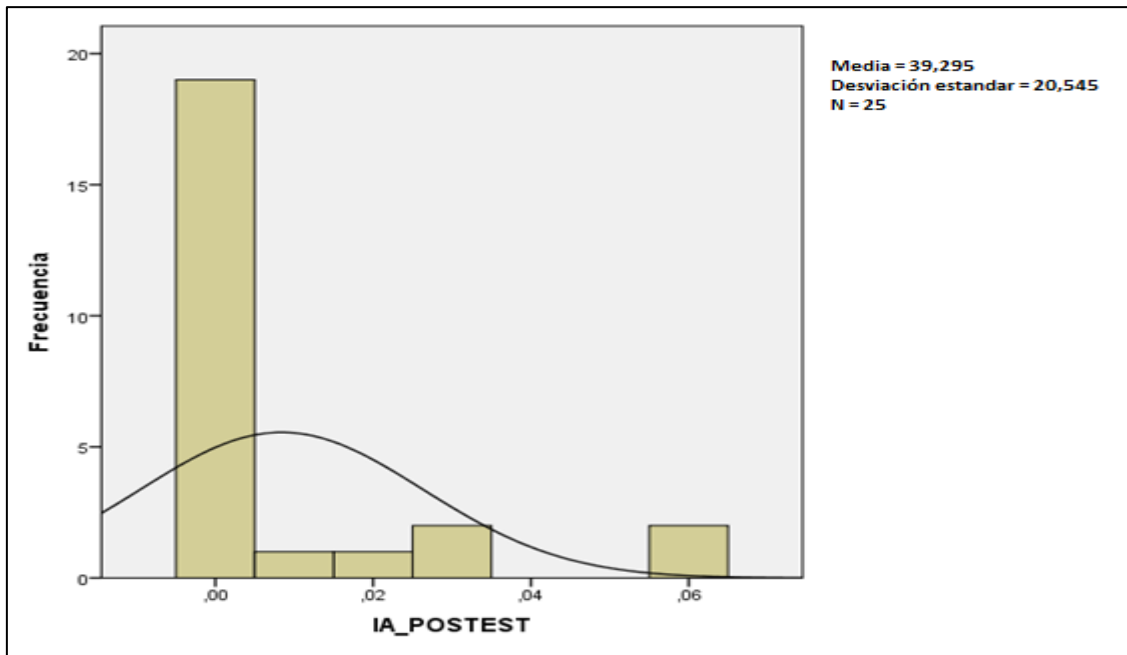


Figura 20. Índice de ausentismo después de implementar la aplicación móvil

© Fuente: Elaboración propia

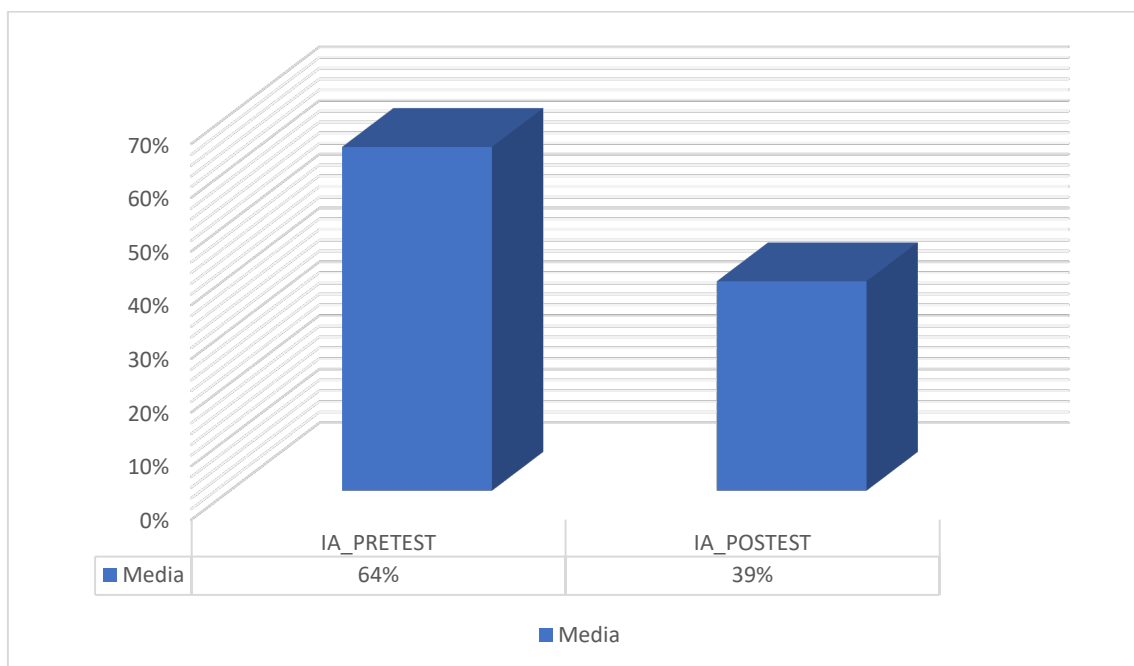


Figura 21. Índice de ausentismo – Comparativa General

Tabla 13. Prueba de T-Student para el índice de ausentismo en el control de personal antes y después de implementar la aplicación móvil

	Media	Desviación estándar	Prueba de T-Student		
			T	gl	Sig. (bilateral)
IA_pretest	64,2150	8.32646	14.965	24	0.04
IA_postest	39,2950				

Fuente: Elaboración propia

Reemplazamos entonces en T:

$$T_c = \frac{x - u}{S / \sqrt{n}}$$

$$T_c = \frac{64.21 - 39,29}{8.32646 \sqrt{25}}$$

$$T_c = 14.965$$

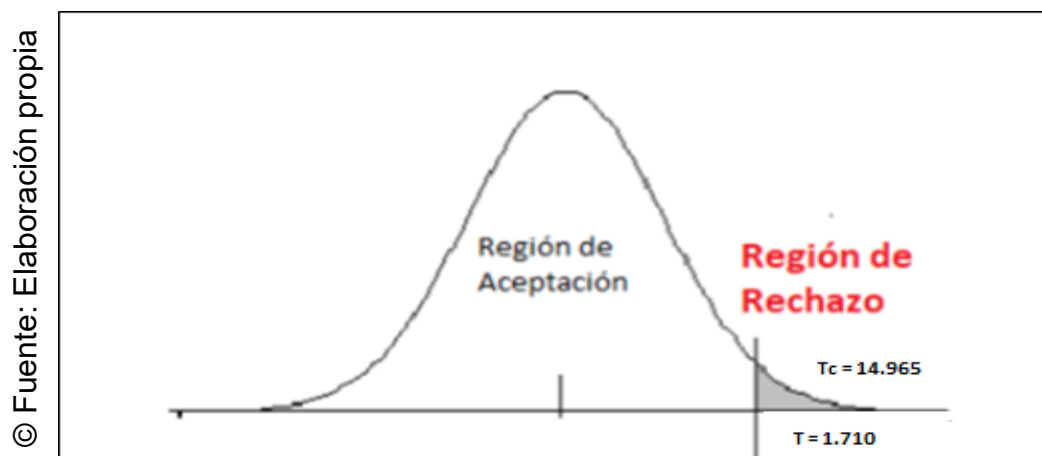


Figura 22. Prueba T-Student del índice de ausentismo

© Fuente: Elaboración propia

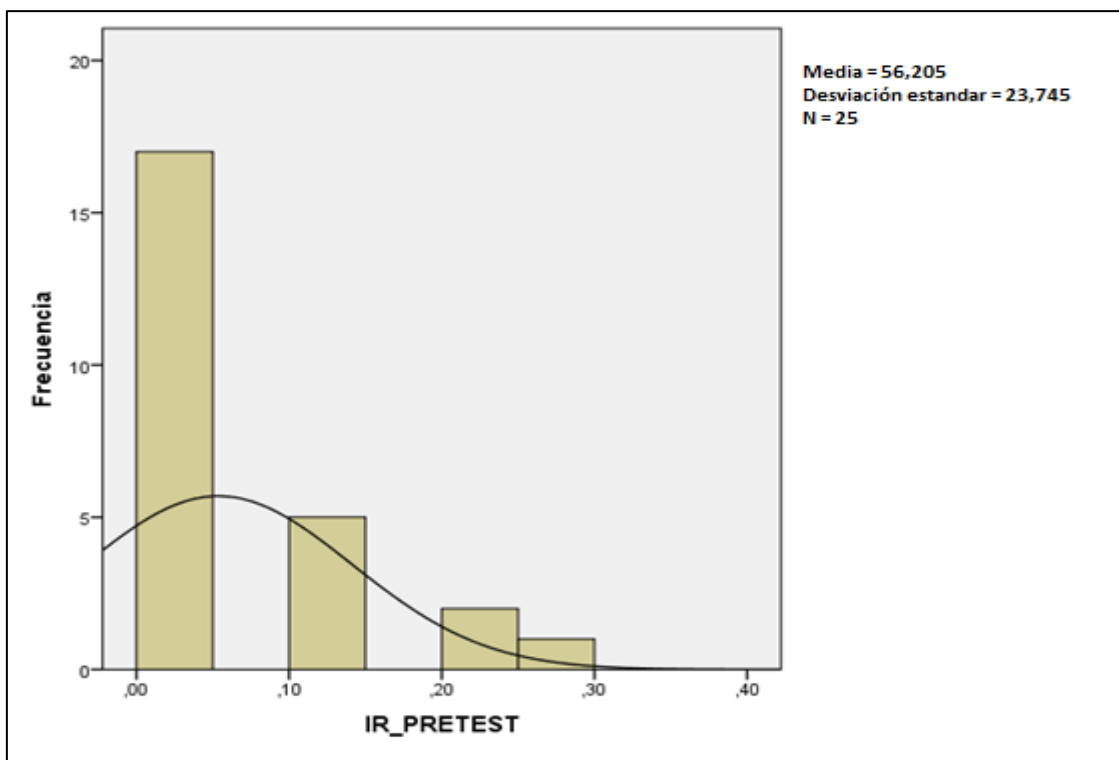


Figura 23. Índice de rotación antes de implementar la aplicación móvil

© Fuente: Elaboración propia

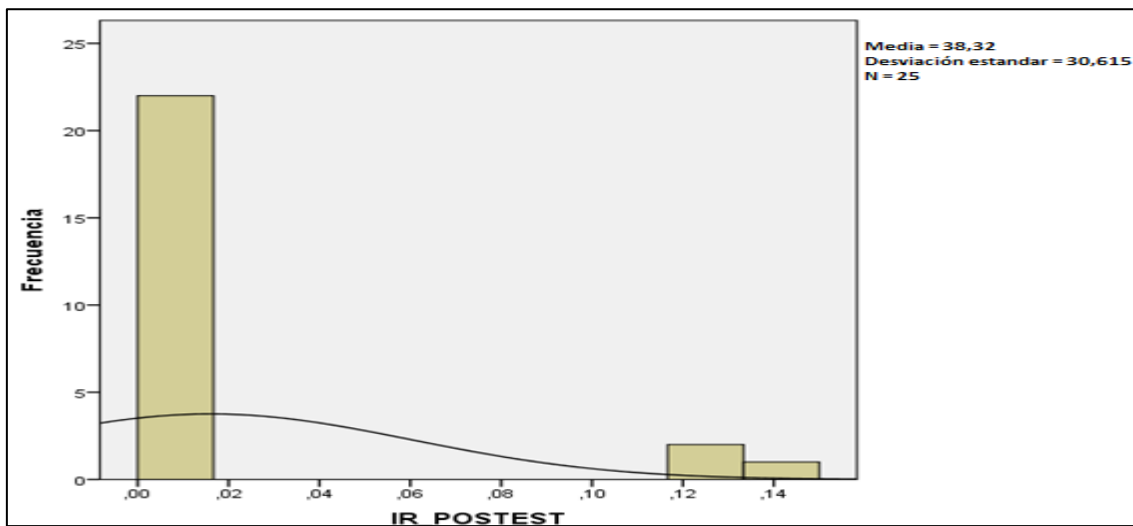


Figura 24. Índice de rotación después de implementar la aplicación móvil

© Fuente: Elaboración propia

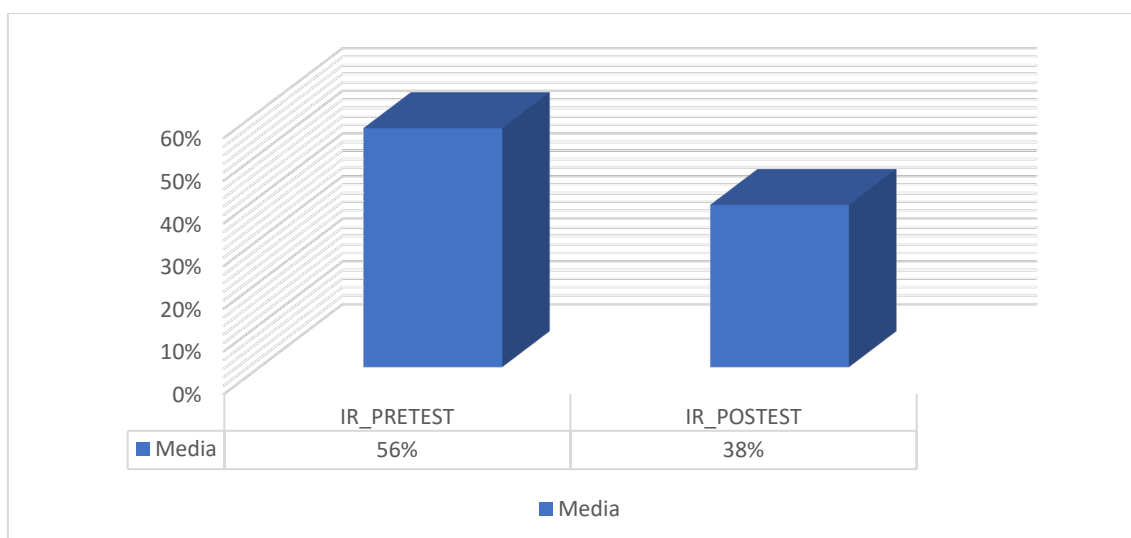


Figura 25. Índice de rotación – Comparativa General

Tabla 14. Prueba de T-Student para el índice de rotación en el control de personal antes y después de implementar la aplicación móvil

	Media	Desviación estándar	Prueba de T-Student		
			T	gl	Sig. (bilateral)
IR_pretest	56,2051	3.54124	25.247	24	0.003
IR_postest	38.3254				

Fuente: Elaboración propia

Reemplazamos entonces en T:

$$T_c = \frac{x - u}{S / \sqrt{n}}$$

$$T_c = \frac{56,20 - 38,32}{3.54124 \sqrt{25}}$$

$$T_c = 25.247$$

© Fuente: Elaboración propia

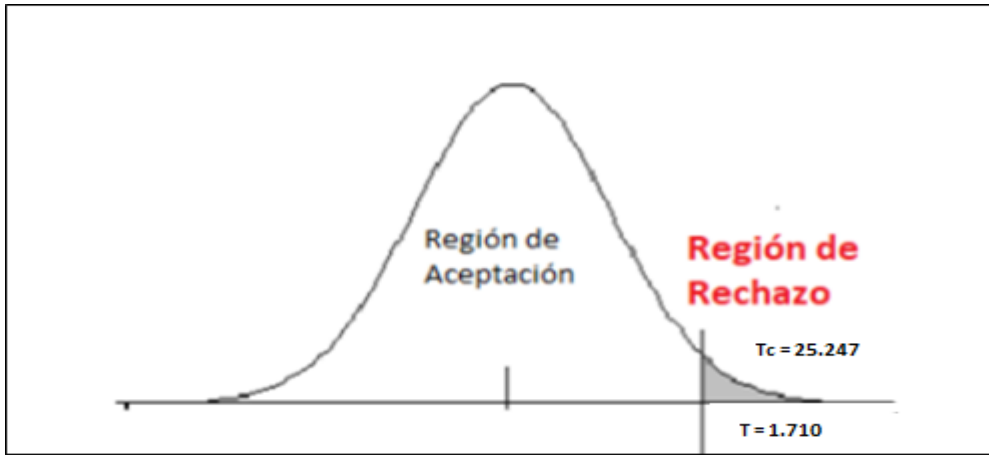


Figura 26. Prueba T-Student del índice de rotación

IV.DISCUSIÓN

Discusión

Se tuvo como resultado que, con la aplicación móvil disminuyó el índice de ausentismo en el control de personal de 64% a un 39%, lo que equivale a una disminución de 25%. Finalmente, también se tuvo como resultado que, con la aplicación móvil disminuyó el índice de rotación en el control de personal de 56% a un 38%, lo que equivale a una disminución de 18%.

La aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., disminuye el índice de ausentismo en un 25% y también disminuye el índice de rotación en 18%.

V.CONSLUSIÓN

Conclusión

En primer lugar, la app móvil mejora el control de personal en la organización Construcciones & Acabados Zamora S.A.C., pues permitió disminuir el índice de ausentismo e índice de rotación, por ende, alcanzó todos los objetivos de la tesis.

En segundo lugar, la app móvil disminuyó el índice de ausentismo en un 25%. Además, se realizó la prueba T-Student, en el cual el valor de T contraste fue de 14.965% y como es mayor a 1.710 se rechazó la hipótesis nula, aceptando la hipótesis alterna con un 95% de confianza. Por consiguiente, la app móvil disminuye el índice de ausentismo en el control de personal.

En tercer lugar, la app móvil disminuyó el índice de rotación en un 18%. Además, se realizó la prueba T-Student, en el cual el valor de T contraste fue de 25.247% y como es mayor a 1.710 se rechazó la hipótesis nula, aceptando la hipótesis alterna con un 95% de confianza. Por consiguiente, la app móvil disminuye el índice de rotación en el control de personal.

VI.RECOMENDACIONES

Recomendaciones

Se recomienda crear un chat online en la aplicación móvil para mejorar la comunicación entre los obreros, administradores y el área de recursos humanos, y así se pueda conocer en tiempo real todas las sugerencias, dudas, preferencias o necesidades que los obreros puedan tener con respecto a su información personal en la organización.

VII.REFERENCIAS BIBLIOGRÁFICAS

BUSTILLOS, Luis y JÁCOME, Gissela. *Aplicación móvil para la gestión de venta de medicamentos en la empresa FarmaMovil.* Tesis (Título de Ingeniero en Informática y Sistemas Computacionales). Latacunga: Universidad Técnica de Cotopaxi, 2016.

CABRALES, Génesis y TOALA, Magno. *Desarrollo e implementación de una aplicación móvil para registro y control de asistencia de estudiantes a clases para Instituto Tecnológico Superior Vicente Rocafuerte.* Tesis (Título de Ingeniero en Sistemas Computacionales). Guayaquil: Universidad de Guayaquil, 2019.

CHÁVEZ, Jorge. *Implementación de un sistema informático para el control de asistencia del personal en la Municipalidad de San Clemente.* Tesis (Título Profesional de Ingeniero de Sistemas e Informática). Lima: Universidad Alas Peruanas, 2014.

CHIAVENATO, Idalberto. *Gestión del Talento Humano.* 3.^a ed. México: McGraw-Hill, 2009. ISBN: 978-970-10-7340-7

CHIRIBOGA, Carlos y MAYORGA, María. *Diseño, Desarrollo e Implementación de un Sistema de Control de Asistencia para los docentes de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Sede Quito.* Tesis (Título de Ingeniero e Ingeniero e Ingeniera de Sistemas). Quito: Universidad Politécnica Salesiana Sede Quito, 2017.

DE LA TORRE, Julio. *¿Cuánto me cuesta el absentismo laboral?* [en línea]. 2018 [fecha de consulta: 20 de septiembre de 2018].

Disponible en: http://www.rrhhdigital.com/secciones/mercado-laboral/131634/Cuanto-me-cuesta-el-absentismo-laboral--?target=_self

DIAZ, Jueph y FLORES, Gyver. *Diseño e implementación de prototipo de un sistema biométrico para mejorar el control de asistencia del personal docente en la Facfym.* Tesis (Título Profesional de Ingeniero Electrónico). Lambayeque: Universidad Nacional Pedro Ruiz Gallo, 2019.

DIAZ, Melina. *Por qué tu negocio debe tener una aplicación web o de escritorio [en línea].* 2017 [fecha de consulta: 20 de septiembre de 2018].

Disponible en: <https://www.fuegoyamana.com/blog/aplicacion-web-o-de-escritorio-para-tu-negocio/>

FERNÁNDEZ, Mario. *El Control, Fundamento de la Gestión por Procesos Y la Calidad Total.* 2.^a ed. Madrid: ESIC Editorial, 2003. ISBN: 84-7356-351-4

FUNDAMENTOS de sistemas operativos por Wolf Gunnar [et al.]. México D.F.: Editorial de la Universidad Nacional Autónoma de México, 2015.

ISBN: 978-607-02-6544-0

GARDNER, Howard y DAVIS, Katie. *La generación App cómo los jóvenes gestionan su identidad, su privacidad y su imaginación en el mundo digital.* Barcelona: Espasa Libros, S.L.U., 2014. ISBN: 978-84-493-2985-2

GONZÁLEZ, Fabiola, ESCOTO, María y CHÁVEZ, Joanna. *Estadística aplicada en Psicología y Ciencias de la salud.* México: Editorial El Manual Moderno S.A., 2017. ISBN: 978-607-448-640-7

HERNÁNDEZ, Roberto, FERNÁNDEZ, Carlos y BAPTISTA, Pilar. *Metodología de la investigación*. 6.^a ed. México: McGraw-Hill, 2014. ISBN: 978-1-4562-2396-0

IZQUIERDO, Judit. *¿Qué es el XP Programming?* [en línea]. 2014. [fecha de consulta: 28 de octubre de 2018].

Disponible en: <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/>

MARTELL, Abel y SANTA CRUZ, Danny. *Sistema de información web de control de personal y planillas para mejorar la gestión de recursos humanos del Gobierno Provincial de Bagua Grande*. Tesis (Título de Ingeniero de Sistemas). Trujillo: Universidad Nacional de Trujillo, 2016.

MARTÍNEZ, Catalina y GONZÁLEZ, Arturo. *Técnicas e instrumentos de recogida y Análisis de datos*. Madrid: Universidad Nacional de Educación a Distancia, 2014. ISBN: 978-84-362-6822-5

MENZINSKY, Alexander, LÓPEZ, Gertrudis y PALACIO, Juan. *Historias de usuario Ingeniería de requisitos ágil* [en línea]. 2018 [fecha de consulta: 28 de septiembre 2018].

Disponible en: http://scrummanager.net/files/historias_usuario_scrum_manager.pdf

PABÓN, Pedro. *Metodología de Desarrollo Tradicional RUP* [en línea]. 2018. [fecha de consulta 28 de octubre de 2018].

Disponible en: <https://smartsoftcolombia.com/portal/index.php/blog/49-rup>

PAREDES, Alipio y PAREDES, Wagner. *Sistema de localización basado en dispositivos móviles para el control y monitoreo del personal en el campamento de la empresa minera VanessaSac.* Tesis (Título Profesional de Ingeniero de Sistemas). Puno: Universidad Nacional del Altiplano, 2017.

PAZMIÑO, Kerly y RAMÍREZ, Guillermo. *Aplicación móvil para control de acceso y asistencia en la empresa Ecuador On Rails mediante reconocimiento facial y códigos QR utilizando el framework React Native y Tecnología Numato.* Tesis (Título de Ingeniero en Sistemas Computacionales). Guayaquil: Universidad de Guayaquil, 2019.

PIMIENTA, Pedro. *Tipos de aplicaciones móviles y sus características* [en línea]. 2014 [fecha de consulta: 28 de septiembre 2018]. Disponible en:

<https://deideaaapp.org/tipos-de-aplicaciones-moviles-y-sus-caracteristicas/>

RIVERA, Iván. *Desarrollo e implementación de un sistema de código de barras con la metodología XP para optimizar el control de asistencia en la Junta Administradora de Servicios de Saneamiento Quilcas.* Tesis (Título Profesional de Ingeniero de Sistemas y Computación). Huancayo: Universidad Peruana Los Andes, 2017.

ROCA, Carlos y VILLAFUERTE, Arturo. *Aplicación móvil para registro de asistencias de la Universidad Central del Ecuador.* Tesis (Título de Ingeniero Informático). Quito: Universidad Central del Ecuador, 2017.

SCHIAFFARINO, Andrés. *Modelo cliente servidor* [en línea]. 2019 [fecha de consulta: 10 de agosto de 2019].

Disponible en: <https://blog.infranetworking.com/modelo-cliente-servidor/>

SCRUMstudy. *Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK)*. 3.^a ed. Arizona: VMEdU, Inc., 2017. ISBN: 978-0-9899252-0-4

SCHWABER, Ken y SUTHERLAND, Jeff. *La Guía Definitiva de Scrum: Las Reglas del Juego* [en línea]. 2017 [fecha de consulta: 10 de octubre de 2018].

Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>

SOLÍS, Walter. *Sistema de Información Web para el Proceso de Control de Asistencia del Personal Administrativo y Asistencial de la Red de Salud Lima Este Metropolitana*. Tesis (Título Profesional de Ingeniero de Sistemas). Lima: Universidad César Vallejo, 2017

TANDAZO, Hipatia y CHAMBA Cristian. *Diseño y desarrollo de un prototipo de solución web para el apoyo y automatización del proceso en el departamento de recursos humanos que permita el control del personal docente y administrativo de la Asociación Fe y Alegría Ecuador*. Tesis (Título de Ingeniero e Ingeniero de Sistemas). Quito: Universidad Politécnica Salesiana Sede Quito, 2016.

VALLEJO, Luz. *Gestión del talento humano*. Riobamba: Escuela Superior Politécnica de Chimborazo, 2015. ISBN: 978-9942-14-271-9

VILALTA, Carlos. *Análisis de datos*. México: Colección Docencia, 2016. ISBN: 978-607-9367-91-6

ANEXOS

ANEXO 1: Matriz de consistencia

PROBLEMAS	OBJETIVOS	HIPOTESIS	VARIABLES	DIMENSIONES	INDICADORES	FORMULA	METODOLOGIA
<p>Problema General PG: ¿De qué manera influye una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.?</p>	<p>Objetivo General OG: Determinar la influencia de una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.</p>	<p>Hipótesis General HG: La aplicación móvil mejora el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.</p>	<p>Variable Independiente</p> <p>Aplicación móvil</p> <p>Variable Dependiente</p>				<p>Tipo de Investigación: Aplicada</p> <p>Diseño de Investigación: Pre- experimental</p> <p>Nivel de investigación: Explicativa</p>
<p>Problemas Específicos PE1: ¿De qué manera una aplicación móvil influye en el índice de ausentismo en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.?</p>	<p>Objetivos Específicos OE1: Determinar la influencia de una aplicación móvil en el índice de ausentismo en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.</p>	<p>Hipótesis Específicas HE1: La aplicación móvil disminuye el índice de ausentismo en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.</p>		Registro de remuneración	índice de ausentismo	$IA = \frac{TP/HP}{TP/HL}$ <p>Donde: IA = índice de ausentismo TP = Total de personas HP = Horas perdidas HT = Horas laboradas</p>	<p>Método de Investigación: Hipotético - deductivo</p> <p>Población: 400 obreros por los indicadores</p> <p>índice de ausentismo e índice de rotación, estratificados por días en un mes en 25 fichas de registro</p>
<p>PE2: ¿De qué manera una aplicación móvil influye en el índice de rotación en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.?</p>	<p>OE2: Determinar la influencia de una aplicación móvil en el índice de rotación en el control de personal en la empresa Construcciones y Acabados Zamora S.A.C.</p>	<p>HE2: La aplicación móvil disminuye el índice de rotación en el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.</p>	Control de personal	Registro de personal	índice de rotación	$IR = \frac{NES}{NEA}$ <p>Donde: IR = índice de rotación NES = Número de empleados separados NEA = Número de empleados asistidos</p>	<p>Muestra: 100 obreros, estratificados por días en un mes en 25 fichas de registro</p> <p>Técnicas: Fichaje</p> <p>Instrumentos: Ficha de registro</p> <p>Unidad de medida: Porcentaje</p>

ANEXO 2: Ficha técnica. Instrumento de recolección de datos

Autores	Escurra Colquis, Maurihño Enrique Zamora Noreña, Luis Fernando	
Nombre del instrumento	Ficha de Registro	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Fecha de aplicación	01 de agosto de 2018	
Objetivo	Determinar la influencia de una aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.	
Tiempo de duración	25 días (lunes a sábados)	
Elección de técnica e instrumento		
Variable	Técnica	Instrumento
Variable dependiente Control de personal	Fichaje	Ficha de registro
Variable independiente Aplicación móvil	_____	_____

Fuente: Elaboración propia

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 3: Instrumento de investigación

Ficha de Registro Pre-Test			
Tipo de Prueba		Pre-Test	
Empresa Investigada		Construcciones & Acabados Zamora S.A.C	
Motivo de		Indice de ausentismo	
Fecha de Inicio	2018-10-01	Fecha Final	2018-10-29

Variable	Indicador	Medida	Fórmula
Control de Personal	Indice de ausentismo	Porcentaje	$IA = (TP / HP) / (TP/HL)$

Ítem	Fecha de registro de ausentismos	Obras	Total de personas(TP)	Horas perdidas(HP)	Horas laboradas(HL)	Indice de ausentismo(IA)
1	2018-10-01	Obras 1	9	11	79	0.07
2	2018-10-02		8	37	53	0.01
3	2018-10-03		8	13	77	0.06
4	2018-10-04		7	20	70	0.04
5	2018-10-05		7	37	53	0.01
6	2018-10-06		8	13	77	0.06
7	2018-10-08		8	20	70	0.04
8	2018-10-09		7	26	64	0.02
9	2018-10-10		7	21	69	0.03
10	2018-10-11		7	26	64	0.02
11	2018-10-12		9	26	64	0.02
12	2018-10-13		7	20	70	0.04
13	2018-10-15		9	70	20	0.00
14	2018-10-16		7	20	70	0.04
15	2018-10-17		9	11	79	0.07
16	2018-10-18		9	0	90	0.00
17	2018-10-19		7	20	70	0.04
18	2018-10-20		8	10	80	0.08
19	2018-10-22		7	26	64	0.02
20	2018-10-23		8	21	69	0.03
21	2018-10-24		8	13	77	0.06
22	2018-10-25		7	26	64	0.02
23	2018-10-26		9	3	87	0.29
24	2018-10-27		8	21	69	0.03
25	2018-10-29		8	26	64	0.02
			196			1.12

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL


ANEXO 3: Instrumento de investigación en el índice de ausentismo

Ficha de Registro Post-test						
Tipo de Prueba				Post-test		
Empresa Investigada		Construcciones & Acabados Zamora S.A.C				
Motivo de Investigación		Índice de ausentismo				
Fecha de Inicio		2019-10-01	Fecha Final	2019-10-29		

Variable Dependiente	Indicador	Medida	Fórmula
Control de Personal	Índice de ausentismo	Porcentaje	$IA = (TP / HP) / (TP/HL)$

Ítem	Fecha de registro de ausentismos	Obras	Total de personas(TP)	Horas perdidas(HP)	Horas laboradas(HL)	Índice de ausentismo(IA)
1	2019-10-01	Obras 1	8	11	69	0.06
2	2019-10-02		8	0	80	0.00
3	2019-10-03		8	0	80	0.00
4	2019-10-04		8	0	80	0.00
5	2019-10-05		8	0	80	0.00
6	2019-10-07		8	0	80	0.00
7	2019-10-08		8	0	80	0.00
8	2019-10-09		8	0	80	0.00
9	2019-10-10		7	24	56	0.02
10	2019-10-11		8	0	80	0.00
11	2019-10-12		8	0	80	0.00
12	2019-10-14		8	0	80	0.00
13	2019-10-15		8	0	80	0.00
14	2019-10-16		8	18	62	0.03
15	2019-10-17		7	11	69	0.06
16	2019-10-18		8	0	80	0.00
17	2019-10-19		8	18	62	0.03
18	2019-10-21		8	0	80	0.00
19	2019-10-22		8	0	80	0.00
20	2019-10-23		8	0	80	0.00
21	2019-10-24		8	0	80	0.00
22	2019-10-25		7	33	47	0.01
23	2019-10-26		8	0	80	0.00
24	2019-10-28		8	0	80	0.00
25	2019-10-29		7	10	70	0.07
			196			0.28

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.


 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 3: Instrumento de investigación en el índice de rotación

Ficha de Registro Pre-Test				
Tipo de Prueba		Tipo de Prueba	Pre-Test	
Empresa Investigada	Construcciones & Acabados Zamora S.A.C			
Motivo de	Indice de rotación			
Fecha de Inicio	2018-10-01	Fecha Final	Fecha final	2018-10-29

Variable Dependiente	Indicador	Medida	Fórmula
Control de Personal	Indice de rotación	Porcentaje	Indice de rotación = NES / NEA

Item	Ficha de registro de indice de rotación	Obras	Numero de empleados separados(NES)	Numero de empleados asistidos(NEA)	Indice de rotación
1	2018-10-01	Obra 1	1	9	0.11
2	2018-10-02		1	8	0.13
3	2018-10-03		0	8	0.00
4	2018-10-04		0	7	0.00
5	2018-10-05		2	7	0.29
6	2018-10-06		1	8	0.13
7	2018-10-08		0	8	0.00
8	2018-10-09		0	7	0.00
9	2018-10-10		0	7	0.00
10	2018-10-11		0	7	0.00
11	2018-10-12		2	9	0.22
12	2018-10-13		0	7	0.00
13	2018-10-15		1	9	0.11
14	2018-10-16		0	7	0.00
15	2018-10-17		2	9	0.22
16	2018-10-18		0	9	0.00
17	2018-10-19		0	7	0.00
18	2018-10-20		0	8	0.00
19	2018-10-22		0	7	0.00
20	2018-10-23		1	8	0.13
21	2018-10-24		0	8	0.00
22	2018-10-25		0	7	0.00
23	2018-10-26		0	9	0.00
24	2018-10-27		0	8	0.00
25	2018-10-29		0	8	0.00
			196	1.34	

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C



 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 3: Instrumento de investigación en el índice de rotación

Ficha de Registro Post-Test					
Tipo de Prueba		Tipo de Prueba		Post-test	
Empresa Investigada	Construcciones & Acabados Zamora S.A.C				
Motivo de	Indice de rotación				
Fecha de Inicio	2019-10-01	Fecha Final	Fecha final	2019-10-29	

Variable Dependiente	Indicador	Medida	Fórmula
Control de Personal	Indice de rotación	Porcentaje	Indice de rotación = NES / NEA

Item	Ficha de registro de indice de rotación	Obras	Numero de empleados separados(NES)	Numero de empleados asistidos(NEA)	Indice de rotación
1	2019-10-01	Obra 1	0	8	0.00
2	2019-10-02		0	8	0.00
3	2019-10-03		0	8	0.00
4	2019-10-04		0	8	0.00
5	2019-10-05		1	8	0.13
6	2019-10-07		0	8	0.00
7	2019-10-08		0	8	0.00
8	2019-10-09		0	8	0.00
9	2019-10-10		0	7	0.00
10	2019-10-11		0	8	0.00
11	2019-10-12		1	8	0.13
12	2019-10-14		0	8	0.00
13	2019-10-15		0	8	0.00
14	2019-10-16		0	8	0.00
15	2019-10-17		1	7	0.14
16	2019-10-18		0	8	0.00
17	2019-10-19		0	8	0.00
18	2019-10-21		0	8	0.00
19	2019-10-22		0	8	0.00
20	2019-10-23		0	8	0.00
21	2019-10-24		0	8	0.00
22	2019-10-25		0	7	0.00
23	2019-10-26		0	8	0.00
24	2019-10-28		0	8	0.00
25	2019-10-29		0	7	0.00
			196		0.40

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 4: Base de datos Experimental

Índice de Ausentismo		
Nº	PRE-TEST	POST-TEST
1	0.07	0.06
2	0.01	0.00
3	0.06	0.00
4	0.04	0.00
5	0.01	0.00
6	0.06	0.00
7	0.04	0.00
8	0.02	0.00
9	0.03	0.02
10	0.02	0.00
11	0.02	0.00
12	0.04	0.00
13	0.00	0.00
14	0.04	0.03
15	0.07	0.06
16	0.00	0.00
17	0.04	0.03
18	0.08	0.00
19	0.02	0.00
20	0.03	0.00
21	0.06	0.00
22	0.02	0.01
23	0.29	0.00
24	0.03	0.00
25	0.02	0.00

Índice de Rotación		
Nº	PRE-TEST	POST-TEST
1	0.11	0.00
2	0.13	0.00
3	0.00	0.00
4	0.00	0.00
5	0.29	0.13
6	0.13	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	0.22	0.13
12	0.00	0.00
13	0.11	0.00
14	0.00	0.00
15	0.22	0.14
16	0.00	0.00
17	0.00	0.00
18	0.00	0.00
19	0.00	0.00
20	0.13	0.00
21	0.00	0.00
22	0.00	0.00
23	0.00	0.00
24	0.00	0.00
25	0.00	0.00

CONSTRUCCIONES Y ACABADOS S.A.S.


 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 5: Resultados de la Confiabilidad del instrumento

Ficha de Registro Test						
Tipo de Prueba				Test		
Empresa Investigada		Construcciones & Acabados Zamora S.A.C				
Motivo de		Indice de ausentismo				
Fecha de Inicio	2018-09-01	Fecha Final	2018-09-29			

Variable	Indicador	Medida	Fórmula
Control de Personal	Indice de ausentismo	Porcentaje	$IA = (TP / HP) / (TP/HL)$

Ítem	Fecha de registro de ausentismos	Obras	Total de personas(TP)	Horas perdidas(HP)	Horas laboradas(HL)	Indice de ausentismo(IA)
1	2018-09-01	Obras 1	8	10	80	0.08
2	2018-09-03		6	30	60	0.02
3	2018-09-04		8	11	79	0.07
4	2018-09-05		8	13	77	0.06
5	2018-09-06		6	30	60	0.02
6	2018-09-07		9	11	79	0.07
7	2018-09-08		9	2	88	0.44
8	2018-09-10		7	21	69	0.03
9	2018-09-11		7	20	70	0.04
10	2018-09-12		9	21	69	0.03
11	2018-09-13		7	21	69	0.03
12	2018-09-14		8	11	79	0.07
13	2018-09-15		7	70	20	0.00
14	2018-09-17		8	11	79	0.07
15	2018-09-18		8	10	80	0.08
16	2018-09-19		8	21	69	0.03
17	2018-09-20		9	13	77	0.06
18	2018-09-21		9	2	88	0.44
19	2018-09-22		8	21	69	0.03
20	2018-09-24		8	21	69	0.03
21	2018-09-25		8	11	79	0.07
22	2018-09-26		7	21	69	0.03
23	2018-09-27		9	2	88	0.44
24	2018-09-28		7	20	70	0.04
25	2018-09-29		8	21	69	0.03
			196			2.31

CONSTRUCCIONES & ACABADOS S.A.C. S.

 JAVIER ZAMORA
 GERENTE GENERAL


ANEXO 5: Resultados de la Confiabilidad del instrumento

Ficha de Registro Re-test						
Tipo de Prueba				Re-test		
Empresa Investigada		Construcciones & Acabados Zamora S.A.C				
Motivo de		Indice de ausentismo				
Fecha de Inicio		2018-08-01	Fecha Final		2018-08-29	

Variable	Indicador	Medida	Fórmula
Control de Personal	Indice de ausentismo	Porcentaje	$IA = (TP / HP) / (TP/HL)$

Ítem	Fecha de registro de ausentismos	Obras	Total de personas(TP)	Horas perdidas(HP)	Horas laboradas(HL)	Indice de ausentismo(IA)
1	2018-08-01	Obras 1	9	1	89	0.89
2	2018-08-02		9	1	89	0.89
3	2018-08-03		9	1	89	0.89
4	2018-08-04		8	10	80	0.08
5	2018-08-06		8	10	80	0.08
6	2018-08-07		8	10	80	0.08
7	2018-08-08		9	1	89	0.89
8	2018-08-09		7	20	70	0.04
9	2018-08-10		8	10	80	0.08
10	2018-08-11		7	20	70	0.04
11	2018-08-13		7	20	70	0.04
12	2018-08-14		8	10	80	0.08
13	2018-08-15		5	40	50	0.01
14	2018-08-16		8	10	80	0.08
15	2018-08-17		9	1	89	0.89
16	2018-08-18		7	20	70	0.04
17	2018-08-20		8	10	80	0.08
18	2018-08-21		9	1	89	0.89
19	2018-08-22		7	20	70	0.04
20	2018-08-23		7	20	70	0.04
21	2018-08-24		8	10	80	0.08
22	2018-08-25		7	20	70	0.04
23	2018-08-27		9	1	89	0.89
24	2018-08-28		8	10	80	0.08
25	2018-08-29		7	20	70	0.04
			196			7.28

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C



JAVIER ZAMORA
GERENTE GENERAL

ANEXO 5: Resultados de la Confiabilidad del instrumento

Correlaciones

		IA_RETEST	IA_TEST
IA_RETEST	Correlación de Pearson	1	,732**
	Sig. (bilateral)		,000
	N	25	25
IA_TEST	Correlación de Pearson	,732**	1
	Sig. (bilateral)	,000	
	N	25	25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Se logra apreciar en esta figura la confiabilidad del índice de ausentismo, ya que el resultado es de 0.732, y se puede deducir que el instrumento de la presente investigación es confiable.

ANEXO 5: Resultados de la Confiabilidad del instrumento

Ficha de Registro Test				
Tipo de Prueba		Tipo de Prueba	Test	
Empresa Investigada	Construcciones & Acabados Zamora S.A.C			
Motivo de	Indice de rotación			
Fecha de Inicio	2018-09-01	Fecha Final	Fecha final	2018-09-29

Variable Dependiente	Indicador	Medida	Fórmula
Control de Personal	Indice de rotación	Porcentaje	Indice de rotación = NES / NEA

Item	Ficha de registro de indice de rotación	Obras	Numeros de empleados separados(NES)	Numero de personas asistidos(NEA)	Indice de rotación
1	2018-09-01	Obra 1	1	8	0.13
2	2018-09-03		1	6	0.17
3	2018-09-04		0	8	0.00
4	2018-09-05		0	8	0.00
5	2018-09-06		2	6	0.33
6	2018-09-07		2	9	0.22
7	2018-09-08		0	9	0.00
8	2018-09-10		1	7	0.14
9	2018-09-11		1	7	0.14
10	2018-09-12		1	9	0.11
11	2018-09-13		2	7	0.29
12	2018-09-14		0	8	0.00
13	2018-09-15		1	7	0.14
14	2018-09-17		1	8	0.13
15	2018-09-18		2	8	0.25
16	2018-09-19		0	8	0.00
17	2018-09-20		1	9	0.11
18	2018-09-21		0	9	0.00
19	2018-09-22		0	8	0.00
20	2018-09-24		2	8	0.25
21	2018-09-25		0	8	0.00
22	2018-09-26		0	7	0.00
23	2018-09-27		1	9	0.11
24	2018-09-28		0	7	0.00
25	2018-09-29		1	8	0.13
			196	2.65	

CONSTRUCCIONES Y ACABADOS ZAMORA S.A.C.


 JAVIER ZAMORA
 GERENTE GENERAL

ANEXO 5: Resultados de la Confiabilidad del instrumento

Ficha de Registro Re Test				
Tipo de Prueba		Tipo de Prueba	Re-test	
Empresa Investigada	Construcciones & Acabados Zamora S.A.C			
Motivo de	Indice de rotacion			
Fecha de Inicio	2018-08-01	Fecha Final	Fecha Final	2018-08-29

Variable Dependiente	Indicador	Medida	Fórmula
Control de Personal	Indice de rotación	Porcentaje	Indice de rotación = NES / NEA

Item	Ficha de registro de indice de rotación	Obras	Numeros de empleados separados(NES)	Numero de empleados asistidos(NEA)	Indice de rotación
1	2018-08-01	Obra 1	2	9	0.22
2	2018-08-02		2	9	0.22
3	2018-08-03		0	9	0.00
4	2018-08-04		0	8	0.00
5	2018-08-06		3	8	0.38
6	2018-08-07		2	8	0.25
7	2018-08-08		0	9	0.00
8	2018-08-09		2	7	0.29
9	2018-08-10		2	8	0.25
10	2018-08-11		1	7	0.14
11	2018-08-13		3	7	0.43
12	2018-08-14		0	8	0.00
13	2018-08-15		1	5	0.20
14	2018-08-16		2	8	0.25
15	2018-08-17		3	9	0.33
16	2018-08-18		0	7	0.00
17	2018-08-20		1	8	0.13
18	2018-08-21		0	9	0.00
19	2018-08-22		0	7	0.00
20	2018-08-23		2	7	0.29
21	2018-08-24		1	8	0.13
22	2018-08-25		0	7	0.00
23	2018-08-27		2	9	0.22
24	2018-08-28		0	8	0.00
25	2018-08-29		1	7	0.14
			196	3.87	

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C

JAVIER ZAMORA
GERENTE GENERAL

ANEXO 5: Resultados de la Confiabilidad del instrumento

Correlaciones

		IR_RETEST	IR_TEST
IR_RETEST	Correlación de Pearson	1	,713**
	Sig. (bilateral)		,000
	N	25	25
IR_TEST	Correlación de Pearson	,713**	1
	Sig. (bilateral)	,000	
	N	25	25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Se logra apreciar en esta figura la confiabilidad del índice de rotación, ya que el resultado es de 0.713, y se puede deducir que el instrumento de la presente investigación es confiable.

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: Galvez Tapra Orleans
 Titulo y/o Grado Académico: Magister en Ingeniería de Sistemas
 Autores: - Escurra Colquis, Mauriño Enrique Fecha: 07-11-2018
 - Zamora Noreña, Luis Fernando

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

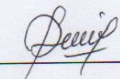
INDICADOR: INDICE DE AUSENTISMO

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.				80%	
Objetividad	Esta expresado en conducta observable.				80%	
Actualidad	Es adecuado al avance de la ciencia y tecnología.				80%	
Organización	Existe una organización lógica.				80%	
Suficiencia	Comprende los aspectos de cantidad y calidad.				80%	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.				80%	
Consistencia	Esta basado en aspectos teóricos y científicos				80%	
Coherencia	Entre los indices e indicadores.				80%	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.				80%	
Pertenencia	El instrumento es adecuado al tipo de investigación.				80%	
Promedio					80%	

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.



 Firma

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: Chumpe Aguilera, Juan B.

Título y/o Grado Académico: Magister

Autores: - Ecurra Colquis, Maurihño Enrique Fecha: 05/11/2010

- Zamora Noreña, Luis Fernando

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

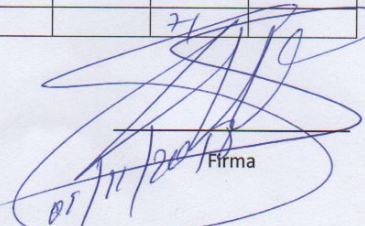
INDICADOR: INDICE DE AUSENTISMO

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.				71	
Objetividad	Esta expresado en conducta observable.				71	
Actualidad	Es adecuado al avance de la ciencia y tecnología.				71	
Organización	Existe una organización lógica.				71	
Suficiencia	Comprende los aspectos de cantidad y calidad.				71	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.				71	
Consistencia	Esta basado en aspectos teóricos y científicos				71	
Coherencia	Entre los índices e indicadores.				71	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.				71	
Pertenencia	El instrumento es adecuado al tipo de investigación.				71	
Promedio					71	

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.


 Firma

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: VARGAS HUAMAN JHONATAN ISAAK

Título y/o Grado Académico: MAGISTER

Autores: - Escurra Colquis, Mauriño Enrique Fecha: 09/11/2018

- Zamora Noreña, Luis Fernando

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.


INDICADOR: INDICE DE AUSENTISMO

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.					85%
Objetividad	Esta expresado en conducta observable.					85%
Actualidad	Es adecuado al avance de la ciencia y tecnología.					85%
Organización	Existe una organización lógica.					85%
Suficiencia	Comprende los aspectos de cantidad y calidad.					85%
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.					85%
Consistencia	Esta basado en aspectos teóricos y científicos					85%
Coherencia	Entre los índices e indicadores.					85%
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.					85%
Pertenencia	El instrumento es adecuado al tipo de investigación.					85%
Promedio						85%

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.



 Firma

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: Gálvez Tapia Orleans
 Titulo y/o Grado Académico: Magister en Ingeniería de Sistemas
 Autores: - Ecurra Colquis, Mauriño Enrique Fecha: 07-11-2018
 - Zamora Noreña, Luis Fernando

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

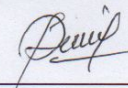
INDICADOR: INDICE DE ROTACION

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.				80%	
Objetividad	Esta expresado en conducta observable.				80%	
Actualidad	Es adecuado al avance de la ciencia y tecnología.				80%	
Organización	Existe una organización lógica.				80%	
Suficiencia	Comprende los aspectos de cantidad y calidad.				80%	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.				80%	
Consistencia	Esta basado en aspectos teóricos y científicos				80%	
Coherencia	Entre los índices e indicadores.				80%	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.				80%	
Pertenencia	El instrumento es adecuado al tipo de investigación.				80%	
Promedio					80%	

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.



Firma

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: Chumpe Acosta, J. S.

Título y/o Grado Académico: Magister

Autores: - Escurra Colquis, Maurihño Enrique Fecha: 07/11/2018

- Zamora Noreña, Luis Fernando

- TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

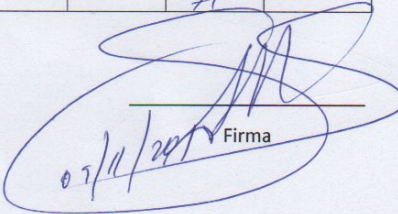
INDICADOR: INDICE DE ROTACION

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.				71	
Objetividad	Esta expresado en conducta observable.				71	
Actualidad	Es adecuado al avance de la ciencia y tecnología.				71	
Organización	Existe una organización lógica.				71	
Suficiencia	Comprende los aspectos de cantidad y calidad.				71	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.				71	
Consistencia	Esta basado en aspectos teóricos y científicos				71	
Coherencia	Entre los índices e indicadores.				71	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.				71	
Pertenencia	El instrumento es adecuado al tipo de investigación.				71	
Promedio					71	

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.


07/11/2018 Firma

ANEXO 6: Validación del instrumento

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL INDICADOR

Apellidos y Nombres del Experto: VARGAS HUAMAN JHONATAN ISAAC

Título y/o Grado Académico: MAGISTER

Autores: - Escurra Colquis, Maurihño Enrique Fecha: 09/11/2018

- Zamora Noreña, Luis Fernando

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

INDICADOR: INDICE DE ROTACION

Mediante la tabla de evaluación de experto, usted tiene la facultad para medir el indicador, mediante una serie de preguntas con puntuaciones especificada al final de la tabla.

Indicadores	Criterios	VALORACIÓN				
		Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0-20%	21-50%	51-70%	71-80%	81-100%
Claridad	Esta fórmula con lenguaje apropiado.					85%
Objetividad	Esta expresado en conducta observable.					85%
Actualidad	Es adecuado al avance de la ciencia y tecnología.					85%
Organización	Existe una organización lógica.					85%
Suficiencia	Comprende los aspectos de cantidad y calidad.					85%
Intencionalidad	Adecuado para valorar aspectos del sistema metodológicos y científico.					85%
Consistencia	Esta basado en aspectos teóricos y científicos					85%
Coherencia	Entre los índices e indicadores.					85%
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr.					85%
Pertenencia	El instrumento es adecuado al tipo de investigación.					85%
Promedio						85%

Aplicabilidad

- El instrumento puede ser aplicado.
 El instrumento no puede ser aplicado.

Firma

ANEXO 6: Validación del instrumento (Selección de la Metodología)

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL DESARROLLO DEL SOFTWARE

Apellidos y Nombres del Experto: Galvez Tapra Orleans
 Título y/o Grado Académico: Magister en Ingeniería de Sistemas
 Fecha de Evaluación: 07/11/2018

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

Mediante la tabla de evaluación de expertos, se determina la selección de la metodología de desarrollo de software en la presente investigación mediante una serie de criterios con puntuaciones específicas según el valor de la tabla de calificaciones, con ayuda de expertos en el rubro.

Muy malo (1)	Malo (2)	Regular (3)	Bueno (4)	Muy Bueno (5)
--------------	----------	-------------	-----------	---------------

Ítem	Criterios	SCRUM	XP	RUP	Observaciones
1	Marco de trabajo para desarrollar, entregar y mantener productos complejos. Desarrollar productos y mejoras.	5	3	5	—
2	Se basa en la teoría de control de procesos empírica.	5	3	5	—
3	Es liviano, fácil de entender y difícil de dominar.	5	3	4	—
4	Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y control de riesgos.	5	3	4	—
5	Se usa para productos, servicios y gestión de la organización matriz.	5	3	4	—
6	Existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas.	5	3	4	—
7	Framework adaptable, iterativo, rápido, flexible y eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto.	5	3	4	—
8	Un Sprint generalmente tienen una duración de una a seis semanas.	5	3	4	—
TOTAL		40	24	34	—

SUGERENCIAS:

Firma del experto:



ANEXO 6: Validación del instrumento (Selección de la Metodología)

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL DESARROLLO DEL SOFTWARE

Apellidos y Nombres del Experto: *Champe Agustín J. Jara*
 Título y/o Grado Académico: *Magister*
 Fecha de Evaluación: *07/11/2018*

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

Mediante la tabla de evaluación de expertos, se determina la selección de la metodología de desarrollo de software en la presente investigación mediante una serie de criterios con puntuaciones específicas según el valor de la tabla de calificaciones, con ayuda de expertos en el rubro.

Muy malo (1)	Malo (2)	Regular (3)	Bueno (4)	Muy Bueno (5)
--------------	----------	-------------	-----------	---------------

Ítem	Criterios	SCRUM	XP	RUP	Observaciones
1	Marco de trabajo para desarrollar, entregar y mantener productos complejos. Desarrollar productos y mejoras.	4	2	5	—
2	Se basa en la teoría de control de procesos empírica.	4	3	4	—
3	Es liviano, fácil de entender y difícil de dominar.	4	4	4	—
4	Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y control de riesgos.	5	3	5	—
5	Se usa para productos, servicios y gestión de la organización matriz.	5	3	5	—
6	Existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas.	5	3	3	—
7	Framework adaptable, iterativo, rápido, flexible y eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto.	5	3	3	—
8	Un Sprint generalmente tienen una duración de una a seis semanas.	5	3	3	—
TOTAL		37	24	32	—

SUGERENCIAS:

Firma del experto:

[Firma manuscrita]
07/11/2018

ANEXO 6: Validación del instrumento (Selección de la Metodología)

TABLA DE EVALUACIÓN DE EXPERTOS PARA EL DESARROLLO DEL SOFTWARE

Apellidos y Nombres del Experto: ... VARGAS HUAMAN JHONATAN ISAAC
 Título y/o Grado Académico: ... MAGISTER
 Fecha de Evaluación: ... 09/11/2018

TESIS

APLICACIÓN MÓVIL PARA EL CONTROL DE PERSONAL EN LA EMPRESA CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

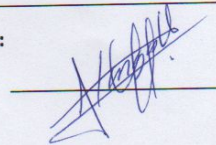
Mediante la tabla de evaluación de expertos, se determina la selección de la metodología de desarrollo de software en la presente investigación mediante una serie de criterios con puntuaciones específicas según el valor de la tabla de calificaciones, con ayuda de expertos en el rubro.

Muy malo (1)	Malo (2)	Regular (3)	Bueno (4)	Muy Bueno (5)
--------------	----------	-------------	-----------	---------------

Ítem	Criterios	SCRUM	XP	RUP	Observaciones
1	Marco de trabajo para desarrollar, entregar y mantener productos complejos. Desarrollar productos y mejoras.	5	4	3	—
2	Se basa en la teoría de control de procesos empírica.	5	3	3	—
3	Es liviano, fácil de entender y difícil de dominar.	5	4	3	—
4	Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y control de riesgos.	5	3	3	—
5	Se usa para productos, servicios y gestión de la organización matriz.	5	5	5	—
6	Existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas.	5	3	3	—
7	Framework adaptable, iterativo, rápido, flexible y eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto.	5	3	3	—
8	Un Sprint generalmente tienen una duración de una a seis semanas.	5	3	3	—
TOTAL		40	28	26	—

SUGERENCIAS:

Firma del experto:



ANEXO 7: Carta de aprobación de la empresa

Carta N° 001-2018-Empresa Construcciones & Acabados Zamora S.A.C.

Lima, 14 de septiembre de 2018

Dra. Lily Doris Salazar Chávez
Directora de la Escuela Profesional de Ingeniería de Sistemas
Universidad César Vallejo

Asunto: Aprobación para realización de la aplicación móvil


Presente. -

De mi mayor consideración:

Mediante la presente es grato dirigirme a usted a fin de saludarle muy cordialmente a nombre de la empresa Construcciones & Acabados Zamora S.A.C., y a su vez, informar la aceptación respectiva para la realización del proyecto de investigación: "Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.", a los estudiantes Escura Colquis Maurihño Enrique y a Zamora Noreña Luis Fernando del IX ciclo de la escuela profesional de Ingeniería de Sistemas, el cual servirá de tesis. Por último, manifestarle que depositamos nuestra absoluta confianza en el desarrollo del proyecto antes mencionado.

Agradeciendo su atención a la presente, es propia la oportunidad para expresarle las muestras de mi consideración y estima.

Atentamente

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.


JAVIER ZAMORA
GERENTE GENERAL

ANEXO 7: Acta de implementación de la Aplicación móvil en la empresa Construcciones & Acabados Zamora S.A.C.

Carta N° 002-2019-Empresa Construcciones & Acabados Zamora S.A.C.

Lima, 19 de noviembre de 2019

Dra. Lily Doris Salazar Chávez
Directora de la Escuela Profesional de Ingeniería de Sistemas
Universidad César Vallejo

Asunto: Entrega de la aplicación móvil

Presente. -

De mi mayor consideración:

Mediante la presente es grato dirigirme a usted a fin de saludarle muy cordialmente a nombre de la empresa Construcciones & Acabados Zamora S.A.C., y a su vez, informar la aplicación móvil para el control de personal que fue presentada como propuesta anteriormente ya se encuentra en producción desde el día 18 de noviembre del presente año. La cual ha sido muy productiva y novedosa a la vez y que nos servirá de mucha ayuda en nuestra empresa.

Agradeciendo su atención a la presente, es propia la oportunidad para expresarle las muestras de mi consideración y estima.

Atentamente

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.


JAVIER ZAMORA
GERENTE GENERAL

ANEXO 8: Entrevista en la empresa Construcciones & Acabados Zamora S.A.C.



CONSTRUCCIONES & ACABADOS ZAMORA

Nombre del Entrevistado	Sr. Javier Antonio Murillo
Cargo	Representante Legal
Fecha	14/09/2018

- 1) **¿Cuál es la misión y visión de la empresa?**
Visión: Nuestra visión es ser una empresa líder en el mercado nacional con proyección internacional en servicios de ingeniería, construcción de casas y edificios, sustentada en el trabajo responsable, dedicado e innovador.

Misión: Somos una empresa constructora dedicada a crear, desarrollar y comercializar proyectos de construcción, nuestra misión es trabajar con el compromiso de satisfacer las necesidades de nuestros clientes generando bienestar y calidad de vida.
- 2) **¿Qué problemática existe en el control de personal?**
El problema que hemos presentado es que no tenemos el control eficiente del personal de campo, para así saber qué es lo que están haciendo en la hora de trabajo, muchas veces los obreros faltan constantemente y no tenemos un detalle de ello, utilizamos el control biométrico, pero genera mucho gasto, ya que se tiene que instalar en cada punto de obra. Además, no hay un acercamiento a los obreros con el área de recursos humanos.
- 3) **¿De qué manera se lleva a cabo el control de personal?**
Actualmente contamos con un sistema biométrico para el registro de las asistencias, del personal y apuntes manuales para el control de las vacaciones.
- 4) **¿Qué requerimientos mencionaría si existiera un sistema de control de personal?**
1. Planificador de turnos: fijos, rotativos, nocturnos.
 2. Gestión de vacaciones, ausentismo y desvinculaciones.
 3. Administración de dispositivos.
 4. Consolidación de horas (atrasos injustificados, permisos, marcaciones erradas).
 5. Administración de usuarios (enrolar personas, autorizar permisos, crear grupos, asignar roles, autorizar lugares de marcación).
 6. Administración de perfiles (administradores de cuenta, supervisores, supervisores de solo lectura, usuarios).



CONSTRUCCIONES & ACABADOS ZAMORA

7. Reportes y estadísticas exactas y en tiempo real que le permitan tomar medidas correctivas a tiempo y mantenerse competitivo en el mercado.

5) **¿Utilizan un sistema biométrico para el control de personal?**

Si, utilizamos un control biométrico para las asistencias del personal

6) **¿De qué manera miden el índice de ausentismo?**

En base al reporte de asistencias que nos brinda el control biométrico, contamos las ausencias de los trabajadores, pero no nos permite saber el motivo.

7) **¿De qué manera miden el índice de rotación en sus trabajadores?**

Contamos con un registro en excel de todo el personal que trabaja, pero no tenemos un control de las desvinculaciones de nuestros empleados.

8) **¿Cuál es la cantidad de empleados que tienen?**

200 empleados

9) **¿Le gustaría llevar el control de personal con una aplicación móvil?**

Sería estupendo, esto nos permitiría tener un mejor control de ellos.

10) **¿Cuántos proyectos de obra tienen al mes? ¿En qué consisten?**

Al mes contamos con 10 obras y consisten en construcciones de casas, edificios, remodelaciones.

CONSTRUCCIONES & ACABADOS ZAMORA S.A.S.

JAVIER ZAMORA
GERENTE GENERAL

ANEXO 9: Desarrollo de la metodología para la Aplicación móvil

En las siguientes páginas de la tesis se aplica la metodología Scrum para la app móvil a implementar en la organización Construcciones & Acabaos Zamora S.A.C.

Índice de tablas de la metodología

Tabla 1: Equipo Scrum (Scrum Team)

Tabla 2: Historia de Usuario 1

Tabla 3: Historia de Usuario 2

Tabla 4: Historia de Usuario 3

Tabla 5: Historia de Usuario 4

Tabla 6: Historia de Usuario 5

Tabla 7: Historia de Usuario 6

Tabla 8: Historia de Usuario 7

Tabla 9: Historia de Usuario 8

Tabla 10: Historia de Usuario 9

Tabla 11: Historia de Usuario 10

Tabla 12: Historia de Usuario 11

Tabla 13: Historia de Usuario 12

Tabla 14: Historia de Usuario 13

Tabla 15: Historia de Usuario 14

Tabla 16: Historia de Usuario 15

Tabla 17: Historia de Usuario 16

Tabla 18: Historia de Usuario 17

Tabla 19: Historia de Usuario 18

Tabla 20: Historia de Usuario 19

Tabla 21: Historia de Usuario 20

Tabla 22: Historia de Usuario 21

Tabla 23: Historia de Usuario 22

Tabla 24: Historia de Usuario 23

Tabla 25: Historia de Usuario 24

Tabla 26: Historia de Usuario 25

Tabla 27: Historia de Usuario 26

Tabla 28: Historia de Usuario 27

Tabla 29: Historia de Usuario 28

Tabla 30: Historia de Usuario 29

Tabla 31: Historia de Usuario 30

Tabla 32: Historia de Usuario 31

Tabla 33: Historia de Usuario 32

Tabla 34: Historia de Usuario 33

Tabla 35: Historia de Usuario 34

Tabla 36: Historia de Usuario 35

Tabla 37: Historia de Usuario 36

Tabla 38: Historia de Usuario 37

Tabla 39: Historia de Usuario 38

Tabla 40: Historia de Usuario 39

Tabla 41: Historia de Usuario 40

Tabla 42: Historia de Usuario 41

Tabla 43: Historia de Usuario 42

Tabla 44: Historia de Usuario 43

Tabla 45: Historia de Usuario 44

Tabla 46: Historia de Usuario 45

Tabla 47: Historia de Usuario 46

Tabla 48: Historia de Usuario 47

Tabla 49: Historia de Usuario 48

Tabla 50: Historia de Usuario 49

Tabla 51: Historia de Usuario 50

Tabla 52: Historia de Usuario 51

Tabla 53: Historia de Usuario 52

Tabla 54: Historia de Usuario 53

Tabla 55: Historia de Usuario 54

Tabla 56: Historia de Usuario 55

Tabla 57: Historia de Usuario 56

Tabla 58: Historia de Usuario 57

Tabla 59: Historia de Usuario 58

Tabla 60: Historia de Usuario 59

Tabla 61: Historia de Usuario 60

Tabla 62: Historia de Usuario 61

Tabla 63: Historia de Usuario 62

Tabla 64: Historia de Usuario 63

Tabla 65: Historia de Usuario 64

Tabla 66: Historia de Usuario 65

Tabla 67: Pila de Producto Inicial

Tabla 68: Pila de Producto Organizada por prioridad

Tabla 69: Requerimientos No Funcionales

Tabla 70: Lista de Sprint

Tabla 71: Sprint 1

Tabla 72: Resumen del Sprint 1

Table 73: Sprint 2

Tabla 74: Resumen del Sprint 2

Tabla 75: Sprint 3

Tabla 76: Resumen del Sprint 3

Tabla 77: Sprint 4

Tabla 78: Resumen del Sprint 4

Índice de figuras de la metodología

- Figura 1: Planificación Sprint 1
- Figura 2: Diagrama lógico de la base de datos RF1
- Figura 3: Diagrama físico de la base de datos RF1
- Figura 4: Prototipo de RF1
- Figura 5: Interface de RF1
- Figura 6: Código de RF1
- Figura 7: Diagrama lógico de la base de datos RF2
- Figura 8: Diagrama físico de la base de datos RF2
- Figura 9: Prototipo de RF2
- Figura 10: Interface de RF2
- Figura 11: Código de RF2
- Figura 12: Diagrama lógico de la base de datos RF3
- Figura 13: Diagrama físico de la base de datos RF3
- Figura 14: Prototipo de RF3
- Figura 15: Interface de RF3
- Figura 16: Código de RF3
- Figura 17: Burndown Chart Sprint 1
- Figura 18: Acta de reunión de entrega del Sprint 1
- Figura 19: Planificación Sprint 2
- Figura 20: Diagrama lógico de la base de datos RF4
- Figura 21: Diagrama físico de la base de datos RF4
- Figura 22: Prototipo de RF4
- Figura 23: Interface de RF4

Figura 24: Código de RF4

Figura 25: Diagrama lógico de la base de datos RF5

Figura 26: Diagrama físico de la base de datos RF5

Figura 27: Prototipo de RF5

Figura 28: Interface de RF5

Figura 29: Código de RF5

Figura 30: Diagrama lógico de la base de datos RF6

Figura 31: Diagrama físico de la base de datos RF6

Figura 32: Prototipo de RF6

Figura 33: Interface de RF6

Figura 34: Código de RF6

Figura 35: Burndown Chart Sprint 2

Figura 36: Acta de reunión de entrega del Sprint 2

Figura 37: Planificación Sprint 3

Figura 38: Diagrama lógico de la base de datos RF7

Figura 39: Diagrama físico de la base de datos RF7

Figura 40: Prototipo de RF7

Figura 41: Interface de RF7

Figura 42: Código de RF7

Figura 43: Diagrama lógico de la base de datos RF8

Figura 44: Diagrama físico de la base de datos RF8

Figura 45: Prototipo de RF8

Figura 46: Interface de RF8

Figura 47: Código de RF8

Figura 48: Diagrama lógico de la base de datos RF9

Figura 49: Diagrama físico de la base de datos RF9

Figura 50: Prototipo de RF9

Figura 51: Interface de RF9

Figura 52: Código de RF9

Figura 53: Diagrama lógico de la base de datos RF10

Figura 54: Diagrama físico de la base de datos RF10

Figura 55: Prototipo de RF10

Figura 56: Interface de RF10

Figura 57: Código de RF10

Figura 58: Diagrama lógico de la base de datos RF11

Figura 59: Diagrama físico de la base de datos RF11

Figura 60: Prototipo de RF11

Figura 61: Interface de RF11

Figura 62: Código de RF11

Figura 63: Burndown Chart Sprint 3

Figura 64: Acta de reunión de entrega del Sprint 3

Figura 65: Planificación Sprint 4

Figura 66: Diagrama lógico de la base de datos RF12

Figura 67: Diagrama físico de la base de datos RF12

Figura 68: Prototipo de RF12

Figura 69: Interface de RF12

Figura 70: Código de RF12

Figura 71: Diagrama lógico de la base de datos RF13

Figura 72: Diagrama físico de la base de datos RF13

Figura 73: Prototipo de RF13

Figura 74: Interface de RF13

Figura 75: Código de RF13

Figura 76: Diagrama lógico de la base de datos RF14

Figura 77: Diagrama físico de la base de datos RF14

Figura 78: Prototipo de RF14

Figura 79: Interface de RF14

Figura 80: Código de RF14

Figura 81: Diagrama lógico de la base de datos RF15

Figura 82: Diagrama físico de la base de datos RF15

Figura 83: Prototipo de RF15

Figura 84: Interface de RF15

Figura 85: Código de RF15

Figura 86: Diagrama lógico de la base de datos RF16

Figura 87: Diagrama físico de la base de datos RF16

Figura 88: Prototipo de RF16

Figura 89: Interface de RF16

Figura 90: Código de RF16

Figura 91: Diagrama lógico de la base de datos RF17

Figura 92: Diagrama físico de la base de datos RF17

Figura 93: Prototipo de RF17

Figura 94: Interface de RF17

Figura 95: Código de RF17

Figura 96: Diagrama lógico de la base de datos RF18

Figura 97: Diagrama físico de la base de datos RF18

Figura 98: Prototipo de RF18

Figura 99: Interface de RF18

Figura 100: Código de RF18

Figura 101: Diagrama lógico de la base de datos RF19

Figura 102: Diagrama físico de la base de datos RF19

Figura 103: Prototipo de RF19

Figura 104: Interface de RF19

Figura 105: Código de RF19

Figura 106: Diagrama lógico de la base de datos RF20

Figura 107: Diagrama físico de la base de datos RF20

Figura 108: Prototipo de RF20

Figura 109: Interface de RF20

Figura 110: Código de RF20

Figura 111: Diagrama lógico de la base de datos RF21

Figura 112: Diagrama físico de la base de datos RF21

Figura 113: Prototipo de RF21

Figura 114: Interface de RF21

Figura 115: Código de RF21

Figura 116: Diagrama lógico de la base de datos RF22

Figura 117: Diagrama físico de la base de datos RF22

Figura 118: Prototipo de RF22

Figura 119: Interface de RF22

Figura 120: Código de RF22

Figura 121: Diagrama lógico de la base de datos RF23

Figura 122: Diagrama físico de la base de datos RF23

Figura 123: Prototipo de RF23

Figura 124: Interface de RF23

Figura 125: Código de RF23

Figura 126: Diagrama lógico de la base de datos RF24

Figura 127: Diagrama físico de la base de datos RF24

Figura 128: Prototipo de RF24

Figura 129: Interface de RF24

Figura 130: Código de RF24

Figura 131: Diagrama lógico de la base de datos RF25

Figura 132: Diagrama físico de la base de datos RF25

Figura 133: Prototipo de RF25

Figura 134: Interface de RF25

Figura 135: Código de RF25

Figura 136: Diagrama lógico de la base de datos RF26

Figura 137: Diagrama físico de la base de datos RF26

Figura 138: Prototipo de RF26

Figura 139: Interface de RF26

Figura 140: Código de RF26

Figura 141: Diagrama lógico de la base de datos RF27

Figura 142: Diagrama físico de la base de datos RF27

Figura 143: Prototipo de RF27

Figura 144: Interface de RF27

Figura 145: Código de RF27

Figura 146: Diagrama lógico de la base de datos RF28

Figura 147: Diagrama físico de la base de datos RF28

Figura 148: Prototipo de RF28

Figura 149: Interface de RF28

Figura 150: Código de RF28

Figura 151: Diagrama lógico de la base de datos RF29

Figura 152: Diagrama físico de la base de datos RF29

Figura 153: Prototipo de RF29

Figura 154: Interface de RF29

Figura 155: Código de RF29

Figura 156: Diagrama lógico de la base de datos RF30

Figura 157: Diagrama físico de la base de datos RF30

Figura 158: Prototipo de RF30

Figura 159: Interface de RF30

Figura 160: Código de RF30

Figura 161: Diagrama lógico de la base de datos RF31

Figura 162: Diagrama físico de la base de datos RF31

Figura 163: Prototipo de RF31

Figura 164: Interface de RF31

Figura 165: Código de RF31

Figura 166: Diagrama lógico de la base de datos RF32

Figura 167: Diagrama físico de la base de datos RF32

Figura 168: Prototipo de RF32

Figura 169: Interface de RF32

Figura 170: Código de RF32

Figura 171: Diagrama lógico de la base de datos RF33

Figura 172: Diagrama físico de la base de datos RF33

Figura 173: Prototipo de RF33

Figura 174: Interface de RF33

Figura 175: Código de RF33

Figura 176: Diagrama lógico de la base de datos RF34

Figura 177: Diagrama físico de la base de datos RF34

Figura 178: Prototipo de RF34

Figura 179: Interface de RF34

Figura 180: Código de RF34

Figura 181: Diagrama lógico de la base de datos RF35

Figura 182: Diagrama físico de la base de datos RF35

Figura 183: Prototipo de RF35

Figura 184: Interface de RF35

Figura 185: Código de RF35

Figura 186: Diagrama lógico de la base de datos RF36

Figura 187: Diagrama físico de la base de datos RF36

Figura 188: Prototipo de RF36

Figura 189: Interface de RF36

Figura 190: Código de RF36

Figura 191: Diagrama lógico de la base de datos RF37

Figura 192: Diagrama físico de la base de datos RF37

Figura 193: Prototipo de RF37

Figura 194: Interface de RF37

Figura 195: Código de RF37

Figura 196: Diagrama lógico de la base de datos RF38

Figura 197: Diagrama físico de la base de datos RF38

Figura 198: Prototipo de RF38

Figura 199: Interface de RF38

Figura 200: Código de RF38

Figura 201: Diagrama lógico de la base de datos RF39

Figura 202: Diagrama físico de la base de datos RF39

Figura 203: Prototipo de RF39

Figura 204: Interface de RF39

Figura 205: Código de RF39

Figura 206: Diagrama lógico de la base de datos RF40

Figura 207: Diagrama físico de la base de datos RF40

Figura 208: Prototipo de RF40

Figura 209: Interface de RF40

Figura 210: Código de RF40

Figura 211: Diagrama lógico de la base de datos RF41

Figura 212: Diagrama físico de la base de datos RF41

Figura 213: Prototipo de RF41

Figura 214: Interface de RF41

Figura 215: Código de RF41

Figura 216: Diagrama lógico de la base de datos RF42

Figura 217: Diagrama físico de la base de datos RF42

Figura 218: Prototipo de RF42

Figura 219: Interface de RF42

Figura 220: Código de RF42

Figura 221: Diagrama lógico de la base de datos RF43

Figura 222: Diagrama físico de la base de datos RF43

Figura 223: Prototipo de RF43

Figura 224: Interface de RF43

Figura 225: Código de RF43

Figura 226: Diagrama lógico de la base de datos RF44

Figura 227: Diagrama físico de la base de datos RF44

Figura 228: Prototipo de RF44

Figura 229: Interface de RF44

Figura 230: Código de RF44

Figura 231: Diagrama lógico de la base de datos RF45

Figura 232: Diagrama físico de la base de datos RF45

Figura 233: Prototipo de RF45

Figura 234: Interface de RF45

Figura 235: Código de RF45

Figura 236: Diagrama lógico de la base de datos RF46

Figura 237: Diagrama físico de la base de datos RF46

Figura 238: Prototipo de RF46

Figura 239: Interface de RF46

Figura 240: Código de RF46

Figura 241: Diagrama lógico de la base de datos RF47

Figura 242: Diagrama físico de la base de datos RF47

Figura 243: Prototipo de RF47

Figura 244: Interface de RF47

Figura 245: Código de RF47

Figura 246: Diagrama lógico de la base de datos RF48

Figura 247: Diagrama físico de la base de datos RF48

Figura 248: Prototipo de RF48

Figura 249: Interface de RF48

Figura 250: Código de RF48

Figura 251: Diagrama lógico de la base de datos RF49

Figura 252: Diagrama físico de la base de datos RF49

Figura 253: Prototipo de RF49

Figura 254: Interface de RF49

Figura 255: Código de RF49

Figura 256: Diagrama lógico de la base de datos RF50

Figura 257: Diagrama físico de la base de datos RF50

Figura 258: Prototipo de RF50

Figura 259: Interface de RF50

Figura 260: Código de RF50

Figura 261: Diagrama lógico de la base de datos RF51

Figura 262: Diagrama físico de la base de datos RF51

Figura 263: Prototipo de RF51

Figura 264: Interface de RF51

Figura 265: Código de RF51

Figura 266: Diagrama lógico de la base de datos RF52

Figura 267: Diagrama físico de la base de datos RF52

Figura 268: Prototipo de RF52

Figura 269: Interface de RF52

Figura 270: Código de RF52

Figura 271: Diagrama lógico de la base de datos RF53

Figura 272: Diagrama físico de la base de datos RF53

Figura 273: Prototipo de RF53

Figura 274: Interface de RF53

Figura 275: Código de RF53

Figura 276: Diagrama lógico de la base de datos RF54

Figura 277: Diagrama físico de la base de datos RF54

Figura 278: Prototipo de RF54

Figura 279: Interface de RF54

Figura 280: Código de RF54

Figura 281: Diagrama lógico de la base de datos RF55

Figura 282: Diagrama físico de la base de datos RF55

Figura 283: Prototipo de RF55

Figura 284: Interface de RF55

Figura 285: Código de RF55

Figura 286: Diagrama lógico de la base de datos RF56

Figura 287: Diagrama físico de la base de datos RF56

Figura 288: Prototipo de RF56

Figura 289: Interface de RF56

Figura 290: Código de RF56

Figura 291: Diagrama lógico de la base de datos RF57

Figura 292: Diagrama físico de la base de datos RF57

Figura 293: Prototipo de RF57

Figura 294: Interface de RF57

Figura 295: Código de RF57

Figura 296: Diagrama lógico de la base de datos RF58

Figura 297: Diagrama físico de la base de datos RF58

Figura 298: Prototipo de RF58

Figura 299: Interface de RF58

Figura 300: Código de RF58

Figura 301: Diagrama lógico de la base de datos RF59

Figura 302: Diagrama físico de la base de datos RF59

Figura 303: Prototipo de RF59

Figura 304: Interface de RF59

Figura 305: Código de RF59

Figura 306: Diagrama lógico de la base de datos RF60

Figura 307: Diagrama físico de la base de datos RF60

Figura 308: Prototipo de RF60

Figura 309: Interface de RF60

Figura 310: Código de RF60

Figura 311: Diagrama lógico de la base de datos RF61

Figura 312: Diagrama físico de la base de datos RF61

Figura 313: Prototipo de RF61

Figura 314: Interface de RF61

Figura 315: Código de RF61

Figura 316: Diagrama lógico de la base de datos RF62

Figura 317: Diagrama físico de la base de datos RF62

Figura 318: Prototipo de RF62

Figura 319: Interface de RF62

Figura 320: Código de RF62

Figura 321: Diagrama lógico de la base de datos RF63

Figura 322: Diagrama físico de la base de datos RF63

Figura 323: Prototipo de RF63

Figura 324: Interface de RF63

Figura 325: Código de RF63

Figura 326: Diagrama lógico de la base de datos RF64

Figura 327: Diagrama físico de la base de datos RF64

Figura 328: Prototipo de RF64

Figura 329: Interface de RF64

Figura 330: Código de RF64

Figura 331: Diagrama lógico de la base de datos RF65

Figura 332: Diagrama físico de la base de datos RF65

Figura 333: Prototipo de RF65

Figura 334: Interface de RF65

Figura 335: Código de RF65

Figura 336: Burndown Chart Sprint 4

Figura 337: Acta de reunión de entrega del Sprint 4

Figura 338: Diagrama lógico de la base de datos – General

Figura 339: Diagrama físico de la base de datos – General

Figura 340: Acta de reunión 1

Figura 341: Acta de reunión 2

Figura 342: Acta de reunión 3

Figura 343: Acta de reunión 4

Figura 344: Acta de reunión 5

Figura 345: Tablero Scrum Inicial

Figura 346: Tablero Scrum con Actividades

Figura 347: Herramienta GitKraken

Figura 348: Commit del Repositorio Bitbucket de la aplicación

Figura 349: Arquitectura Cliente – Servidor

Figura 350: Interoperabilidad de aplicaciones híbridas

Figura 351: Acta de Pruebas

Figura 352: Acta de Pase a Producción

Figura 353: Construcción de casa

Equipo Scrum

Tabla 1: *Equipo Scrum (Scrum Team)*

Persona	Cargo	Contacto	Rol
Javier Zamora	Product Owner	javierzamora@gmail.com	Product Owner
Lorenzo Palacios	Scrum Master	lorenzo_palacios@gmail.com	Scrum Master
Luis Zamora	Programador	luiszamoranorena@gmail.com	Team Member
Maurihño Ecurra	Analista	mescurrac123@gmail.com	Team Member
Julio Sosa	Calidad	julio_sosa@gmail.com	Team Member

Fuente: Elaboración propia

Aquí está todo el equipo Scrum por su cargo, por su nombre, rol y contacto.

Historias de Usuario

Aquí se va a presentar cada Historia de Usuario de la app móvil, con un ID, con cada nombre, con una descripción, una estimación, una prioridad y con sus propias restricciones o criterios de aceptación.

Tabla 2: *Historia de Usuario 1*

ID: H1	Historia de Usuario 1
Nombre	Acceder a la aplicación
Prioridad	1
Estimación	3
Descripción: Para que puedan hacer uso de la aplicación deberán haber adquirido un usuario y contraseña.	
Restricciones: Pueden entrar a la app el usuario administrador, obrero y recursos humanos. Validar los campos de usuarios y contraseña. Mostrar confirmación de “Acceso Satisfactorio”. Flujos erróneos: Si en caso el usuario del empleado no existe, se mostrará un mensaje de “Usuario no existe”. Si el empleado esta desvinculado, se mostrará un mensaje “Este empleado ha sido desvinculado, por lo tanto, no tiene acceso”.	

Fuente: Elaboración propia

Tabla 3: *Historia de Usuario 2*

ID: H2	Historia de Usuario 2
Nombre	Registrar asistencia
Prioridad	1
Estimación	3
Descripción: La aplicación permitirá al usuario obrero y administrador registrar sus asistencias.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas registrar una asistencia?”, y muestra un mensaje “La asistencia ha sido registrada correctamente” después de registrar una asistencia. Flujo erróneo: Si el empleado nunca ha tenido una obra asignada, se mostrará un mensaje “No puede marcar una asistencia, porque todavía no tiene una obra asignada”. Si el empleado tiene una obra asignada que ya finalizó, se mostrará un mensaje “No puede marcar una asistencia, debe asignarle nuevamente una obra”.	

Fuente: Elaboración propia

Tabla 4: *Historia de Usuario 3*

ID: H3	Historia de Usuario 3
Nombre	Cerrar sesión
Prioridad	1
Estimación	1
Descripción: La aplicación permitirá a los usuarios cerrar sesión.	
Restricciones: Mostrar una alerta para confirmar si desea cerrar sesión el usuario.	

Fuente: Elaboración propia

Tabla 5: *Historia de Usuario 4*

ID: H4	Historia de Usuario 4
Nombre	Crear obra
Prioridad	2
Estimación	3
Descripción: La aplicación permitirá crear obras de trabajo.	
Restricciones: Validar campos de formulario. Mostrar una alerta para confirmar si desea crear la obra y muestra mensaje “Obra creada correctamente”, después de crear la obra. Flujo erróneo: Si la fecha de inicio es mayor a la fecha de fin de la obra, se mostrará un mensaje “La fecha de inicio no puede ser mayor a la fecha de fin”. Si una obra a crear tiene el mismo nombre de una ya creada, se mostrará un mensaje “No se puede crear otra obra con el mismo nombre”.	

Fuente: Elaboración propia

Tabla 6: *Historia de Usuario 5*

ID: H5	Historia de Usuario 5
Nombre	Asignar obra
Prioridad	2
Estimación	3
Descripción: La aplicación permitirá asignar una obra a cada obrero.	
Restricciones: Validar campos de formulario. Antes de asignar una obra, muestra una alerta de confirmación “¿Desea asignar una obra?”, y muestra un mensaje “Obra asignada correctamente” después de asignar una obra. Flujo erróneo: Si el obrero tiene asignada una obra que aún no termina, se mostrará un mensaje “El obrero tiene una obra asignada que aún no finaliza”. Si todas las obras han culminado, se muestra un mensaje en pantalla “No hay obras a asignar” y no se podrá asignar una obra a nadie hasta que se extienda la fecha de fin de la obra o se crea una nueva.	

Fuente: Elaboración propia

Tabla 7: *Historia de Usuario 6*

ID: H6	Historia de Usuario 6
Nombre	Actualizar obra
Prioridad	2
Estimación	3
Descripción: La aplicación permitirá actualizar los datos de cada obra.	
Restricciones: Validar campos de formulario. Antes de actualizar una obra, muestra una alerta de confirmación “¿Desea actualizar la obra?” y muestra un mensaje “Obra actualizada correctamente” después de actualizar una obra correctamente.	

Fuente: Elaboración propia

Tabla 8: *Historia de Usuario 7*

ID: H7	Historia de Usuario 7
Nombre	Eliminar obra
Prioridad	3
Estimación	3
Descripción: La aplicación permitirá eliminar una obra.	
Restricciones: Solamente se puede eliminar las obras que no tienen empleados asignados a esta. Mostrar alerta de confirmación “¿Desea eliminar la obra?” antes de eliminar la obra. Mostrar mensaje “Obra eliminada correctamente”, después de eliminar la obra. Flujo erróneo: Si la obra tiene obreros asignados, muestra un mensaje “No se puede eliminar la obra porque tiene obreros asignados” y la obra no será eliminada.	

Fuente: Elaboración propia

Tabla 9: *Historia de Usuario 8*

ID: H8	Historia de Usuario 8
Nombre	Registrar empleado
Prioridad	3
Estimación	3
Descripción: La aplicación permitirá registrar un empleado, administrador, obrero o recursos humanos.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea crear un empleado?”. Mostrar mensaje “Empleado registrado correctamente”, después de registrar un empleado con éxito. Flujos erróneos: Si el dni ya existe se mostrará un mensaje: “Este dni ya existe, por favor escriba otro dni”. Si el usuario ya existe se mostrará un mensaje: “El usuario ya existe, por favor escriba otro usuario”, cuando se ingresa un usuario existente. Si el email ya existe se mostrará un mensaje: “El email ya existe, por favor escriba otro email”.	

Fuente: Elaboración propia

Tabla 10: *Historia de Usuario 9*

ID: H9	Historia de Usuario 9
Nombre	Actualizar empleado
Prioridad	3
Estimación	3
Descripción: La aplicación permitirá actualizar un empleado.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar un empleado?”. Muestra mensaje “Empleado actualizado correctamente”, después de actualizar un empleado con éxito. Flujos erróneos: Si el dni ya existe se mostrará un mensaje: “Este dni ya existe, por favor escriba otro dni”. Si el usuario ya existe se mostrará un mensaje: “El usuario ya existe, por favor escriba otro usuario”, cuando se ingresa un usuario existente. Si el email ya existe se mostrará un mensaje: “El email ya existe, por favor escriba otro email”.	

Fuente: Elaboración propia

Tabla 11: *Historia de Usuario 10*

ID: H10	Historia de Usuario 10
Nombre	Eliminar empleado
Prioridad	3
Estimación	3
Descripción: La aplicación permitirá eliminar un empleado.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar un empleado?”. Mostrar mensaje “Empleado eliminado correctamente”, después de eliminar un empleado con éxito.	

Fuente: Elaboración propia

Tabla 12: *Historia de Usuario 11*

ID: H11	Historia de Usuario 11
Nombre	Desvincular empleado
Prioridad	3
Estimación	3
Descripción: Esta funcionalidad permitirá colocar en un estado de desvinculado a los empleados.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas desvincular este empleado?”. Mostrar mensaje “Empleado Desvinculado correctamente”, después de desvincular el empleado con éxito. Flujo erróneo: Si no hay empleados desvinculados, se mostrará un mensaje en pantalla “No hay empleados desvinculados”.	

Fuente: Elaboración propia

Tabla 13: *Historia de Usuario 12*

ID: H12	Historia de Usuario 12
Nombre	Consultar marcaciones
Prioridad	4
Estimación	5
Descripción: El sistema proporcionará a los usuarios información de las marcaciones de asistencia del obrero.	
Restricciones: Validar campos. Seleccionar empleado. Seleccionar fecha inicio y fecha fin. Al pulsar el botón Consultar Marcaciones se visualizará un reporte en PDF de las marcaciones de los empleados. Los flujos erróneos son: Si no existe ninguna asistencia en el rango de fechas seleccionado saldrá un mensaje “No existe ninguna asistencia en el rango de fecha seleccionado”. Si la fecha de inicio es mayor a la fecha de fin, se mostrará un mensaje “La fecha de fin no puede ser menor a la fecha de inicio”. Si el obrero no tiene ninguna obra asignada, se mostrará un mensaje “No puede consultar una marcación, porque no tiene una obra asignada”.	

Fuente: Elaboración propia

Tabla 14: *Historia de Usuario 13*

ID: H13	Historia de Usuario 13
Nombre	Visualizar reporte de ausentismo
Prioridad	4
Estimación	8
Descripción: Este módulo permitirá visualizar un reporte de ausentismo de todos los empleados que están en una obra.	
Restricciones: Validar campos. Seleccionar obra, se carga automáticamente la fecha inicio y fecha fin. Al pulsar el botón Reporte de Ausentismo se visualizará un reporte en PDF de las ausencias de los empleados. El flujo erróneo es: Si la obra no tiene empleados saldrá un mensaje “No existen empleados en esta obra”.	

Fuente: Elaboración propia

Tabla 15: *Historia de Usuario 14*

ID: H14	Historia de Usuario 14
Nombre	Visualizar reporte de rotación
Prioridad	4
Estimación	8
Descripción: Este módulo permitirá visualizar un reporte de todos los empleados por obra que se hayan desvinculado de la empresa.	
Restricciones: Validar campos. Seleccionar obra, se carga automáticamente la fecha inicio y fecha fin. Al pulsar el botón Reporte de Rotación se visualizará un reporte en PDF de las rotaciones o desvinculaciones de los empleados. El flujo erróneo es: Si la obra no tiene empleados saldrá un mensaje “No existen empleados en esta obra”.	

Fuente: Elaboración propia

Tabla 16: *Historia de Usuario 15*

ID: H15	Historia de Usuario 15
Nombre	Llamar a recursos humanos
Prioridad	4
Estimación	1
Descripción: La aplicación permitirá al obrero y administrador comunicarse con al área de recursos humanos para hacerle consultas.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 17: *Historia de Usuario 16*

ID: H16	Historia de Usuario 16
Nombre	Visualizar obra asignada
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá mostrar al usuario obrero en que obra está asignado.	
Restricciones: Para el flujo normal sale los datos de la obra asignada del obrero. Flujo erróneo: Si el obrero no tiene ninguna obra asignada actualmente, se muestra un mensaje en pantalla “Usted no tiene ninguna obra asignada”.	

Fuente: Elaboración propia

Tabla 18: *Historia de Usuario 17*

ID: H17	Historia de Usuario 17
Nombre	Listar empleados
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá mostrar al usuario administrador un listado de los empleados: obreros, recursos humanos y administrador.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 19: *Historia de Usuario 18*

ID: H18	Historia de Usuario 18
Nombre	Listar obras
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá mostrar un listado de todas las obras al usuario administrador.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 20: *Historia de Usuario 19*

ID: H19	Historia de Usuario 19
Nombre	Ver detalle de obra
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador ver el detalle de cada obra seleccionada.	
Restricciones: Seleccionar una obra previamente.	

Fuente: Elaboración propia

Tabla 21: *Historia de Usuario 20*

ID: H20	Historia de Usuario 20
Nombre	Visualizar empleados asignados a una obra
Prioridad	4
Estimación	5
Descripción: La aplicación permitirá al usuario administrador visualizar los empleados asignados a una obra.	
Restricciones: Seleccionar una obra previamente.	

Fuente: Elaboración propia

Tabla 22: *Historia de Usuario 21*

ID: H21	Historia de Usuario 21
Nombre	Listar empleados desvinculados
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador visualizar los empleados desvinculados de la empresa.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 23: *Historia de Usuario 22*

ID: H22	Historia de Usuario 22
Nombre	Ver detalle de empleados desvinculados
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador ver el detalle de los empleados desvinculados de la empresa.	
Restricciones: Seleccionar un empleado desvinculado previamente.	

Fuente: Elaboración propia

Tabla 24: *Historia de Usuario 23*

ID: H23	Historia de Usuario 23
Nombre	Eliminar empleados desvinculados
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador eliminar a los empleados desvinculados de la empresa.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar un empleado desvinculado?”. Mostrar mensaje “La desvinculación del empleado se ha eliminado correctamente”, después de eliminar un empleado desvinculado con éxito.	

Fuente: Elaboración propia

Tabla 25: *Historia de Usuario 24*

ID: H24	Historia de Usuario 24
Nombre	Registrar salida
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario obrero y administrador registrar sus salidas.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas registrar una salida?”, y muestra un mensaje “La salida ha sido registrada correctamente” después de registrar una salida.	

Fuente: Elaboración propia

Tabla 26: *Historia de Usuario 25*

ID: H25	Historia de Usuario 25
Nombre	Visualizar estrategias de negocio
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obrero visualizar estrategias de negocio de la empresa.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 27: *Historia de Usuario 26*

ID: H26	Historia de Usuario 26
Nombre	Seleccionar tipos de estrategias
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obrero seleccionar los tipos de estrategias de negocio de la empresa: ausentismo o rotación.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 28: *Historia de Usuario 27*

ID: H27	Historia de Usuario 27
Nombre	Listar estrategias de ausentismo
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de ausentismo.	
Restricciones: Seleccionar estrategia de ausentismo previamente. Flujo erróneo: Si no tiene estrategias de ausentismo, se muestra un mensaje en pantalla “No existen estrategias actualmente”.	

Fuente: Elaboración propia

Tabla 29: *Historia de Usuario 28*

ID: H28	Historia de Usuario 28
Nombre	Listar estrategias de rotación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de rotación.	
Restricciones: Seleccionar estrategia de rotación previamente. Flujo erróneo: Si no tiene estrategias de rotación, se muestra un mensaje en pantalla “No existen estrategias actualmente”.	

Fuente: Elaboración propia

Tabla 30: *Historia de Usuario 29*

ID: H29	Historia de Usuario 29
Nombre	Crear estrategias
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador crear estrategias de negocio ya sean estrategias de ausentismo o rotación.	
Restricciones: Seleccionar estrategia de negocio previamente. Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea crear una estrategia?”. Mostrar mensaje “Estrategia registrado correctamente”, después de crear una estrategia de negocio con éxito.	

Fuente: Elaboración propia

Tabla 31: *Historia de Usuario 30*

ID: H30	Historia de Usuario 30
Nombre	Actualizar estrategias de ausentismo
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador actualizar estrategias de ausentismo.	
Restricciones: Seleccionar estrategia de ausentismo previamente. Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea actualizar la estrategia?”. Mostrar mensaje “Estrategia actualizada correctamente”, después de actualizar una estrategia de ausentismo con éxito.	

Fuente: Elaboración propia

Tabla 32: *Historia de Usuario 31*

ID: H31	Historia de Usuario 31
Nombre	Eliminar estrategias de ausentismo
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador eliminar estrategias de ausentismo.	
Restricciones: Seleccionar estrategia de ausentismo previamente. Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea eliminar una estrategia?”. Mostrar mensaje “Estrategia eliminado correctamente”, después de eliminar una estrategia de ausentismo con éxito.	

Fuente: Elaboración propia

Tabla 33: *Historia de Usuario 32*

ID: H32	Historia de Usuario 32
Nombre	Actualizar estrategias de rotación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador actualizar estrategias de rotación.	
Restricciones: Seleccionar estrategia de rotación previamente. Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea actualizar la estrategia?”. Mostrar mensaje “Estrategia actualizada correctamente”, después de actualizar una estrategia de rotación con éxito.	

Fuente: Elaboración propia

Tabla 34: *Historia de Usuario 33*

ID: H33	Historia de Usuario 33
Nombre	Eliminar estrategias de rotación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador eliminar estrategias de rotación.	
Restricciones: Seleccionar estrategia de rotación previamente. Validar campos de formulario. Mostrar una alerta de confirmación “¿Desea eliminar una estrategia?”. Mostrar mensaje “Estrategia eliminado correctamente”, después de eliminar una estrategia de rotación con éxito.	

Fuente: Elaboración propia

Tabla 35: *Historia de Usuario 34*

ID: H34	Historia de Usuario 34
Nombre	Listar reglamentos
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá a los usuarios visualizar el listado de reglamentos de la empresa.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 36: *Historia de Usuario 35*

ID: H35	Historia de Usuario 35
Nombre	Crear reglamentos
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos crear reglamentos de la empresa.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas crear un reglamento?”. Mostrar mensaje “Reglamento creado correctamente” después de crear un reglamento.	

Fuente: Elaboración propia

Tabla 37: *Historia de Usuario 36*

ID: H36	Historia de Usuario 36
Nombre	Actualizar reglamentos
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos actualizar reglamentos de la empresa.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar un reglamento?”. Mostrar mensaje “Reglamento actualizado correctamente” después de actualizar un reglamento.	

Fuente: Elaboración propia

Tabla 38: *Historia de Usuario 37*

ID: H37	Historia de Usuario 37
Nombre	Eliminar reglamentos
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos eliminar reglamentos de la empresa.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar un reglamento?”. Mostrar mensaje “Reglamento eliminado correctamente” después de eliminar un reglamento.	

Fuente: Elaboración propia

Tabla 39: *Historia de Usuario 38*

ID: H38	Historia de Usuario 38
Nombre	Listar gratificaciones
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá a los usuarios visualizar el listado de gratificaciones de la empresa.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 40: *Historia de Usuario 39*

ID: H39	Historia de Usuario 39
Nombre	Crear gratificaciones
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos crear gratificaciones de la empresa.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas crear una gratificación?”. Mostrar mensaje “Gratificación creado correctamente” después de crear una gratificación. Además, solo se puede crear dos gratificaciones y si creas un tercero saldrá un mensaje “Solo puede crear 2 gratificaciones”.	

Fuente: Elaboración propia

Tabla 41: *Historia de Usuario 40*

ID: H40	Historia de Usuario 40
Nombre	Actualizar gratificaciones
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos actualizar gratificaciones de la empresa.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar una gratificación?”. Mostrar mensaje “Gratificación actualizado correctamente” después de actualizar una gratificación.	

Fuente: Elaboración propia

Tabla 42: *Historia de Usuario 41*

ID: H41	Historia de Usuario 41
Nombre	Eliminar gratificaciones
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y recursos humanos eliminar gratificaciones de la empresa.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar una gratificación?”. Mostrar mensaje “Gratificación eliminado correctamente” después de eliminar una gratificación.	

Fuente: Elaboración propia

Tabla 43: *Historia de Usuario 42*

ID: H42	Historia de Usuario 42
Nombre	Pulsar justificación
Prioridad	4
Estimación	1
Descripción: La aplicación permitirá a los usuarios pulsar justificaciones en la barra lateral.	
Restricciones: En el caso de administrador y obrero saldrá un contador por la cantidad de justificaciones creadas.	

Fuente: Elaboración propia

Tabla 44: *Historia de Usuario 43*

ID: H43	Historia de Usuario 43
Nombre	Listar empleado justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no justificaciones.	
Restricciones: En el caso de que falte aprobar una justificación se mostrará un contador en el empleado con la cantidad de justificación a aprobar.	

Fuente: Elaboración propia

Tabla 45: *Historia de Usuario 44*

ID: H44	Historia de Usuario 44
Nombre	Seleccionar empleado justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no justificaciones.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 46: *Historia de Usuario 45*

ID: H45	Historia de Usuario 45
Nombre	Listar justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá a los usuarios visualizar el listado de las justificaciones de los empleados de la empresa.	
Restricciones: En el caso de que los empleados no tengan justificaciones, se mostrará en la pantalla “No hay solicitudes de justificaciones”. En el caso de obreros y administradores les saldrá un estado “SI” que fue aprobado o “NO” una vez creada su justificación.	

Fuente: Elaboración propia

Tabla 47: *Historia de Usuario 46*

ID: H46	Historia de Usuario 46
Nombre	Aprobar justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos aprobar las justificaciones de los empleados de la empresa.	
Restricciones: Si desea aprueba o no las justificaciones de los empleados.	

Fuente: Elaboración propia

Tabla 48: *Historia de Usuario 47*

ID: H47	Historia de Usuario 47
Nombre	Crear justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros crear sus justificaciones.	
Restricciones: Validar campos de formulario. Seleccionar fecha inicio y fecha fin. Mostrar una alerta de confirmación “¿Deseas crear una justificación?”. Mostrar mensaje “Justificación creada correctamente” después de crear una justificación. Además, la justificación aún no estará aprobada por recursos humanos. Flujo erróneo: Si la fecha de inicio es mayor a la fecha de fin, se mostrará un mensaje “La fecha de inicio no puede ser mayor a la fecha de fin”.	

Fuente: Elaboración propia

Tabla 49: *Historia de Usuario 48*

ID: H48	Historia de Usuario 48
Nombre	Actualizar justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros actualizar sus justificaciones.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar una justificación?”. Mostrar mensaje “Justificación actualizada correctamente” después de actualizar una justificación.	

Fuente: Elaboración propia

Tabla 50: *Historia de Usuario 49*

ID: H49	Historia de Usuario 49
Nombre	Eliminar justificación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros eliminar sus justificaciones.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar una justificación?”. Mostrar mensaje “Justificación eliminado correctamente” después de eliminar una justificación.	

Fuente: Elaboración propia

Tabla 51: *Historia de Usuario 50*

ID: H50	Historia de Usuario 50
Nombre	Pulsar vacación
Prioridad	4
Estimación	1
Descripción: La aplicación permitirá a los usuarios pulsar vacaciones en la barra lateral.	
Restricciones: En el caso de administrador y obrero saldrá un contador por la cantidad de vacaciones creadas.	

Fuente: Elaboración propia

Tabla 52: *Historia de Usuario 51*

ID: H51	Historia de Usuario 51
Nombre	Listar empleado vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no vacaciones.	
Restricciones: En el caso de que falte aprobar una vacación se mostrará un contador en el empleado con la cantidad de vacaciones a aprobar.	

Fuente: Elaboración propia

Tabla 53: *Historia de Usuario 52*

ID: H52	Historia de Usuario 52
Nombre	Seleccionar empleado vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no vacaciones.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 54: *Historia de Usuario 53*

ID: H53	Historia de Usuario 53
Nombre	Listar vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá a los usuarios visualizar el listado de las vacaciones de los empleados de la empresa.	
Restricciones: En el caso de que los empleados no tengan vacaciones, se mostrará en la pantalla “No hay solicitud de vacaciones”. En el caso de obreros y administradores les saldrá un estado “SI” que fue aprobado o “NO” una vez creada sus vacaciones.	

Fuente: Elaboración propia

Tabla 55: *Historia de Usuario 54*

ID: H54	Historia de Usuario 54
Nombre	Aprobar vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos aprobar las vacaciones de los empleados de la empresa.	
Restricciones: Si desea aprueba o no las vacaciones de los empleados.	

Fuente: Elaboración propia

Tabla 56: *Historia de Usuario 55*

ID: H55	Historia de Usuario 55
Nombre	Crear vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros crear sus vacaciones.	
Restricciones: Validar campos de formulario. Seleccionar fecha inicio y fecha fin. Mostrar una alerta de confirmación “¿Deseas crear una vacación?”. Mostrar mensaje “Vacación creada correctamente” después de crear una vacación. Además, la vacación aún no estará aprobada por recursos humanos. Flujo erróneo: Si la fecha de inicio es mayor a la fecha de fin, se mostrará un mensaje “La fecha de inicio no puede ser mayor a la fecha de fin”. Si en caso la vacación del empleado no ha sido aprobada aún por recursos humanos, se muestra un mensaje “Ya tiene una solíc. de vacación creada y aún no ha sido aprobada”. Si en caso la vacación del empleado ya ha sido aprobada y quiere registrar otra vacación, se muestra un mensaje “Solo puede realizar una solíc. de vacación por año”.	

Fuente: Elaboración propia

Tabla 57: *Historia de Usuario 56*

ID: H56	Historia de Usuario 56
Nombre	Actualizar vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros actualizar sus vacaciones.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar una vacación?”. Mostrar mensaje “Solic. Vacación actualizada correctamente” después de actualizar una vacación.	

Fuente: Elaboración propia

Tabla 58: *Historia de Usuario 57*

ID: H57	Historia de Usuario 57
Nombre	Eliminar vacación
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario administrador y obreros eliminar sus vacaciones.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar una vacación?”. Mostrar mensaje “Solic. vacación eliminada correctamente” después de eliminar una vacación.	

Fuente: Elaboración propia

Tabla 59: *Historia de Usuario 58*

ID: H58	Historia de Usuario 58
Nombre	Pulsar remuneración
Prioridad	4
Estimación	1
Descripción: La aplicación permitirá a los usuarios pulsar remuneraciones en la barra lateral.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 60: *Historia de Usuario 59*

ID: H59	Historia de Usuario 59
Nombre	Listar empleado remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no remuneración.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 61: *Historia de Usuario 60*

ID: H60	Historia de Usuario 60
Nombre	Seleccionar empleado remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no remuneraciones.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Tabla 62: *Historia de Usuario 61*

ID: H61	Historia de Usuario 61
Nombre	Listar remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá a los usuarios visualizar el listado de las remuneraciones de los empleados de la empresa.	
Restricciones: En el caso de que los empleados no tengan remuneraciones se mostrará en la pantalla “No hay remuneración para este empleado”.	

Fuente: Elaboración propia

Tabla 63: *Historia de Usuario 62*

ID: H62	Historia de Usuario 62
Nombre	Crear remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos crear las remuneraciones.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas crear una remuneración?”. Mostrar mensaje “Remuneración creada correctamente” después de crear una remuneración. Flujo erróneo: Si vuelves a crear la remuneración al mismo empleado se mostrará un mensaje “Solo puede crear una remuneración por empleado”.	

Fuente: Elaboración propia

Tabla 64: *Historia de Usuario 63*

ID: H63	Historia de Usuario 63
Nombre	Actualizar remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos actualizar las remuneraciones.	
Restricciones: Validar campos de formulario. Mostrar una alerta de confirmación “¿Deseas actualizar una remuneración?”. Mostrar mensaje “Remuneración actualizada correctamente” después de actualizar una remuneración.	

Fuente: Elaboración propia

Tabla 65: *Historia de Usuario 64*

ID: H64	Historia de Usuario 64
Nombre	Eliminar remuneración
Prioridad	4
Estimación	3
Descripción: La aplicación permitirá al usuario de recursos humanos eliminar las remuneraciones.	
Restricciones: Mostrar una alerta de confirmación “¿Deseas eliminar una remuneración?”. Mostrar mensaje “Remuneración eliminado correctamente” después de eliminar una remuneración.	

Fuente: Elaboración propia

Tabla 66: *Historia de Usuario 65*

ID: H65	Historia de Usuario 65
Nombre	Redirigir al home
Prioridad	4
Estimación	1
Descripción: La aplicación permitirá a los usuarios redirigir a la pantalla de inicio (Home) al pulsar en “Personalapp”.	
Restricciones: Ninguna.	

Fuente: Elaboración propia

Pila de Producto (Product Backlog)

Tabla 67: *Pila de Producto Inicial*

ID	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
RF1	Para que puedan hacer uso de la aplicación deberán haber adquirido un usuario y contraseña.	H1	3	1
RF2	La aplicación permitirá al usuario obrero y administrador registrar sus asistencias.	H2	3	1
RF3	La aplicación permitirá a los usuarios cerrar sesión.	H3	1	1
RF4	La aplicación permitirá crear obras de trabajo.	H4	3	2
RF5	La aplicación permitirá asignar una obra a cada obrero.	H5	3	2
RF6	La aplicación permitirá actualizar los datos de cada obra.	H6	3	2
RF7	La aplicación permitirá eliminar una obra.	H7	3	3
RF8	La aplicación permitirá registrar un empleado, administrador, obrero o recursos humanos.	H8	3	3
RF9	La aplicación permitirá actualizar un empleado.	H9	3	3
RF10	La aplicación permitirá eliminar un empleado.	H10	3	3
RF11	Esta funcionalidad permitirá colocar en un estado de desvinculado a los empleados.	H11	3	3

RF12	El sistema proporcionará a los usuarios información de las marcaciones de asistencia del obrero.	H12	5	4
RF13	Este módulo permitirá visualizar un reporte de ausentismo de todos los empleados que están en una obra.	H13	8	4
RF14	Este módulo permitirá visualizar un reporte de todos los empleados por obra que se hayan desvinculado de la empresa.	H14	8	4
RF15	La aplicación permitirá al obrero y administrador comunicarse con al área de recursos humanos para hacerle consultas.	H15	1	4
RF16	La aplicación permitirá mostrar al usuario obrero en que obra está asignado.	H16	3	4
RF17	La aplicación permitirá mostrar al usuario administrador un listado de los empleados: obreros, recursos humanos y administrador.	H17	3	4
RF18	La aplicación permitirá mostrar un listado de todas las obras al usuario administrador.	H18	3	4
RF19	La aplicación permitirá al usuario administrador ver el detalle de cada obra seleccionada.	H19	3	4

RF20	La aplicación permitirá al usuario administrador visualizar los empleados asignados a una obra.	H20	5	4
RF21	La aplicación permitirá al usuario administrador visualizar los empleados desvinculados de la empresa.	H21	3	4
RF22	La aplicación permitirá al usuario administrador ver el detalle de los empleados desvinculados de la empresa.	H22	3	4
RF23	La aplicación permitirá al usuario administrador eliminar a los empleados desvinculados de la empresa.	H23	3	4
RF24	La aplicación permitirá al usuario obrero y administrador registrar sus salidas.	H24	3	4
RF25	La aplicación permitirá al usuario administrador y obrero visualizar estrategias de negocio de la empresa.	H25	3	4
RF26	La aplicación permitirá al usuario administrador y obrero seleccionar los tipos de estrategias de negocio de la empresa: ausentismo o rotación.	H26	3	4
RF27	La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de ausentismo.	H27	3	4

RF28	La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de rotación.	H28	3	4
RF29	La aplicación permitirá al usuario administrador crear estrategias de negocio ya sean estrategias de ausentismo o rotación.	H29	3	4
RF30	La aplicación permitirá al usuario administrador actualizar estrategias de ausentismo.	H30	3	4
RF31	La aplicación permitirá al usuario administrador eliminar estrategias de ausentismo.	H31	3	4
RF32	La aplicación permitirá al usuario administrador actualizar estrategias de rotación.	H32	3	4
RF33	La aplicación permitirá al usuario administrador eliminar estrategias de rotación.	H33	3	4
RF34	La aplicación permitirá a los usuarios visualizar el listado de reglamentos de la empresa.	H34	3	4
RF35	La aplicación permitirá al usuario administrador y recursos humanos crear reglamentos de la empresa.	H35	3	4
RF36	La aplicación permitirá al usuario administrador y recursos humanos actualizar reglamentos de la empresa.	H36	3	4
RF37	La aplicación permitirá al usuario administrador y recursos	H37	3	4

	humanos eliminar reglamentos de la empresa.			
RF38	La aplicación permitirá a los usuarios visualizar el listado de gratificaciones de la empresa.	H38	3	4
RF39	La aplicación permitirá al usuario administrador y recursos humanos crear gratificaciones de la empresa.	H39	3	4
RF40	La aplicación permitirá al usuario administrador y recursos humanos actualizar gratificaciones de la empresa.	H40	3	4
RF41	La aplicación permitirá al usuario administrador y recursos humanos eliminar gratificaciones de la empresa.	H41	3	4
RF42	La aplicación permitirá a los usuarios pulsar justificaciones en la barra lateral.	H42	1	4
RF43	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no justificaciones.	H43	3	4
RF44	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no justificaciones.	H44	3	4
RF45	La aplicación permitirá a los usuarios visualizar el listado de las justificaciones de los empleados de la empresa.	H45	3	4

RF46	La aplicación permitirá al usuario de recursos humanos aprobar las justificaciones de los empleados de la empresa.	H46	3	4
RF47	La aplicación permitirá al usuario administrador y obreros crear sus justificaciones.	H47	3	4
RF48	La aplicación permitirá al usuario administrador y obreros actualizar sus justificaciones.	H48	3	4
RF49	La aplicación permitirá al usuario administrador y obreros eliminar sus justificaciones.	H49	3	4
RF50	La aplicación permitirá a los usuarios pulsar vacaciones en la barra lateral.	H50	1	4
RF51	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no vacaciones.	H51	3	4
RF52	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no vacaciones.	H52	3	4
RF53	La aplicación permitirá a los usuarios visualizar el listado de las vacaciones de los empleados de la empresa.	H53	3	4
RF54	La aplicación permitirá al usuario de recursos humanos aprobar las vacaciones de los empleados de la empresa.	H54	3	4

RF55	La aplicación permitirá al usuario administrador y obreros crear sus vacaciones.	H55	3	4
RF56	La aplicación permitirá al usuario administrador y obreros actualizar sus vacaciones.	H56	3	4
RF57	La aplicación permitirá al usuario administrador y obreros eliminar sus vacaciones.	H57	3	4
RF58	La aplicación permitirá a los usuarios pulsar remuneraciones en la barra lateral.	H58	1	4
RF59	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no remuneración.	H59	3	4
RF60	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no remuneraciones.	H60	3	4
RF61	La aplicación permitirá a los usuarios visualizar el listado de las remuneraciones de los empleados de la empresa.	H61	3	4
RF62	La aplicación permitirá al usuario de recursos humanos crear las remuneraciones.	H62	3	4
RF63	La aplicación permitirá al usuario de recursos humanos actualizar las remuneraciones.	H63	3	4

RF64	La aplicación permitirá al usuario de recursos humanos eliminar las remuneraciones.	H64	3	4
RF65	La aplicación permitirá a los usuarios redirigir a la pantalla de inicio (Home) al pulsar en “Personalapp”.	H65	1	4

Fuente: Elaboración propia

Aquí están todas las descripciones de las funcionalidades de la app móvil, la prioridad por cada sprint y las estimaciones que el equipo de desarrollo realiza el puntaje correspondiente. Esta app móvil cuenta con diez módulos divididos por diferentes funcionalidades.

Pila de Producto (Product Backlog)

Tabla 68: Pila de Producto Organizada por prioridad

ID	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
RF1	Para que puedan hacer uso de la aplicación deberán haber adquirido un usuario y contraseña.	H1	3	1
RF2	La aplicación permitirá al usuario obrero y administrador registrar sus asistencias.	H2	3	1
RF3	La aplicación permitirá a los usuarios cerrar sesión.	H3	1	1
RF4	La aplicación permitirá crear obras de trabajo.	H4	3	2
RF5	La aplicación permitirá asignar una obra a cada obrero.	H5	3	2

RF6	La aplicación permitirá actualizar los datos de cada obra.	H6	3	2
RF7	La aplicación permitirá eliminar una obra.	H7	3	3
RF8	La aplicación permitirá registrar un empleado, administrador, obrero o recursos humanos.	H8	3	3
RF9	La aplicación permitirá actualizar un empleado.	H9	3	3
RF10	La aplicación permitirá eliminar un empleado.	H10	3	3
RF11	Esta funcionalidad permitirá colocar en un estado de desvinculado a los empleados.	H11	3	3
RF12	El sistema proporcionará a los usuarios información de las marcaciones de asistencia del obrero.	H12	5	4
RF13	Este módulo permitirá visualizar un reporte de ausentismo de todos los empleados que están en una obra.	H13	8	4
RF14	Este módulo permitirá visualizar un reporte de todos los empleados por obra que se hayan desvinculado de la empresa.	H14	8	4
RF15	La aplicación permitirá al obrero y administrador comunicarse con al área de recursos humanos para hacerle consultas.	H15	1	4

RF16	La aplicación permitirá mostrar al usuario obrero en que obra está asignado.	H16	3	4
RF17	La aplicación permitirá mostrar al usuario administrador un listado de los empleados: obreros, recursos humanos y administrador.	H17	3	4
RF18	La aplicación permitirá mostrar un listado de todas las obras al usuario administrador.	H18	3	4
RF19	La aplicación permitirá al usuario administrador ver el detalle de cada obra seleccionada.	H19	3	4
RF20	La aplicación permitirá al usuario administrador visualizar los empleados asignados a una obra.	H20	5	4
RF21	La aplicación permitirá al usuario administrador visualizar los empleados desvinculados de la empresa.	H21	3	4
RF22	La aplicación permitirá al usuario administrador ver el detalle de los empleados desvinculados de la empresa.	H22	3	4
RF23	La aplicación permitirá al usuario administrador eliminar a los empleados desvinculados de la empresa.	H23	3	4
RF24	La aplicación permitirá al usuario obrero y administrador registrar sus salidas.	H24	3	4

RF25	La aplicación permitirá al usuario administrador y obrero visualizar estrategias de negocio de la empresa.	H25	3	4
RF26	La aplicación permitirá al usuario administrador y obrero seleccionar los tipos de estrategias de negocio de la empresa: ausentismo o rotación.	H26	3	4
RF27	La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de ausentismo.	H27	3	4
RF28	La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de rotación.	H28	3	4
RF29	La aplicación permitirá al usuario administrador crear estrategias de negocio ya sean estrategias de ausentismo o rotación.	H29	3	4
RF30	La aplicación permitirá al usuario administrador actualizar estrategias de ausentismo.	H30	3	4
RF31	La aplicación permitirá al usuario administrador eliminar estrategias de ausentismo.	H31	3	4
RF32	La aplicación permitirá al usuario administrador actualizar estrategias de rotación.	H32	3	4
RF33	La aplicación permitirá al usuario administrador eliminar estrategias de rotación.	H33	3	4

RF34	La aplicación permitirá a los usuarios visualizar el listado de reglamentos de la empresa.	H34	3	4
RF35	La aplicación permitirá al usuario administrador y recursos humanos crear reglamentos de la empresa.	H35	3	4
RF36	La aplicación permitirá al usuario administrador y recursos humanos actualizar reglamentos de la empresa.	H36	3	4
RF37	La aplicación permitirá al usuario administrador y recursos humanos eliminar reglamentos de la empresa.	H37	3	4
RF38	La aplicación permitirá a los usuarios visualizar el listado de gratificaciones de la empresa.	H38	3	4
RF39	La aplicación permitirá al usuario administrador y recursos humanos crear gratificaciones de la empresa.	H39	3	4
RF40	La aplicación permitirá al usuario administrador y recursos humanos actualizar gratificaciones de la empresa.	H40	3	4
RF41	La aplicación permitirá al usuario administrador y recursos humanos eliminar gratificaciones de la empresa.	H41	3	4

RF42	La aplicación permitirá a los usuarios pulsar justificaciones en la barra lateral.	H42	1	4
RF43	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no justificaciones.	H43	3	4
RF44	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no justificaciones.	H44	3	4
RF45	La aplicación permitirá a los usuarios visualizar el listado de las justificaciones de los empleados de la empresa.	H45	3	4
RF46	La aplicación permitirá al usuario de recursos humanos aprobar las justificaciones de los empleados de la empresa.	H46	3	4
RF47	La aplicación permitirá al usuario administrador y obreros crear sus justificaciones.	H47	3	4
RF48	La aplicación permitirá al usuario administrador y obreros actualizar sus justificaciones.	H48	3	4
RF49	La aplicación permitirá al usuario administrador y obreros eliminar sus justificaciones.	H49	3	4
RF50	La aplicación permitirá a los usuarios pulsar vacaciones en la barra lateral.	H50	1	4

RF51	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no vacaciones.	H51	3	4
RF52	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no vacaciones.	H52	3	4
RF53	La aplicación permitirá a los usuarios visualizar el listado de las vacaciones de los empleados de la empresa.	H53	3	4
RF54	La aplicación permitirá al usuario de recursos humanos aprobar las vacaciones de los empleados de la empresa.	H54	3	4
RF55	La aplicación permitirá al usuario administrador y obreros crear sus vacaciones.	H55	3	4
RF56	La aplicación permitirá al usuario administrador y obreros actualizar sus vacaciones.	H56	3	4
RF57	La aplicación permitirá al usuario administrador y obreros eliminar sus vacaciones.	H57	3	4
RF58	La aplicación permitirá a los usuarios pulsar remuneraciones en la barra lateral.	H58	1	4
RF59	La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no remuneración.	H59	3	4

RF60	La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no remuneraciones.	H60	3	4
RF61	La aplicación permitirá a los usuarios visualizar el listado de las remuneraciones de los empleados de la empresa.	H61	3	4
RF62	La aplicación permitirá al usuario de recursos humanos crear las remuneraciones.	H62	3	4
RF63	La aplicación permitirá al usuario de recursos humanos actualizar las remuneraciones.	H63	3	4
RF64	La aplicación permitirá al usuario de recursos humanos eliminar las remuneraciones.	H64	3	4
RF65	La aplicación permitirá a los usuarios redirigir a la pantalla de inicio (Home) al pulsar en “Personalapp”.	H65	1	4

Fuente: Elaboración propia

Aquí están todas las descripciones de las funcionalidades de la app móvil organizadas por prioridad y las estimaciones que el equipo de desarrollo realiza el puntaje correspondiente. Esta app móvil cuenta con diez módulos divididos por diferentes funcionalidades.

Requerimientos No Funcionales

Tabla 69: *Requerimientos No Funcionales*

Código	Requerimiento No Funcional	Descripción
RNF1	Usabilidad	La aplicación móvil tiene una Interfaz ágil y amigable, fácil de usar. Con ayudas e interfaces intuitivas.
RNF2	Seguridad	El ingreso de la aplicación móvil estará restringido bajo contraseñas y usuarios definidos.
RNF3	Portabilidad	La aplicación móvil deberá brindar comodidad al empleado y a otras áreas que trabajan o necesitan del área de personal.
RNF4	Rendimiento	La aplicación móvil deberá soportar el flujo de la información requerida durante su proceso.
RNF5	Desempeño	La aplicación móvil no presentará problemas para su manejo.
RNF6	Soporte	La aplicación móvil debe ser fácil de analizar y modificar para corregir posibles fallas.
RNF7	Fiabilidad	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado.
RNF8	Disponibilidad	El sistema debe estar 100% disponible al personal de la empresa.

Fuente: Elaboración propia

Aquí están todos los requerimientos no funcionalidades de la app móvil a implementar. Es muy importante que en una aplicación se vea la parte de usabilidad, seguridad, soporte, etc

Entregables por Sprint

En este punto se detalle la cantidad de Sprints, las descripciones de los requerimientos funcionales de la Pila de Producto, sus respectivas prioridades que es por cada sprint y tiempos estimados que el equipo de desarrollo realiza el puntaje correspondiente

Lista de Sprint (Sprint Backlog)

Tabla 70: *Lista de Sprint*

N° Sprint	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
Sprint 1	RF1: Para que puedan hacer uso de la aplicación deberán haber adquirido un usuario y contraseña.	H1	3	1
	RF2: La aplicación permitirá al usuario obrero y administrador registrar sus asistencias.	H2	3	1
	RF3: La aplicación permitirá a los usuarios cerrar sesión.	H3	1	1
Sprint 2	RF4: La aplicación permitirá crear obras de trabajo.	H4	3	2
	RF5: La aplicación permitirá asignar una obra a cada obrero.	H5	3	2
	RF6: La aplicación permitirá actualizar los datos de cada obra.	H6	3	2
Sprint 3	RF7: La aplicación permitirá eliminar una obra.	H7	3	3
	RF8: La aplicación permitirá registrar un empleado, administrador, obrero o recursos humanos.	H8	3	3

	RF9: La aplicación permitirá actualizar un empleado.	H9	3	3
	RF10: La aplicación permitirá eliminar un empleado.	H10	3	3
	RF11: Esta funcionalidad permitirá colocar en un estado de desvinculado a los empleados.	H11	3	3
	RF12: El sistema proporcionará a los usuarios información de las marcaciones de asistencia del obrero.	H12	5	4
	RF13: Este módulo permitirá visualizar un reporte de ausentismo de todos los empleados que están en una obra.	H13	8	4
	RF14: Este módulo permitirá visualizar un reporte de todos los empleados por obra que se hayan desvinculado de la empresa.	H14	8	4
	RF15: La aplicación permitirá al obrero y administrador comunicarse con al área de recursos humanos para hacerle consultas.	H15	1	4
	RF16: La aplicación permitirá mostrar al usuario obrero en que obra está asignado.	H16	3	4
	RF17: La aplicación permitirá mostrar al usuario administrador un listado de los empleados: obreros, recursos humanos y administrador.	H17	3	4

Sprint 4	RF18: La aplicación permitirá mostrar un listado de todas las obras al usuario administrador.	H18	3	4
	RF19: La aplicación permitirá al usuario administrador ver el detalle de cada obra seleccionada.	H19	3	4
	RF20: La aplicación permitirá al usuario administrador visualizar los empleados asignados a una obra.	H20	5	4
	RF21: La aplicación permitirá al usuario administrador visualizar los empleados desvinculados de la empresa.	H21	3	4
	RF22: La aplicación permitirá al usuario administrador ver el detalle de los empleados desvinculados de la empresa.	H22	3	4
	RF23: La aplicación permitirá al usuario administrador eliminar a los empleados desvinculados de la empresa.	H23	3	4
	RF24: La aplicación permitirá al usuario obrero y administrador registrar sus salidas.	H24	3	4
	RF25: La aplicación permitirá al usuario administrador y obrero visualizar estrategias de negocio de la empresa.	H25	3	4
	RF26: La aplicación permitirá al usuario administrador y obrero seleccionar los tipos de estrategias	H26	3	4

de negocio de la empresa: ausentismo o rotación.			
RF27: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de ausentismo.	H27	3	4
RF28: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de rotación.	H28	3	4
RF29: La aplicación permitirá al usuario administrador crear estrategias de negocio ya sean estrategias de ausentismo o rotación.	H29	3	4
RF30: La aplicación permitirá al usuario administrador actualizar estrategias de ausentismo.	H30	3	4
RF31: La aplicación permitirá al usuario administrador eliminar estrategias de ausentismo.	H31	3	4
RF32: La aplicación permitirá al usuario administrador actualizar estrategias de rotación.	H32	3	4
RF33: La aplicación permitirá al usuario administrador eliminar estrategias de rotación.	H33	3	4
RF34: La aplicación permitirá a los usuarios visualizar el listado de reglamentos de la empresa.	H34	3	4
RF35: La aplicación permitirá al usuario administrador y recursos	H35	3	4

humanos crear reglamentos de la empresa.			
RF36: La aplicación permitirá al usuario administrador y recursos humanos actualizar reglamentos de la empresa.	H36	3	4
RF37: La aplicación permitirá al usuario administrador y recursos humanos eliminar reglamentos de la empresa.	H37	3	4
RF38: La aplicación permitirá a los usuarios visualizar el listado de gratificaciones de la empresa.	H38	3	4
RF39: La aplicación permitirá al usuario administrador y recursos humanos crear gratificaciones de la empresa.	H39	3	4
RF40: La aplicación permitirá al usuario administrador y recursos humanos actualizar gratificaciones de la empresa.	H40	3	4
RF41: La aplicación permitirá al usuario administrador y recursos humanos eliminar gratificaciones de la empresa.	H41	3	4
RF42: La aplicación permitirá a los usuarios pulsar justificaciones en la barra lateral.	H42	1	4
RF43: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los	H43	3	4

empleados que tengan o no justificaciones.			
RF44: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no justificaciones.	H44	3	4
RF45: La aplicación permitirá a los usuarios visualizar el listado de las justificaciones de los empleados de la empresa.	H45	3	4
RF46: La aplicación permitirá al usuario de recursos humanos aprobar las justificaciones de los empleados de la empresa.	H46	3	4
RF47: La aplicación permitirá al usuario administrador y obreros crear sus justificaciones.	H47	3	4
RF48: La aplicación permitirá al usuario administrador y obreros actualizar sus justificaciones.	H48	3	4
RF49: La aplicación permitirá al usuario administrador y obreros eliminar sus justificaciones.	H49	3	4
RF50: La aplicación permitirá a los usuarios pulsar vacaciones en la barra lateral.	H50	1	4
RF51: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no vacaciones.	H51	3	4

RF52: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no vacaciones.	H52	3	4
RF53: La aplicación permitirá a los usuarios visualizar el listado de las vacaciones de los empleados de la empresa.	H53	3	4
RF54: La aplicación permitirá al usuario de recursos humanos aprobar las vacaciones de los empleados de la empresa.	H54	3	4
RF55: La aplicación permitirá al usuario administrador y obreros crear sus vacaciones.	H55	3	4
RF56: La aplicación permitirá al usuario administrador y obreros actualizar sus vacaciones.	H56	3	4
RF57: La aplicación permitirá al usuario administrador y obreros eliminar sus vacaciones.	H57	3	4
RF58: La aplicación permitirá a los usuarios pulsar remuneraciones en la barra lateral.	H58	1	4
RF59: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no remuneración.	H59	3	4
RF60: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que	H60	3	4

	tengan o no remuneraciones.			
	RF61: La aplicación permitirá a los usuarios visualizar el listado de las remuneraciones de los empleados de la empresa.	H61	3	4
	RF62: La aplicación permitirá al usuario de recursos humanos crear las remuneraciones.	H62	3	4
	RF63: La aplicación permitirá al usuario de recursos humanos actualizar las remuneraciones.	H63	3	4
	RF64: La aplicación permitirá al usuario de recursos humanos eliminar las remuneraciones.	H64	3	4
	RF65: La aplicación permitirá a los usuarios redirigir a la pantalla de inicio (Home) al pulsar en “Personalapp”.	H65	1	4

Fuente: Elaboración propia

Aquí están todas las descripciones de las funcionalidades de la app móvil por sprints, las estimaciones que realiza el equipo de desarrollo y las prioridades por cada sprint. Esta app móvil cuenta con diez módulos divididos por diferentes funcionalidades.

Lista de pendientes de Sprint (Sprint Backlog)

Sprint 1

Tabla 71: *Sprint 1*

Nº Sprint	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
Sprint 1	RF1: Para que puedan hacer uso de la aplicación deberán haber adquirido un usuario y contraseña.	H1	3	1
	RF2: La aplicación permitirá al usuario obrero y administrador registrar sus asistencias.	H2	3	1
	RF3: La aplicación permitirá a los usuarios cerrar sesión.	H3	1	1

Fuente: Elaboración propia

Aquí está todas las funcionalidades a realizar en el sprint 1.

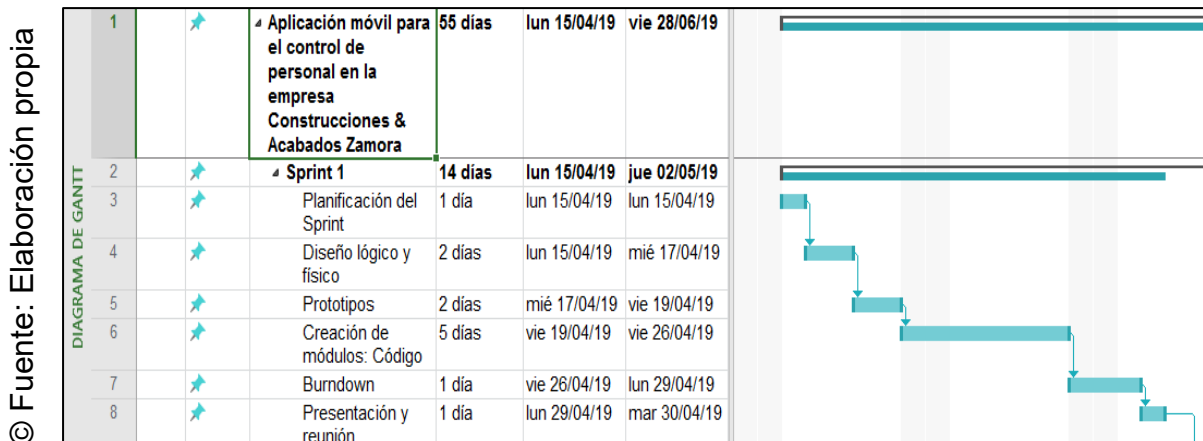


Figura 1. Planificación Sprint 1

Aquí en esta figura 1 está el cronograma de las actividades a realizar en el sprint 1.

A continuación, en este apartado de nuestra tesis, se podrá evidenciar todo el proceso de la implementación de la app móvil divididos por cada funcionalidad del sprint 1. En el cual nosotros hemos elaborado un texto descriptivo y cuatro imágenes por funcionalidad para que ustedes como lectores puedan entender bien todo el proceso de implementación de la aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., todas las imágenes que hemos elaborado están en orden secuencial y por cada funcionalidad pero no todas las imágenes están con sus respectivas descripciones, por eso vamos a explicar en este texto como es el orden y descripción de cada imagen por funcionalidad: una vez explicado el objetivo del requerimiento funcional podrán observar en las dos imágenes siguientes la base de datos lógico y físico de las tablas involucradas de dicha funcionalidad (el proceso para llegar al diseño de la base de datos lógico y físico y tablas involucradas que son las entidades fue que se obtuvo durante el análisis de la segunda actividad que se puede visualizar en el cronograma del sprint 1 y de las descripciones de las historias de usuario), luego se mostrará la imagen del diseño de los prototipos elaborados con la herramienta Balsamiq Mockups (estos diseños fueron validados y aceptados por la empresa), luego ustedes podrán observar la imagen de implementación de las interfaces y por último puedan ver la imagen del código fuente de dicha funcionalidad de la app móvil. Todos los requerimientos funcionales de este sprint 1 están enumerados y por ende se sobre entiende que las imágenes que van a observar pertenecen a las funcionalidades ya que están todas enumeradas.

Requerimiento RF1 – Acceder a la aplicación

El requerimiento 1, tiene como objetivo que la aplicación debe permitir hacer login a los usuarios “Administrador”, “Recursos Humanos” y “Obrero”, dichos usuarios deben poder acceder a la aplicación y visualizar módulos según su rol. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 1).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 198 de la presente tesis ya que ahí hay un texto que deben de leer.

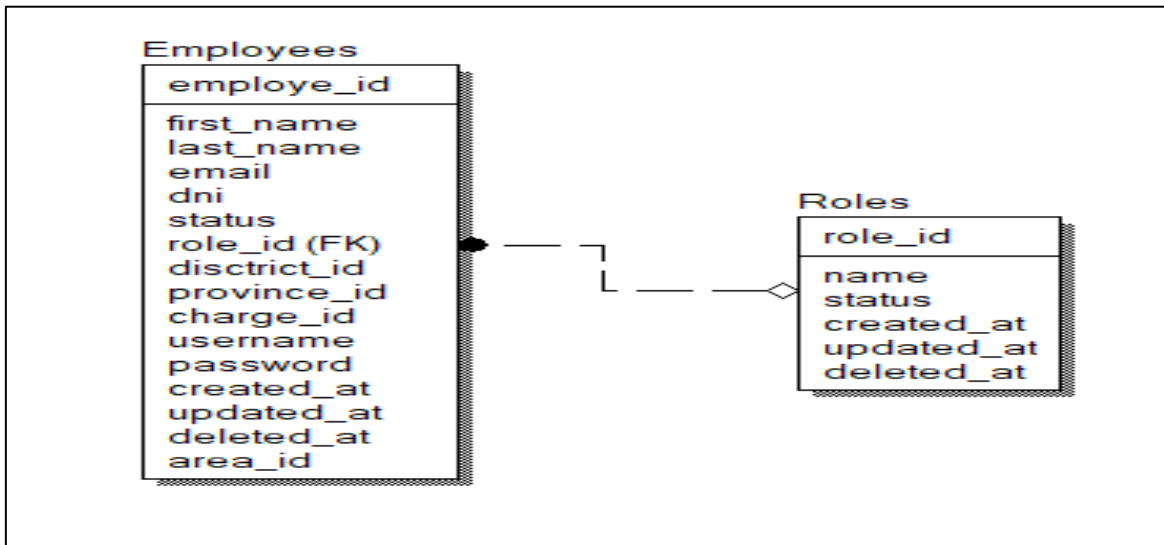


Figura 2. Diagrama lógico de la base de datos RF1

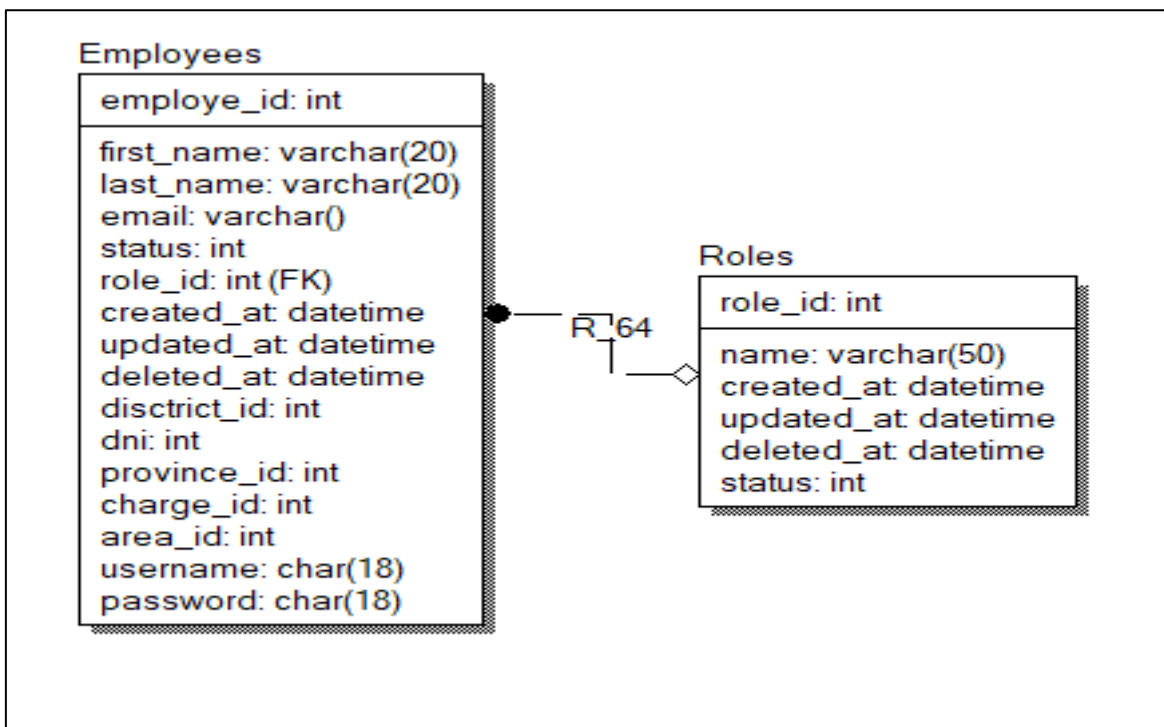


Figura 3. Diagrama físico de la base de datos RF1

Diseño

© Fuente: Elaboración propia



Figura 4. Prototipo de RF1

En la figura 4 se muestra el prototipo para la GUI de acceder a la aplicación aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



Figura 5. Interface de RF1

En la figura 5 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto

Código

© Fuente: Elaboración propia

```
public function authenticate(User $user) {
    $this->validate($this->request, [
        'username' => 'required',
        'password' => 'required'
    ]);

    $user = User::where('username', $this->request->input(key: 'username'))->where('password', $this->request->input(key: 'password'))->first();

    if($user){
        return response()->json([
            'msg' => 'Acceso satisfactorio'
        ], status: 200);
    }else{
        return response()->json([
            'msg' => 'No existe usuario'
        ], status: 400);
    }
}
```

Figura 6. Código de RF1

En esta imagen 6 de la presente tesis se puede observar el código fuente del requerimiento funcional 1.

Requerimiento RF2 – Registrar asistencia

El requerimiento 2, tiene como objetivo que la aplicación debe registrar una asistencia, solo a los usuarios “Administrador” y “Obrero”, dicha asistencia debe guardar la hora de entrada. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 2)

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 198 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

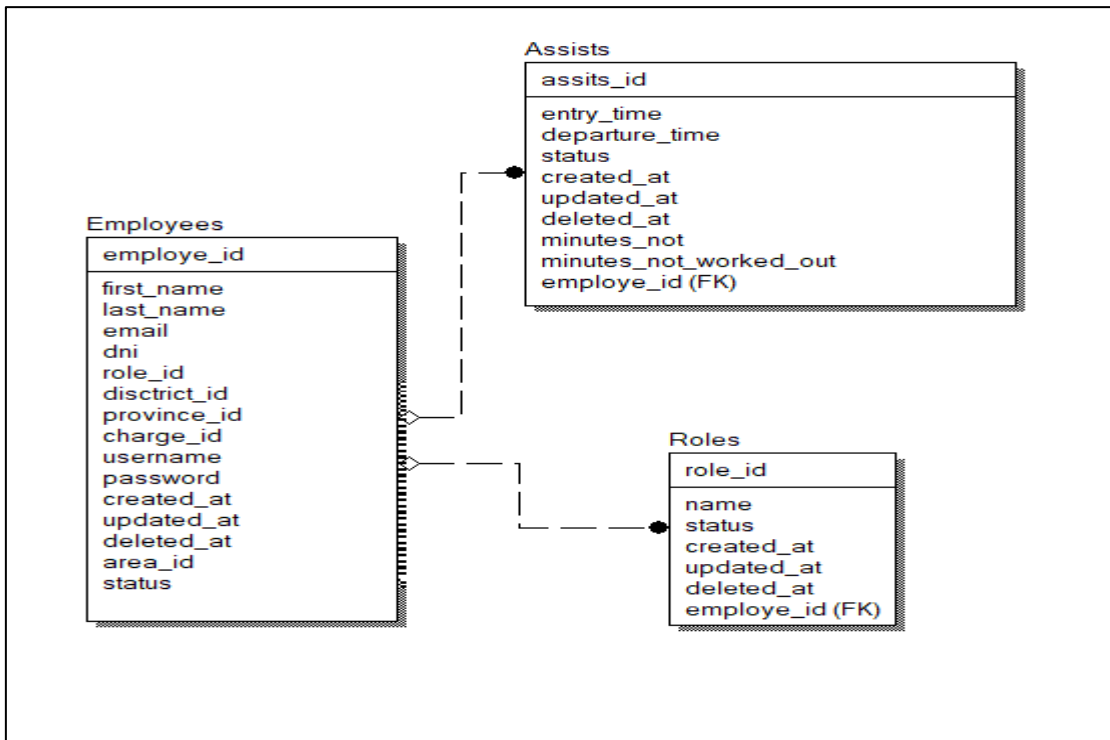


Figura 7. Diagrama lógico de la base de datos RF2

© Fuente: Elaboración propia

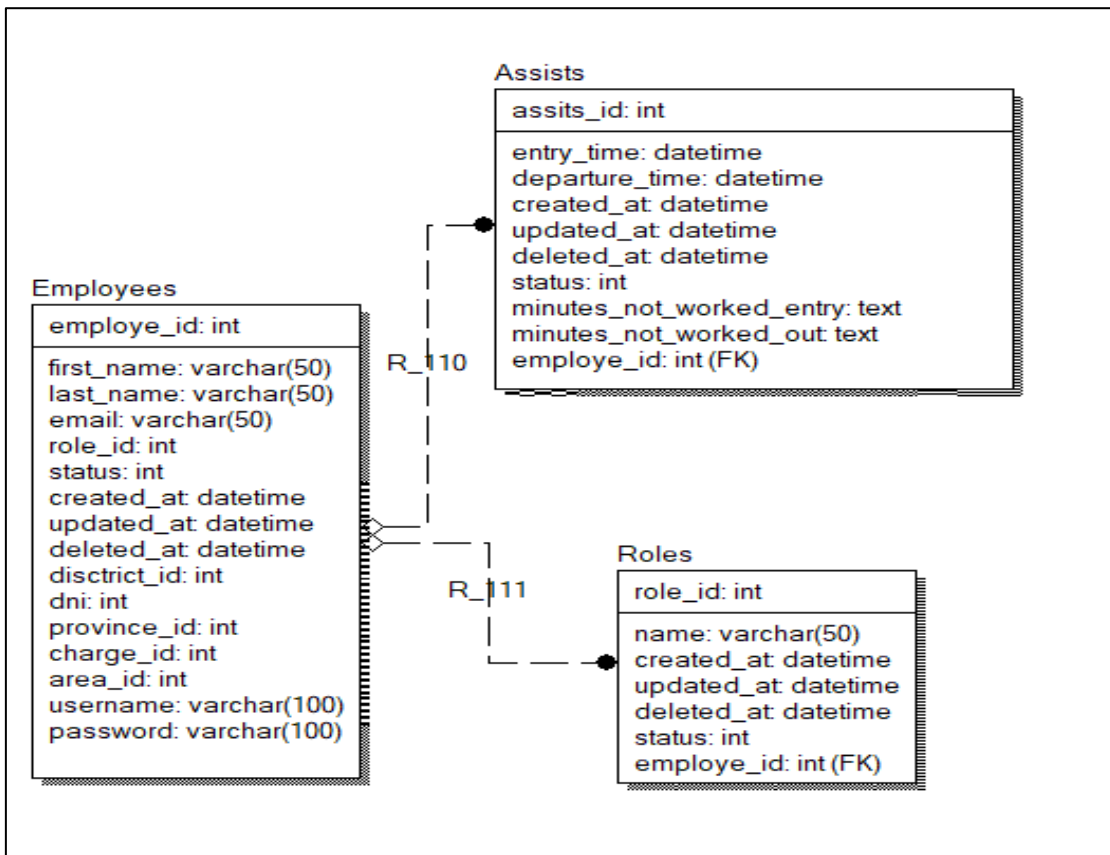


Figura 8. Diagrama físico de la base de datos RF2

Diseño

© Fuente: Elaboración propia



Figura 9. Prototipo de RF2

En la figura 9 se muestra el prototipo para la GUI de registrar asistencia aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

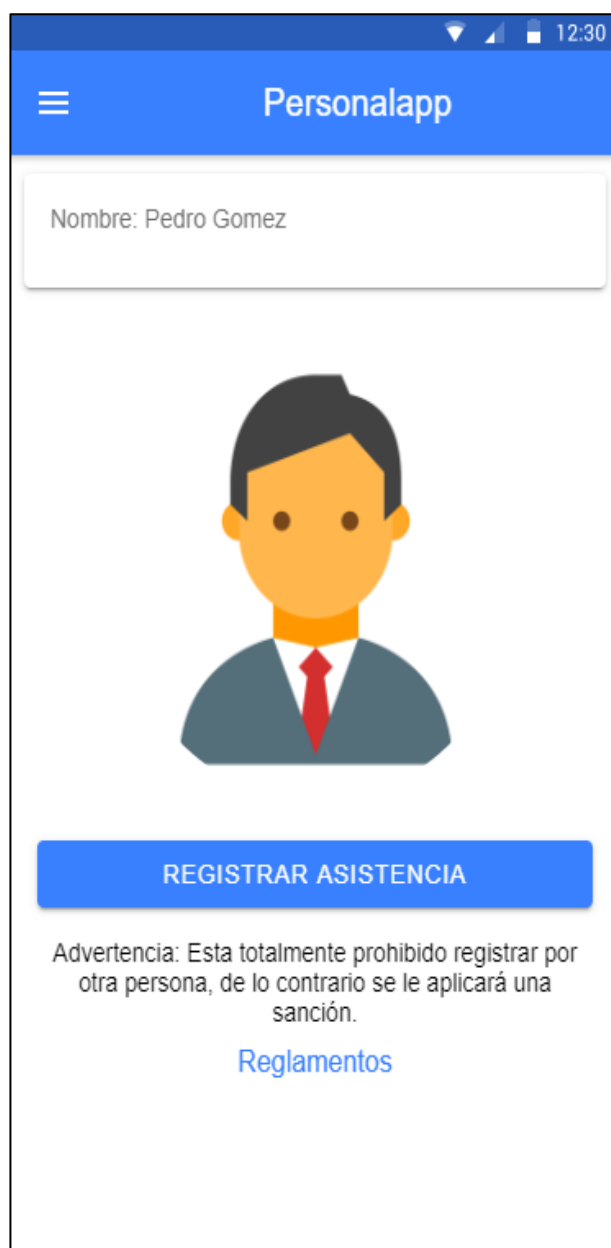


Figura 10. Interface de RF2

En la figura 10 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function assist(Request $request){
    $mytime = Carbon::now( tz: 'America/Lima');
    $fecha_actual= $mytime->createFromFormat( format: 'Y-m-d H:i:s', $mytime)->format( format: 'Y-m-d');
    $hora_de_entrada_establecida = Carbon::parse( time: $fecha_actual.'.'.'08:00:00', tz: 'America/Lima');
    if($mytime->toDateTimeString() < $fecha_actual.'.'.'08:00:00' ){
        $tiempo = 0;
    }else{
        $tiempo = $hora_de_entrada_establecida->diffInHours($mytime) . ':' . $mytime->diff($hora_de_entrada_establecida)
            ->format( format: '%I:%S');
    }
    $employee = new Assist();

    $employee->minutes_not_worked_entry = $tiempo;
    $employee->entry_time = $mytime->toDateTimeString();
    $employee->employee_id = $this->request->input( key: 'id');
    $employee->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 11. Código de RF2

En esta imagen 11 de la presente tesis se puede observar el código fuente del requerimiento funcional 2.

Requerimiento RF3 – Cerrar sesión

El requerimiento 3, tiene como objetivo que la aplicación debe permitir cerrar sesión. Dicho requerimiento debe hacerlo los usuarios “Administrador”, “Recursos Humanos” y “Obrero” que hayan iniciado sesión. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 3).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 198 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

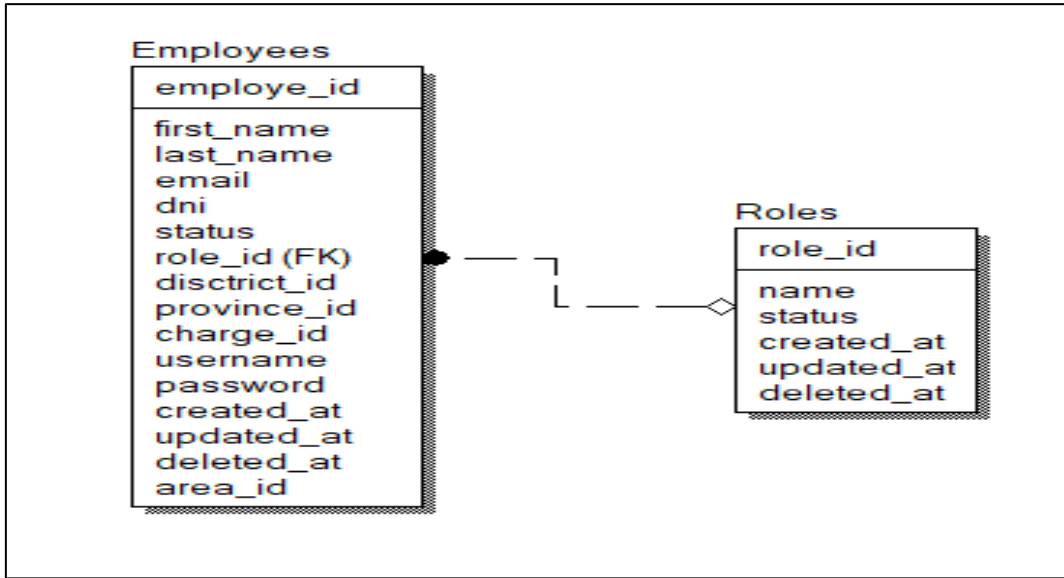


Figura 12. Diagrama lógico de la base de datos RF3

© Fuente: Elaboración propia

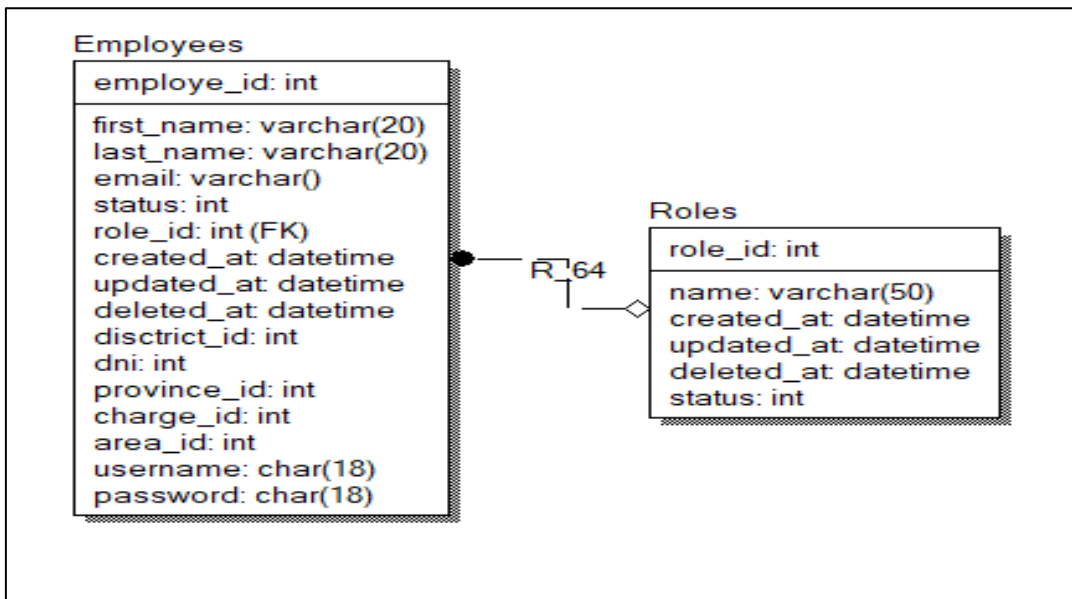


Figura 13. Diagrama físico de la base de datos RF3

Diseño

© Fuente: Elaboración propia

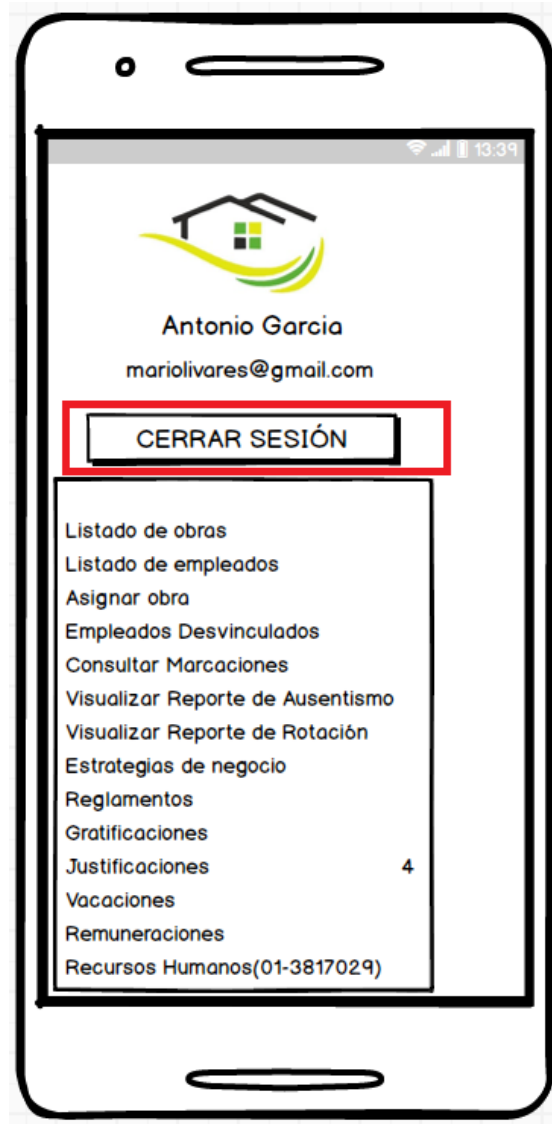


Figura 14. Prototipo de RF3

En la figura 14 se muestra el prototipo para la GUI de cerrar sesión de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

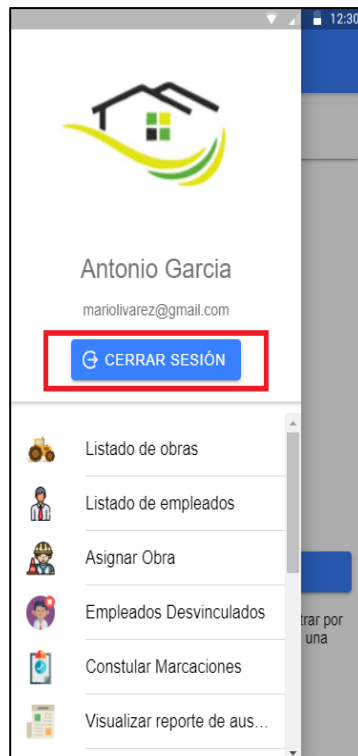


Figura 15. Interface de RF3

En la figura 15 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
async logout() {
  const alert = await this.alertController.create({
    header: 'Confirmación',
    message: 'Deseas salir de la aplicación?',
    buttons: [
      {
        text: 'Cancelar',
        role: 'Cancelar',
        cssClass: 'secondary',
        handler: (blah) => {
          console.log('Confirm Cancel: blah');
        }
      }, {
        text: 'Si',
        handler: () => {
          this.storage.clear();
          this.router.navigateByUrl('/login');
        }
      }
    ]
  });
  await alert.present();
}
```

Figura 16. Código de RF3

En esta imagen 16 de la presente tesis se puede observar el código fuente del requerimiento funcional 3.

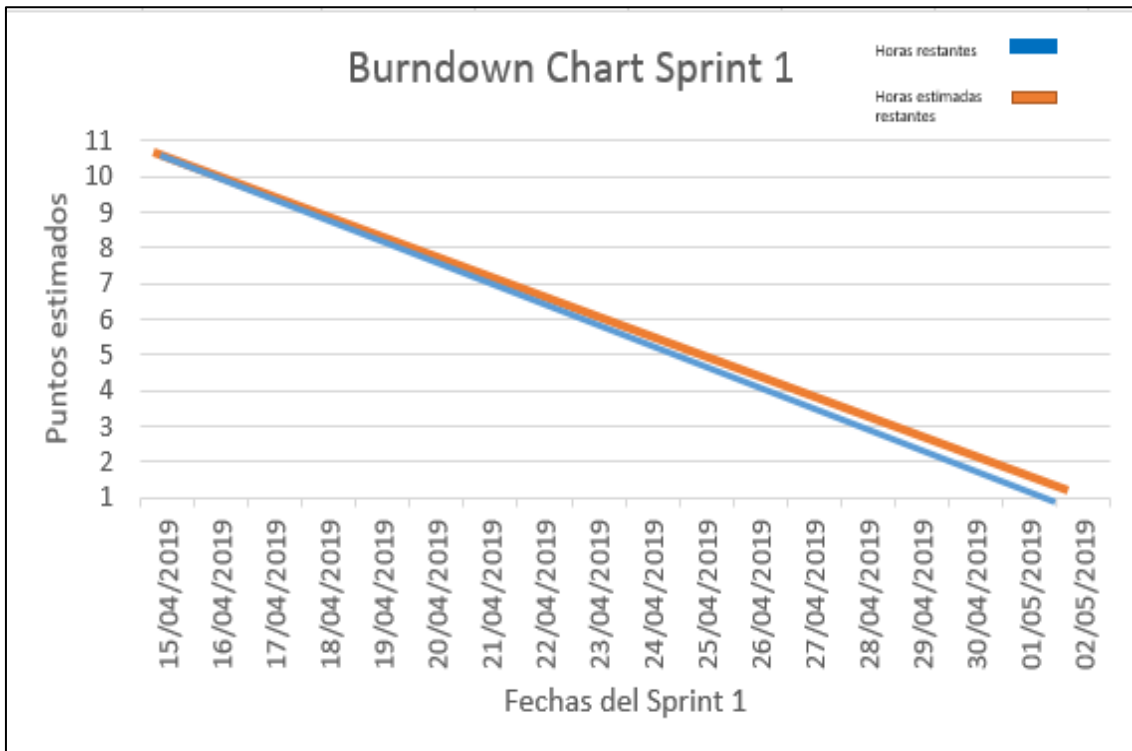


Figura 17. Burndown Chart Sprint 1


Para este sprint, se puede apreciar en esta figura 17, que el tiempo real de Scrum (línea naranja, ya que así dice en su documentación y nosotros que estamos laborando en consultoría también utilizamos este Burndown Chart ya que trabajamos con Scrum y sabemos cómo se usa) va acorde al tiempo planificado (línea azul, en el cual es el tiempo que se demora el equipo de desarrollo en acabar sus actividades), e incluso se terminó mucho antes de lo planificado. Bajo este escenario se puede determinar que se ha terminado todo correctamente, dando paso al siguiente sprint.



CONSTRUCCIONES & ACABADOS ZAMORA

Acta de reunión de entrega del Sprint 1

Acta de reunión de entrega del Sprint 1		
Fecha	29/04/2019	
Hora	3:00 pm	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Responsable	Lorenzo Palacios	Scrum Master
Participantes	Javier Zamora	Product Owner
	Maurihño Ecurra	Analista
	Luis Zamora	Programador
	Julio Sosa	Calidad
Tema	Presentación del Sprint 1	
Resumen de la reunión		
En esta reunión se mostraron las pantallas de la aplicación a nivel funcional de acuerdo a las historias pactadas en la Planificación del Sprint 1.		
El cliente quedo satisfecho con el trabajo realizado y dio el visto bueno, dando paso al siguiente sprint.		

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

JAVIER ZAMORA
FRONTE GENERAL

© Fuente: Elaboración propia

Figura 18. Acta de reunión de entrega del Sprint 1

El tema de reunión de esta acta es sobre la presentación de entrega del sprint 1.

Resumen del Sprint 1

Tabla 72. *Resumen del Sprint 1*

Total, de historias	3
Historias terminadas	3
Historias por terminar	0
Avance	100%

Fuente: Elaboración propia

Las pruebas realizadas a las funcionalidades durante este sprint fueron exitosas sin ningún error de diseño ni de código fuente.

Retrospectiva del Sprint 1

Al final del Sprint, el equipo Scrum se reunió para recibir la respuesta del Scrum Master, para saber cómo le fue en la reunión con el Product Owner, resulta que el producto se entregó sin problemas y con el cliente satisfecho.

Cosas Positivas:

- Lograron cumplir los resultados
- El apoyo del equipo en todo momento

Cosas Negativas:

- Ninguna

Lista de Pendientes de Sprint (Sprint Backlog)

Sprint 2

Tabla 73: *Sprint 2*

Nº Sprint	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
Sprint 2	RF4: La aplicación permitirá crear obras de trabajo.	H4	3	2
	RF5: La aplicación permitirá asignar una obra a cada obrero.	H5	3	2
	RF6: La aplicación permitirá actualizar los datos de cada obra.	H6	3	2

Fuente: Elaboración propia

Aquí está todas las funcionalidades a realizar en el sprint 2.

© Fuente: Elaboración propia

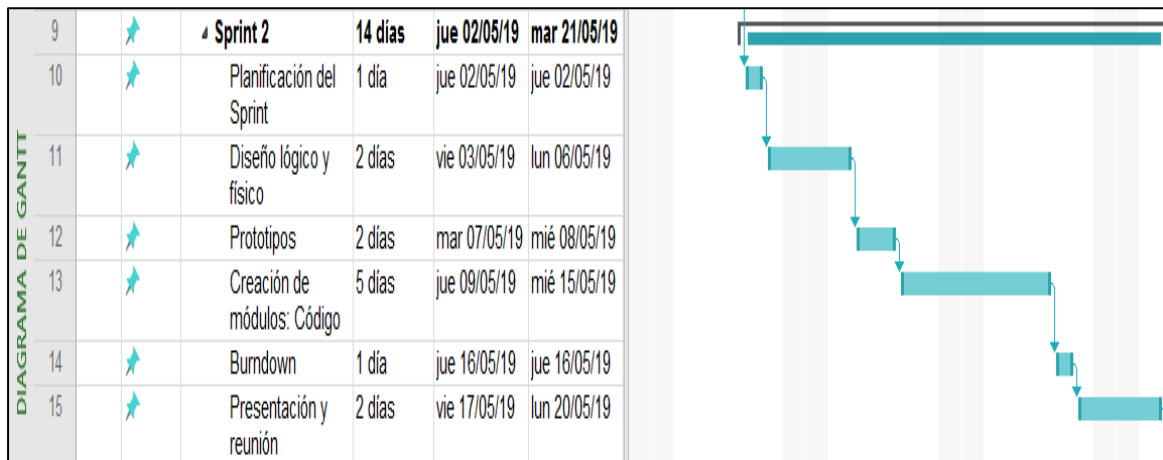


Figura 19. Planificación Sprint 2

Aquí en esta figura 19 está el cronograma de las actividades a realizar en el sprint 2.

A continuación, en este apartado de nuestra tesis, se podrá evidenciar todo el proceso de la implementación de la app móvil divididos por cada funcionalidad del sprint 2. En el cual nosotros hemos elaborado un texto descriptivo y cuatro imágenes por funcionalidad para que ustedes como lectores puedan entender bien todo el proceso de implementación de la aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., todas las imágenes que hemos elaborado están en orden secuencial y por cada funcionalidad pero no todas las imágenes están con sus respectivas descripciones, por eso vamos a explicar en este texto como es el orden y descripción de cada imagen por funcionalidad: una vez explicado el objetivo del requerimiento funcional podrán observar en las dos imágenes siguientes la base de datos lógico y físico de las tablas involucradas de dicha funcionalidad (el proceso para llegar al diseño de la base de datos lógico y físico y tablas involucradas que son las entidades fue que se obtuvo durante el análisis de la segunda actividad que se puede visualizar en el cronograma del sprint 2 y de las descripciones de las historias de usuario), luego se mostrará la imagen del diseño de los prototipos elaborados con la herramienta Balsamiq Mockups (estos diseños fueron validados y aceptados por la empresa), luego ustedes podrán observar la imagen de implementación de las interfaces y por último puedan ver la imagen del código fuente de dicha funcionalidad de la app móvil. Todos los requerimientos funcionales de este sprint 2 están enumerados y por ende se sobre entiende que las imágenes que van a observar pertenecen a las funcionalidades ya que están todas enumeradas.

Requerimiento RF4 – Crear obra

El requerimiento 4, tiene como objetivo que la aplicación debe permitir crear una obra a los usuarios “Administrador” con una fecha de inicio y fin. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 4).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 214 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

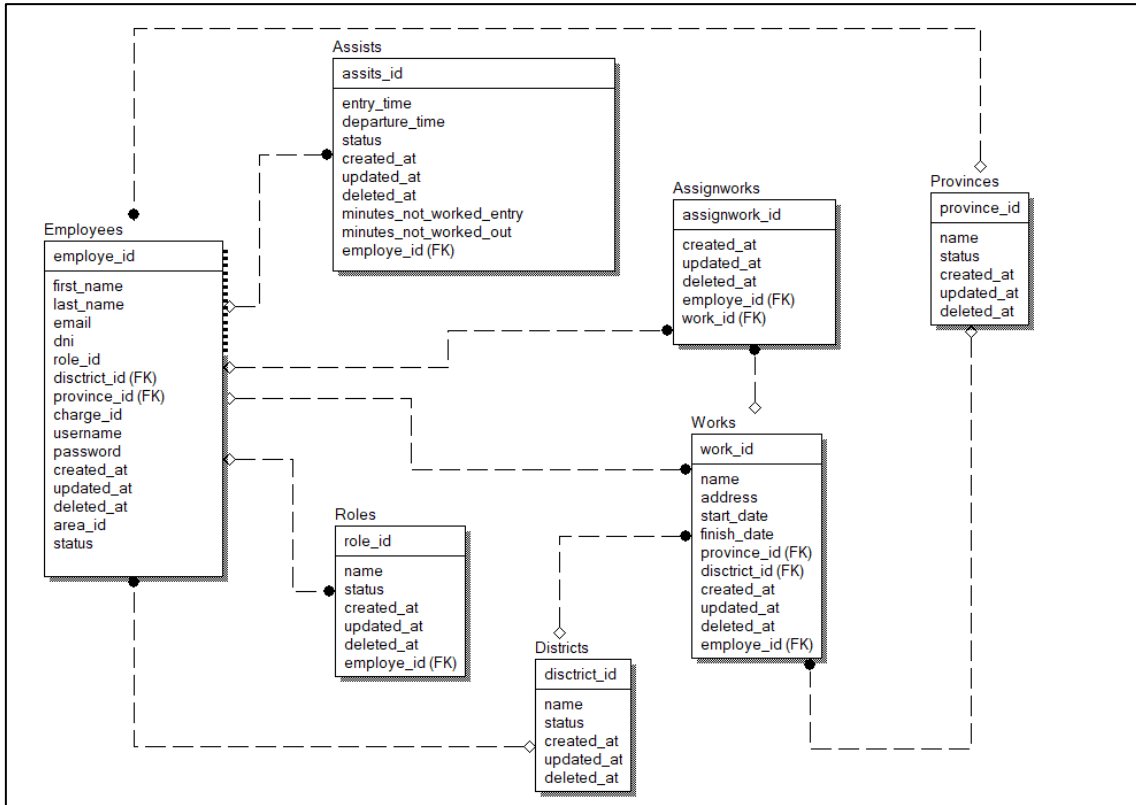


Figura 20. Diagrama lógico de la base de datos RF4

© Fuente: Elaboración propia

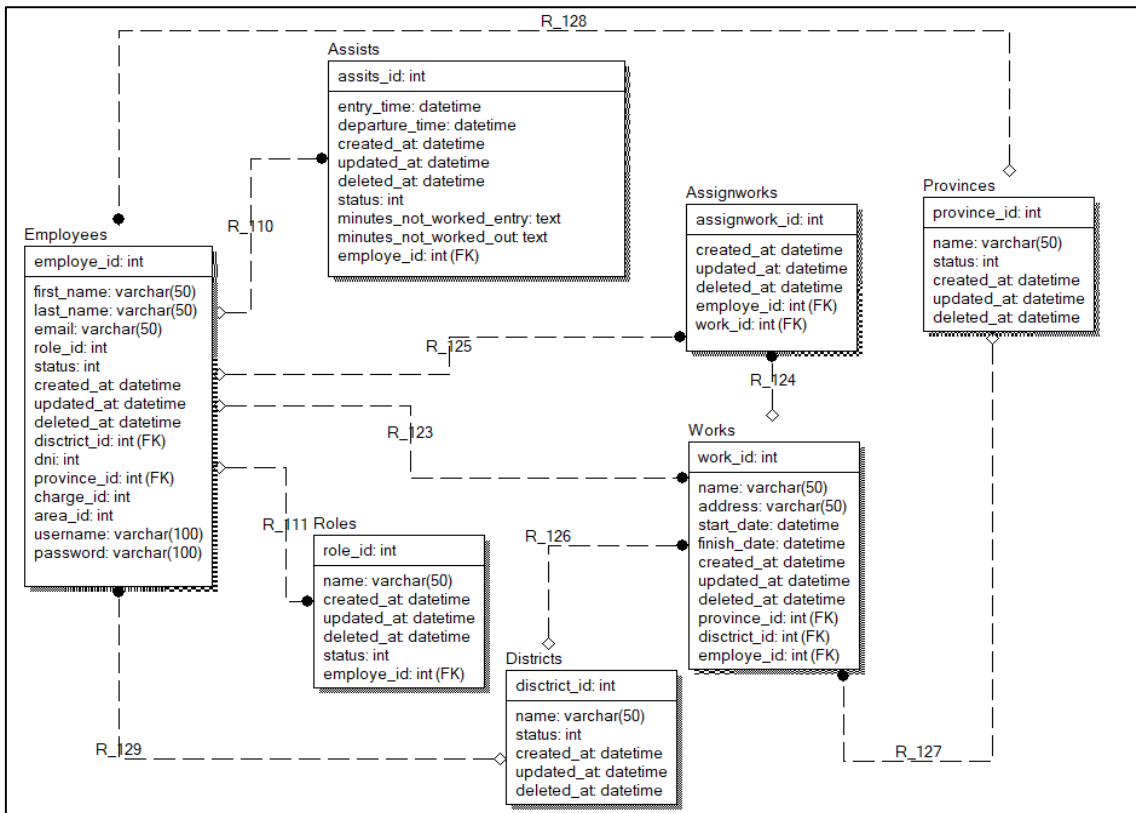


Figura 21. Diagrama físico de la base de datos RF4

Diseño

© Fuente: Elaboración propia

El prototipo muestra una pantalla de un teléfono móvil con el título "CREAR OBRA" en la parte superior. A la izquierda hay un icono de menú (tres líneas horizontales). El formulario contiene los siguientes campos:

- name:
- address:
- start date:
- finish date:
- provincia:
- distrito:

En la parte inferior del formulario hay un botón que dice "CREAR OBRA".

Figura 22. Prototipo de RF4

En la figura 22 se muestra el prototipo para la GUI de crear obra aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

The screenshot shows a mobile application interface titled 'Personalapp' with a blue header. Below the header, the main title is 'Crear Obra'. The form contains several input fields: 'Nombre' (with a placeholder 'Nombre'), 'Direccion' (with a placeholder 'Dirección'), 'Fecha de inicio', and 'Fecha de fin'. There are two dropdown menus for 'Provincia' and 'Distrito', both with the placeholder text 'Seleccione...'. At the bottom of the form is a blue button labeled 'CREAR OBRA'.

Figura 23. Interface de RF4

En la figura 23 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function save(Request $request){
    $work = new Work();
    $work->name = $request->full_name;
    $work->address = $request->direction;
    $work->start_date = $request->date_start;
    $work->finish_date = $request->date_end;
    $work->province_id = $request->province_id;
    $work->district_id = $request->district_id;
    $work->employe_id = 1;
    $work->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 24. Código de RF4

En esta imagen 24 de la presente tesis se puede observar el código fuente del requerimiento funcional 4.

Requerimiento RF5 – Asignar obra

El requerimiento 5, tiene como objetivo que la aplicación debe permitir asignar solo una obra en ese momento a cada obrero, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 4).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 214 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

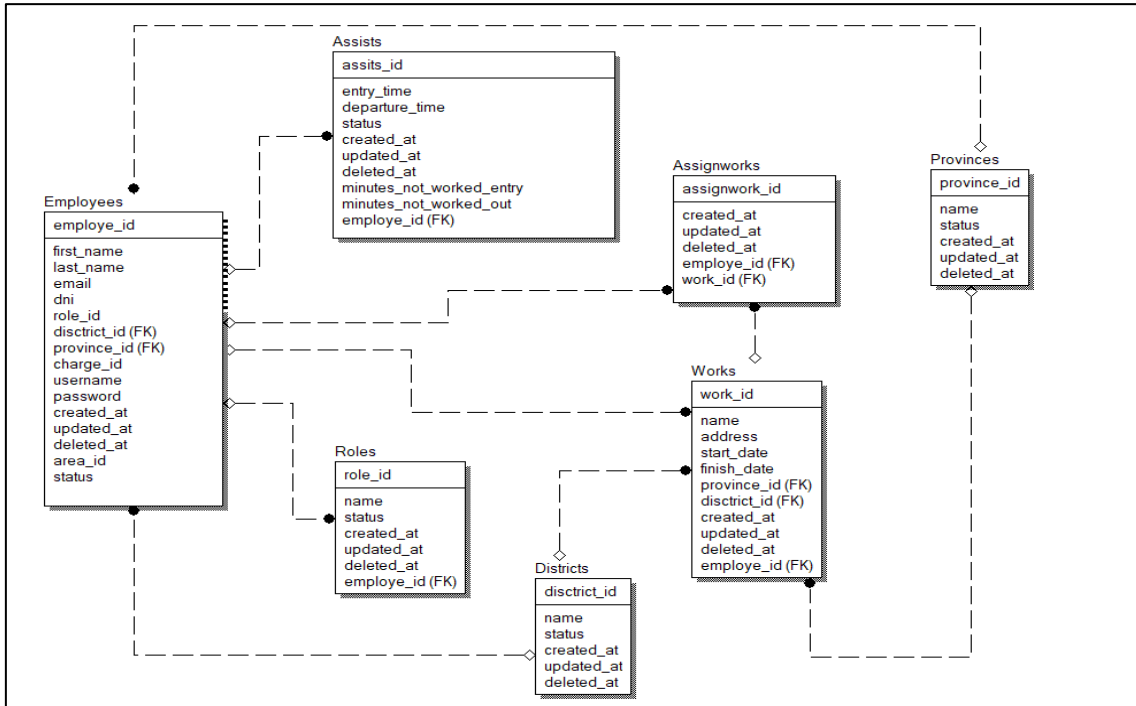


Figura 25. Diagrama lógico de la base de datos RF5

© Fuente: Elaboración propia

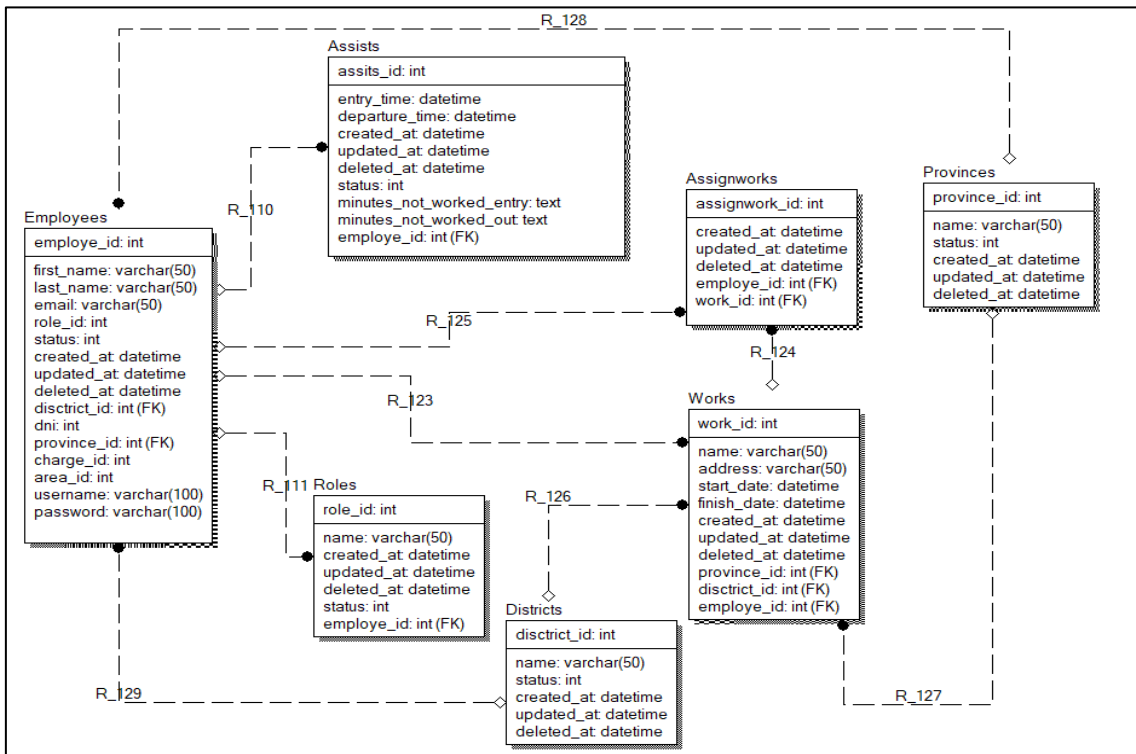


Figura 26. Diagrama físico de la base de datos RF5

Diseño

© Fuente: Elaboración propia

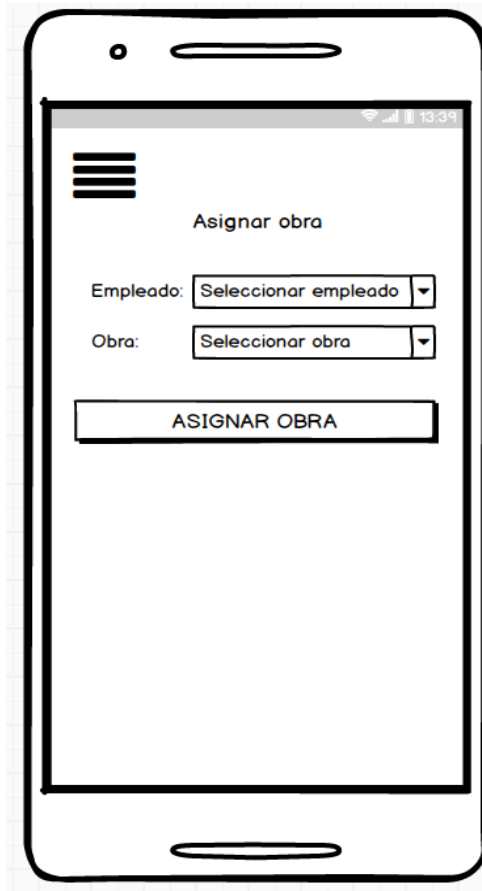
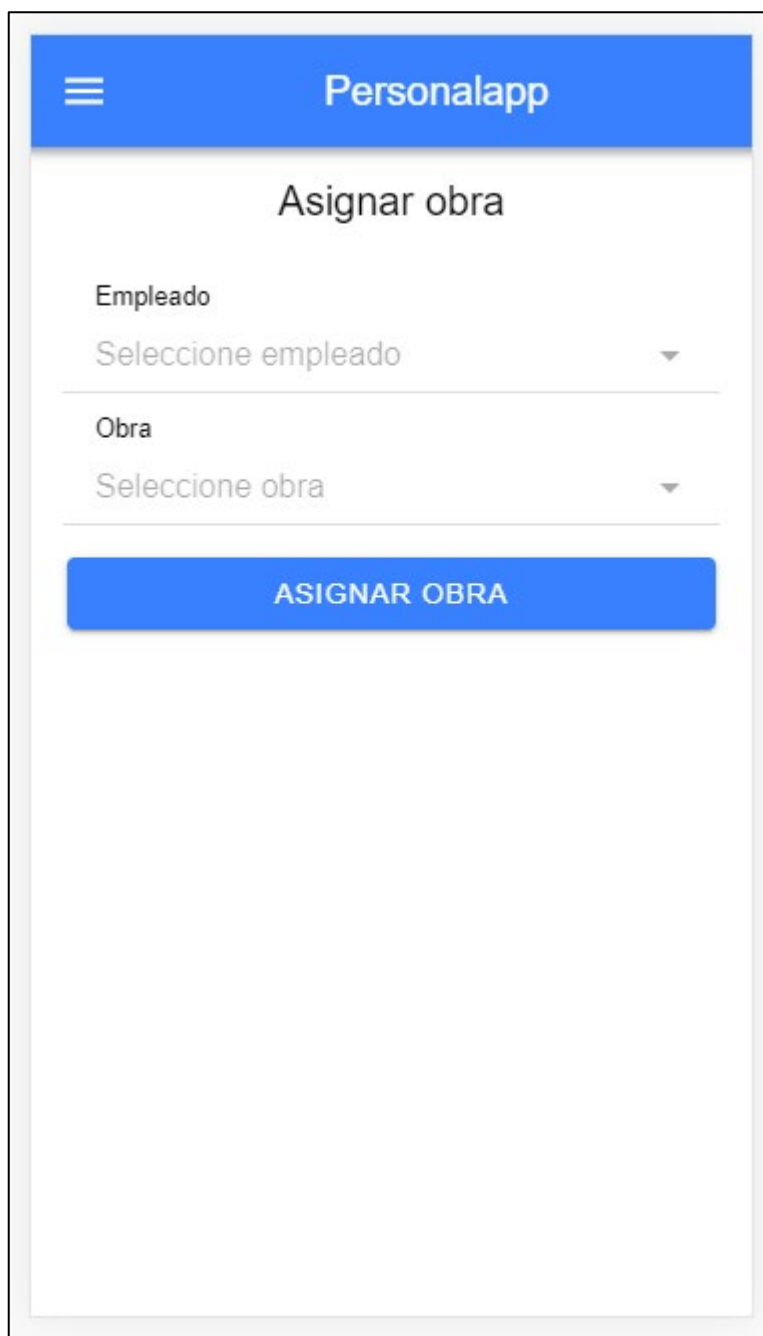


Figura 27. Prototipo de RF5

En la figura 27 se muestra el prototipo para la GUI de asignar obra aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. The main heading is 'Asignar obra'. Below this, there are two dropdown menus: 'Empleado' with the placeholder text 'Seleccione empleado' and 'Obra' with the placeholder text 'Seleccione obra'. At the bottom of the form is a blue button labeled 'ASIGNAR OBRA'.

Figura 28. Interface de RF5

En la figura 28 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function assignWork(Request $request) {
    $work = new Assignwork();
    $work->employee_id = $request->employee_id;
    $work->work_id = $request->work_id;
    $work->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 29. Código de RF5

En esta imagen 29 de la presente tesis se puede observar el código fuente del requerimiento funcional 5.

Requerimiento RF6 – Actualizar obra

El requerimiento 6, tiene como objetivo que la aplicación debe permitir actualizar la información de una obra que ya ha sido creada anteriormente, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 6).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 214 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

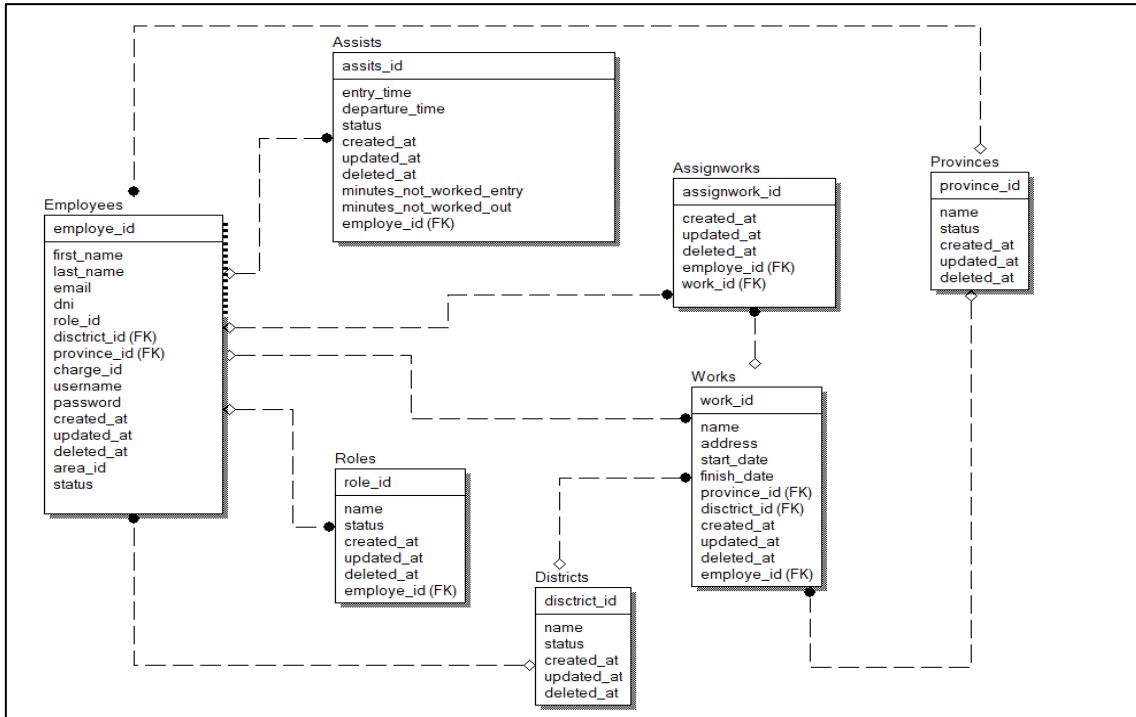


Figura 30. Diagrama lógico de la base de datos RF6

© Fuente: Elaboración propia

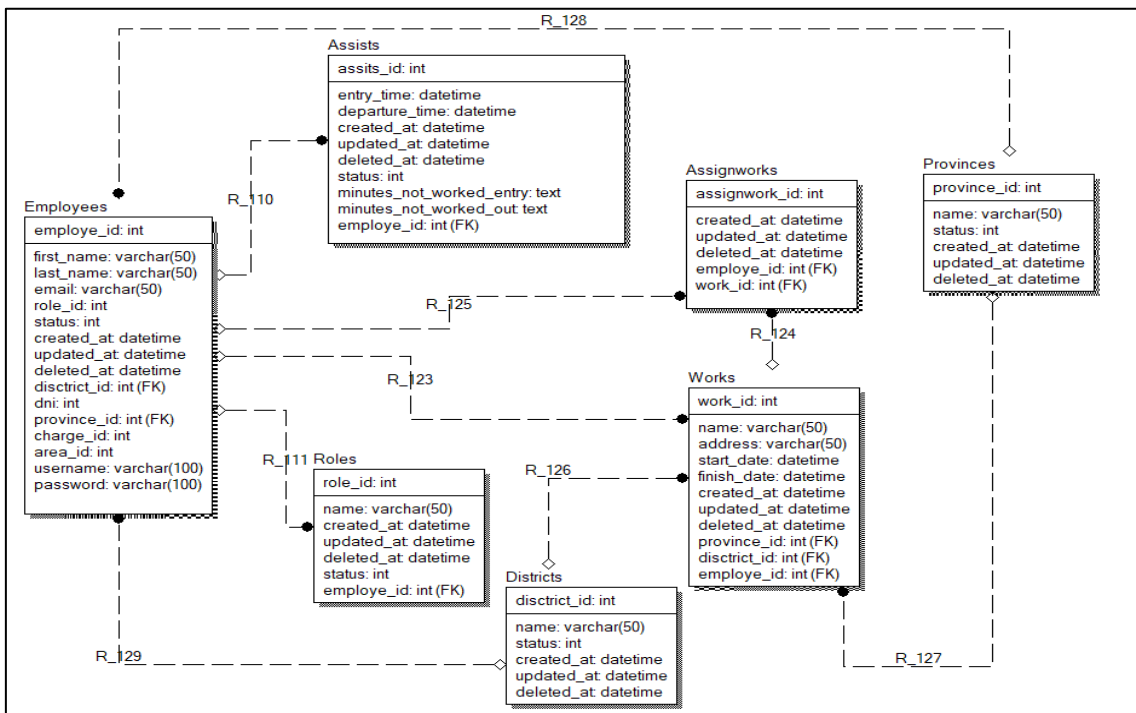


Figura 31. Diagrama físico de la base de datos RF6

Diseño

© Fuente: Elaboración propia

ACTUALIZAR OBRA

Nombre:

Dirección:

Fecha inicio:

Fecha fin:

Provincia: ▼

Distrito: ▼

The image shows a mobile application prototype for updating a work record. The screen is titled 'ACTUALIZAR OBRA'. It contains several input fields: 'Nombre' with the value 'Andahuaylas', 'Dirección' with 'Jr Huarmey #768', 'Fecha inicio' with '18/05/2019' and a calendar icon, 'Fecha fin' with '25/07/2020' and a calendar icon, 'Provincia' with a dropdown menu showing 'Ancash', and 'Distrito' with a dropdown menu showing 'Apurama'. At the bottom, there is a button labeled 'ACTUALIZAR OBRA'. The entire form is enclosed in a rounded rectangle representing a smartphone screen.

Figura 32. Prototipo de RF6

En la figura 32 se muestra el prototipo para la GUI de actualizar obra aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

Personalapp

Actualizar Obra

Nombre Andahuaylas

Dirección Jr Huarmey #768

Fecha de inicio 2019-05-18

Fecha de fin 2019-05-18

Provincia Ancash ▾

Distrito Apurama ▾

ACTUALIZAR OBRA

Figura 33. Interface de RF6

En la figura 33 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function update(Request $request) {
    $work = new Work();
    $work->name = $request->full_name;
    $work->address = $request->direction;
    $work->start_date = $request->date_start;
    $work->finish_date = $request->date_end;
    $work->province_id = $request->province_id;
    $work->district_id = $request->district_id;
    $work->employe_id = 1;
    $work->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 34. Código de RF6

© Fuente: Elaboración propia

En esta imagen 34 de la presente tesis se puede observar el código fuente del requerimiento funcional 6.

© Fuente: Elaboración propia

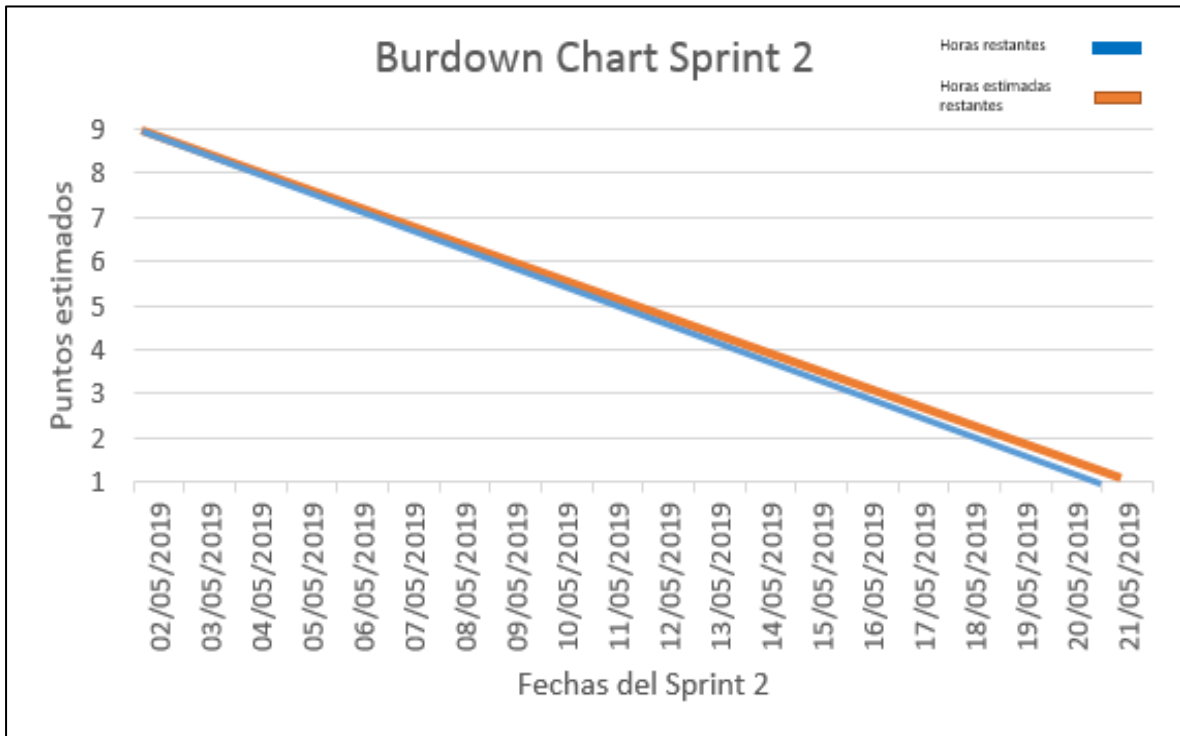


Figura 35. Burndown Chart Sprint 2

Para este sprint, se puede apreciar en esta figura 35, que el tiempo real de Scrum (línea naranja, ya que así dice en su documentación y nosotros que estamos laborando en consultoría también utilizamos este Burndown Chart ya que trabajamos con Scrum y sabemos cómo se usa) va acorde al tiempo planificado (línea azul, en el cual es el tiempo que se demora el equipo de desarrollo en acabar sus actividades), e incluso se terminó mucho antes de lo planificado. Bajo este escenario se puede determinar que se ha terminado todo correctamente, dando paso al siguiente sprint.



Acta de reunión de entrega del Sprint 2

Acta de reunión de entrega del Sprint 2		
Fecha	17/05/2019	
Hora	3:00 pm	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Responsable	Lorenzo Palacios	Scrum Master
Participantes	Javier Zamora	Product Owner
	Maurihño Escurra	Analista
	Luis Zamora	Programador
	Julio Sosa	Calidad
Tema	Presentación del Sprint 2	
Resumen de la reunión		
<p>En esta reunión se mostraron las pantallas de la aplicación a nivel funcional de acuerdo a las historias pactadas en la Planificación del Sprint 2.</p> <p>El cliente quedo satisfecho con el trabajo realizado y dio el visto bueno, dando paso al siguiente sprint.</p>		

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

© Fuente: Elaboración propia

Figura 36. Acta de reunión de entrega del Sprint 2

El tema de reunión de esta acta es sobre la presentación de entrega del sprint 2.

Resumen del Sprint 2

Tabla 74. *Resumen del Sprint 2*

Total, de historias	3
Historias terminadas	3
Historias por terminar	0
Avance	100%

Fuente: Elaboración propia

Las pruebas realizadas a las funcionalidades durante este sprint fueron exitosas sin ningún error de diseño ni de código fuente.

Retrospectiva del Sprint 2

Al final del Sprint, el equipo Scrum se reunió para recibir la respuesta del Scrum Master, para saber cómo le fue en la reunión con el Product Owner, resulta que el producto se entregó sin problemas y con el cliente satisfecho.

Cosas Positivas:

- Lograron cumplir los resultados
- El apoyo del equipo en todo momento

Cosas Negativas:

- Ninguna

Lista de Pendientes de Sprint (Sprint Backlog)

Sprint 3

Tabla 75. *Sprint 3*

Nº Sprint	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
Sprint 3	RF7: La aplicación permitirá eliminar una obra.	H7	3	3
	RF8: La aplicación permitirá registrar un empleado, administrador, obrero o recursos humanos.	H8	3	3
	RF9: La aplicación permitirá actualizar un empleado.	H9	3	3
	RF10: La aplicación permitirá eliminar un empleado.	H10	3	3
	RF11: Esta funcionalidad permitirá colocar en un estado de desvinculado a los empleados.	H11	3	3

Fuente: Elaboración propia

Aquí está todas las funcionalidades a realizar en el sprint 3.

© Fuente: Elaboración propia

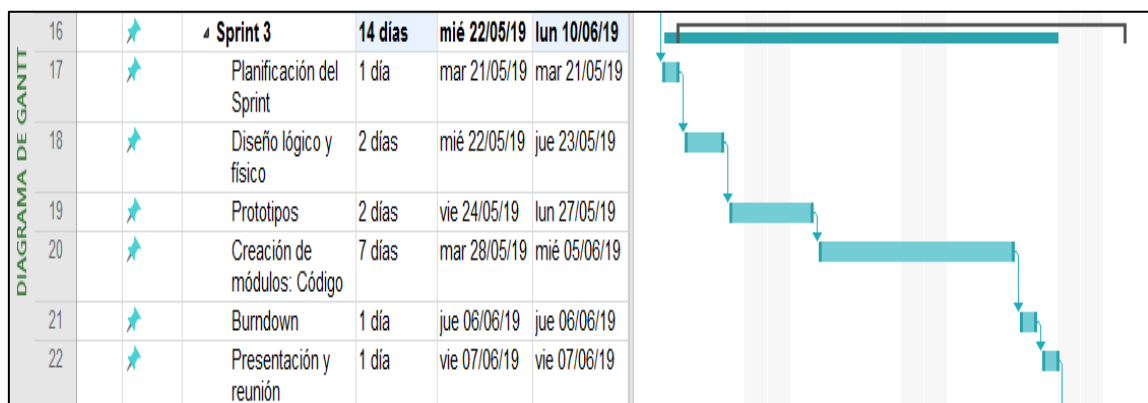


Figura 37. Planificación Sprint 3

Aquí en esta figura 37 está el cronograma de las actividades a realizar en el sprint 3.

A continuación, en este apartado de nuestra tesis, se podrá evidenciar todo el proceso de la implementación de la app móvil divididos por cada funcionalidad del sprint 3. En el cual nosotros hemos elaborado un texto descriptivo y cuatro imágenes por funcionalidad para que ustedes como lectores puedan entender bien todo el proceso de implementación de la aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., todas las imágenes que hemos elaborado están en orden secuencial y por cada funcionalidad pero no todas las imágenes están con sus respectivas descripciones, por eso vamos a explicar en este texto como es el orden y descripción de cada imagen por funcionalidad: una vez explicado el objetivo del requerimiento funcional podrán observar en las dos imágenes siguientes la base de datos lógico y físico de las tablas involucradas de dicha funcionalidad (el proceso para llegar al diseño de la base de datos lógico y físico y tablas involucradas que son las entidades fue que se obtuvo durante el análisis de la segunda actividad que se puede visualizar en el cronograma del sprint 3 y de las descripciones de las historias de usuario), luego se mostrará la imagen del diseño de los prototipos elaborados con la herramienta Balsamiq Mockups (estos diseños fueron validados y aceptados por la empresa), luego ustedes podrán observar la imagen de implementación de las interfaces y por último puedan ver la imagen del código fuente de dicha funcionalidad de la app móvil. Todos los requerimientos funcionales de este sprint 3 están enumerados y por ende se sobre entiende que las imágenes que van a observar pertenecen a las funcionalidades ya que están todas enumeradas.

Requerimiento RF7 – Eliminar obra

El requerimiento 7, tiene como objetivo que la aplicación debe poder eliminar una obra que ya ha sido creada anteriormente, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 7).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 230 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

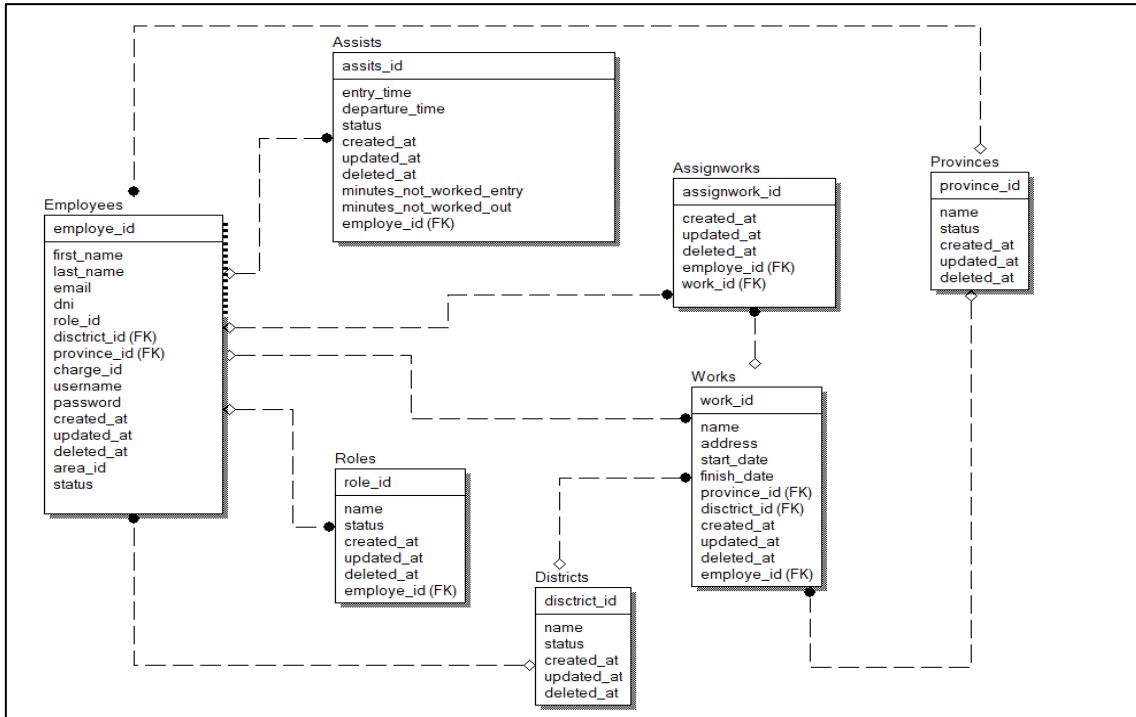


Figura 38. Diagrama lógico de la base de datos RF7

© Fuente: Elaboración propia

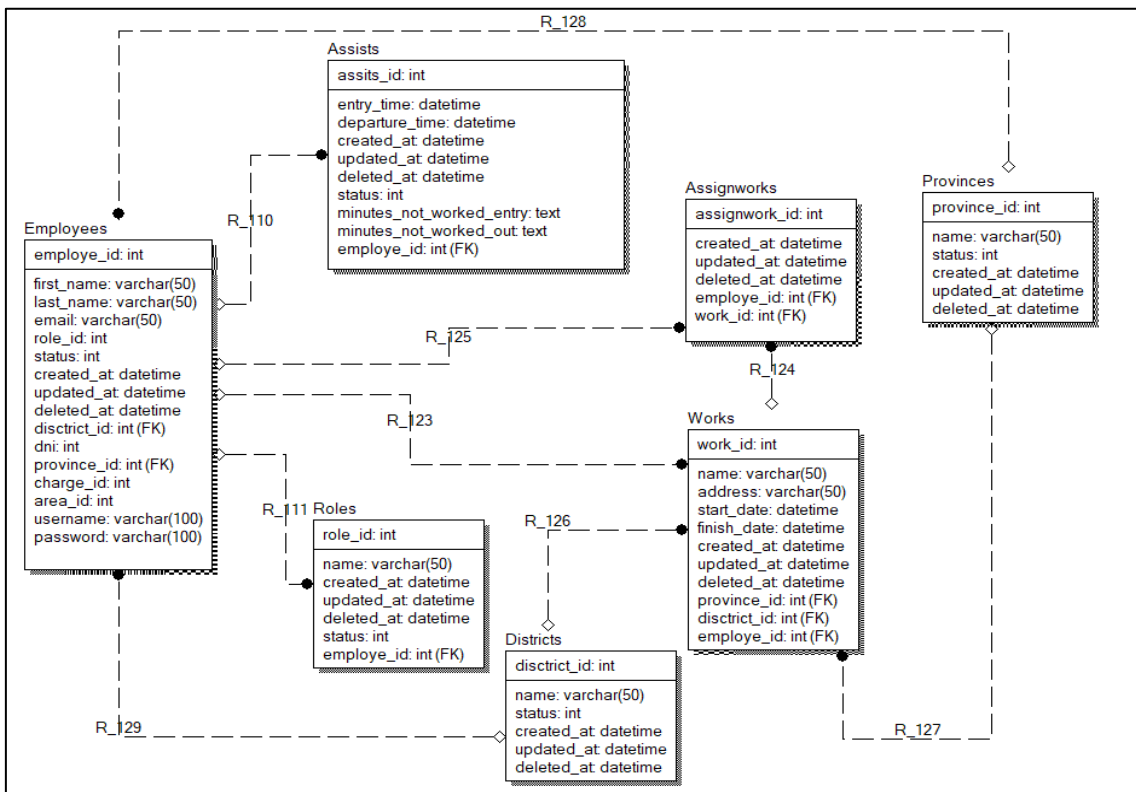


Figura 39. Diagrama físico de la base de datos RF7

Diseño

© Fuente: Elaboración propia

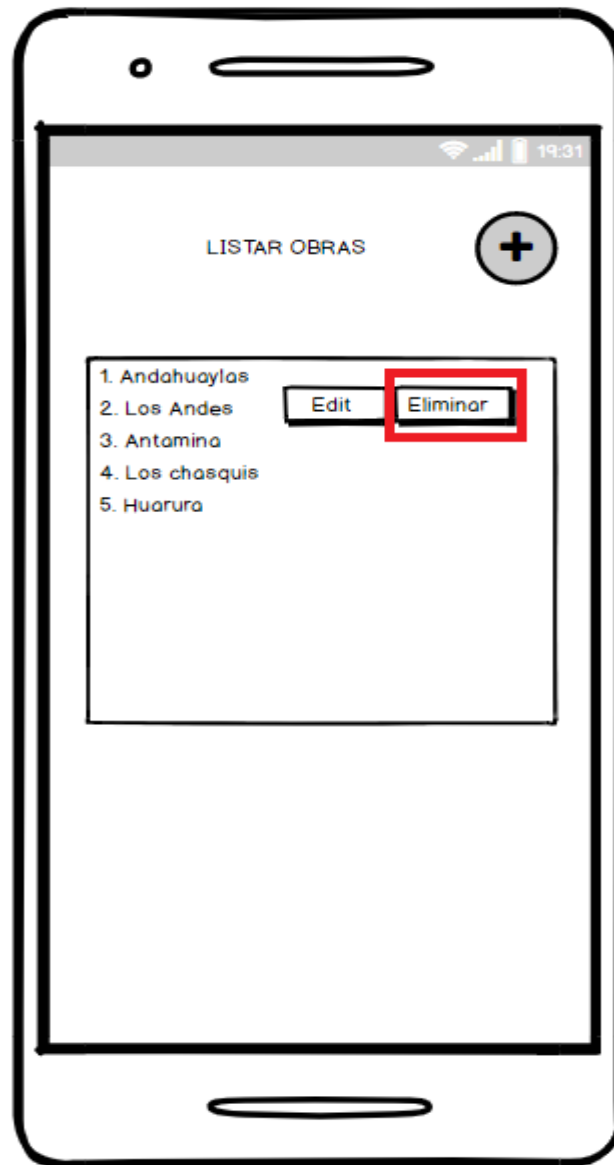


Figura 40. Prototipo de RF7

En la figura 40 se muestra el prototipo para la GUI de eliminar obra aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

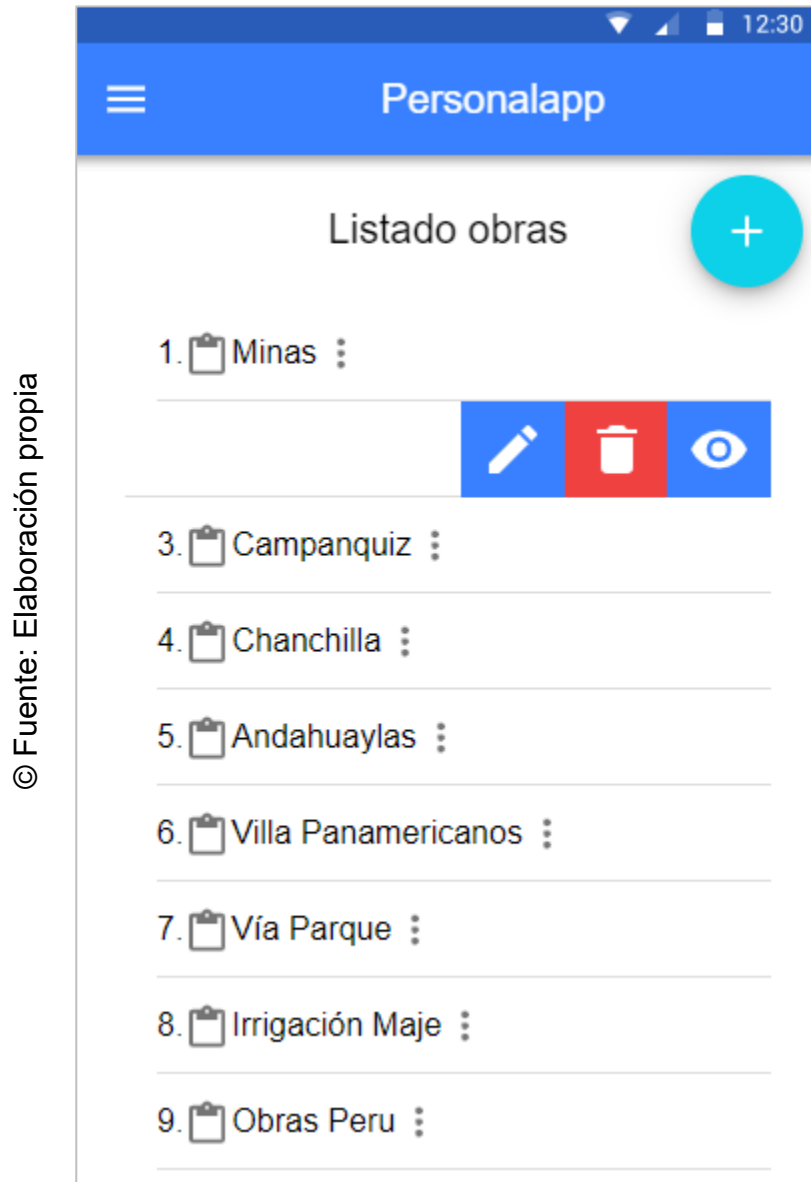


Figura 41. Interface de RF7

En la figura 41 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function deleteWork(Request $request){
    Work::destroy($this->request->input('key: work_id'));
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 42. Código de RF7

En esta imagen 42 de la presente tesis se puede observar el código fuente del requerimiento funcional 7.

Requerimiento RF8 – Registrar empleado

El requerimiento 8, tiene como objetivo que la aplicación debe poder registrar un nuevo empleado, con todos sus datos necesarios y a la misma vez se debe poder crear un usuario y una contraseña para que pueda ingresar a la aplicación, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 8).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 230 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

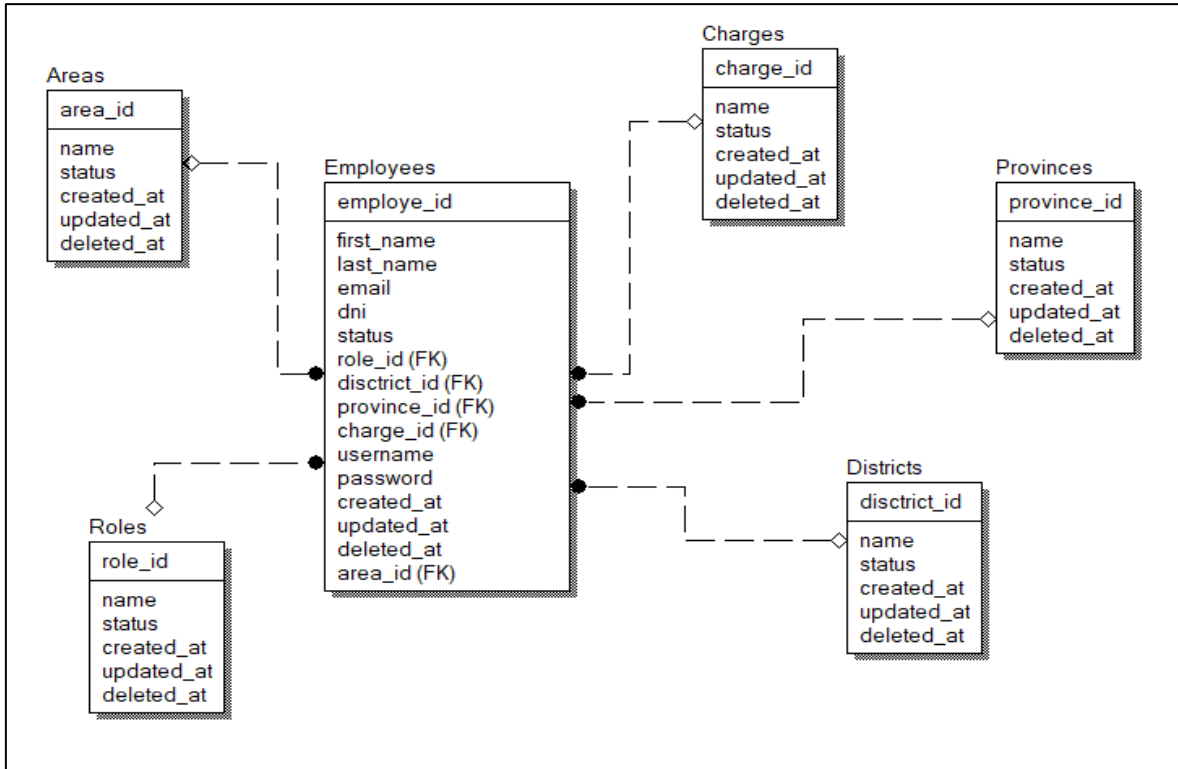


Figura 43. Diagrama lógico de la base de datos RF8

© Fuente: Elaboración propia

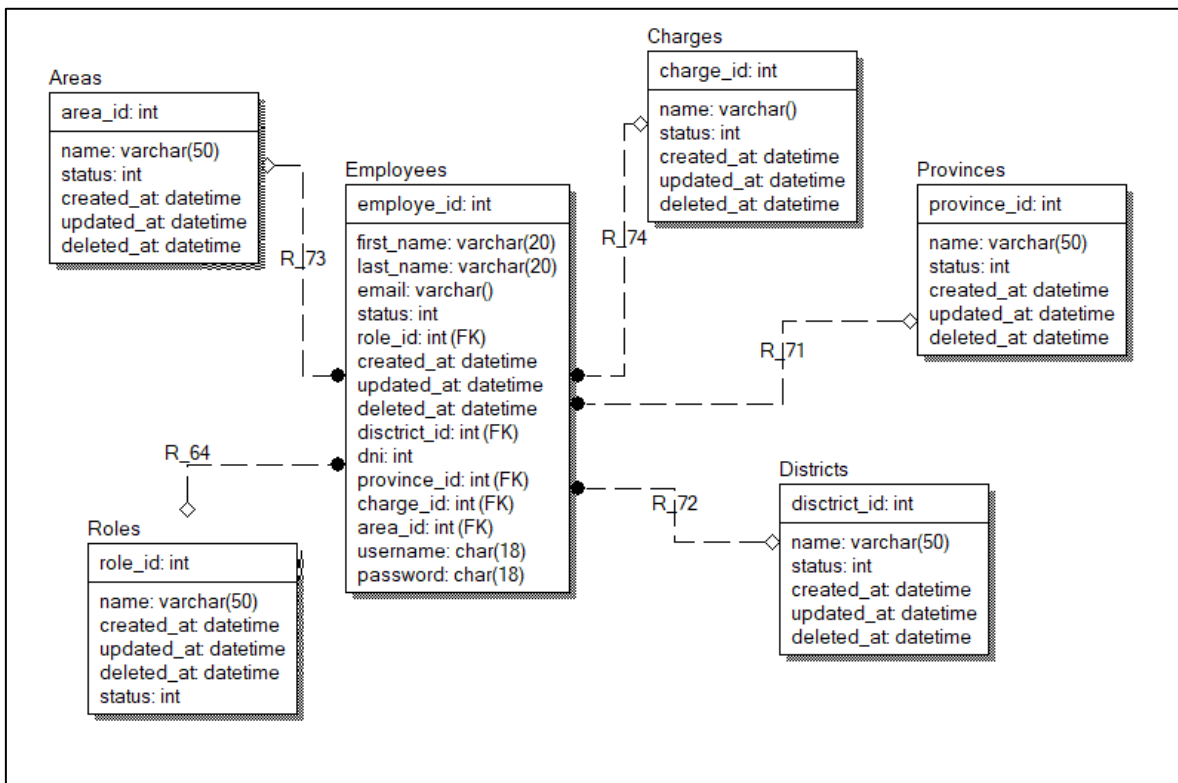
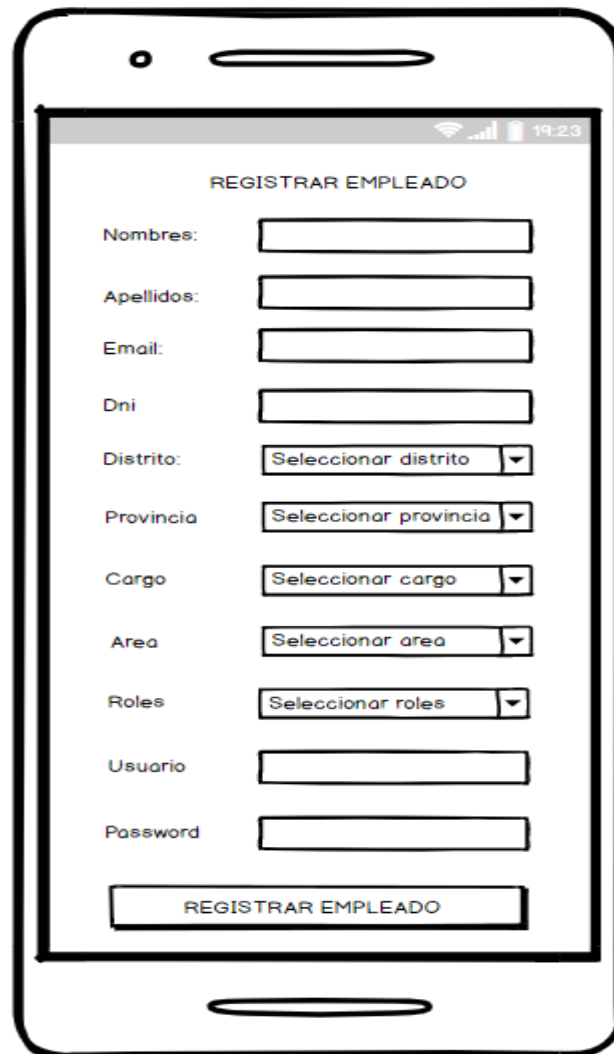


Figura 44. Diagrama físico de la base de datos RF8

Diseño

© Fuente: Elaboración propia



REGISTRAR EMPLEADO

Nombres:

Apellidos:

Email:

Dni:

Distrito:

Provincia:

Cargo:

Area:

Roles:

Usuario:

Password:

Figura 45. Prototipo de RF8

En la figura 45 se muestra el prototipo para la GUI de registrar empleado aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

The screenshot shows a mobile application interface titled 'Personalapp' with a blue header bar containing a back arrow and the title. Below the header, the main title 'Registrar Empleado' is centered. The form consists of several input fields and dropdown menus, each with a label above and a placeholder text below:

- Nombres**: Input field with placeholder 'Nombre'.
- Apellidos**: Input field with placeholder 'Dirección'.
- Email**: Input field with placeholder 'email'.
- Dni**: Input field with placeholder 'dni'.
- Distrito**: Dropdown menu with placeholder 'Seleccione distrito'.
- Provincia**: Dropdown menu with placeholder 'Seleccione provincia'.
- Cargo**: Dropdown menu with placeholder 'Seleccione cargo'.
- Area**: Dropdown menu with placeholder 'Seleccione area'.
- Roles**: Dropdown menu with placeholder 'Seleccione roles'.
- Usuario**: Input field with placeholder 'Usuario'.
- Contraseña**: Input field with placeholder 'Contraseña'.

At the bottom of the form is a blue button labeled 'CREAR EMPLEADO'.

Figura 46. Interface de RF8

En la figura 46 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function save(Request $request) {
    $employee = new Employee();
    $employee->first_name = $request->first_name;
    $employee->last_name = $request->last_name;
    $employee->email = $request->email;
    $employee->status = 1;
    $employee->district_id = $request->district_id;
    $employee->dni = $request->dni;
    $employee->province_id = $request->province_id;
    $employee->charge_id = $request->charge_id;
    $employee->area_id = $request->area_id;
    $employee->user_id = $request->user_id;
    $employee->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 47. Código de RF8

En esta imagen 47 de la presente tesis se puede observar el código fuente del requerimiento funcional 8.

Requerimiento RF9 – Actualizar empleado

El requerimiento 9, tiene como objetivo que la aplicación debe poder actualizar un empleado ya creado anteriormente, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 9).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 230 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

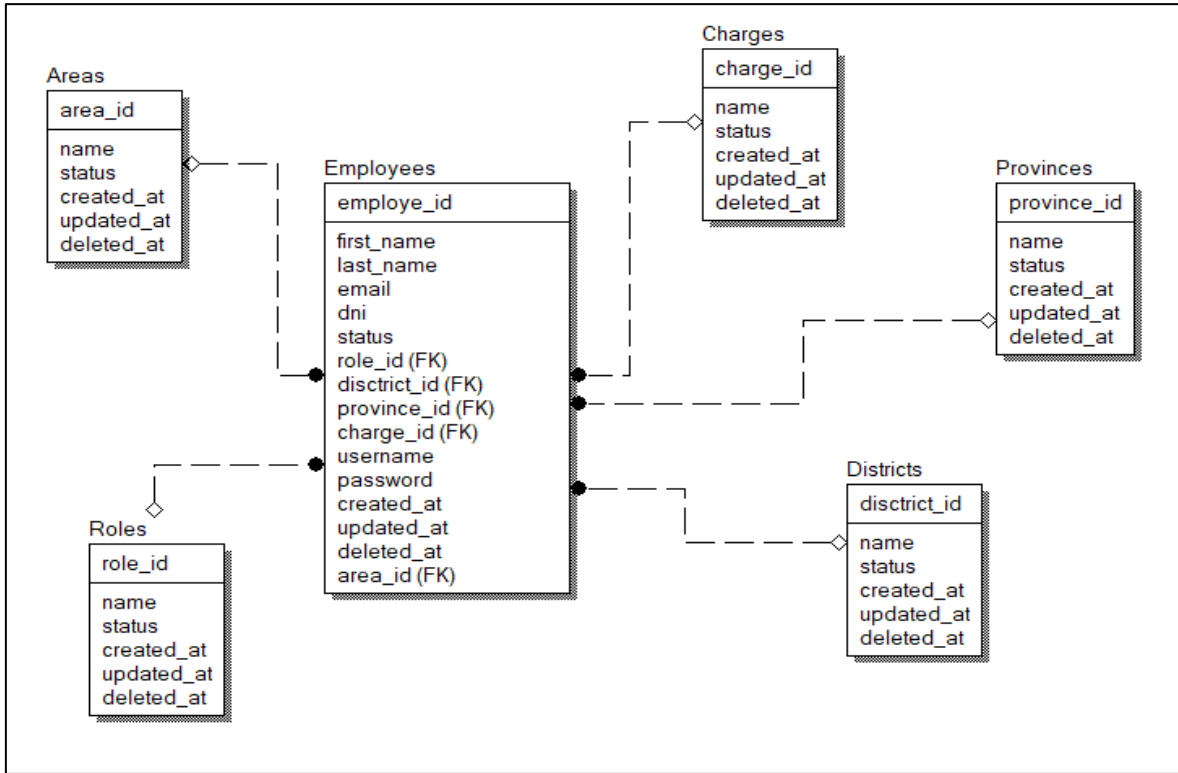


Figura 48. Diagrama lógico de la base de datos RF9

© Fuente: Elaboración propia

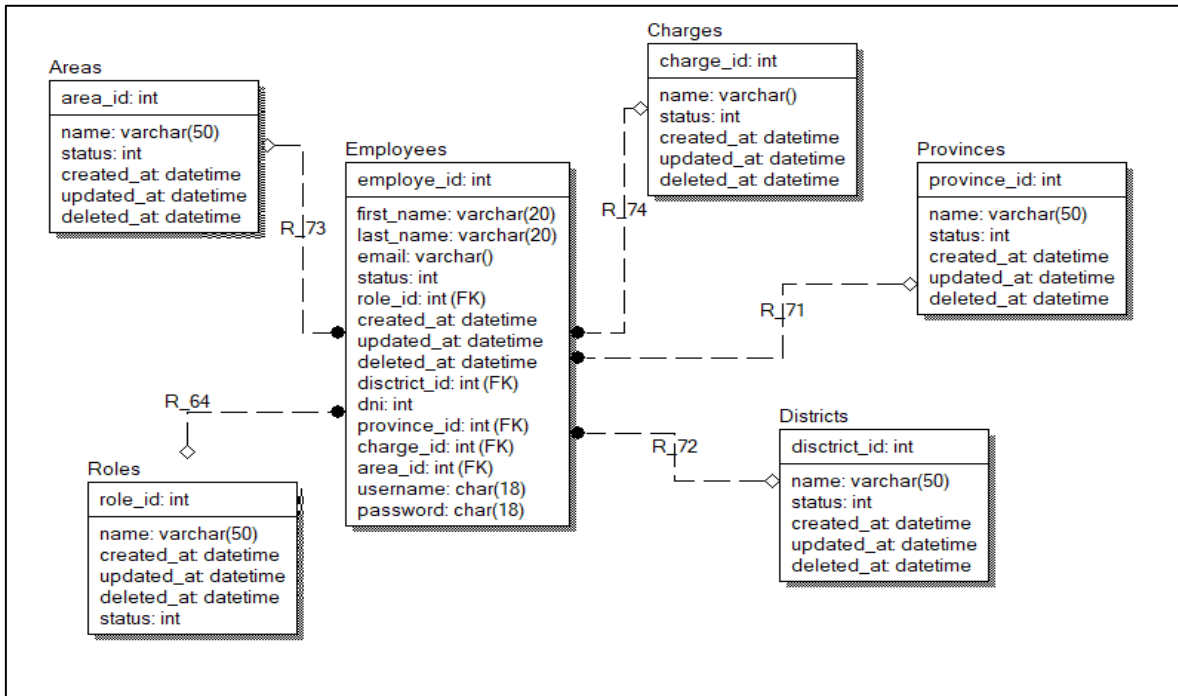
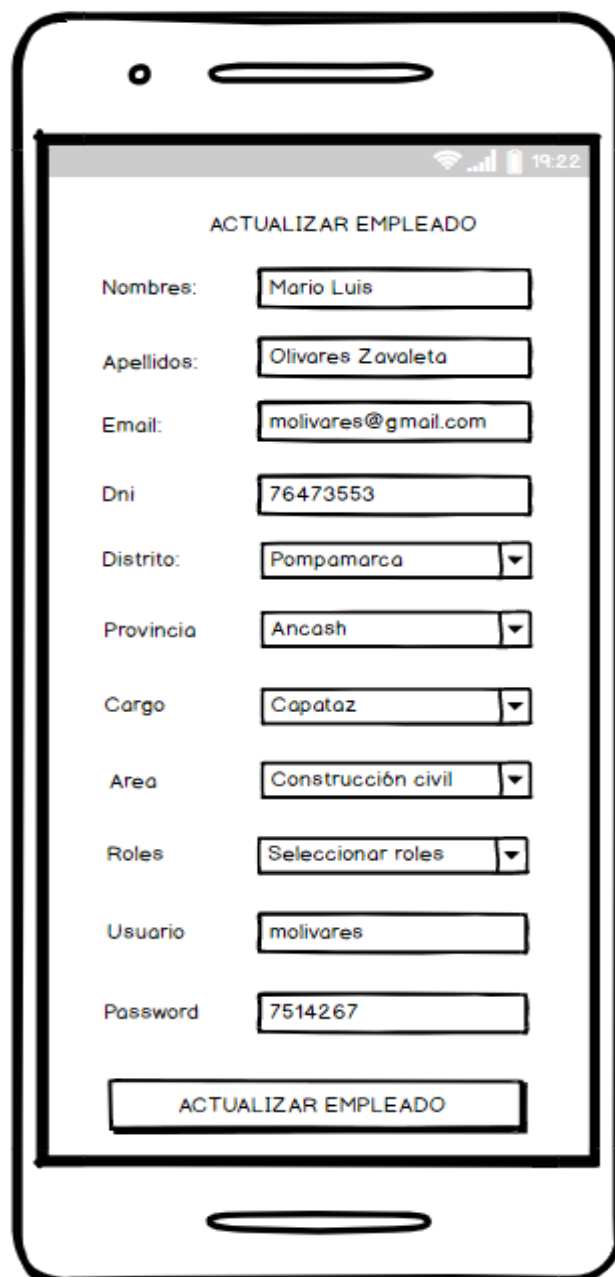


Figura 49. Diagrama físico de la base de datos RF9

Diseño

© Fuente: Elaboración propia



ACTUALIZAR EMPLEADO

Nombres:

Apellidos:

Email:

Dni:

Distrito:

Provincia:

Cargo:

Area:

Roles:

Usuario:

Password:

Figura 50. Prototipo de RF9

En la figura 50 se muestra el prototipo para la GUI de registrar empleado aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

Personalapp

Actualizar empleado

Nombres
Mario

Apellidos
Olivarez

Email
mariolivarez@gmail.com

Dni
76473553

Rol
Obreros

Distrito
Jesús María

Provincia
Independencia

Cargo
Capataz

Area
Construcción

Usuario
zamoranorena

Contraseña
mailito

ACTUALIZAR EMPLEADO

Figura 51. Interface de RF9

En la figura 51 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function update(Request $request, $id) {
    $employee = Employee::findOrFail($id);
    $employee->update([
        'first_name'=>$request->input('first_name'),
        'last_name'=>$request->input('last_name'),
        'email'=>$request->input('email'),
        'status'=>$request->input('status'),
        'district_id'=>$request->input('district_id'),
        'dni'=>$request->input('dni'),
        'province_id'=>$request->input('province_id'),
        'charge_id'=>$request->input('charge_id'),
        'province_id'=>$request->input('province_id'),
        'area_id'=>$request->input('area_id'),
        'user_id'=>$request->input('user_id')
    ]);
}
```

Figura 52. Código de RF9

En esta imagen 52 de la presente tesis se puede observar el código fuente del requerimiento funcional 9.

Requerimiento RF10 – Eliminar empleado

El requerimiento 10, tiene como objetivo que la aplicación debe poder eliminar un empleado ya creado anteriormente, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 10).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 230 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

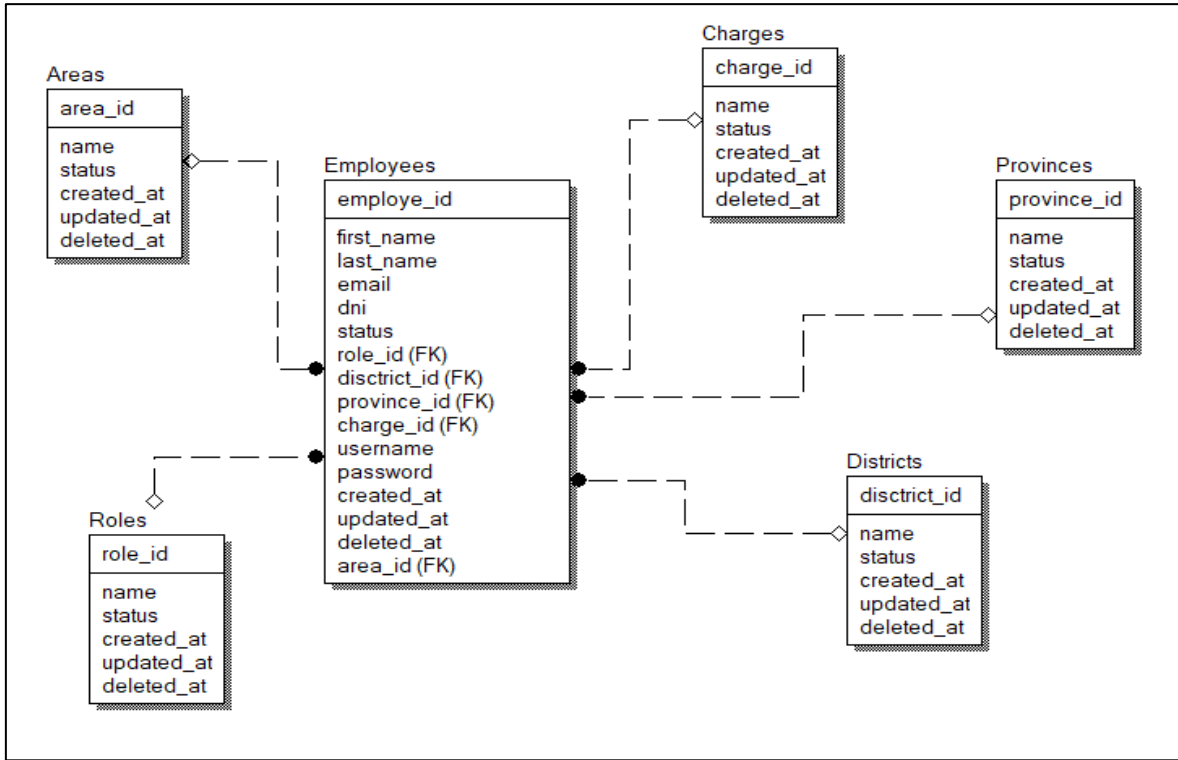


Figura 53. Diagrama lógico de la base de datos RF10

© Fuente: Elaboración propia

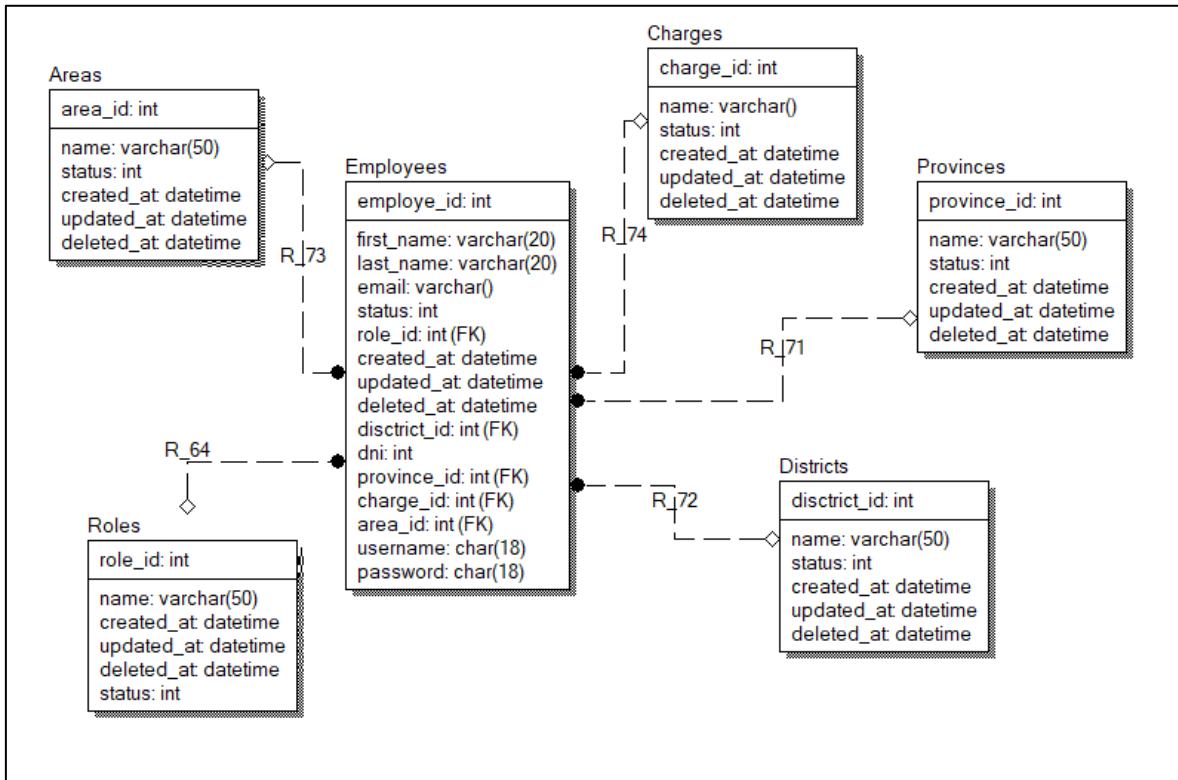


Figura 54. Diagrama físico de la base de datos RF10

Diseño

© Fuente: Elaboración propia

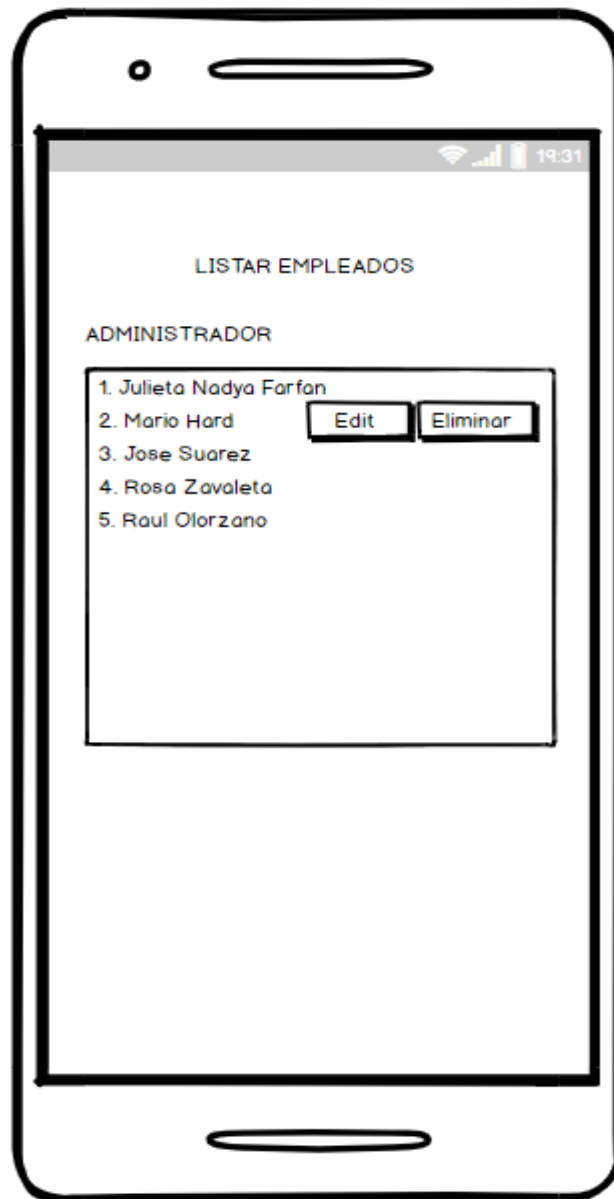


Figura 55. Prototipo de RF10

En la figura 55 se muestra el prototipo para la GUI de eliminar empleado aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

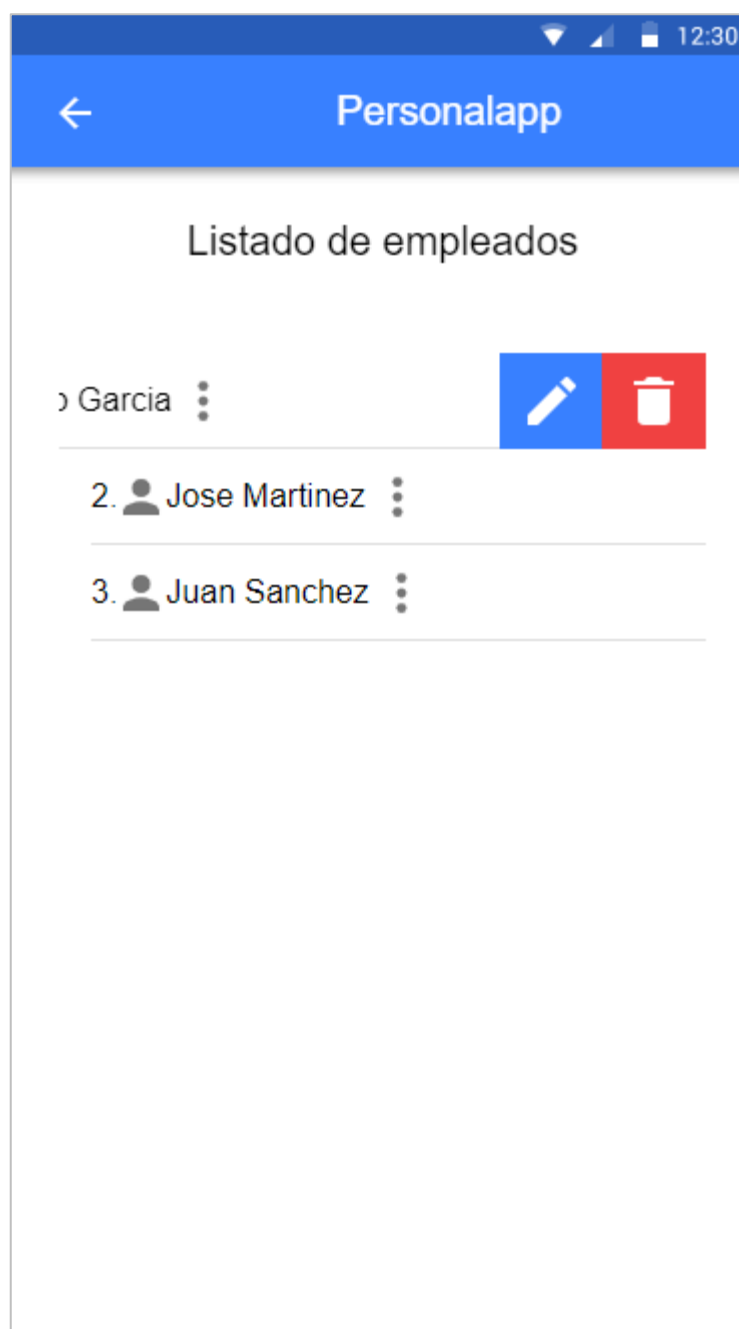


Figura 56. Interface de RF10

En la figura 56 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function deleteEmployee(Request $request){
    Employee::destroy($this->request->input('employee_id'));
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 57. Código de RF10

En esta imagen 57 de la presente tesis se puede observar el código fuente del requerimiento funcional 10.

Requerimiento RF11 – Desvincular empleado

El requerimiento 11, tiene como objetivo que la aplicación debe poder desvincular un empleado de la empresa cuando ya no esté trabajando, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 11).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 230 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

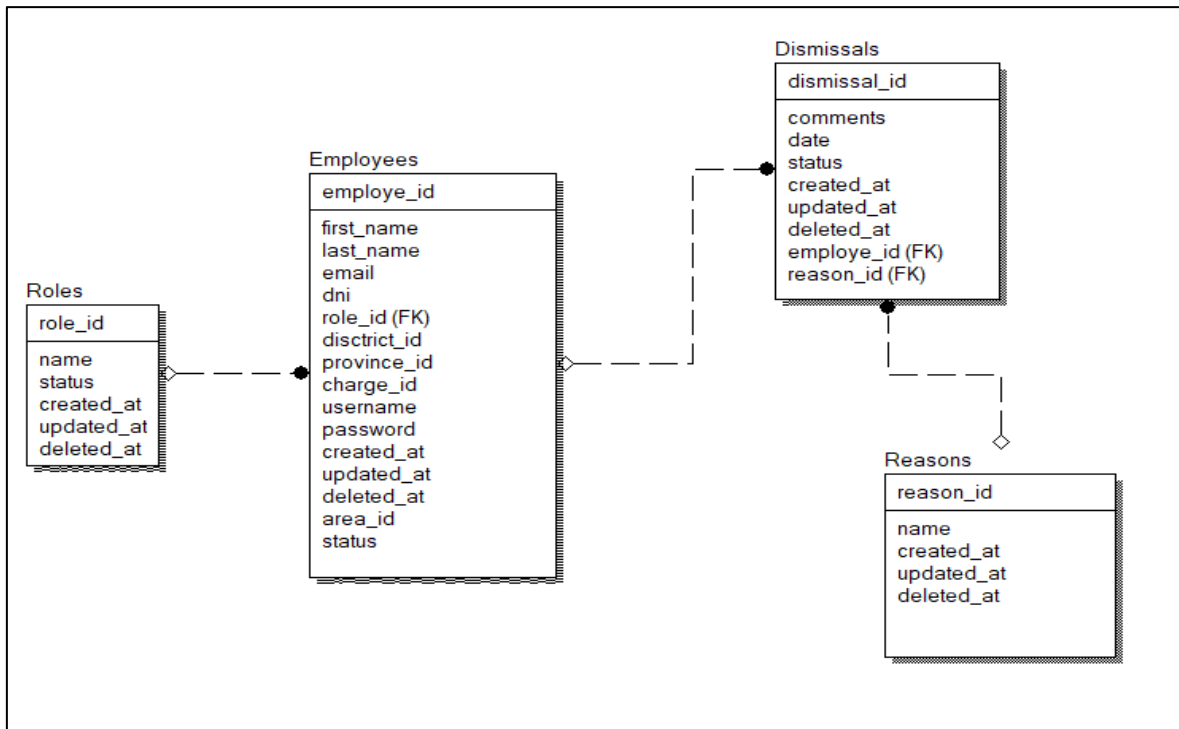


Figura 58. Diagrama lógico de la base de datos RF11

© Fuente: Elaboración propia

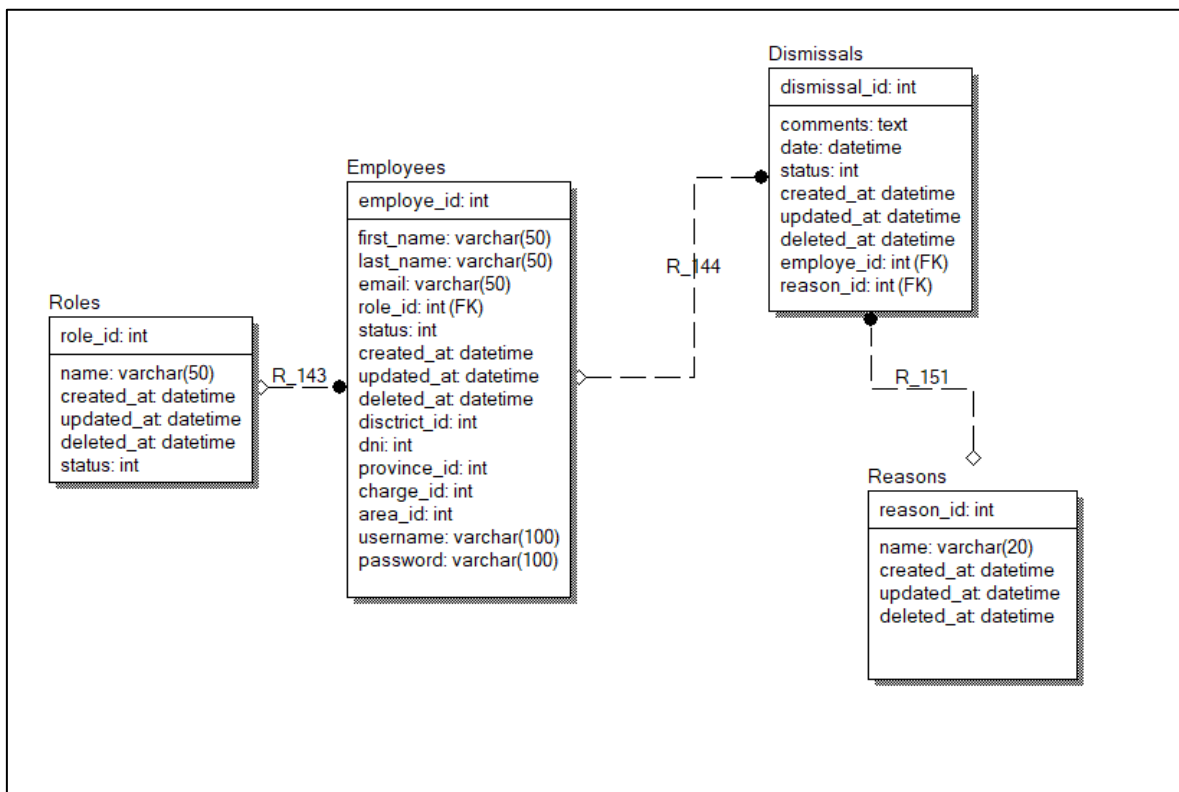


Figura 59. Diagrama físico de la base de datos RF11

Diseño

© Fuente: Elaboración propia

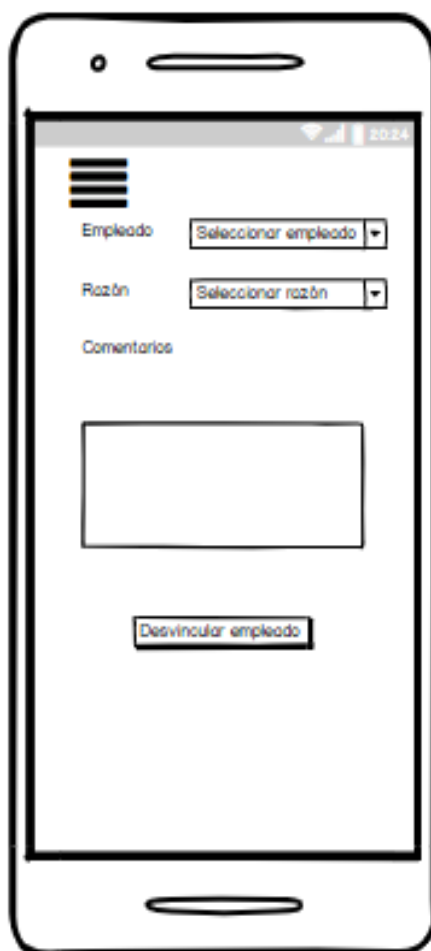


Figura 60. Prototipo de RF11

En la figura 60 se muestra el prototipo para la GUI de desvincular empleado aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

The screenshot shows a mobile application interface titled 'Personalapp'. The main heading is 'Desvincular empleado'. Below this, there are two dropdown menus: 'Empleado' and 'Razón', both with 'Selecciona...' as the placeholder text. A text input field labeled 'Comentarios' is positioned below the dropdowns. At the bottom of the form, there is a prominent blue button with the text 'DESVINCLAR EMPLEADO'.

Figura 61. Interface de RF11

En la figura 61 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function desvincularEmployee(Request $request, $id) {
    $employee = Employee::findOrFail($id);
    $employee->update([
        'status'=> 0
    ]);
}
```

Figura 62. Código de RF11

En esta imagen 62 de la presente tesis se puede observar el código fuente del requerimiento funcional 11.

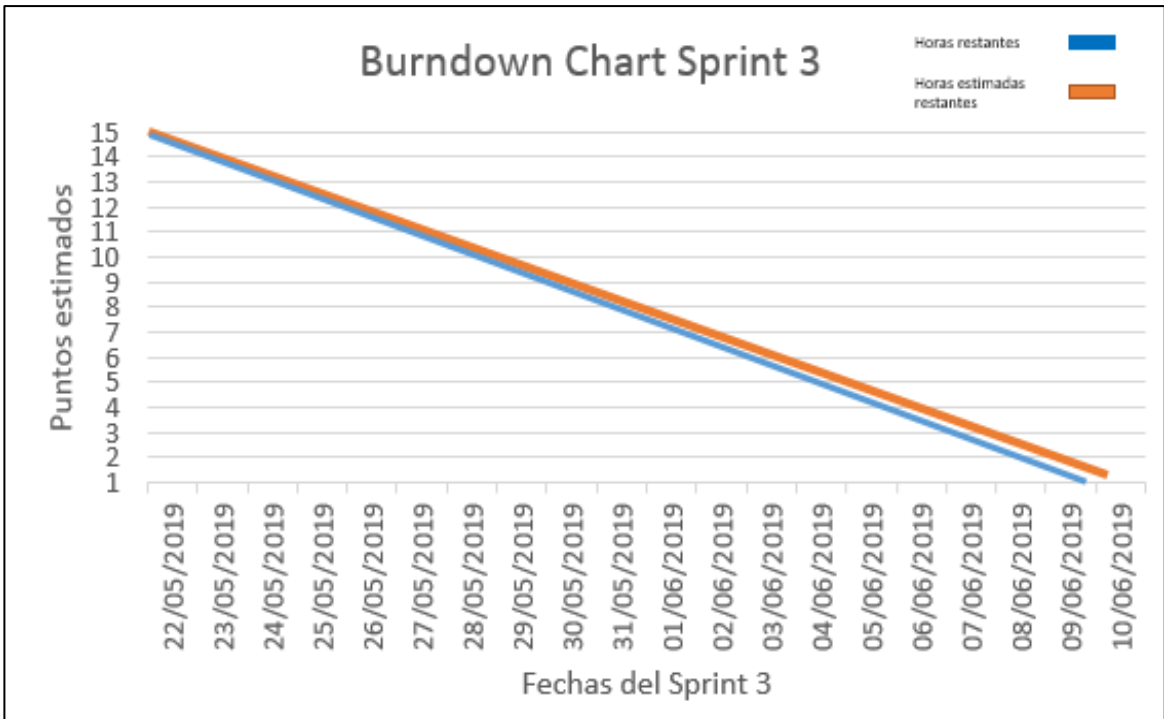


Figura 63. Burndown Chart Sprint 3

Para este sprint, se puede apreciar en esta figura 63, que el tiempo real de Scrum (línea naranja, ya que así dice en su documentación y nosotros que estamos laborando en consultoría también utilizamos este Burndown Chart ya que trabajamos con Scrum y sabemos cómo se usa) va acorde al tiempo planificado (línea azul, en el cual es el tiempo que se demora el equipo de desarrollo en acabar sus actividades), e incluso se terminó mucho antes de lo planificado. Bajo este escenario se puede determinar que se ha terminado todo correctamente, dando paso al siguiente sprint.



Acta de reunión de entrega del Sprint 3

Acta de reunión de entrega del Sprint 3		
Fecha	07/06/2019	
Hora	3:00 pm	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Responsable	Lorenzo Palacios	Scrum Master
Participantes	Javier Zamora	Product Owner
	Maurihño Escurra	Analista
	Luis Zamora	Programador
	Julio Sosa	Calidad
Tema	Presentación del Sprint 3	
Resumen de la reunión		
<p>En esta reunión se mostraron las pantallas de la aplicación a nivel funcional de acuerdo a las historias pactadas en la Planificación del Sprint 3.</p> <p>El cliente quedo satisfecho con el trabajo realizado y dio el visto bueno, dando paso al siguiente sprint.</p>		


CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

Figura 64. Acta de reunión de entrega del Sprint 3

El tema de reunión de esta acta es sobre la presentación de entrega del sprint 3.

Resumen del Sprint 3

Tabla 76: *Resumen del Sprint 3*

Total, de historias	5
Historias terminadas	5
Historias por terminar	0
Avance	100%

Fuente: Elaboración propia

Las pruebas realizadas a las funcionalidades durante este sprint fueron exitosas sin ningún error de diseño ni de código fuente.

Retrospectiva del Sprint 3

Al final del Sprint, el equipo Scrum se reunió para recibir la respuesta del Scrum Master, para saber cómo le fue en la reunión con el Product Owner, resulta que el producto se entregó sin problemas y con el cliente satisfecho.

Cosas Positivas:

- Lograron cumplir los resultados
- El apoyo del equipo en todo momento

Cosas Negativas:

- Ninguna

Lista de Pendientes de Sprint (Sprint Backlog)

Sprint 4

Tabla 77: *Sprint 4*

Nº Sprint	Requerimientos Funcionales	Historias de Usuario	Tiempo Estimado	Prioridad
Sprint 4	RF12: El sistema proporcionará a los usuarios información de las marcaciones de asistencia del obrero.	H12	5	4
	RF13: Este módulo permitirá visualizar un reporte de ausentismo de todos los empleados que están en una obra.	H13	8	4
	RF14: Este módulo permitirá visualizar un reporte de todos los empleados por obra que se hayan desvinculado de la empresa.	H14	8	4
	RF15: La aplicación permitirá al obrero y administrador comunicarse con el área de recursos humanos para hacerle consultas.	H15	1	4
	RF16: La aplicación permitirá mostrar al usuario obrero en que obra está asignado.	H16	3	4
	RF17: La aplicación permitirá mostrar al usuario administrador un listado de los empleados: obreros, recursos humanos y administrador.	H17	3	4

RF18: La aplicación permitirá mostrar un listado de todas las obras al usuario administrador.	H18	3	4
RF19: La aplicación permitirá al usuario administrador ver el detalle de cada obra seleccionada.	H19	3	4
RF20: La aplicación permitirá al usuario administrador visualizar los empleados asignados a una obra.	H20	5	4
RF21: La aplicación permitirá al usuario administrador visualizar los empleados desvinculados de la empresa.	H21	3	4
RF22: La aplicación permitirá al usuario administrador ver el detalle de los empleados desvinculados de la empresa.	H22	3	4
RF23: La aplicación permitirá al usuario administrador eliminar a los empleados desvinculados de la empresa.	H23	3	4
RF24: La aplicación permitirá al usuario obrero y administrador registrar sus salidas.	H24	3	4
RF25: La aplicación permitirá al usuario administrador y obrero visualizar estrategias de negocio de la empresa.	H25	3	4
RF26: La aplicación permitirá al usuario administrador y obrero seleccionar los tipos de	H26	3	4

	estrategias de negocio de la empresa: ausentismo o rotación.			
	RF27: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de ausentismo.	H27	3	4
	RF28: La aplicación permitirá al usuario administrador y obrero visualizar las estrategias de rotación.	H28	3	4
	RF29: La aplicación permitirá al usuario administrador crear estrategias de negocio ya sean estrategias de ausentismo o rotación.	H29	3	4
	RF30: La aplicación permitirá al usuario administrador actualizar estrategias de ausentismo.	H30	3	4
	RF31: La aplicación permitirá al usuario administrador eliminar estrategias de ausentismo.	H31	3	4
	RF32: La aplicación permitirá al usuario administrador actualizar estrategias de rotación.	H32	3	4
	RF33: La aplicación permitirá al usuario administrador eliminar estrategias de rotación.	H33	3	4
	RF34: La aplicación permitirá a los usuarios visualizar el listado de reglamentos de la empresa.	H34	3	4
	RF35: La aplicación permitirá al usuario administrador y recursos	H35	3	4

humanos crear reglamentos de la empresa.			
RF36: La aplicación permitirá al usuario administrador y recursos humanos actualizar reglamentos de la empresa.	H36	3	4
RF37: La aplicación permitirá al usuario administrador y recursos humanos eliminar reglamentos de la empresa.	H37	3	4
RF38: La aplicación permitirá a los usuarios visualizar el listado de gratificaciones de la empresa.	H38	3	4
RF39: La aplicación permitirá al usuario administrador y recursos humanos crear gratificaciones de la empresa.	H39	3	4
RF40: La aplicación permitirá al usuario administrador y recursos humanos actualizar gratificaciones de la empresa.	H40	3	4
RF41: La aplicación permitirá al usuario administrador y recursos humanos eliminar gratificaciones de la empresa.	H41	3	4
RF42: La aplicación permitirá a los usuarios pulsar justificaciones en la barra lateral.	H42	1	4
RF43: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los	H43	3	4

empleados que tengan o no justificaciones.			
RF44: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no justificaciones.	H44	3	4
RF45: La aplicación permitirá a los usuarios visualizar el listado de las justificaciones de los empleados de la empresa.	H45	3	4
RF46: La aplicación permitirá al usuario de recursos humanos aprobar las justificaciones de los empleados de la empresa.	H46	3	4
RF47: La aplicación permitirá al usuario administrador y obreros crear sus justificaciones.	H47	3	4
RF48: La aplicación permitirá al usuario administrador y obreros actualizar sus justificaciones.	H48	3	4
RF49: La aplicación permitirá al usuario administrador y obreros eliminar sus justificaciones.	H49	3	4
RF50: La aplicación permitirá a los usuarios pulsar vacaciones en la barra lateral.	H50	1	4
RF51: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no vacaciones.	H51	3	4

RF52: La aplicación permitirá al usuario de recursos humanos seleccionar a los empleados que tengan o no vacaciones.	H52	3	4
RF53: La aplicación permitirá a los usuarios visualizar el listado de las vacaciones de los empleados de la empresa.	H53	3	4
RF54: La aplicación permitirá al usuario de recursos humanos aprobar las vacaciones de los empleados de la empresa.	H54	3	4
RF55: La aplicación permitirá al usuario administrador y obreros crear sus vacaciones.	H55	3	4
RF56: La aplicación permitirá al usuario administrador y obreros actualizar sus vacaciones.	H56	3	4
RF57: La aplicación permitirá al usuario administrador y obreros eliminar sus vacaciones.	H57	3	4
RF58: La aplicación permitirá a los usuarios pulsar remuneraciones en la barra lateral.	H58	1	4
RF59: La aplicación permitirá al usuario de recursos humanos visualizar el listado de los empleados que tengan o no remuneración.	H59	3	4
RF60: La aplicación permitirá al usuario de recursos humanos	H60	3	4

	seleccionar a los empleados que tengan o no remuneraciones.			
	RF61: La aplicación permitirá a los usuarios visualizar el listado de las remuneraciones de los empleados de la empresa.	H61	3	4
	RF62: La aplicación permitirá al usuario de recursos humanos crear las remuneraciones.	H62	3	4
	RF63: La aplicación permitirá al usuario de recursos humanos actualizar las remuneraciones.	H63	3	4
	RF64: La aplicación permitirá al usuario de recursos humanos eliminar las remuneraciones.	H64	3	4
	RF65: La aplicación permitirá a los usuarios redirigir a la pantalla de inicio (Home) al pulsar en “Personalapp”.	H65	1	4

Fuente: Elaboración propia

Aquí está todas las funcionalidades a realizar en el sprint 4.

© Fuente: Elaboración propia

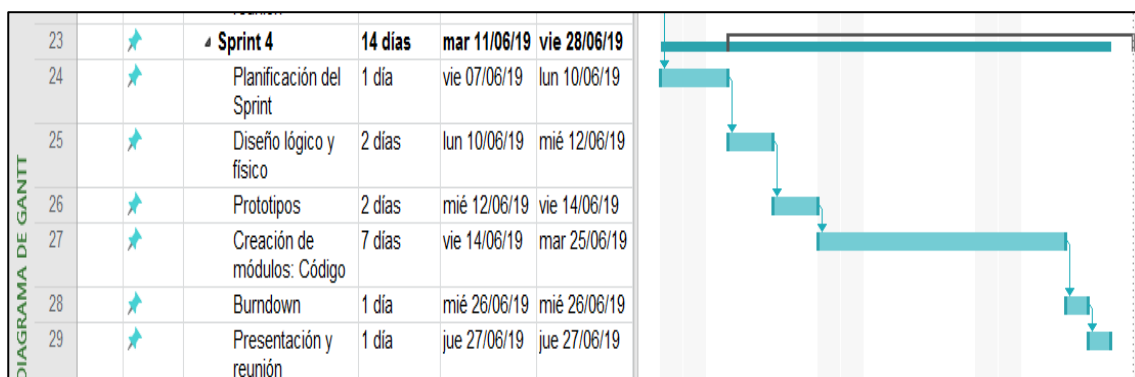


Figura 65. Planificación Sprint 4

Aquí en esta figura 65 está el cronograma de las actividades a realizar en el sprint 4.

A continuación, en este apartado de nuestra tesis, se podrá evidenciar todo el proceso de la implementación de la app móvil divididos por cada funcionalidad del sprint 4. En el cual nosotros hemos elaborado un texto descriptivo y cuatro imágenes por funcionalidad para que ustedes como lectores puedan entender bien todo el proceso de implementación de la aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C., todas las imágenes que hemos elaborado están en orden secuencial y por cada funcionalidad pero no todas las imágenes están con sus respectivas descripciones, por eso vamos a explicar en este texto como es el orden y descripción de cada imagen por funcionalidad: una vez explicado el objetivo del requerimiento funcional podrán observar en las dos imágenes siguientes la base de datos lógico y físico de las tablas involucradas de dicha funcionalidad (el proceso para llegar al diseño de la base de datos lógico y físico y tablas involucradas que son las entidades fue que se obtuvo durante el análisis de la segunda actividad que se puede visualizar en el cronograma del sprint 4 y de las descripciones de las historias de usuario), luego se mostrará la imagen del diseño de los prototipos elaborados con la herramienta Balsamiq Mockups (estos diseños fueron validados y aceptados por la empresa), luego ustedes podrán observar la imagen de implementación de las interfaces y por último puedan ver la imagen del código fuente de dicha funcionalidad de la app móvil. Todos los requerimientos funcionales de este sprint 4 están enumerados y por ende se sobre entiende que las imágenes que van a observar pertenecen a las funcionalidades ya que están todas enumeradas.

Requerimiento RF12 – Consultar marcaciones

El requerimiento 12, tiene como objetivo que la aplicación debe poder hacer una consulta de las marcaciones de cada empleado (hora de entrada - salida), el cual generará un PDF, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Recursos Humanos” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 12).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

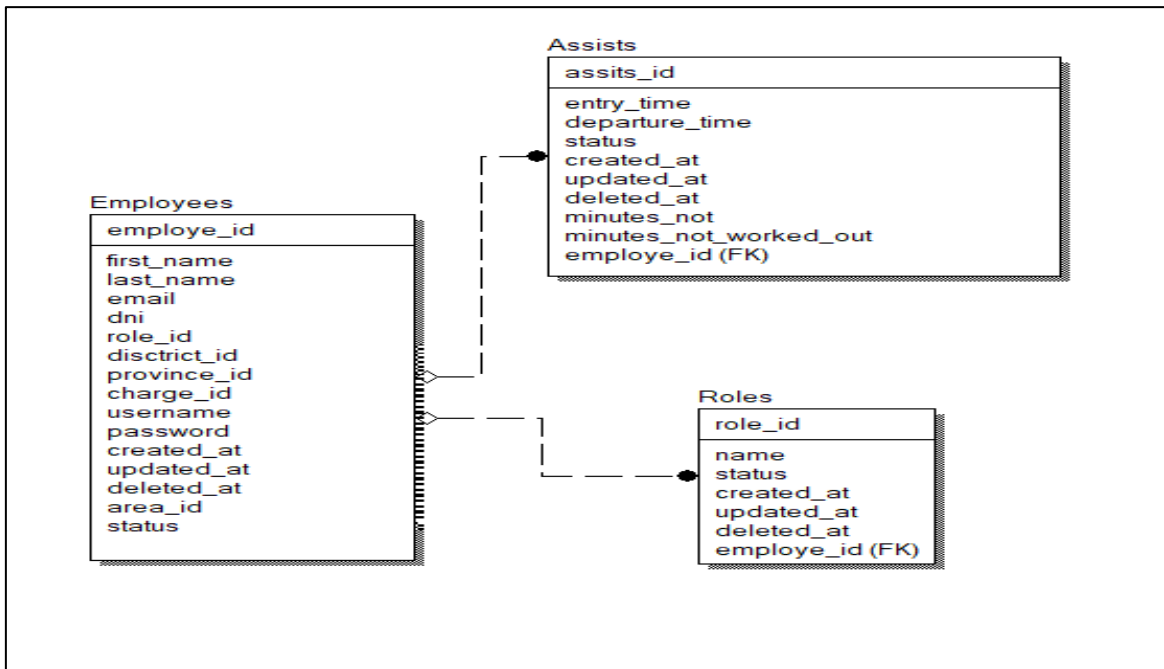


Figura 66. Diagrama lógico de la base de datos RF12

© Fuente: Elaboración propia

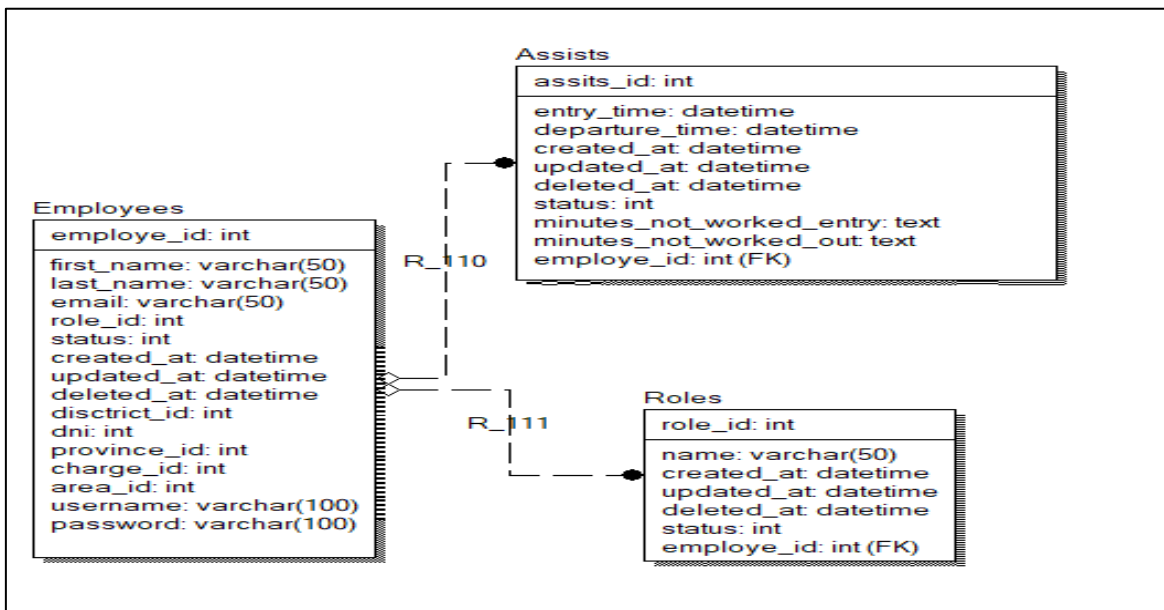


Figura 67. Diagrama físico de la base de datos RF12

Diseño

© Fuente: Elaboración propia

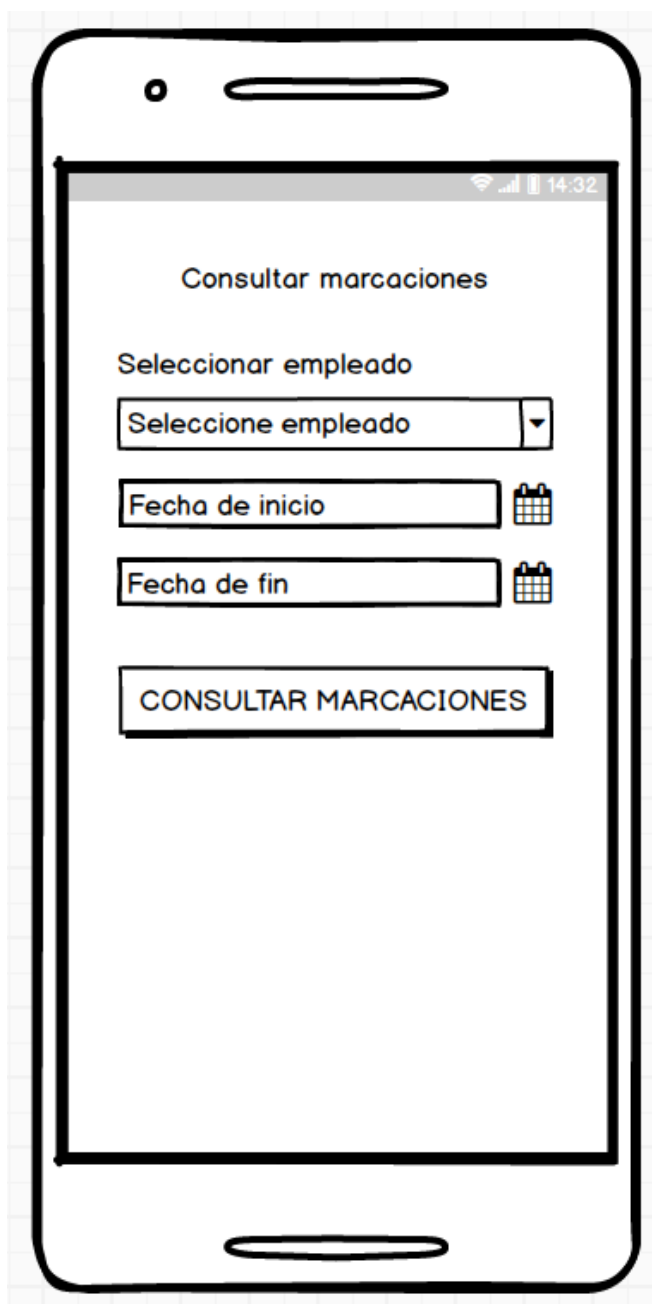


Figura 68. Prototipo de RF12

En la figura 68 se muestra el prototipo para la GUI de consultar marcaciones aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

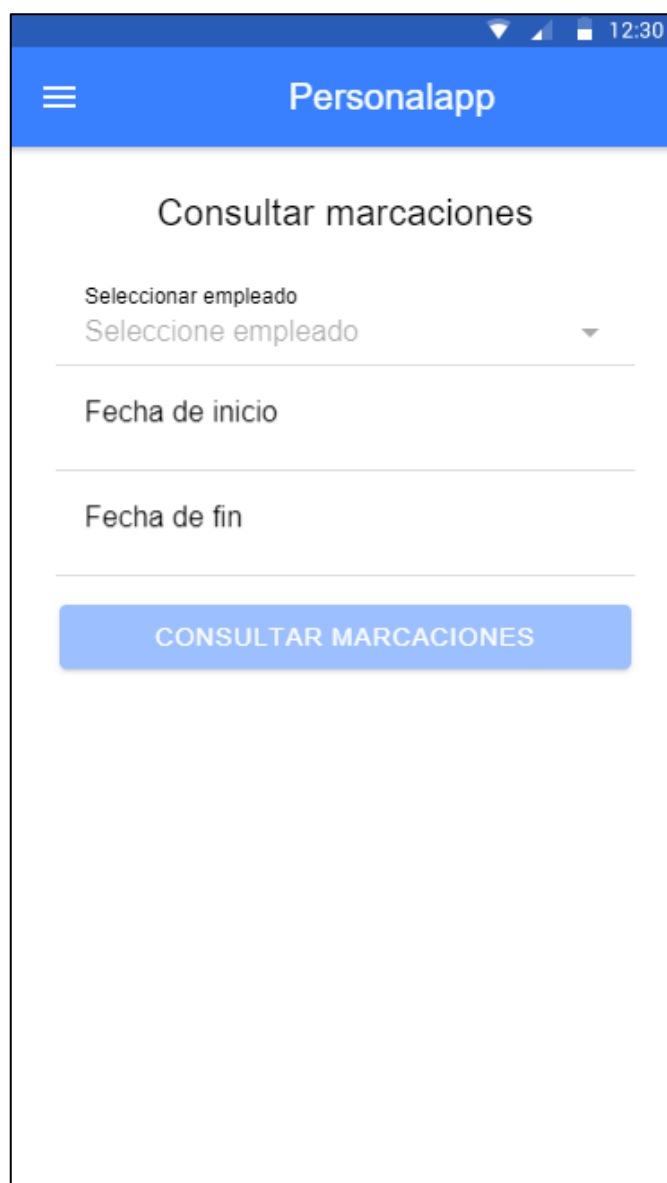


Figura 69. Interface de RF12

En la figura 69 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function marcationForDate(Request $request) {  
    $start_date = $this->request->input( key: 'start_date');  
    $end_date = $this->request->input( key: 'end_date');  
    $asist = Assist::whereBetween('entry_time', [$start_date, $end_date])  
        ->where('employee_id', $this->request->input( key: 'employee_id'))->get();  
    $employee = Employee::find($this->request->input( key: 'employee_id'));  
    if($end_date < $start_date){  
        return response()->json([  
            'msg' => 'dateranger',  
            'employee' => $employee->first_name . ' ' . $employee->last_name  
        ], status: 200);  
    }else if(count($asist) == 0){  
        return response()->json([  
            'msg' => "rangedate",  
            'employee' => $employee->first_name . ' ' . $employee->last_name  
        ], status: 200);  
    }else{  
        return response()->json([  
            'msg' => $asist,  
            'employee' => $employee->first_name . ' ' . $employee->last_name  
        ], status: 200);  
    }  
}
```

Figura 70. Código de RF12

En esta imagen 70 de la presente tesis se puede observar el código fuente del requerimiento funcional 12.

Requerimiento RF13 – Visualizar reporte de ausentismo

El requerimiento 13, tiene como objetivo que la aplicación debe poder visualizar el reporte de ausentismo de cada día según el seleccionado de obra, y se visualizará en un PDF, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos Humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 13).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

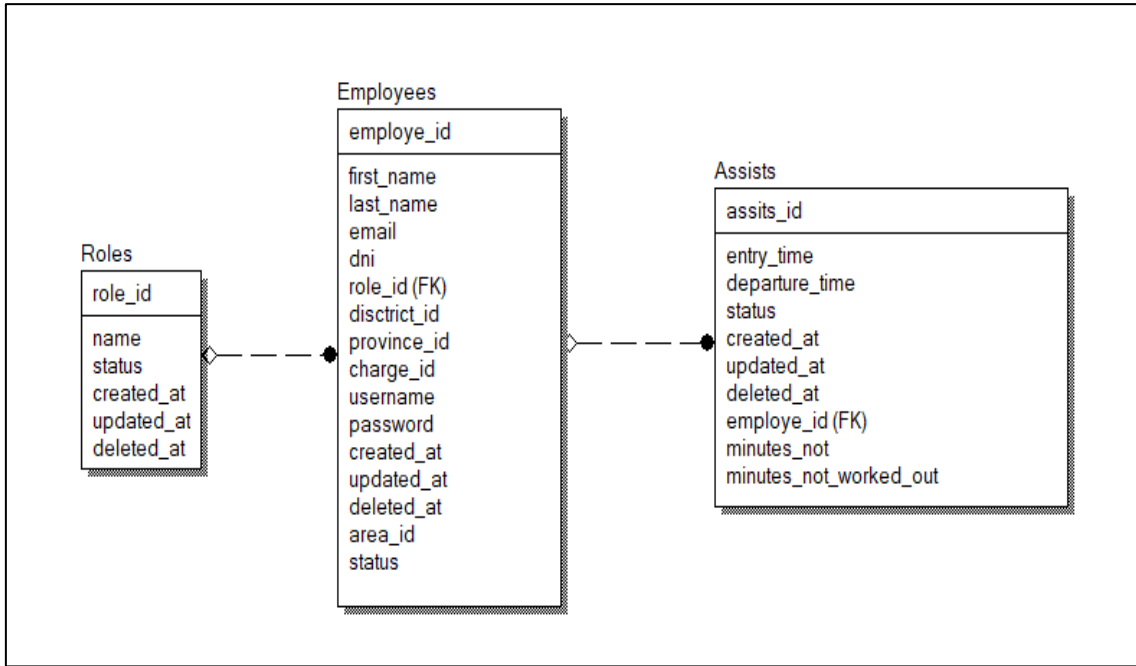


Figura 71. Diagrama lógico de la base de datos RF13

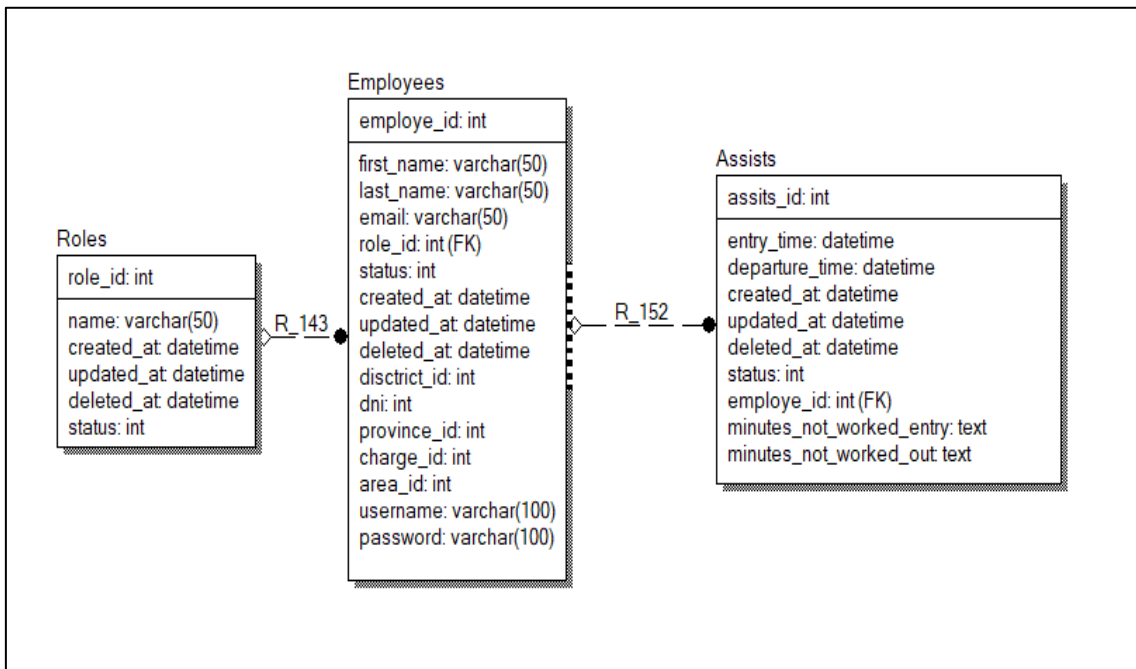


Figura 72. Diagrama físico de la base de datos RF13

Diseño

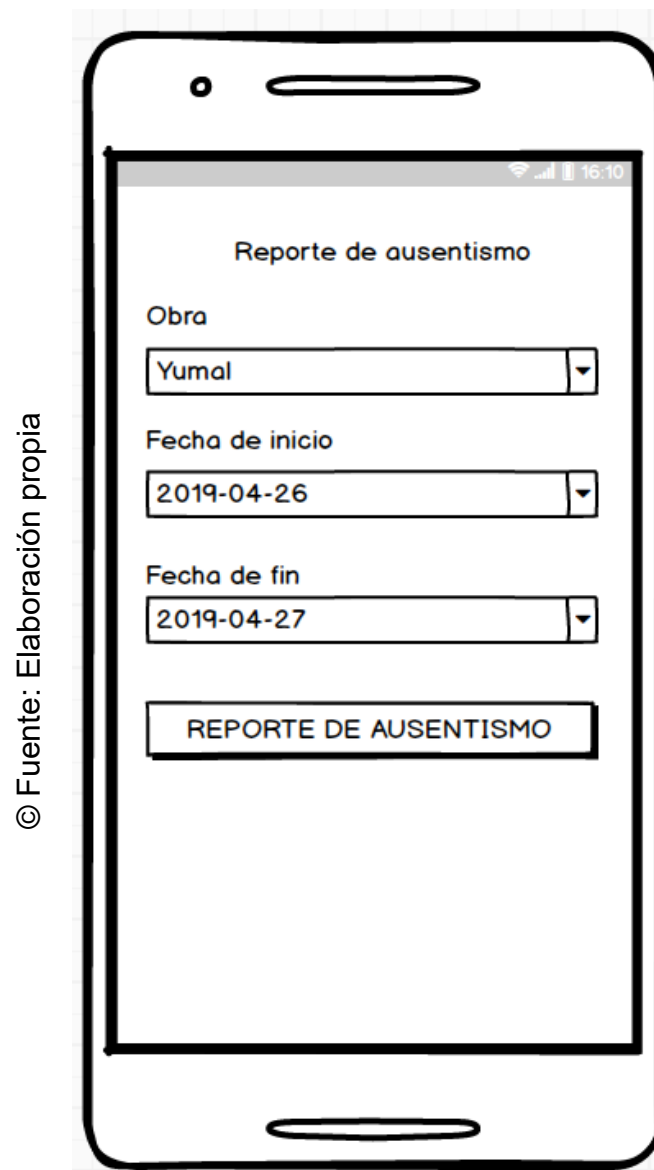


Figura 73. Prototipo de RF13

En la figura 73 se muestra el prototipo para la GUI de visualizar reporte de ausentismo aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. At the top, there is a blue header bar with a back arrow on the left and the text 'Personalapp' in the center. Below the header, the main content area is white and contains the following elements: a title 'Reporte de ausentismo', a dropdown menu labeled 'Obra:' with the text 'Seleccione obra' and a downward arrow, a text input field labeled 'Fecha inicio', another text input field labeled 'Fecha fin', and a blue button at the bottom with the text 'REPORTE DE AUSENTISMO'. The status bar at the top right shows the time as 12:30.

Figura 74. Interface de RF13

En la figura 74 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function reportAusentism(Request $request){
    $array_data = [];
    $responses = Assignwork::where('work_id',$request->work_id)->get();
    foreach ($responses as $respons){
        $array_data[]=$respons->employee_id;
    }
    $reports = DB::table('asists')
    ->selectRaw('date(asists.entry_time) as registro, count(*) as total_de_personas,
    SEC_TO_TIME( SUM( TIME_TO_SEC(ADDTIME(minutes_not_worked_entry,minutes_not_worked_out)))) as horas_perdidas')
    ->whereIn('employee_id', $array_data)
    ->groupBy('registro')
    ->where('asists.minutes_not_worked_entry', '>', 0)
    ->get();
    if(count($reports) > 0){
        foreach ($reports as $report){
            if($report->horas_perdidas != null){
                $horas_trabajo = explode( delimiter: ':', string: 54*4 . ':00:00');
                $data = explode( delimiter: ':',$report->horas_perdidas);
                $minutos = intval( var: 60) - intval($data[1]);
                $segundos = intval( var: 60) - intval($data[2]);
                $report->horas_laboradas = intval($horas_trabajo[0])* $report->total_de_personas - intval($data[0]).':'.
                $minutos.':'.$segundos;
                $datetite = explode( delimiter: ':',$report->horas_laboradas);
                $report->indice_de_ausentismo = number_format( number: ($report->total_de_personas/$data[0])/
                ($report->total_de_personas/$datetite[0]), decimals: 2);
            }else{
                $report->horas_perdidas = '00:00:00';
                $report->horas_laboradas = 54*4 . ':00:00';
                $report->indice_de_ausentismo = 0;
            }
        }
    }
    return response()->json([
        'msg' => $reports
    ], status: 200);
}else{
    return response()->json([
        'msg' => 'vacio'
    ], status: 200);
}
```

Figura 75. Código de RF13

En esta imagen 75 de la presente tesis se puede observar el código fuente del requerimiento funcional 13.

Requerimiento RF14 – Visualizar reporte de rotación

El requerimiento 14, tiene como objetivo que la aplicación debe poder visualizar el reporte de rotación, en donde se encuentra los empleados desvinculados de la empresa, y se visualizará en un PDF, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos Humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 14).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

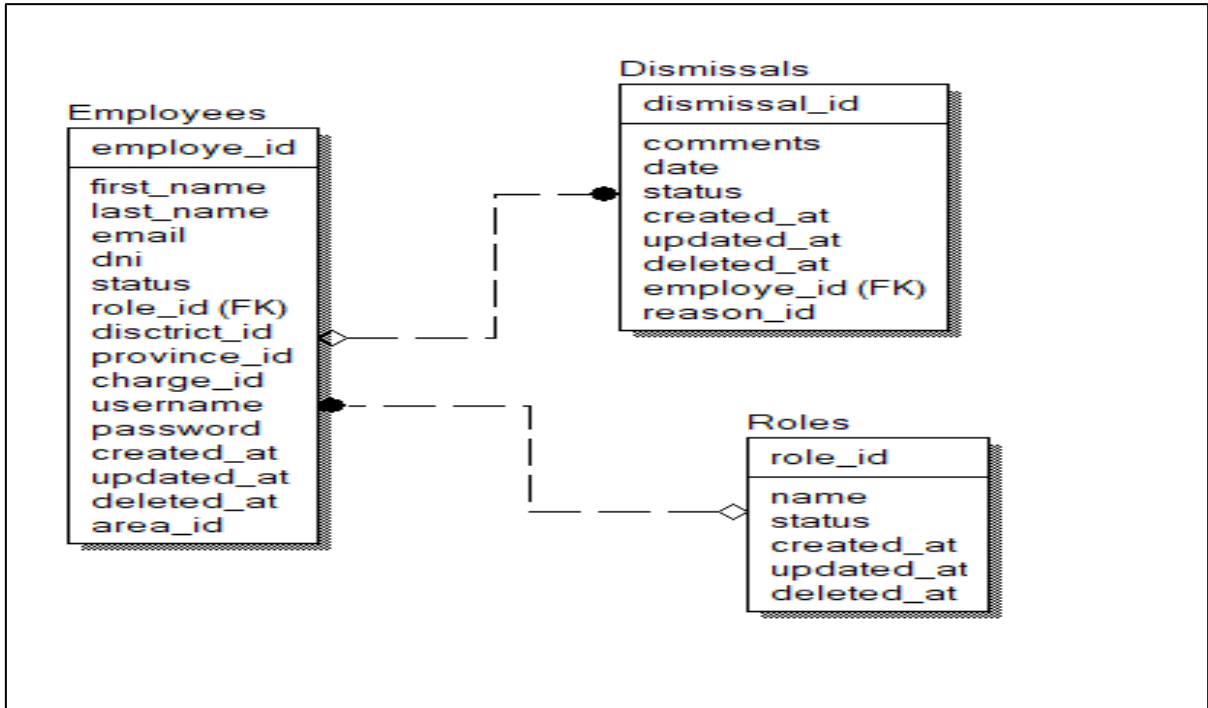


Figura 76. Diagrama lógico de la base de datos RF14

© Fuente: Elaboración propia

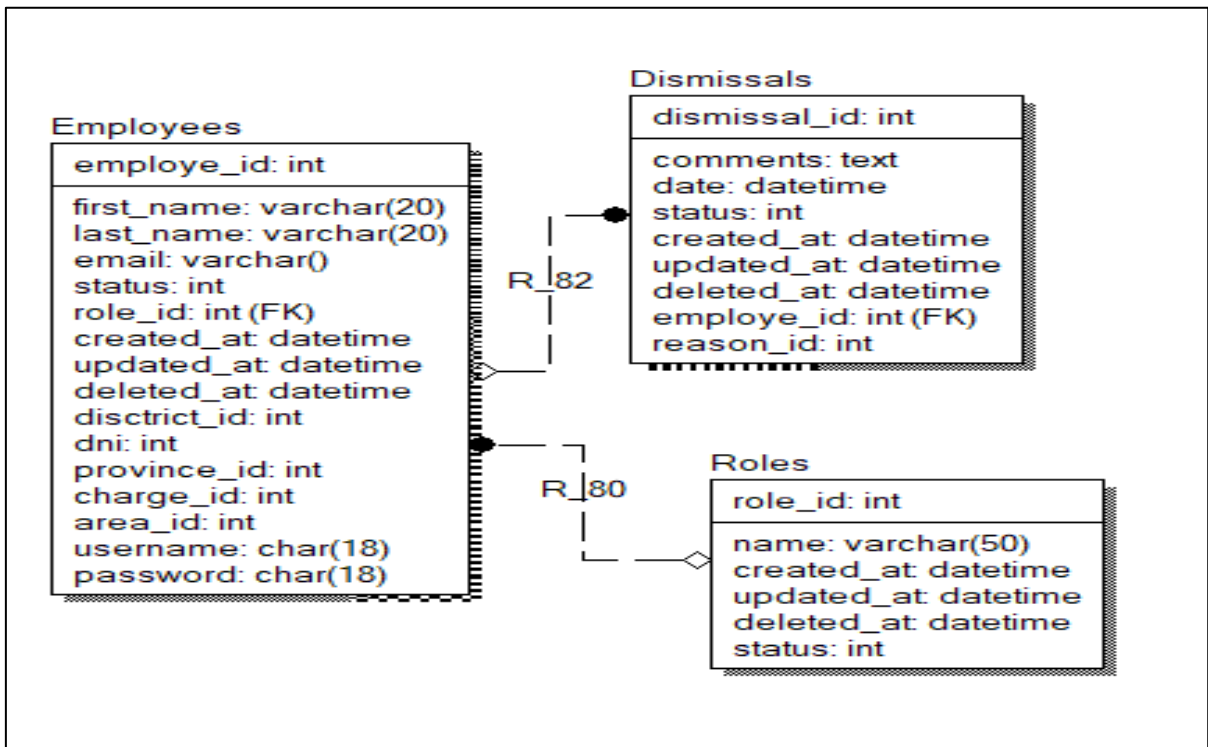
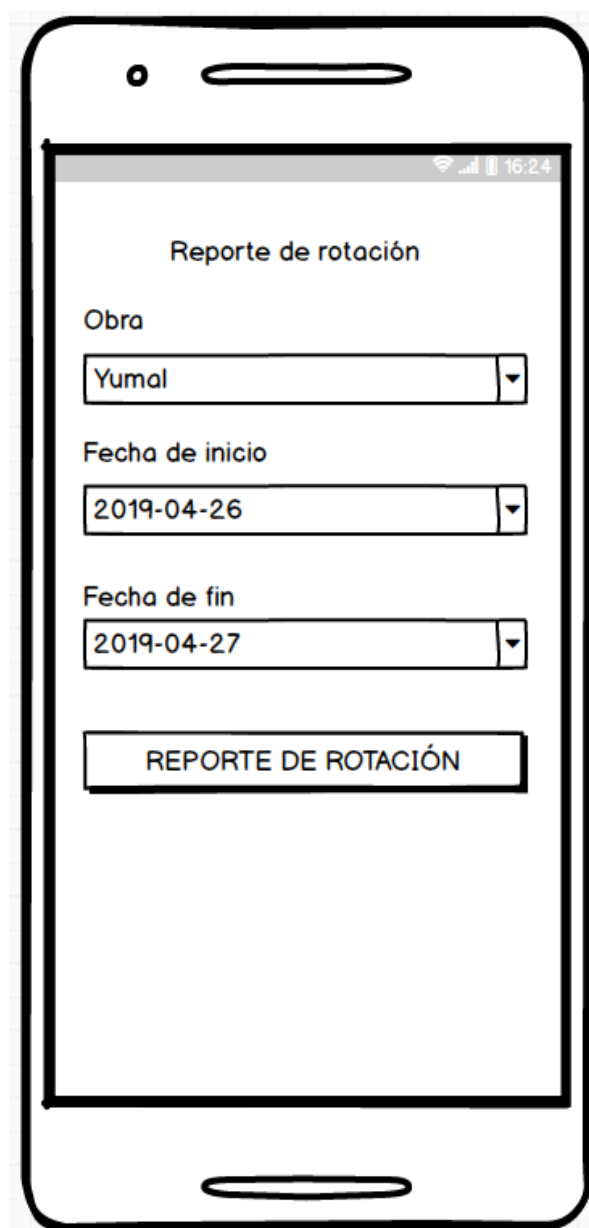


Figura 77. Diagrama físico de la base de datos RF14

Diseño

© Fuente: Elaboración propia



Reporte de rotación

Obra
Yumal

Fecha de inicio
2019-04-26

Fecha de fin
2019-04-27

REPORTE DE ROTACIÓN

The image shows a mobile application prototype for a rotation report. It features a title 'Reporte de rotación' at the top. Below the title are three dropdown menus: 'Obra' with the value 'Yumal', 'Fecha de inicio' with the value '2019-04-26', and 'Fecha de fin' with the value '2019-04-27'. At the bottom of the form is a button labeled 'REPORTE DE ROTACIÓN'. The entire form is displayed within a smartphone frame with a status bar at the top showing signal strength, Wi-Fi, and the time 16:24.

Figura 78. Prototipo de RF14

En la figura 78 se muestra el prototipo para la GUI de visualizar reporte de rotación aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. The main heading is 'Reporte de rotación'. Below this, there is a dropdown menu labeled 'Obras' with the text 'Seleccione obra' and a downward arrow. Underneath the dropdown are two text input fields: 'Fecha inicio' and 'Fecha fin'. At the bottom of the form is a blue button with the text 'REPORTE DE ROTACIÓN'. The top of the screen shows a blue navigation bar with a back arrow and the title 'Personalapp'. The status bar at the very top indicates the time as 12:30.

Figura 79. Interface de RF14

En la figura 79 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function reportRotation(Request $request){
    $array_data = [];
    $responses = Assignwork::where('work_id', $request->work_id->get());
    foreach ($responses as $respons){
        $array_data[]=$respons->employee_id;
    }
    $reports = DB::table('asists')
        ->selectRaw('date(asists.entry_time) as registro, count(*) as numero_de_empleados_asistidos')
        ->whereIn('employee_id', $array_data)
        ->groupBy('registro')
        ->where('asists.entry_time', '!=', null)
        ->get();
    $dismissals = Dismissals::all();
    if(count($reports) > 0){
        foreach ($reports as $report){
            $contador = 0;
            foreach ($dismissals as $dismissal){
                if($report->registro == $dismissal->created_at->format('Y-m-d')){
                    $contador = $contador + 1;
                    $report->numero_de_empleados_separados = $contador;
                }else{
                    $report->numero_de_empleados_separados = 0;
                }
            }
            $report->indice_de_ausentismo = intval($report->numero_de_empleados_separados) / intval($report->numero_de_empleados_asistidos);
        }
        return response()->json([
            'msg' => $reports,
        ], status: 200);
    } else {
        return response()->json([
            'msg' => 'vacio',
        ], status: 200);
    }
    return $reports;
}
```

Figura 80. Código de RF14

En esta imagen 80 de la presente tesis se puede observar el código fuente del requerimiento funcional 14.

Requerimiento RF15 – Llamar a recursos humanos

El requerimiento 15, tiene como objetivo que la aplicación debe poder llamar a recursos humanos, dicho requerimiento solo debe hacerlo los usuarios “Obrero” y “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 15).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

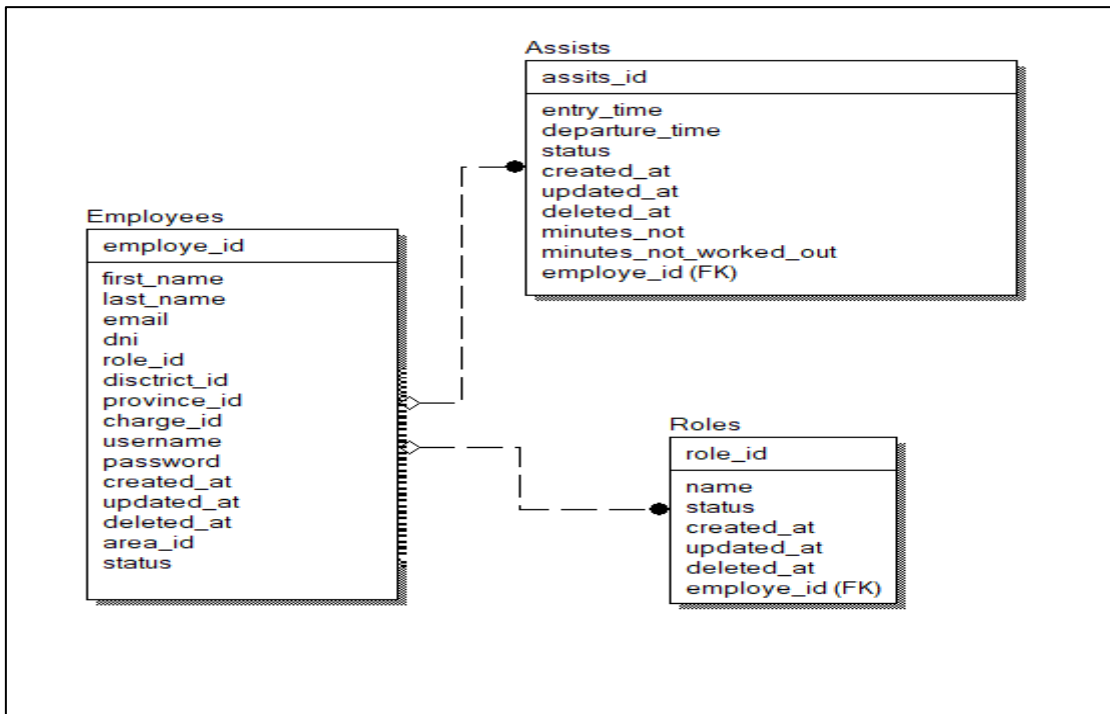


Figura 81. Diagrama lógico de la base de datos RF15

© Fuente: Elaboración propia

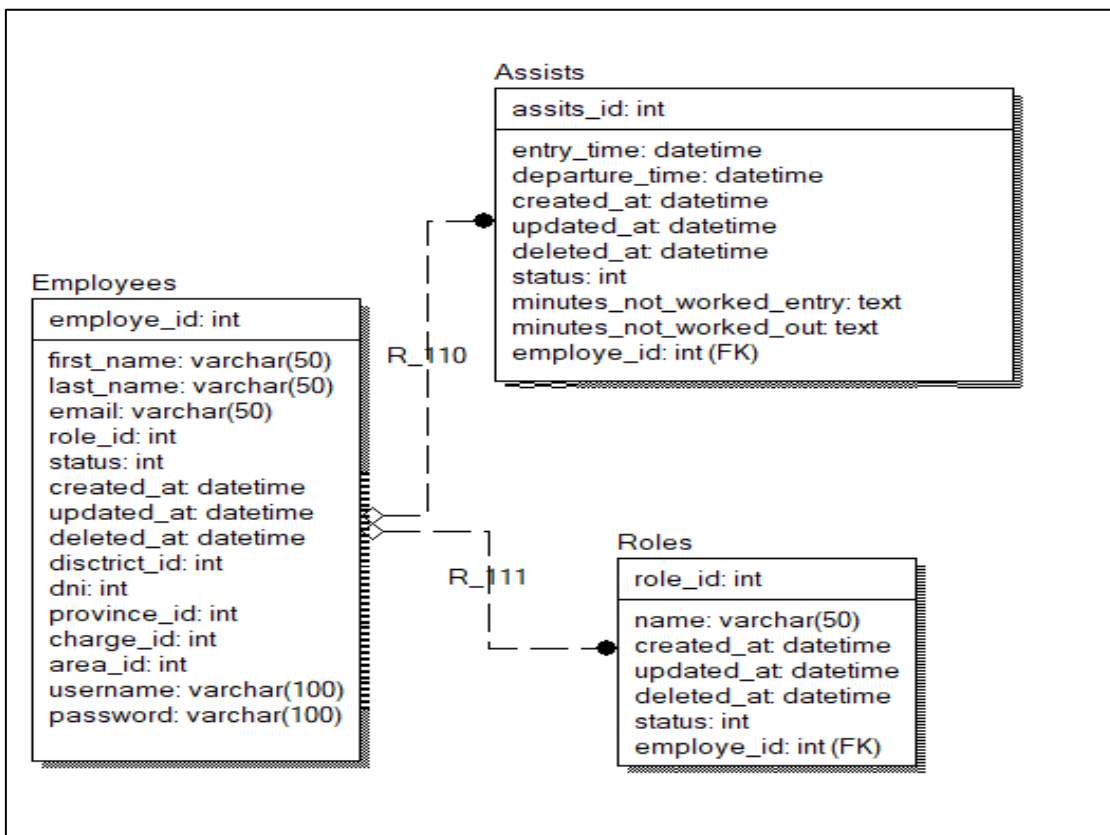


Figura 82. Diagrama físico de la base de datos RF15

Diseño



Figura 83. Prototipo de RF15

En la figura 83 se muestra el prototipo para la GUI de llamar a recursos humanos tanto del usuario obrero como administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

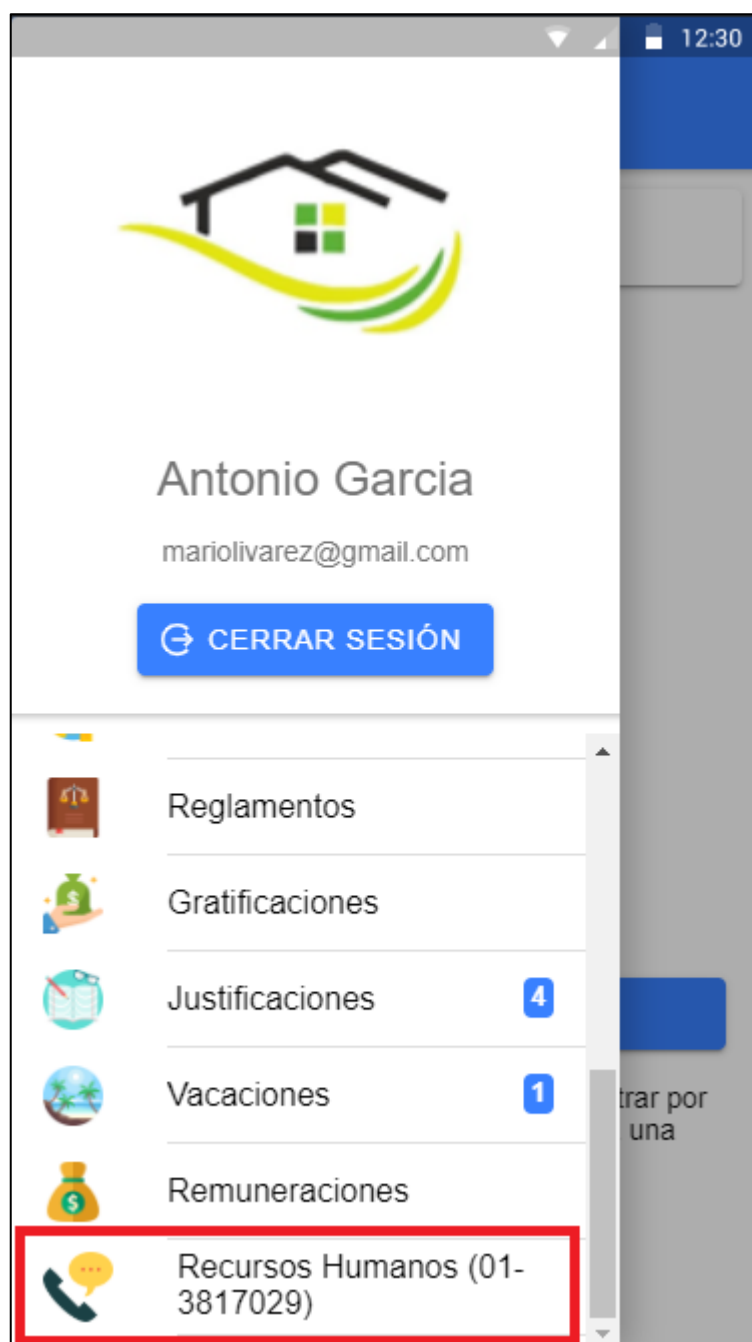


Figura 84. Interface de RF15

En la figura 84 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
<ion-list>
  <ion-menu-toggle auto-hide="false">
    <ion-item [routerDirection]="root" (click)="redirect(p)" *ngFor="let p of pages">
      <ion-icon slot="start" [name]="p.icon"></ion-icon>
      <ion-label>
        | {{p.title}}
      </ion-label>
    </ion-item>
    <ion-item *ngIf="role_id_menu == 1"><ion-icon slot="start" name="call"> </ion-icon> <a href="tel:013817029" class="call-number">Recursos Humanos (01-3817029)</a>
    </ion-item>
    <ion-item *ngIf="role_id_menu == 3"><ion-icon slot="start" name="call"> </ion-icon> <a href="tel:013817029" class="call-number">Recursos Humanos (01-3817029)</a>
    </ion-item>
  </ion-menu-toggle>
</ion-list>
```

Figura 85. Código de RF15

En esta imagen 85 de la presente tesis se puede observar el código fuente del requerimiento funcional 15.

Requerimiento RF16 – Visualizar obra asignada

El requerimiento 16, tiene como objetivo que en la aplicación los obreros deben poder ver la información de la obra que le han asignado, dicho requerimiento solo debe hacerlo el usuario “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 16).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

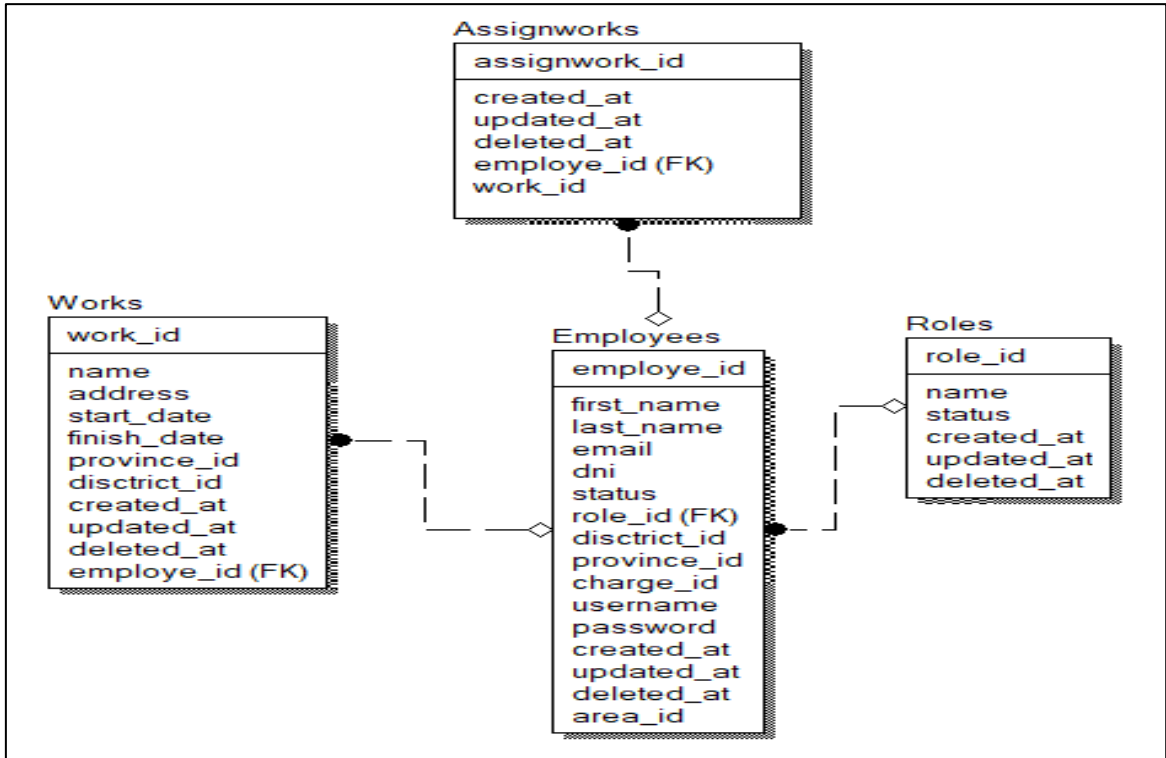


Figura 86. Diagrama lógico de la base de datos RF16

© Fuente: Elaboración propia

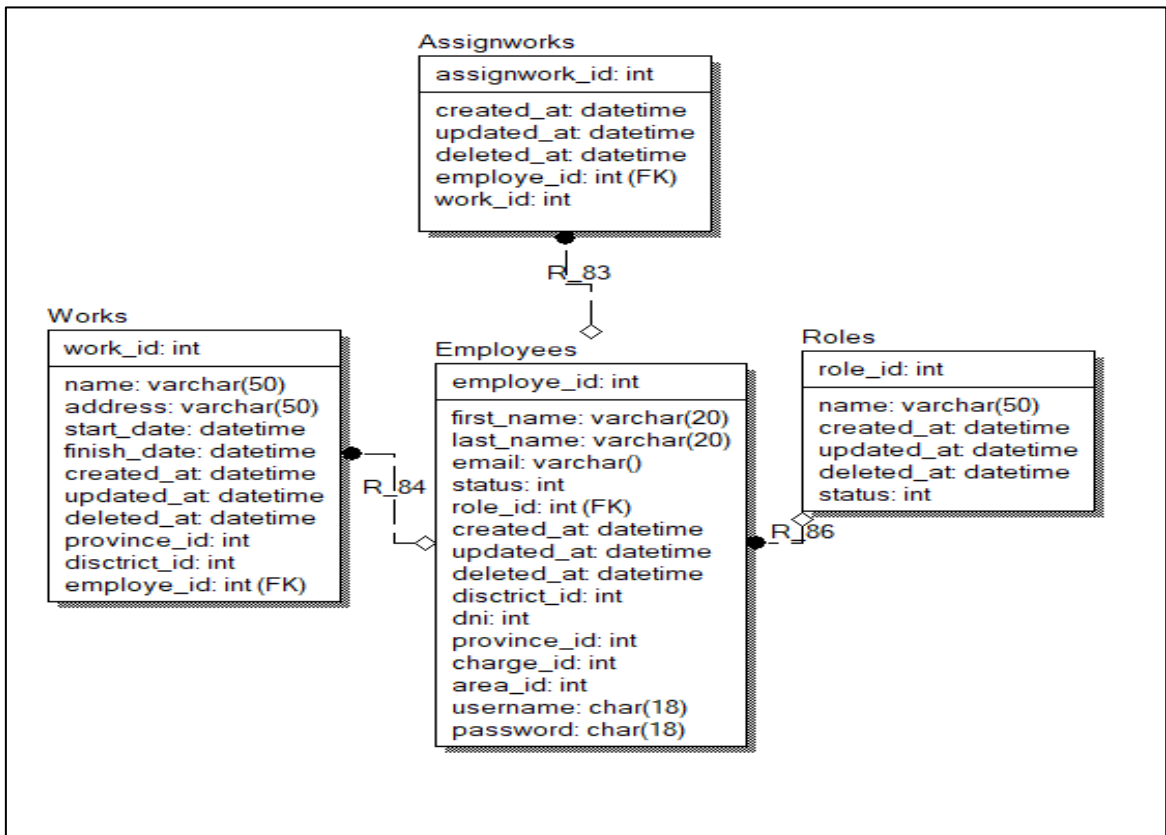


Figura 87. Diagrama físico de la base de datos RF16

Diseño

© Fuente: Elaboración propia

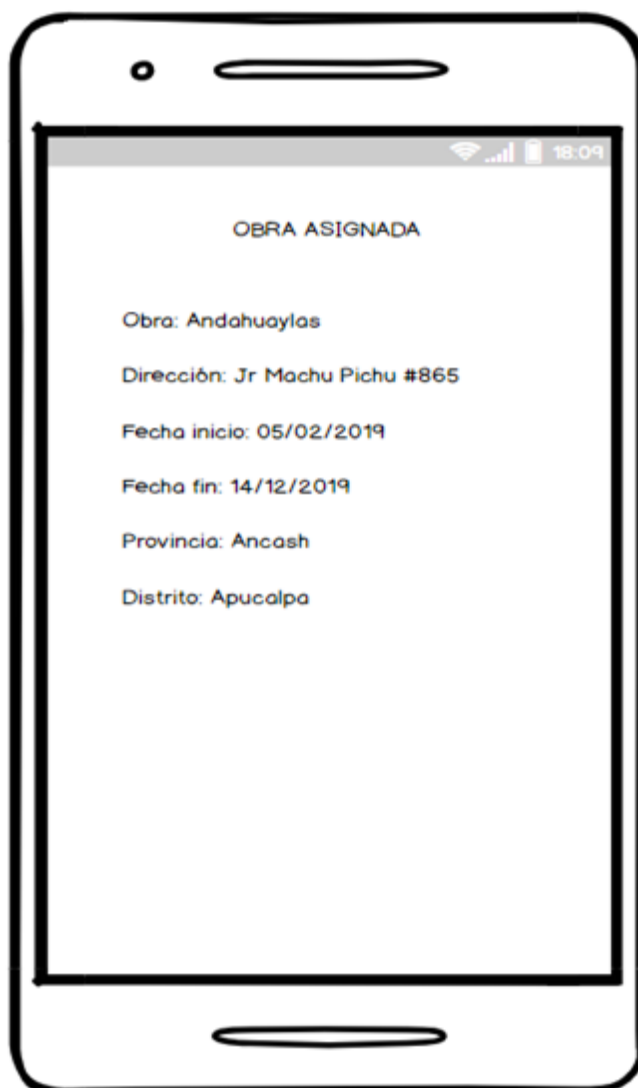


Figura 88. Prototipo de RF16

En la figura 88 se muestra el prototipo para la GUI de visualizar obra asignada del usuario obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



Figura 89. Interface de RF16

En la figura 89 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function showWorkEmployee(Request $request){
    $request = Assignwork::with( relations: 'work')->where( column: 'employee_id', $request->id)->first();
    return $request->work;
}
```

Figura 90. Código de RF16

En esta imagen 90 de la presente tesis se puede observar el código fuente del requerimiento funcional 16.

Requerimiento RF17 – Listar empleados

El requerimiento 17, tiene como objetivo que en la aplicación se debe poder visualizar el listado de todos los empleados que trabajan en la empresa, dicho requerimiento solo debe hacerlo el usuario “Administrador”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

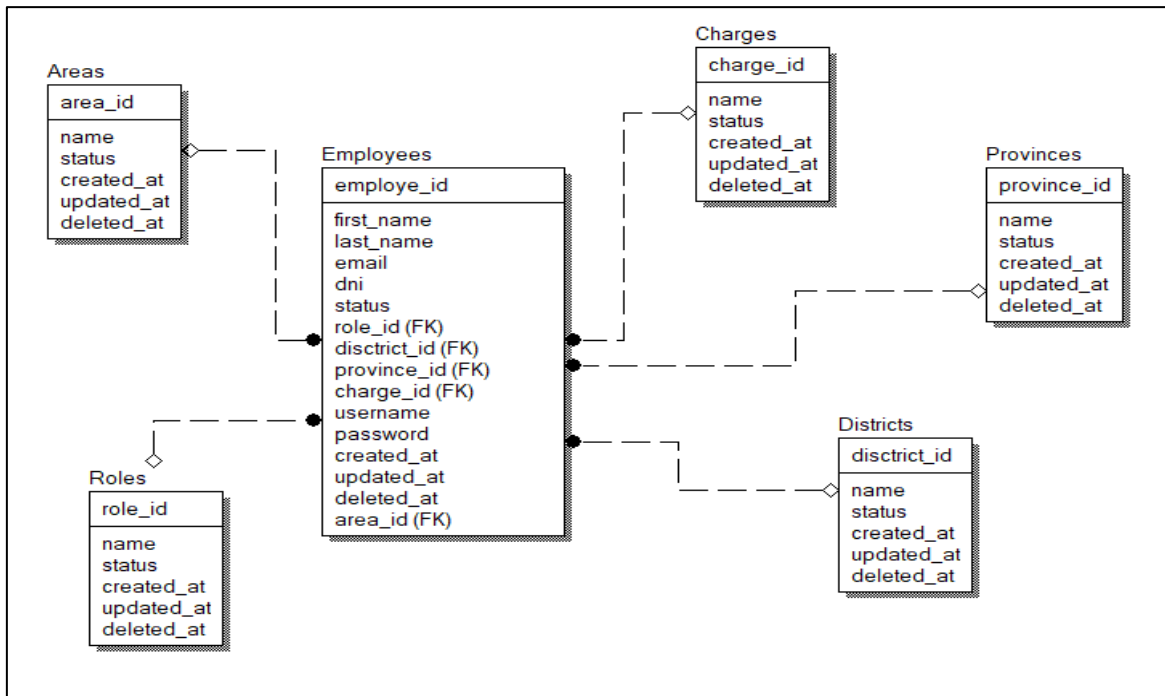


Figura 91. Diagrama lógico de la base de datos RF17

© Fuente: Elaboración propia

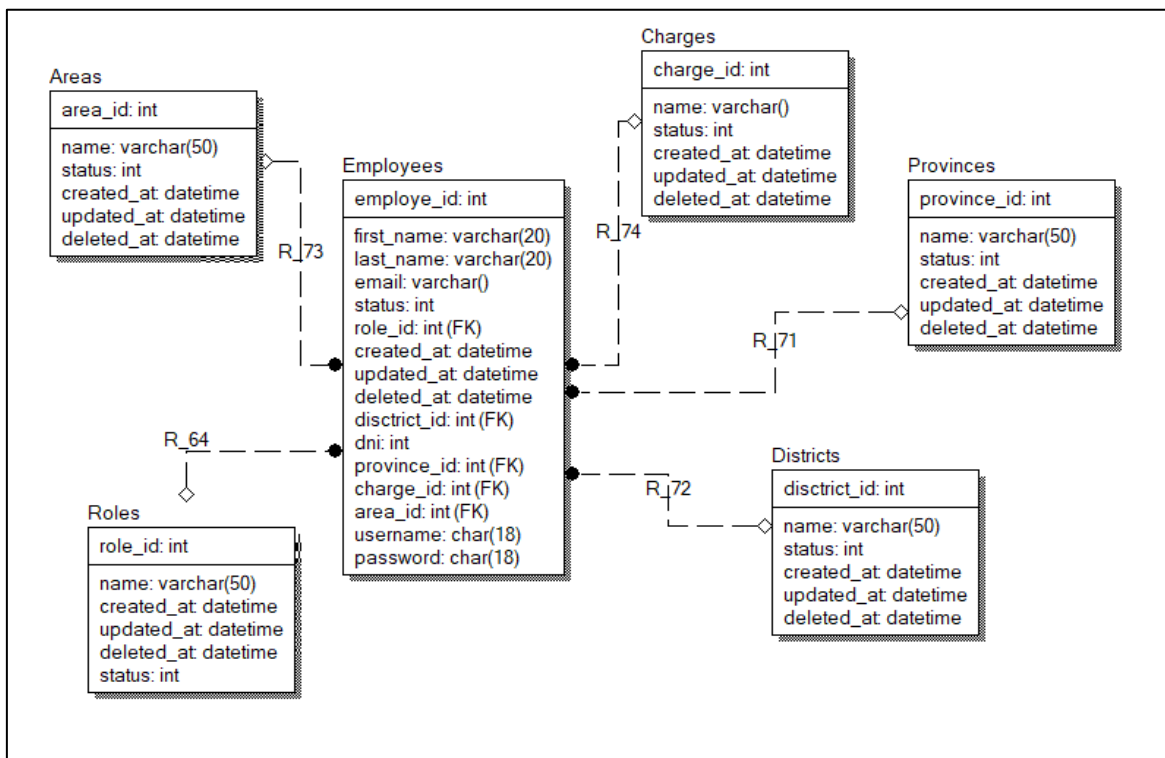


Figura 92. Diagrama físico de la base de datos RF17

Diseño

© Fuente: Elaboración propia

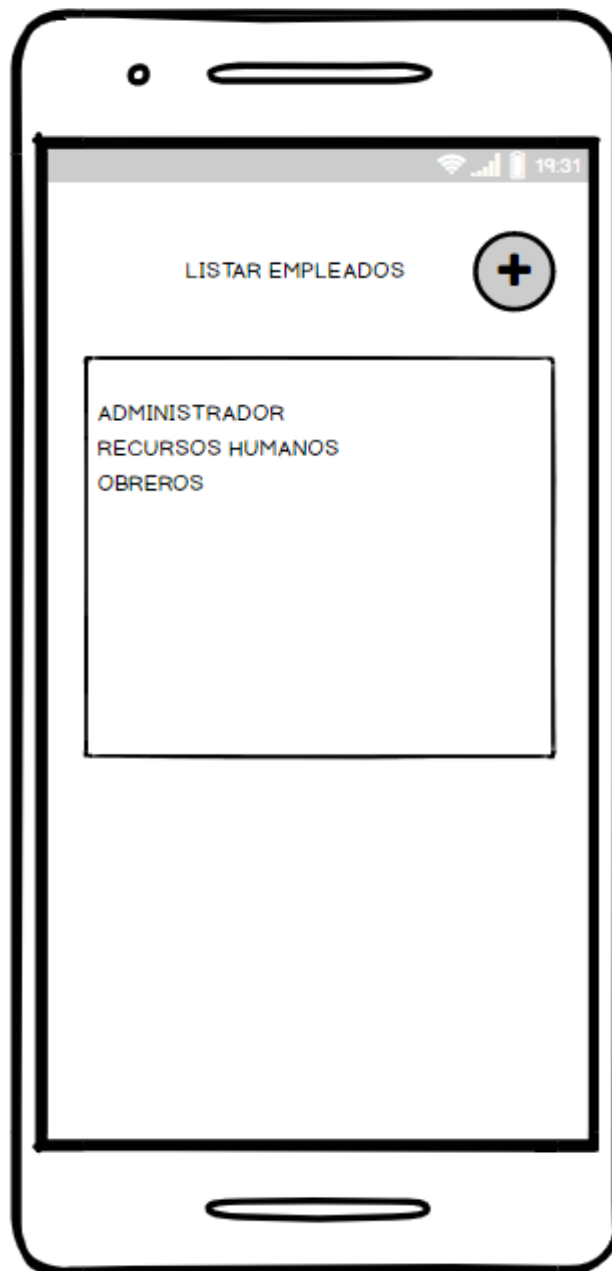


Figura 93. Prototipo de RF17

En la figura 93 se muestra el prototipo para la GUI de listar empleados del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

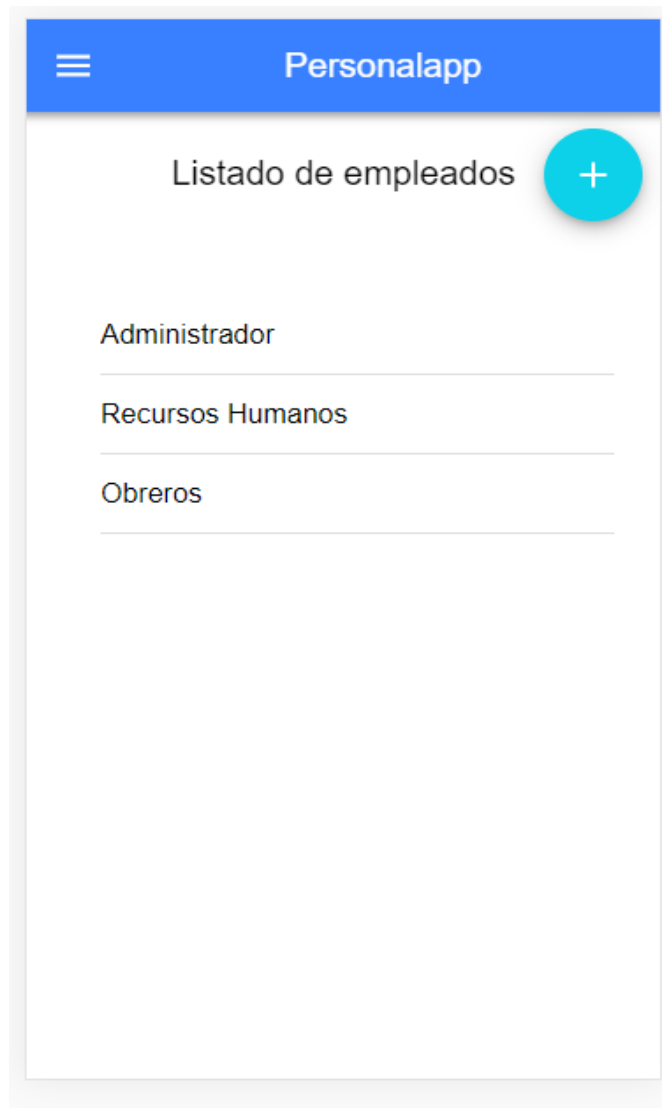


Figura 94. Interface de RF17

En la figura 94 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function index(Request $request) {  
    $employees = Employee::where('role_id', $request->all())->get();  
    return $employees;  
}
```

Figura 95. Código de RF17

En esta imagen 95 de la presente tesis se puede observar el código fuente del requerimiento funcional 17.

Requerimiento RF18 – Listar obras

El requerimiento 18, tiene como objetivo que en la aplicación se debe poder visualizar el listado de todas las obras que la empresa ha trabajado, dicho requerimiento solo debe hacerlo el usuario “Administrador”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

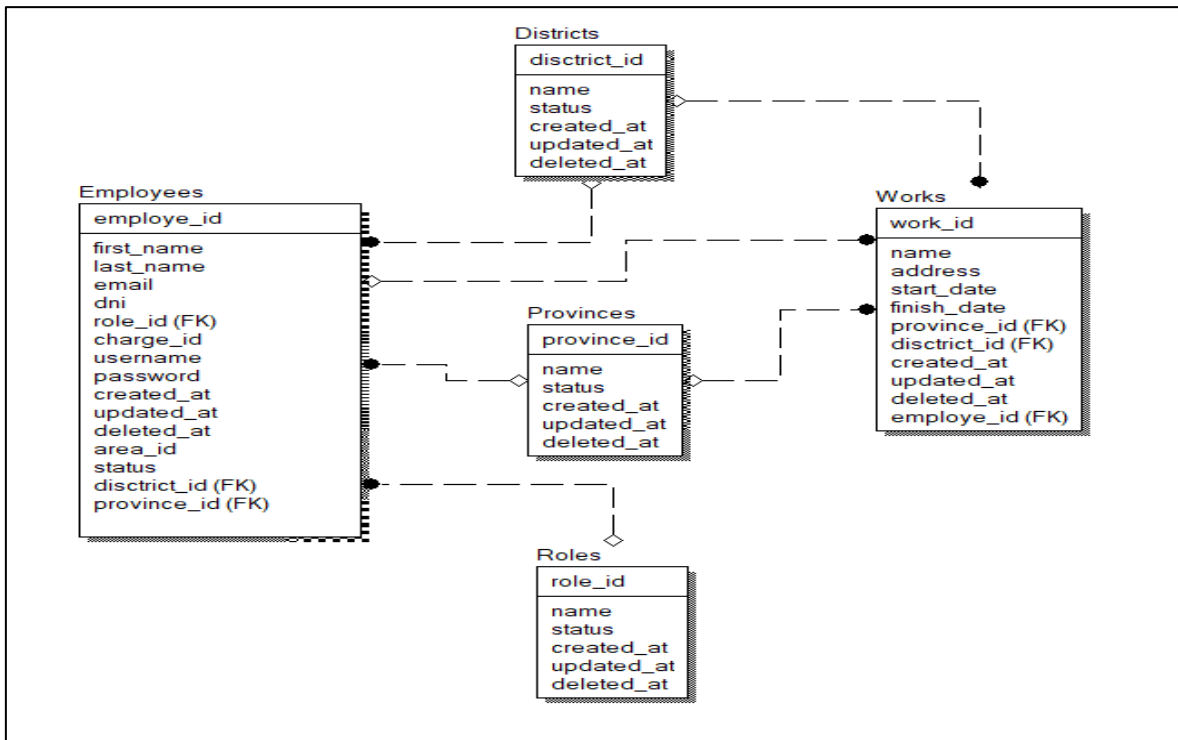


Figura 96. Diagrama lógico de la base de datos RF18

© Fuente: Elaboración propia

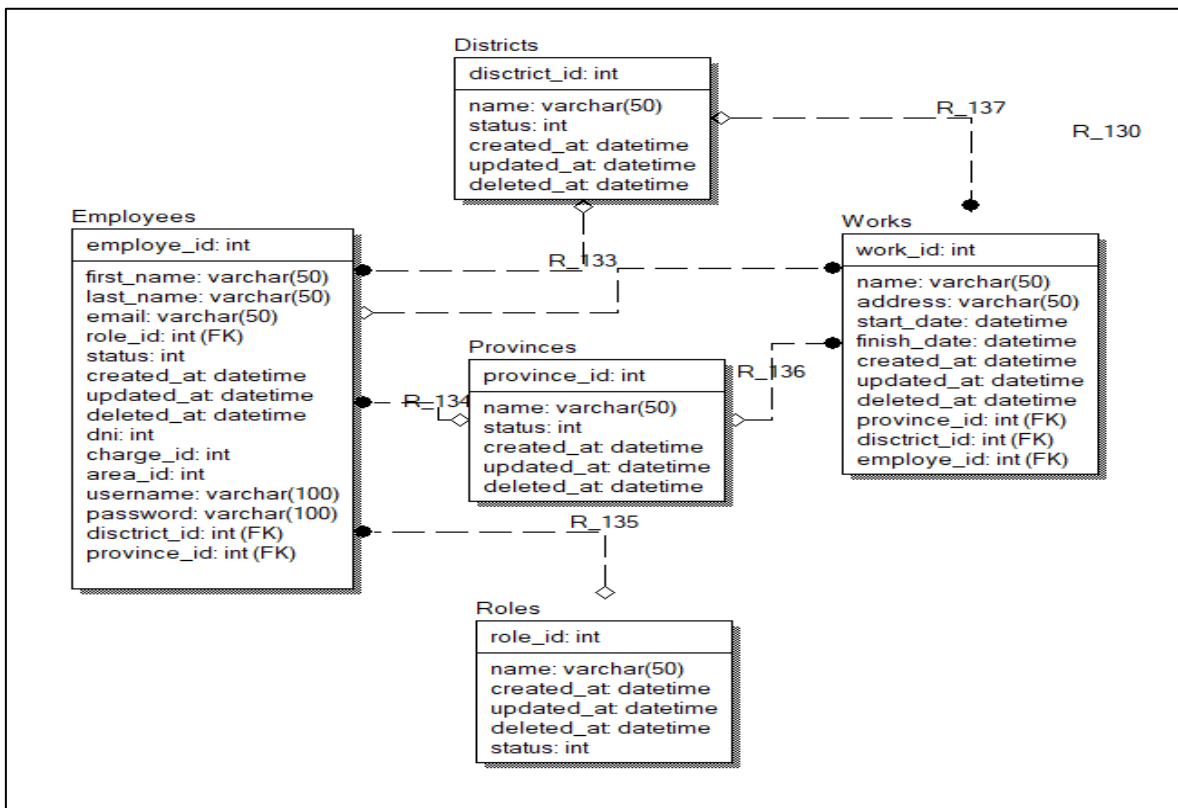


Figura 97. Diagrama físico de la base de datos RF18

Diseño

© Fuente: Elaboración propia

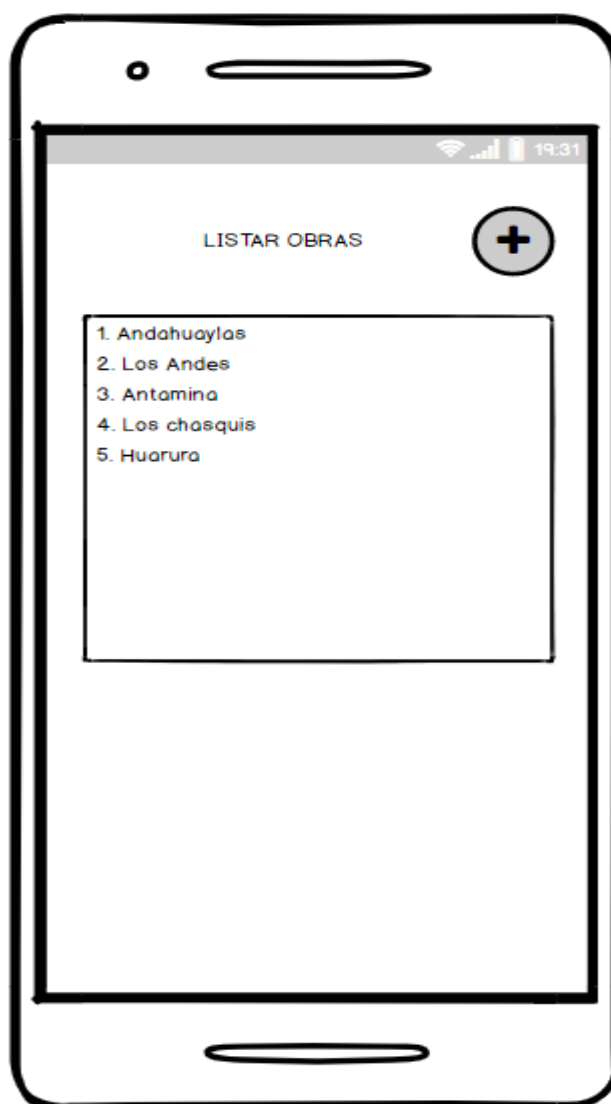


Figura 98. Prototipo de RF18

En la figura 98 se muestra el prototipo para la GUI de listar obras del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

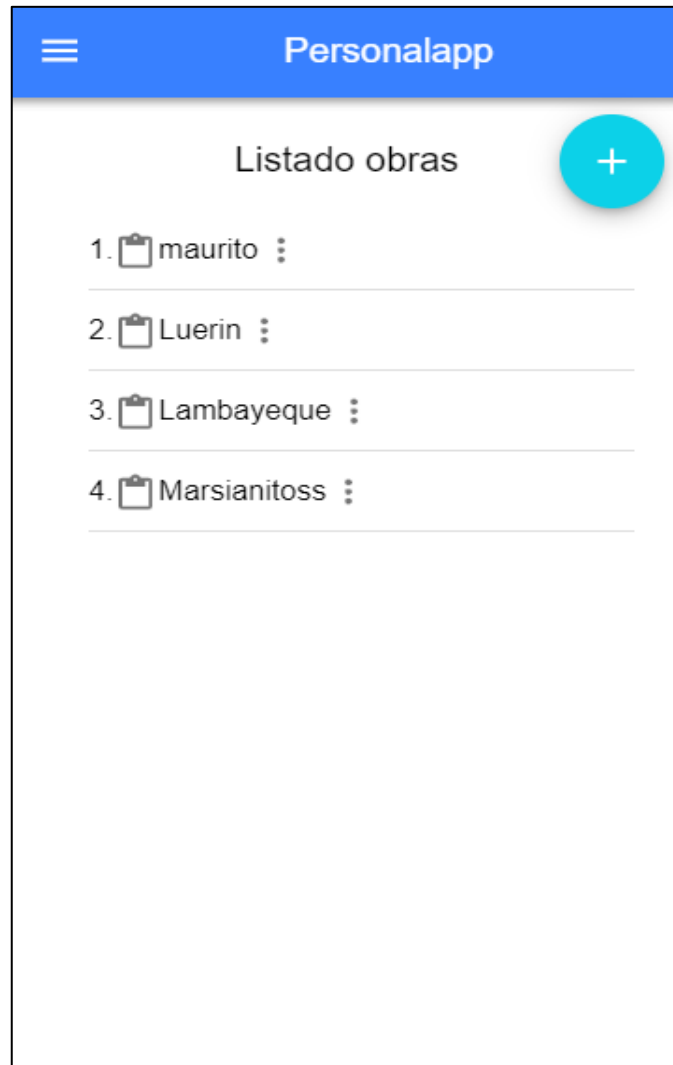


Figura 99. Interface de RF18

En la figura 99 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function index() {  
    $works = Work::all();  
    return $works;  
}
```

Figura 100. Código de RF18

En esta imagen 100 de la presente tesis se puede observar el código fuente del requerimiento funcional 18.

Requerimiento RF19 – Ver detalle obra

El requerimiento 19, tiene como objetivo que en la aplicación se debe poder ver toda la información de la obra, tanto la fecha de inicio como la fecha de fin también, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 19).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

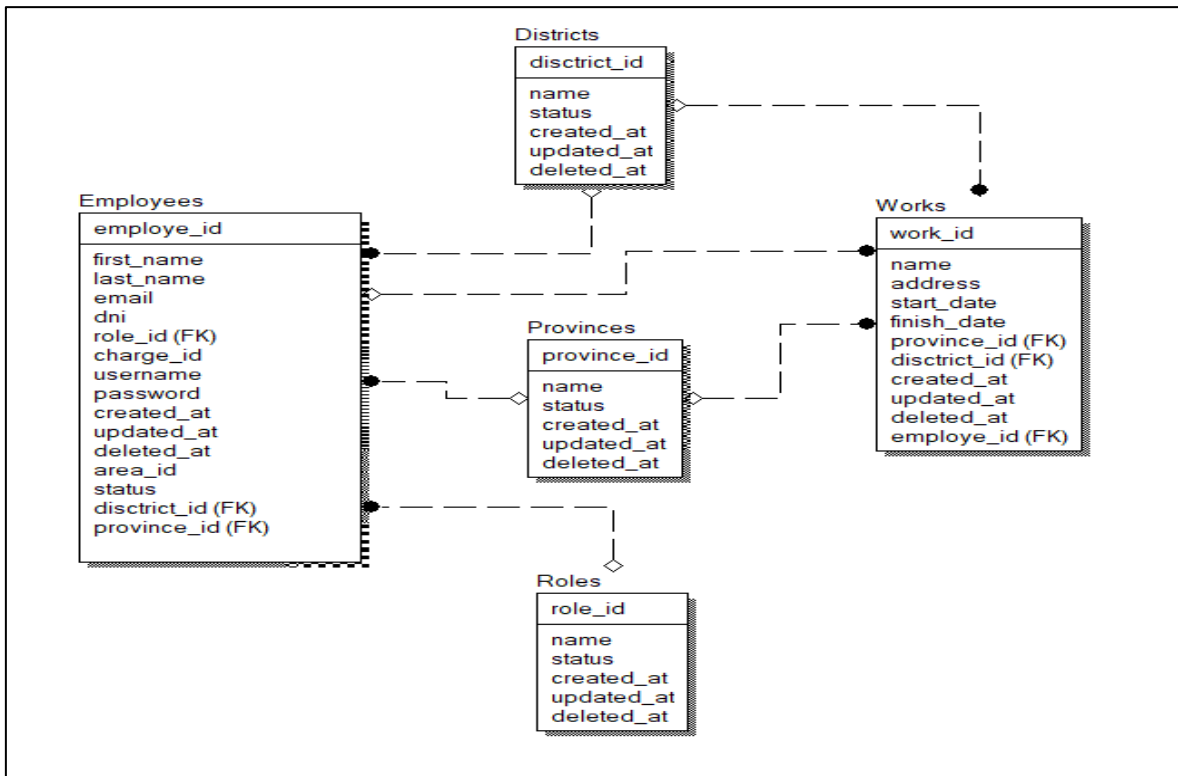


Figura 101. Diagrama lógico de la base de datos RF19

© Fuente: Elaboración propia

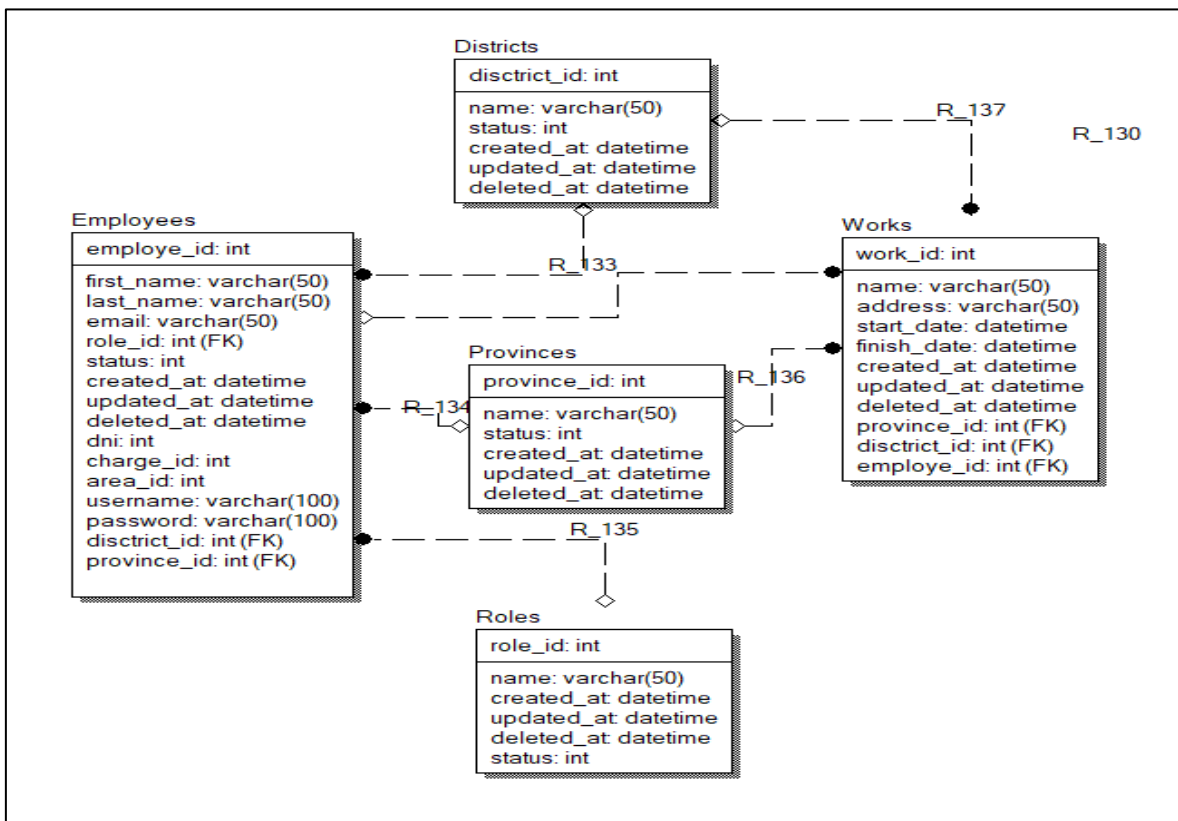


Figura 102. Diagrama físico de la base de datos RF19

Diseño

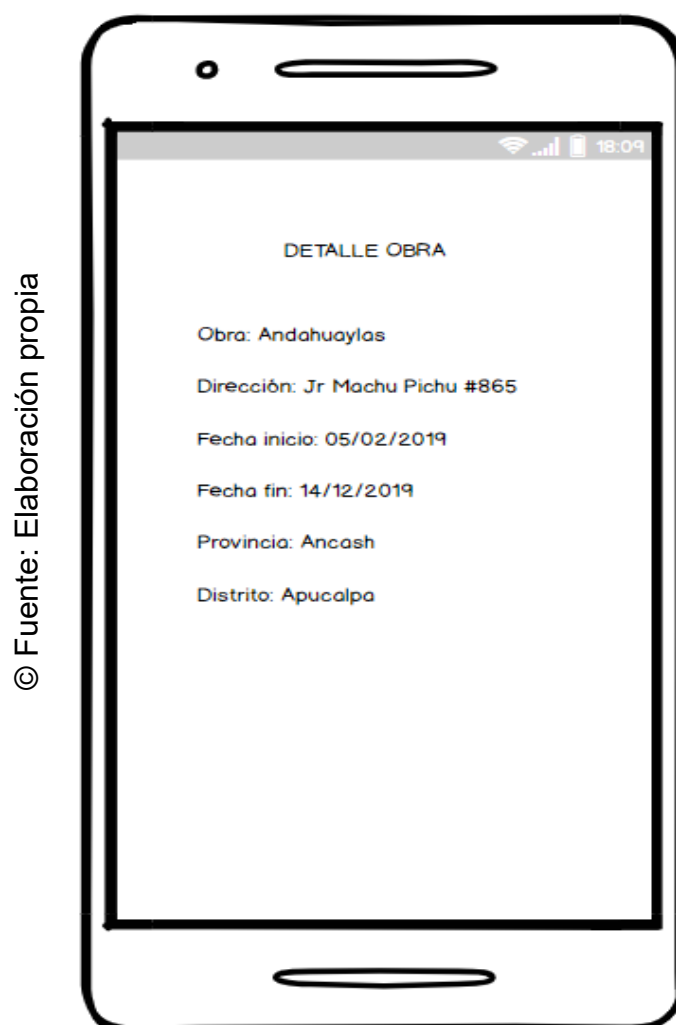


Figura 103. Prototipo de RF19

En la figura 103 se muestra el prototipo para la GUI de ver detalle de obra del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

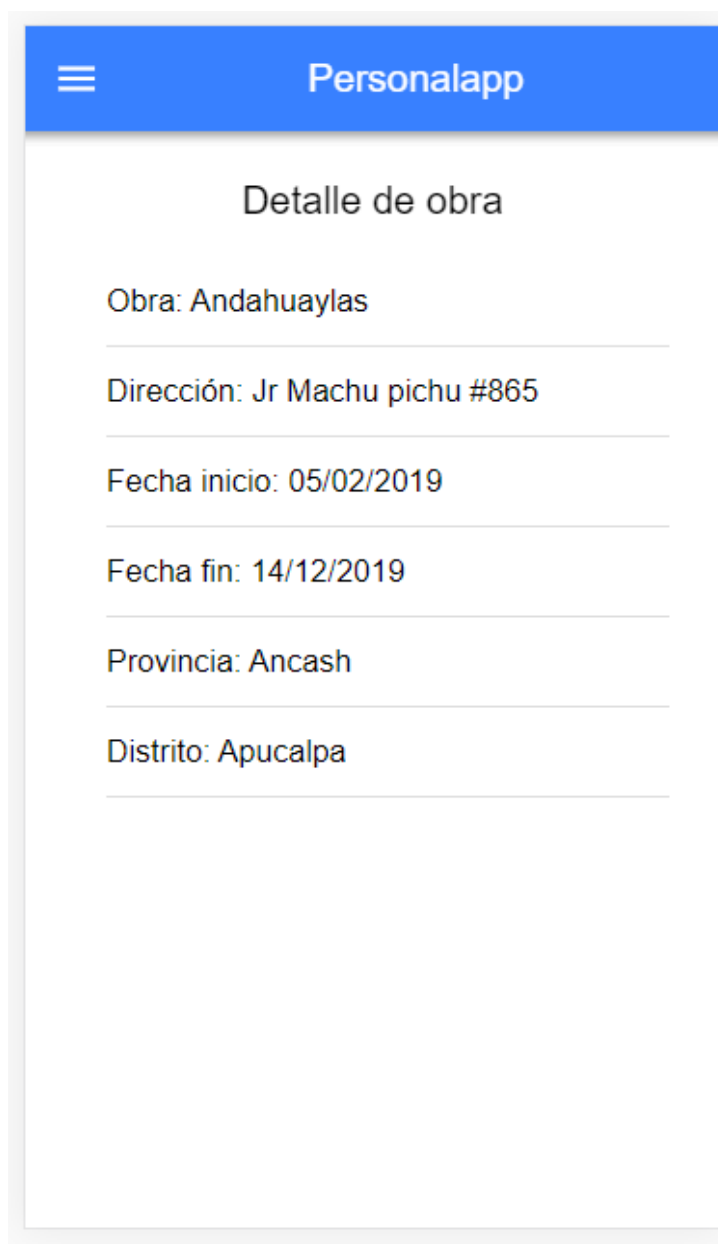


Figura 104. Interface de RF19

En la figura 104 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function detailWork(Request $request) {  
    $work = Work::where('id', $request->id)->first();  
    return $work;  
}
```

Figura 105. Código de RF19

En esta imagen 105 de la presente tesis se puede observar el código fuente del requerimiento funcional 19.

Requerimiento RF20 – Visualizar empleados asignados a obra

El requerimiento 20, tiene como objetivo que en la aplicación se debe poder ver la lista de obreros que están asignados en dicha obra, es decir los que están trabajando, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 20).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

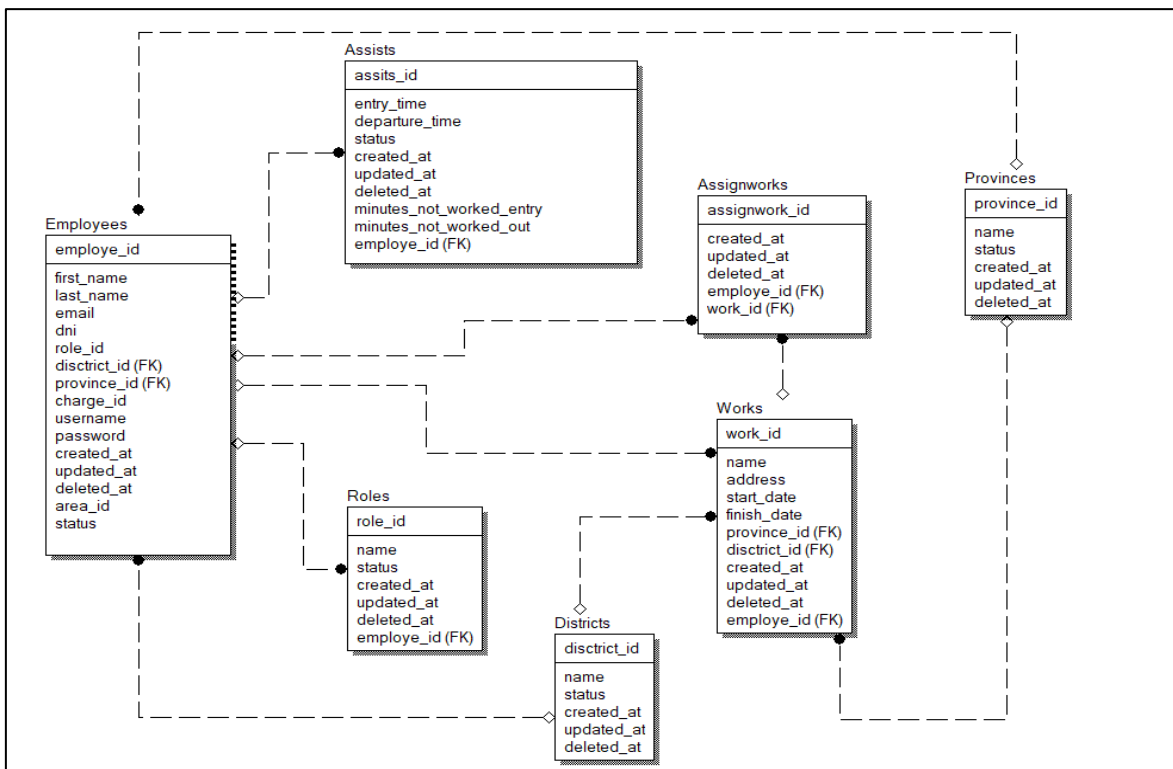


Figura 106. Diagrama lógico de la base de datos RF20

© Fuente: Elaboración propia

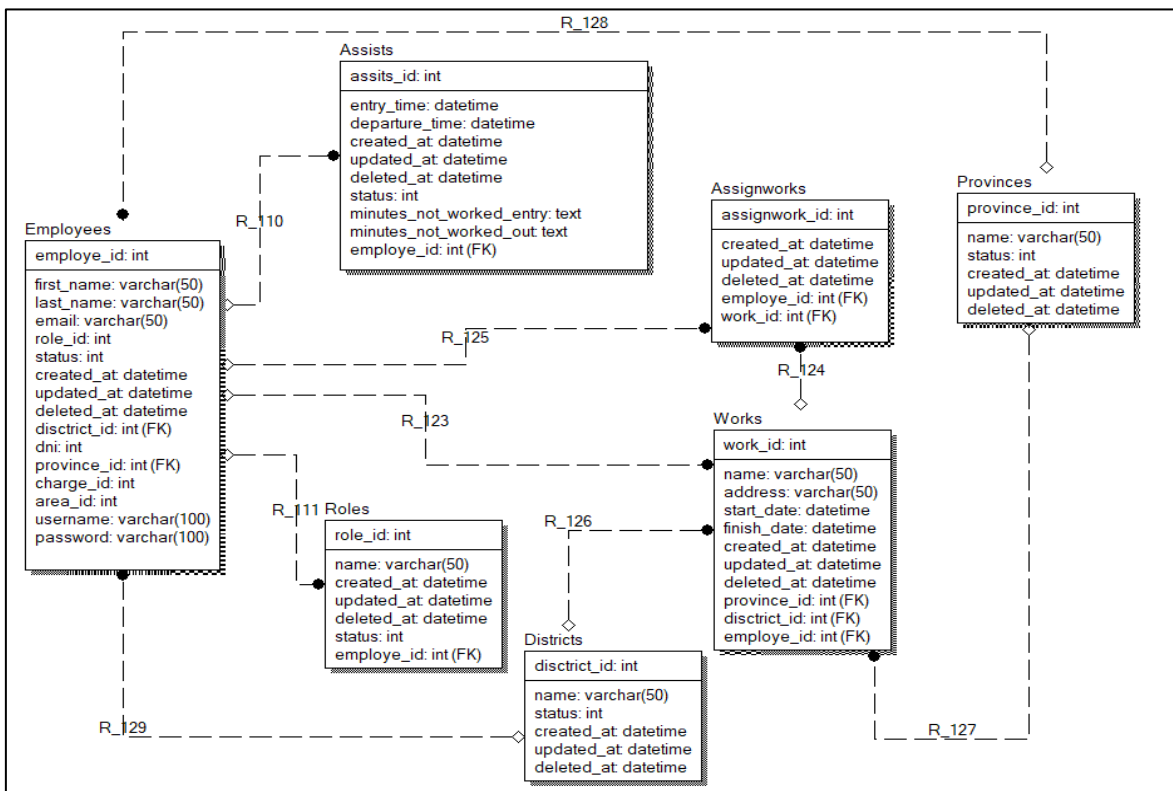


Figura 107. Diagrama físico de la base de datos RF20

Diseño

© Fuente: Elaboración propia

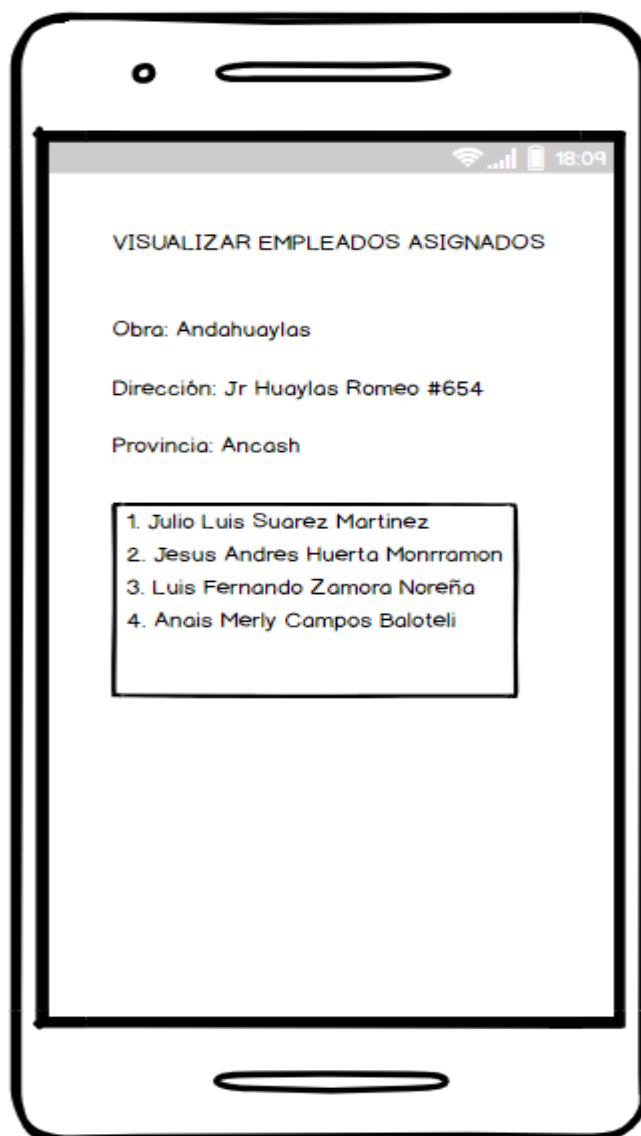


Figura 108. Prototipo de RF20

En la figura 108 se muestra el prototipo para la GUI de visualizar empleados asignados a una obra del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

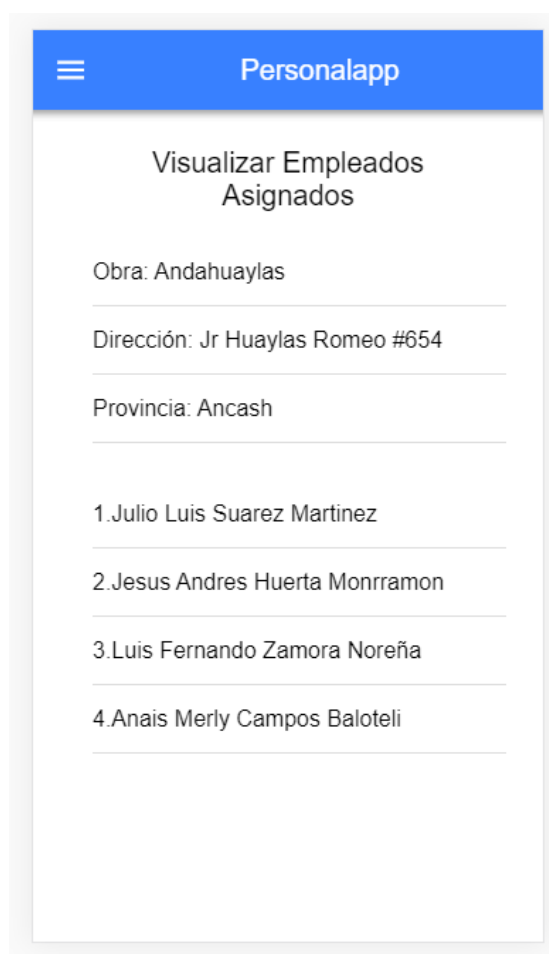


Figura 109. Interface de RF20

En la figura 109 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function employeeAssign(Request $request){
    $request = Assignwork::with( relations: 'employees' )->where( column: 'work_id', $request->id )->first();
    return $request->employees;
}
```

Figura 110. Código de RF20

En esta imagen 110 de la presente tesis se puede observar el código fuente del requerimiento funcional 20.

Requerimiento RF21 – Listar empleados desvinculados

El requerimiento 21, tiene como objetivo que en la aplicación se debe poder ver la lista de empleados desvinculados, es decir empleados que ya no trabajan en la empresa, dicho requerimiento solo debe hacerlo el usuario “Administrador”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

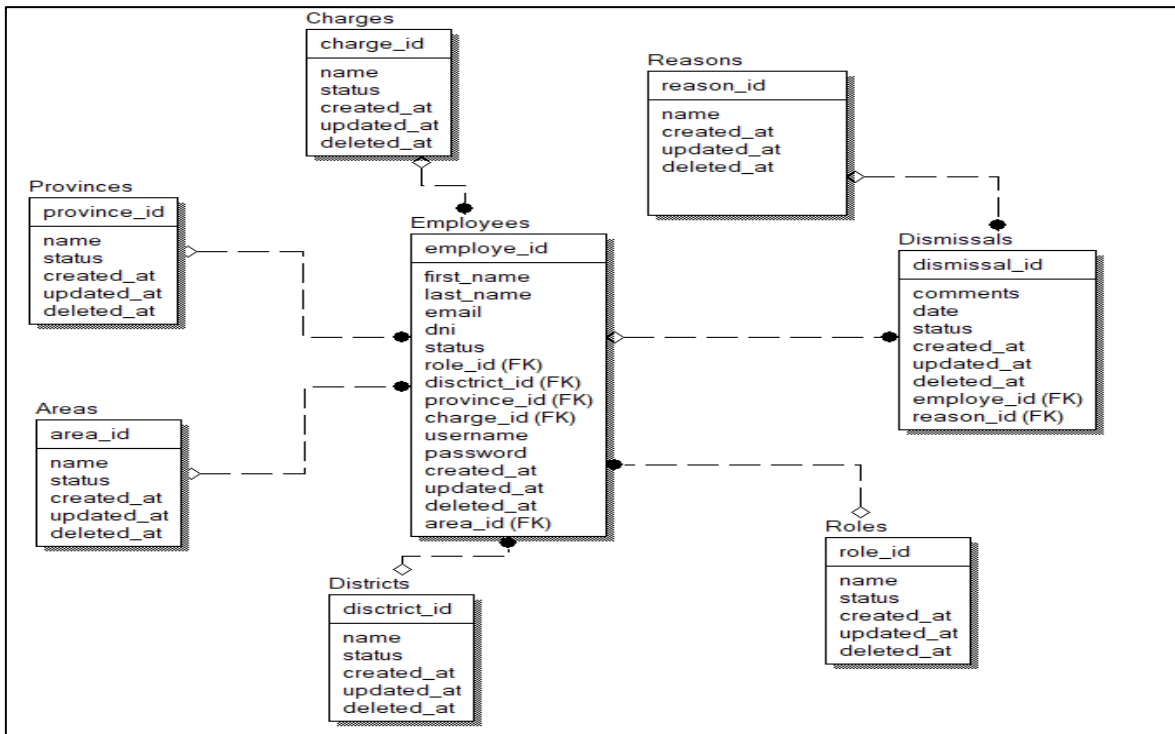


Figura 111. Diagrama lógico de la base de datos RF21

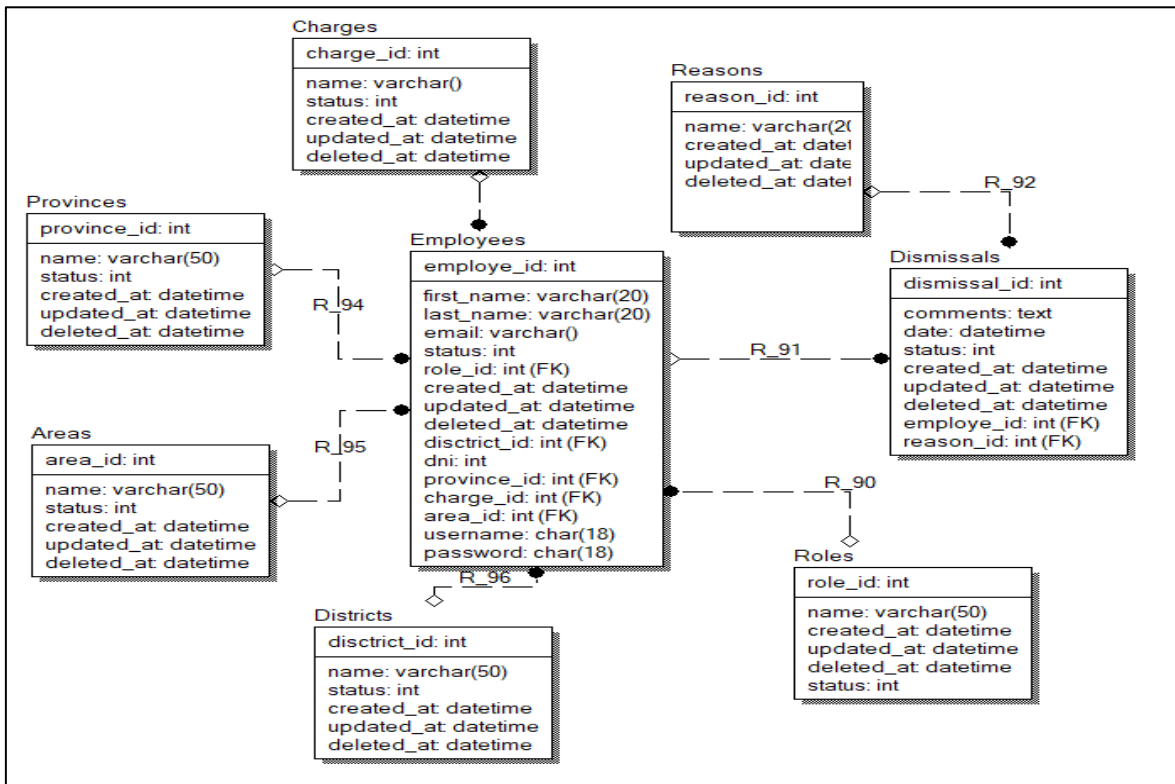


Figura 112. Diagrama físico de la base de datos RF21

Diseño

© Fuente: Elaboración propia

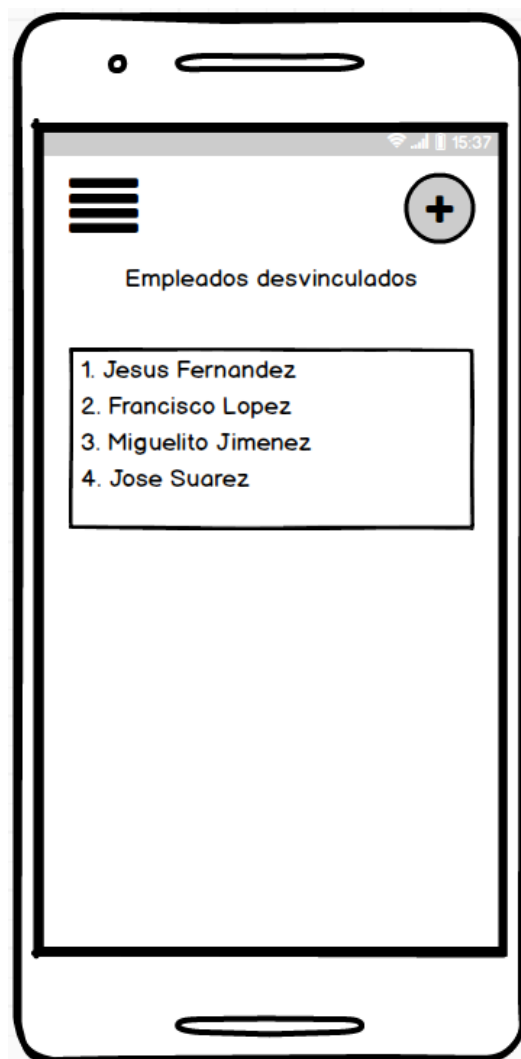


Figura 113. Prototipo de RF21

En la figura 113 se muestra el prototipo para la GUI de listar empleados desvinculados del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

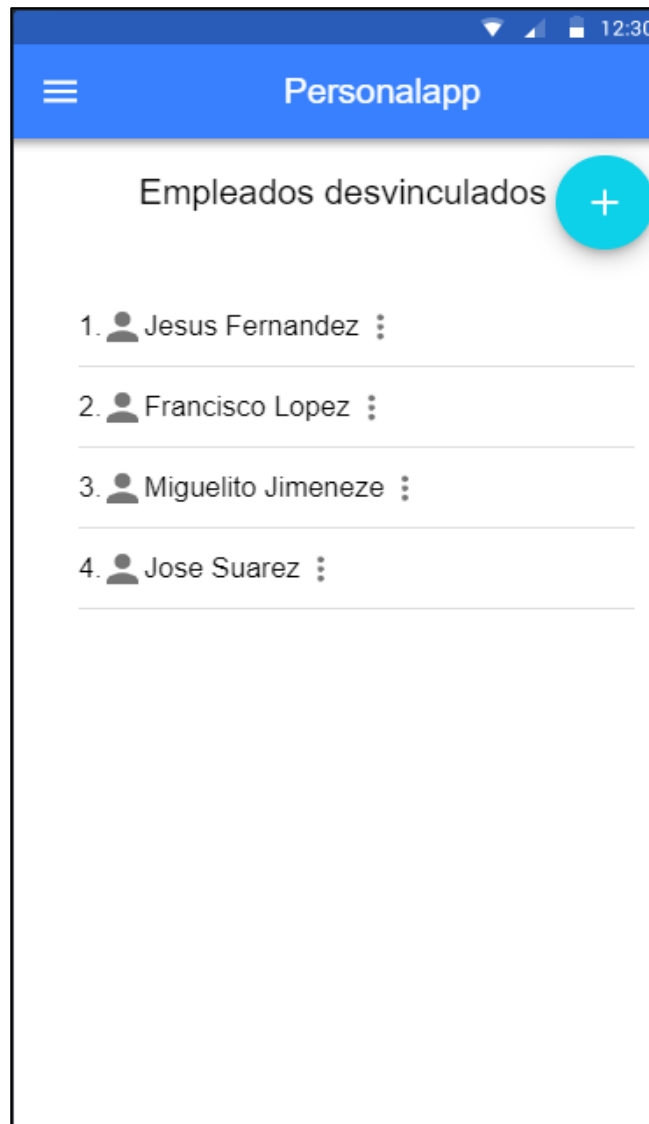


Figura 114. Interface de RF21

En la figura 114 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Funcion para traer empleados desvinculados  
public function dismissal() {  
    $employees = Employee::has('dismissal')->get();  
    return $employees;  
}
```

Figura 115. Código de RF21

En esta imagen 115 de la presente tesis se puede observar el código fuente del requerimiento funcional 21.

Requerimiento RF22 – Ver detalle de empleados desvinculados

El requerimiento 22, tiene como objetivo que en la aplicación se debe poder ver el detalle de los empleados desvinculados, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 22).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

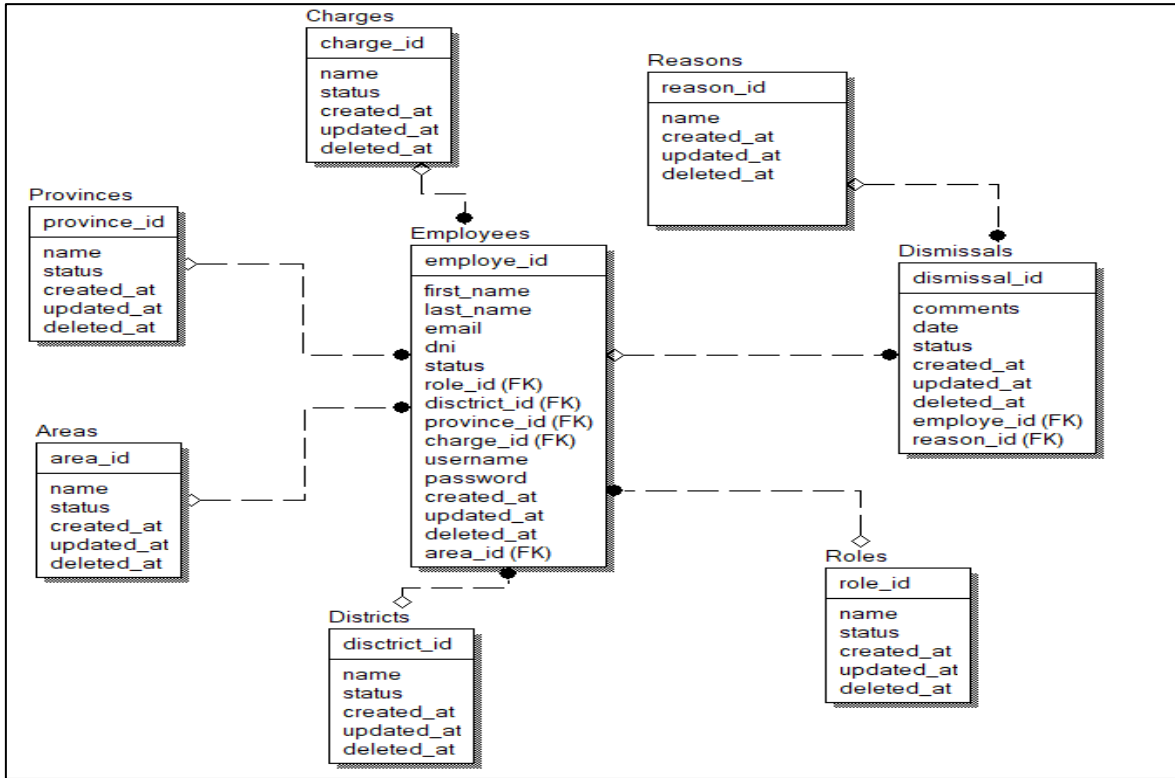


Figura 116. Diagrama lógico de la base de datos RF22

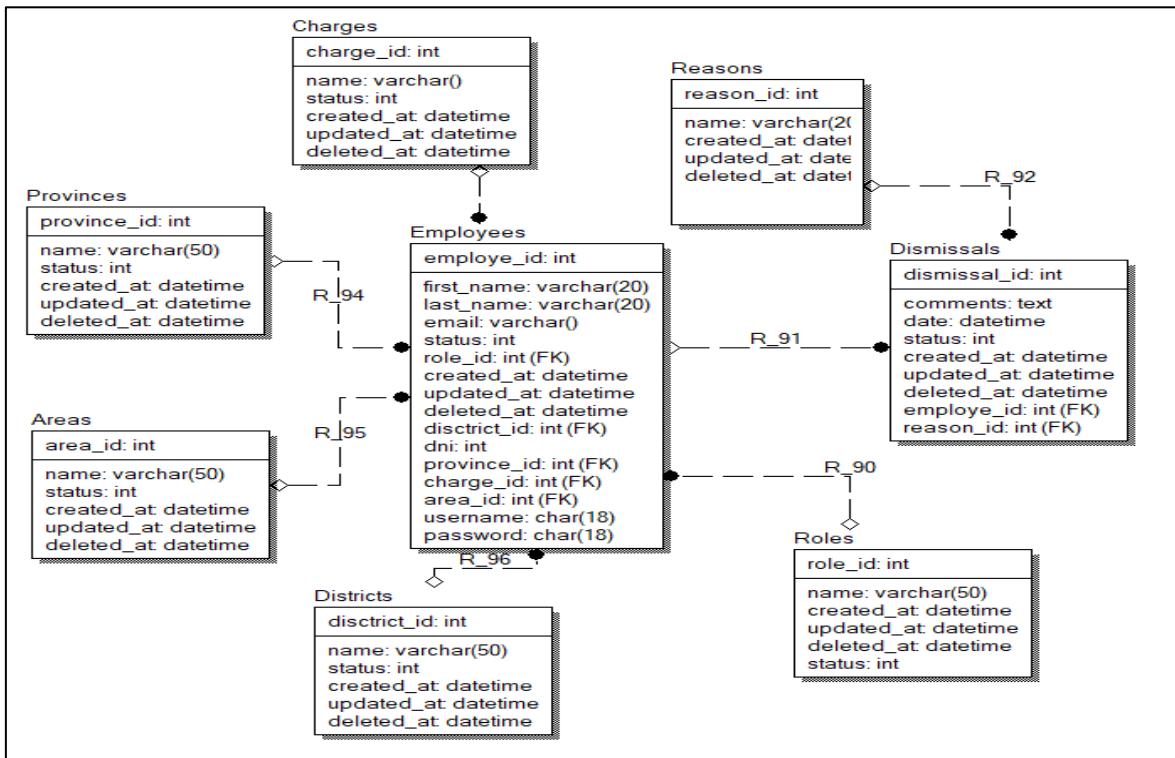


Figura 117. Diagrama físico de la base de datos RF22

Diseño

© Fuente: Elaboración propia .



Figura 118. Prototipo de RF22

En la figura 118 se muestra el prototipo para la GUI de ver detalle de empleados desvinculados del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

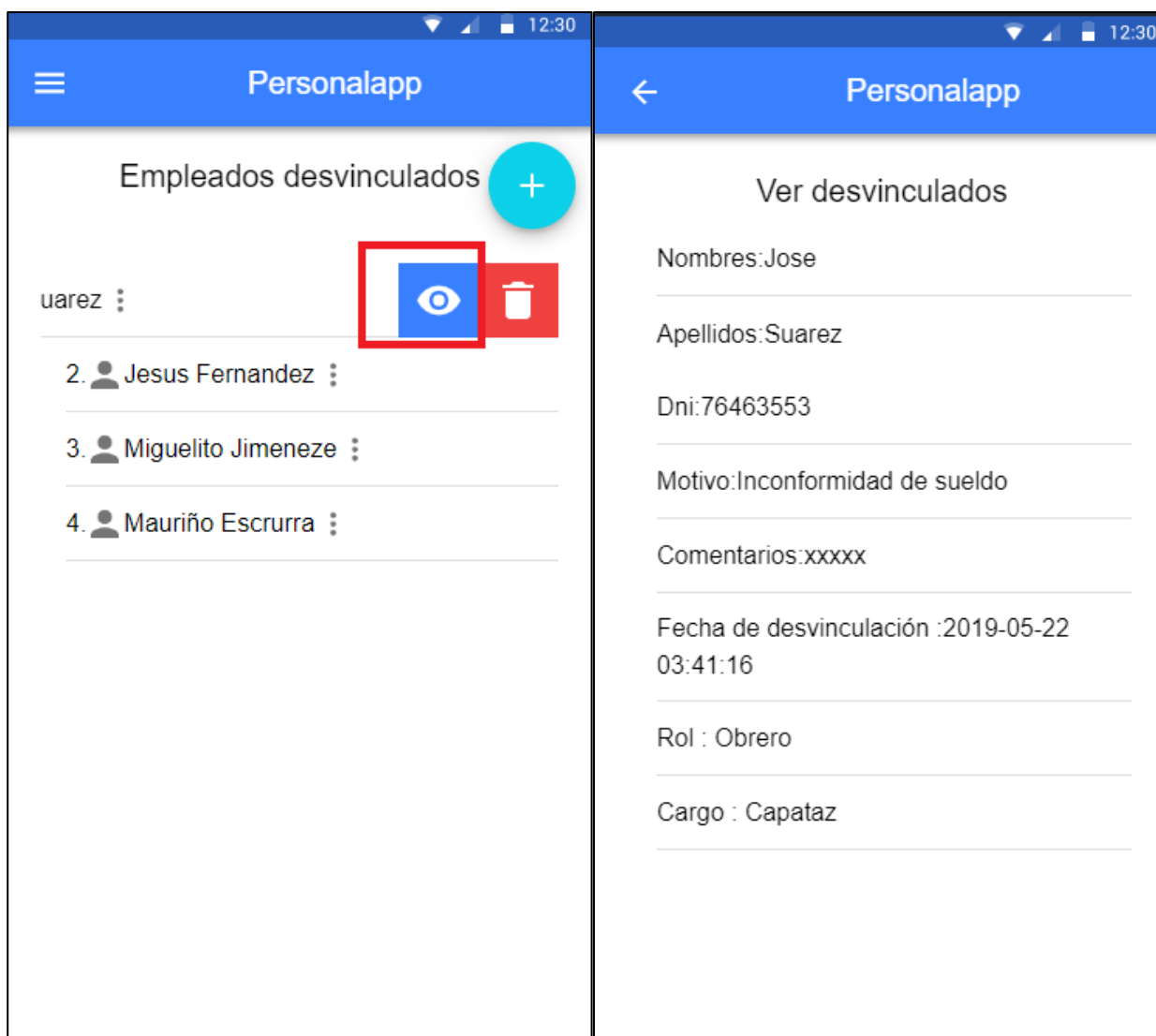


Figura 119. Interface de RF22

En la figura 119 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Funcion para ver el detalle de empleados desvinculados
public function showDismissal($id){
    $employee = Employee::with( relations: 'dismissal.reason', 'charge', 'role' )->findOrFail($id);
    return $employee;
}
```

Figura 120. Código de RF22

En esta imagen 120 de la presente tesis se puede observar el código fuente del requerimiento funcional 22.

Requerimiento RF23 – Eliminar empleados desvinculados

El requerimiento 23, tiene como objetivo que en la aplicación se debe poder eliminar empleados desvinculados, una vez ejecutada esta acción, el empleado seguirá registrado en la aplicación, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 23).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

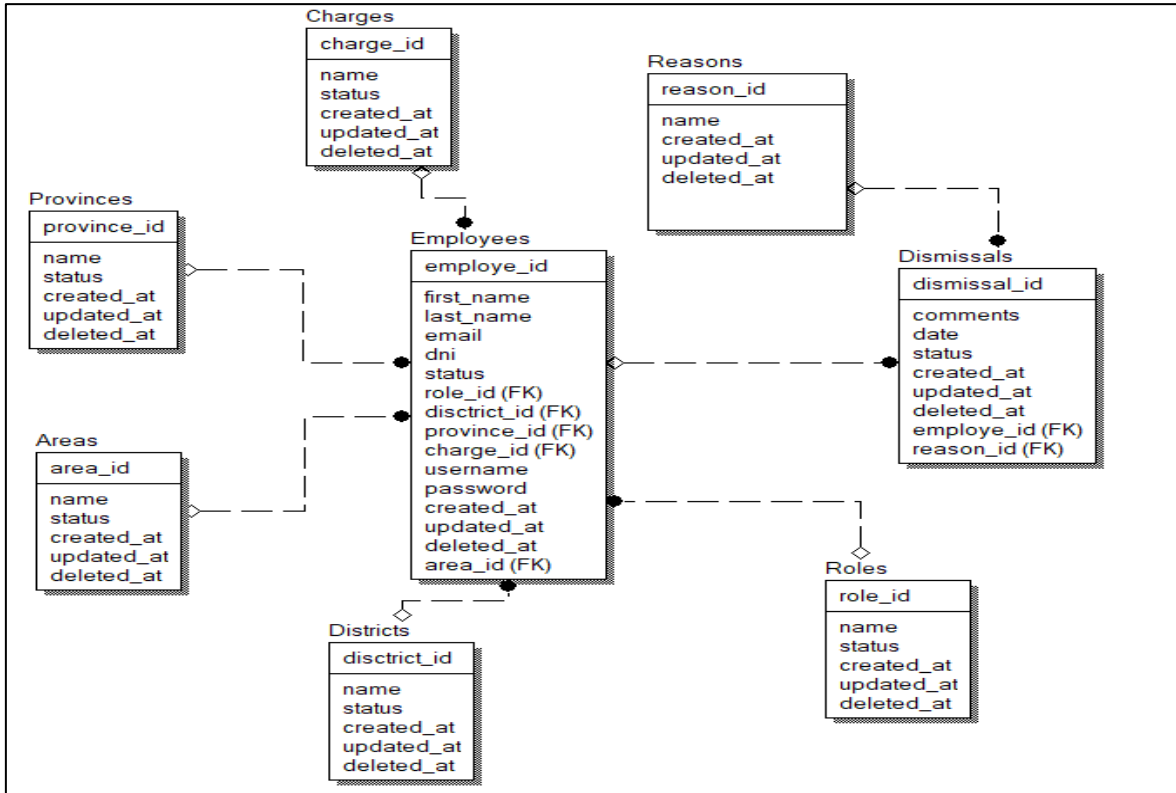


Figura 121. Diagrama lógico de la base de datos RF23

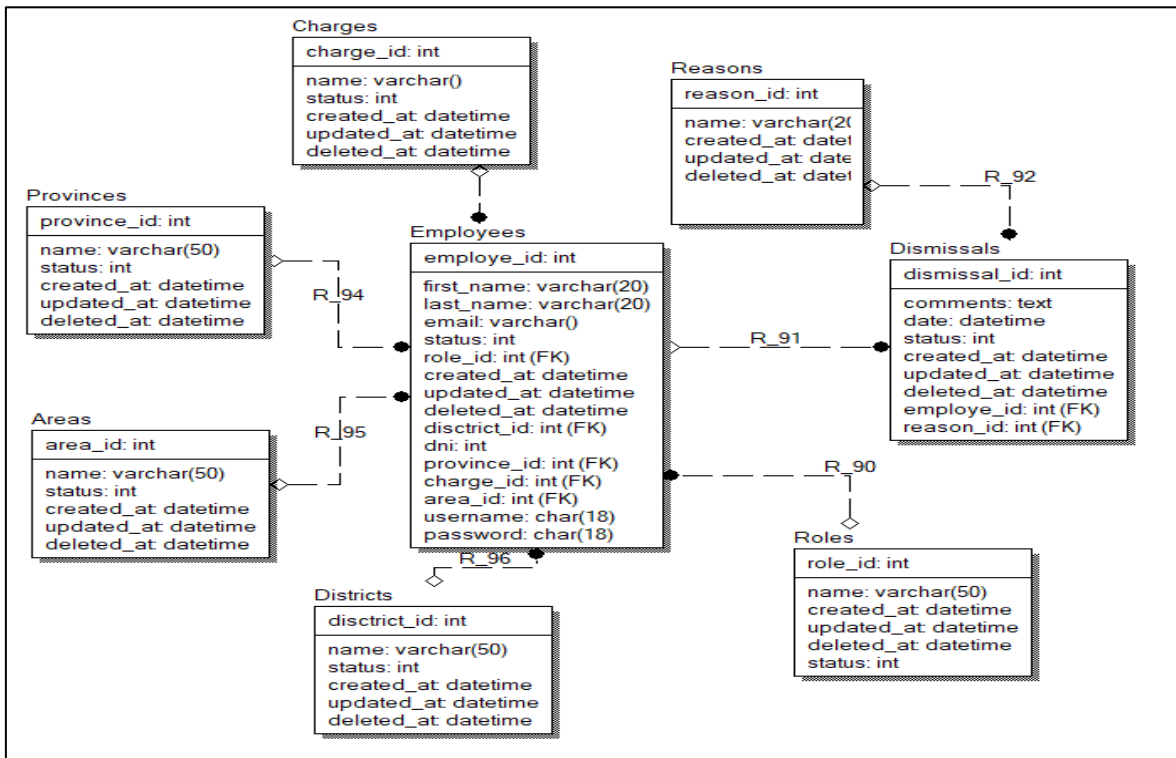


Figura 122. Diagrama físico de la base de datos RF23

Diseño

© Fuente: Elaboración propia

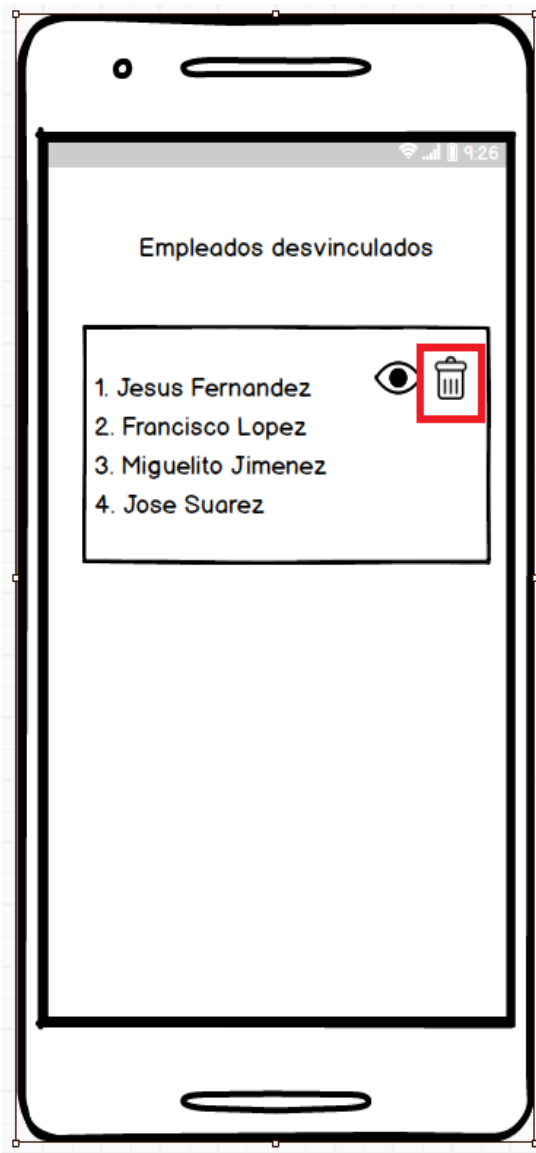


Figura 123. Prototipo de RF23

En la figura 123 se muestra el prototipo para la GUI de eliminar empleados desvinculados del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

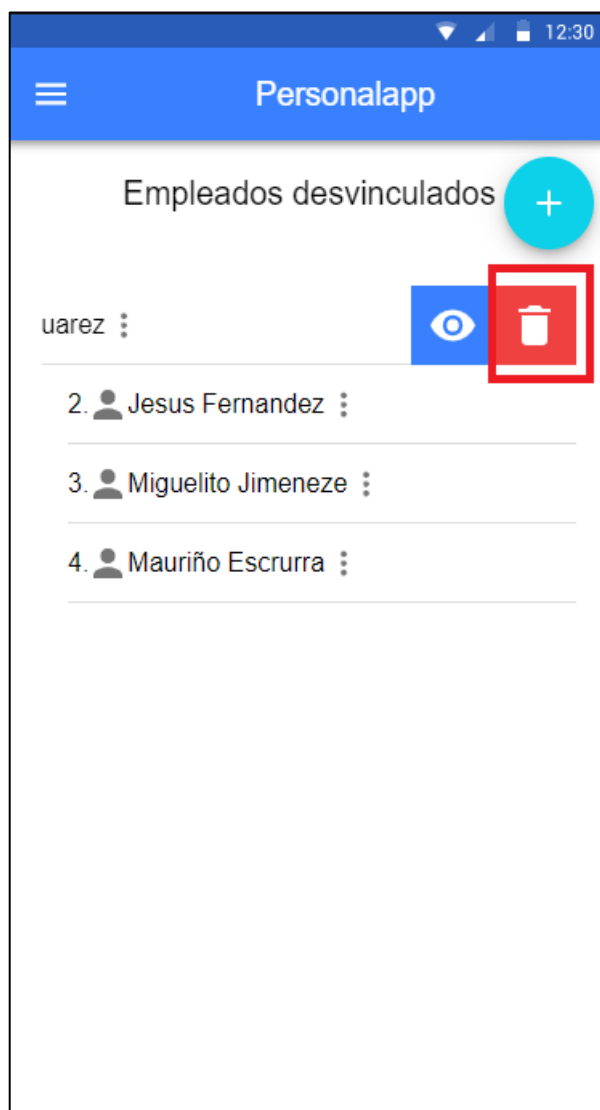


Figura 124. Interface de RF23

En la figura 124 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Función para eliminar desvinculación de empleados
public function dismissalDelete(Request $request) {
    $dismissal = Dismissals::where('employee_id', $this->request->input('key: id'))->delete();
    return $dismissal;
}
```

Figura 125. Código de RF23

En esta imagen 125 de la presente tesis se puede observar el código fuente del requerimiento funcional 23.

Requerimiento RF24 – Registrar salida

El requerimiento 24, tiene como objetivo que en la aplicación se debe poder guardar la hora de salida y se basa en el momento en que el empleado se retira de la empresa, dicho requerimiento solo debe hacerlo el usuario “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 24).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

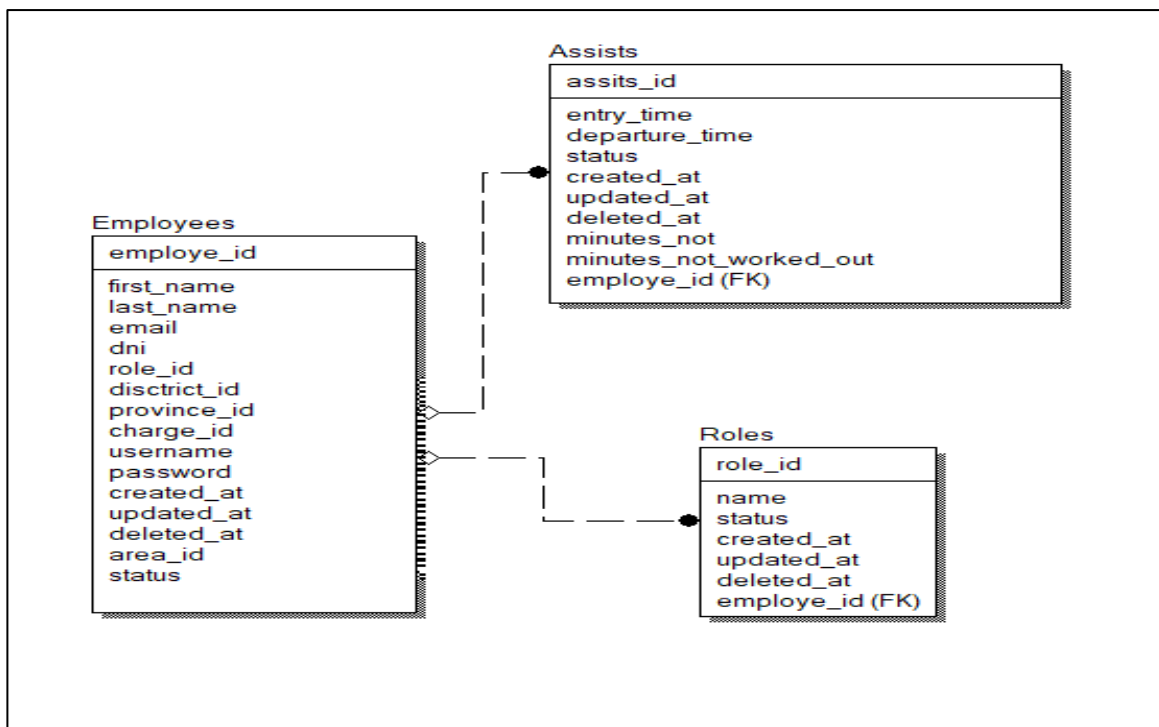


Figura 126. Diagrama lógico de la base de datos RF24

© Fuente: Elaboración propia

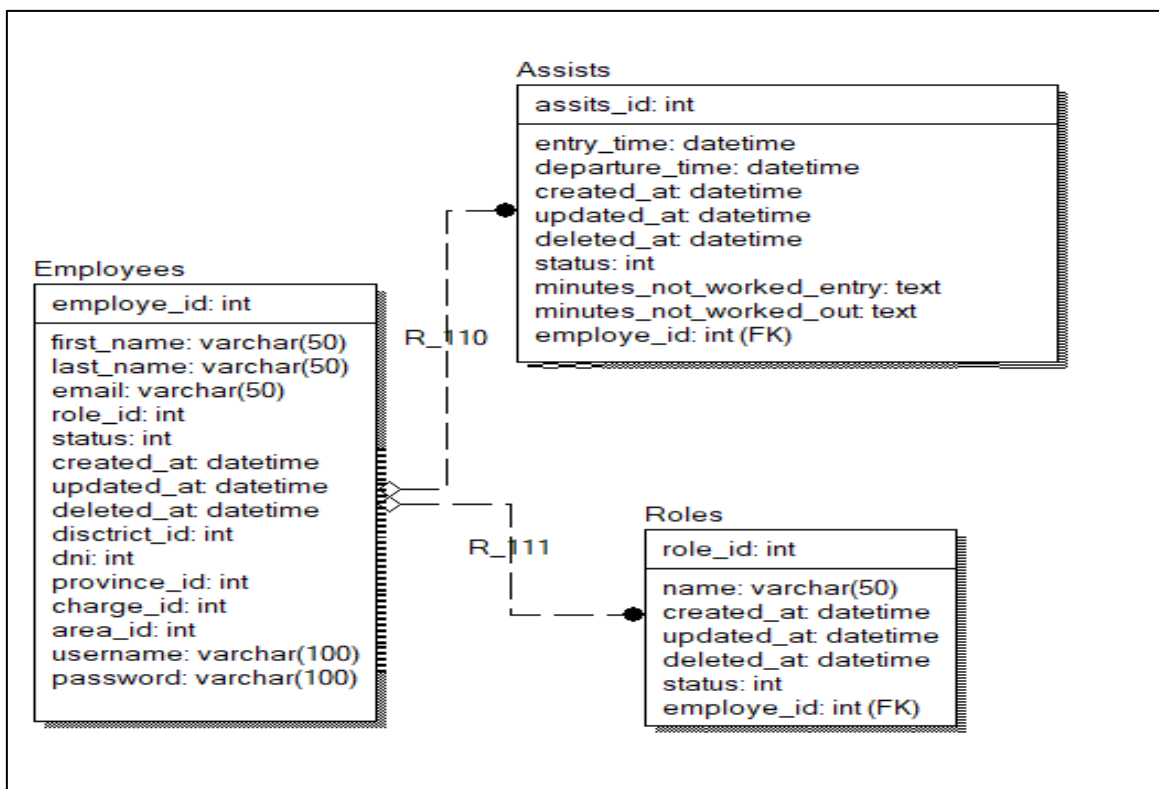


Figura 127. Diagrama físico de la base de datos RF24

Diseño

© Fuente: Elaboración propia



Figura 128. Prototipo de RF24

En la figura 128 se muestra el prototipo para la GUI de registrar salida que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Interface

© Fuente: Elaboración propia

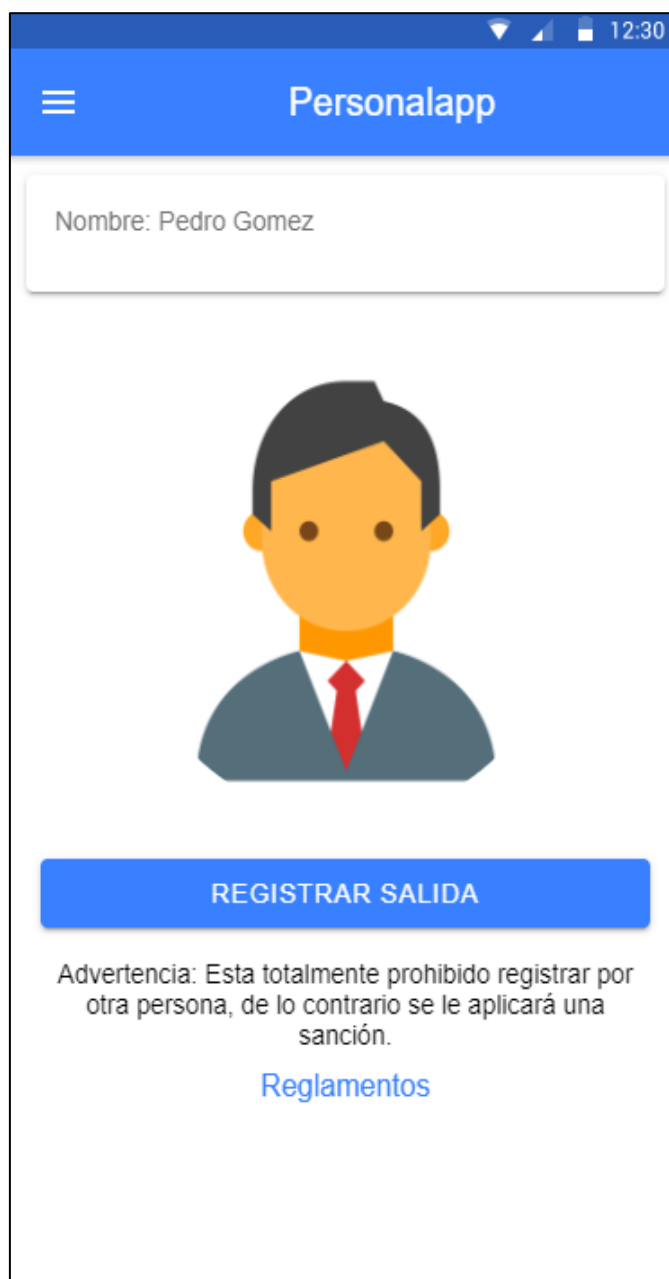


Figura 129. Interface de RF24

En la figura 129 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```

public function salida(Request $request){
    $mytime = Carbon::now( tz: 'America/Lima');
    $fecha_actual= $mytime->createFromFormat( format: 'Y-m-d H:i:s', $mytime)->format( format: 'Y-m-d');
    $hora_de_salida_establecida = Carbon::parse( time: $fecha_actual.'.'. '08:00:00', tz: 'America/Lima');
    if($mytime->toDateTimeString() > $fecha_actual.'.'. '08:00:00' ){
        $tiempo = 0;
        $h_extras = $hora_de_salida_establecida->diffInHours($mytime) . ':' . $mytime
            ->diff($hora_de_salida_establecida)->format( format: '%I:%S');
    }else{
        $tiempo = $hora_de_salida_establecida->diffInHours($mytime) . ':' . $mytime
            ->diff($hora_de_salida_establecida)->format( format: '%I:%S');
        $h_extras = 0;
    }
    $employee = Assist::where('employee_id',$this->request->input( key: 'id'))->whereDate('entry_time', $fecha_actual)->first();
    $employee->minutes_not_worked_out = $tiempo;
    $employee->departure_time = $mytime->toDateTimeString();
    $employee->employee_id = $this->request->input( key: 'id');
    $employee->h_extras = $h_extras;
    $employee->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}

```

Figura 130. Código de RF24

En esta imagen 130 de la presente tesis se puede observar el código fuente del requerimiento funcional 24.

Requerimiento RF25 – Visualizar estrategias de negocio

El requerimiento 25, tiene como objetivo que en la aplicación se debe poder visualizar las estrategias de negocio, dicho requerimiento solo puede hacerlo los usuarios “Administrador” y “Obrero”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

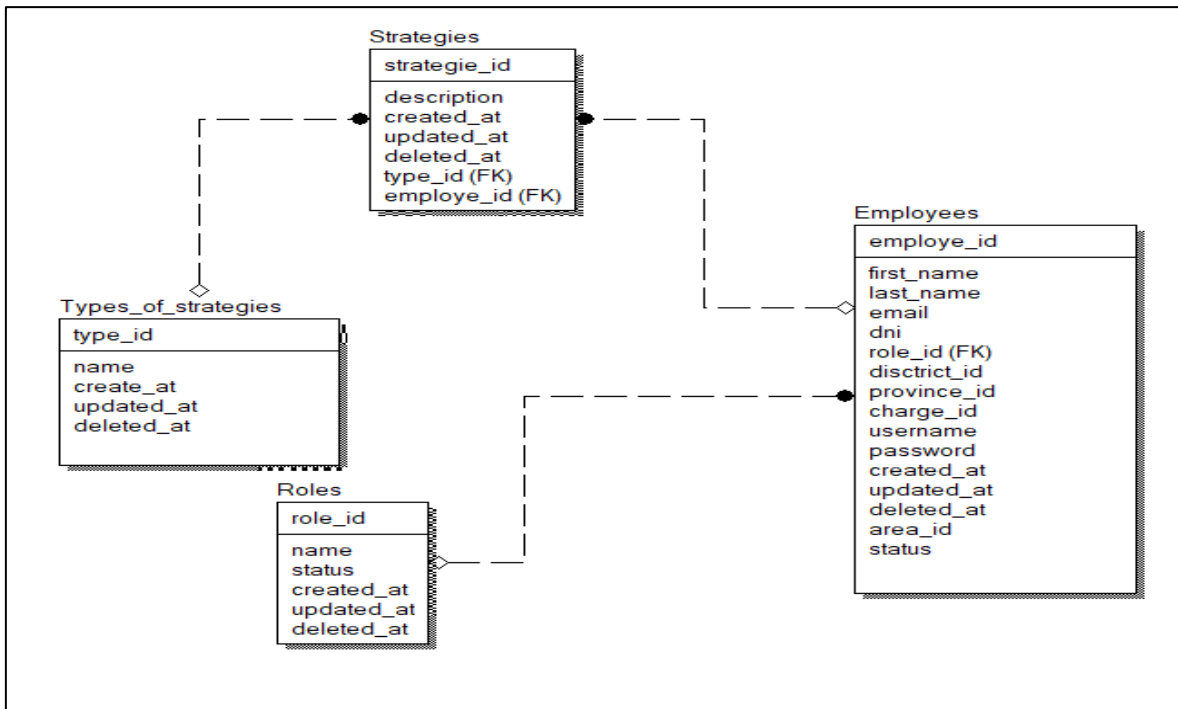


Figura 131. Diagrama lógico de la base de datos RF25

© Fuente: Elaboración propia

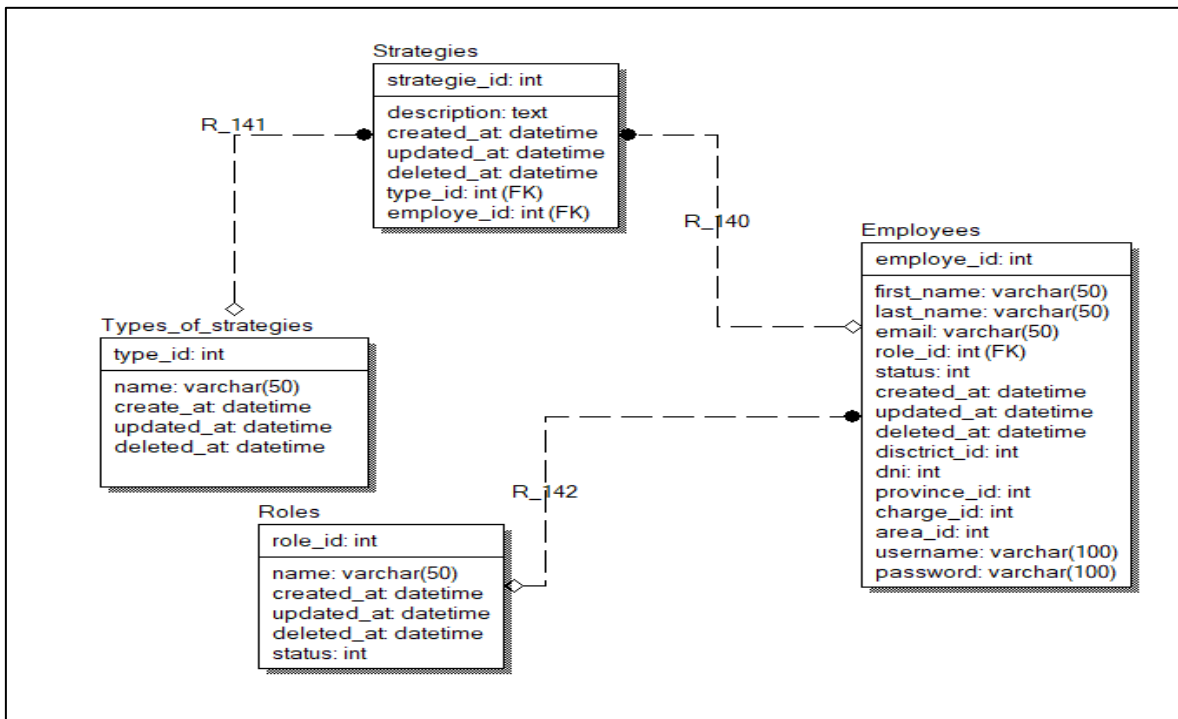


Figura 132. Diagrama físico de la base de datos RF25

Diseño

© Fuente: Elaboración propia

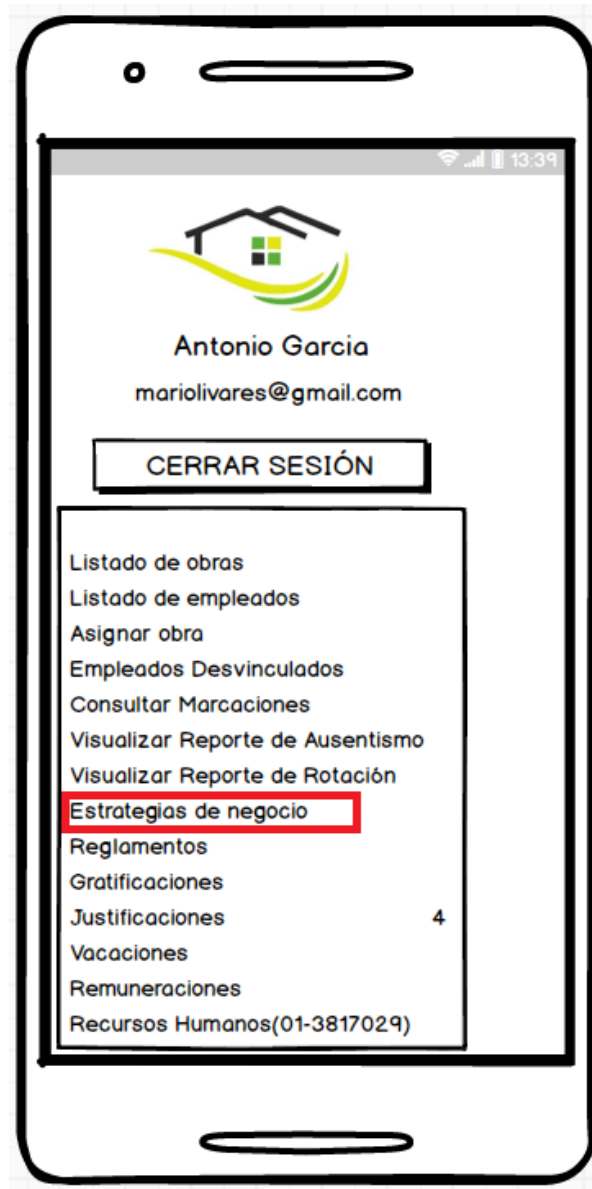


Figura 133. Prototipo de RF25

En la figura 133 se muestra el prototipo para la GUI de visualizar estrategias de negocio del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia .

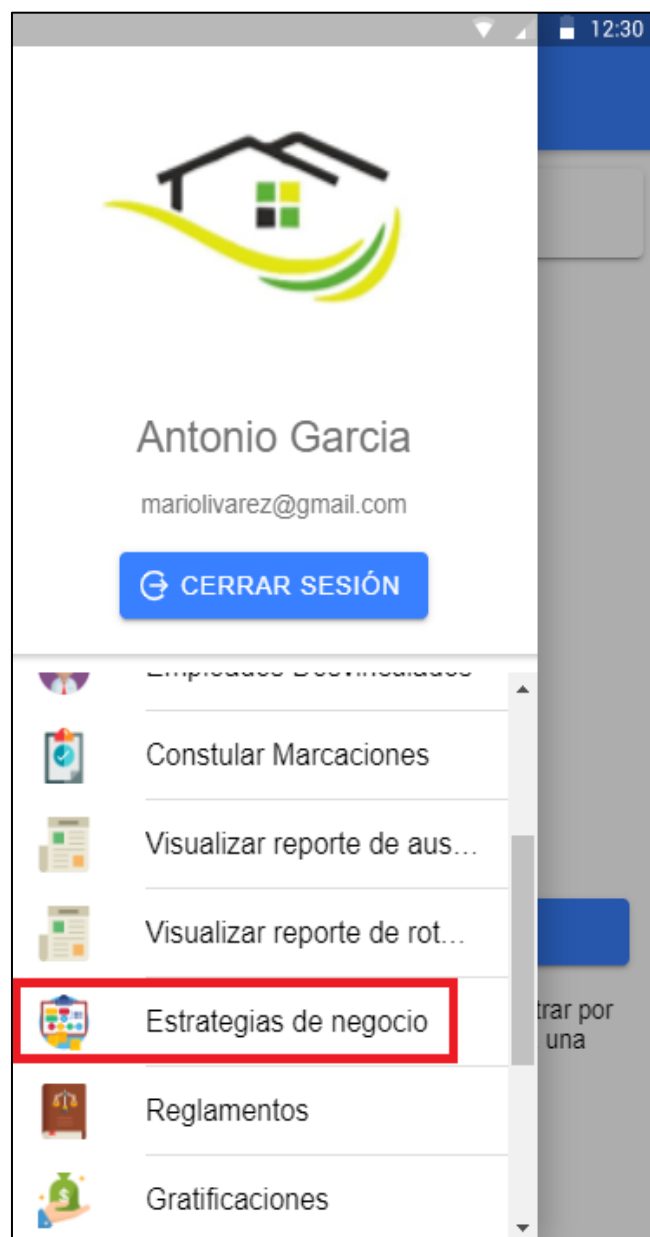


Figura 134. Interface de RF25

En la figura 134 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
if(res == 1){
  this.pages = [
    { title: 'Listado de obras', page: '/workcontroller', icon:'tractor' },
    { title: 'Listado de empleados', page: '/employeecontroller', icon:'employee' },
    { title: 'Asignar Obra', page: '/assignwork', icon:'traffic-cone' },
    { title: 'Empleados Desvinculados', page: '/dismissal', icon:'dismissal' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'approve' },
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'newspaper' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'newspaper' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'marketing-strategy' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'law-book' },
    { title: 'Gratificaciones', page: '/list', icon:'bank' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'notebook' },
    { title: 'Vacaciones', page: '/listvacations', icon:'beach' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'money-bag' }
  ]
}
else if(res == 2){
  this.pages = [
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'newspaper' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'newspaper' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'approve' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'law-book' },
    { title: 'Gratificaciones', page: '/list', icon:'bank' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'notebook' },
    { title: 'Vacaciones', page: '/listvacations', icon:'beach' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'money-bag' }
  ]
}
else if(res == 3){
  this.pages = [
    { title: 'Visualizar Obra Asignada', page: '/workassign', icon:'helmet' },
    { title: 'Consultar Marcaciones', page: '/marcations', icon:'approve' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'marketing-strategy' },
    { title: 'Gratificaciones', page: '/list', icon:'bank' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'notebook' },
    { title: 'Vacaciones', page: '/listvacations', icon:'beach' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'money-bag' }
  ]
}
```

Figura 135. Código de RF25

En esta imagen 135 de la presente tesis se puede observar el código fuente del requerimiento funcional 25.

Requerimiento RF26 – Seleccionar tipos de estrategias

El requerimiento 26, tiene como objetivo que en la aplicación se debe poder seleccionar los tipos de estrategias, tanto ausentismo como de rotación, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

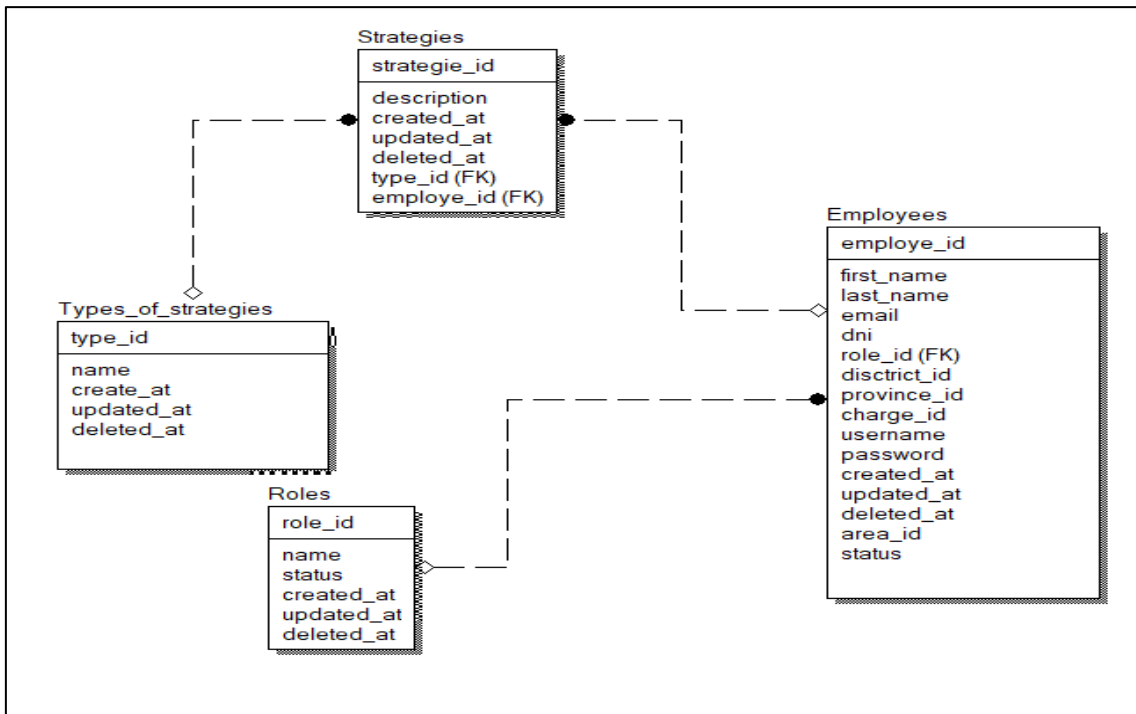


Figura 136. Diagrama lógico de la base de datos RF26

© Fuente: Elaboración propia

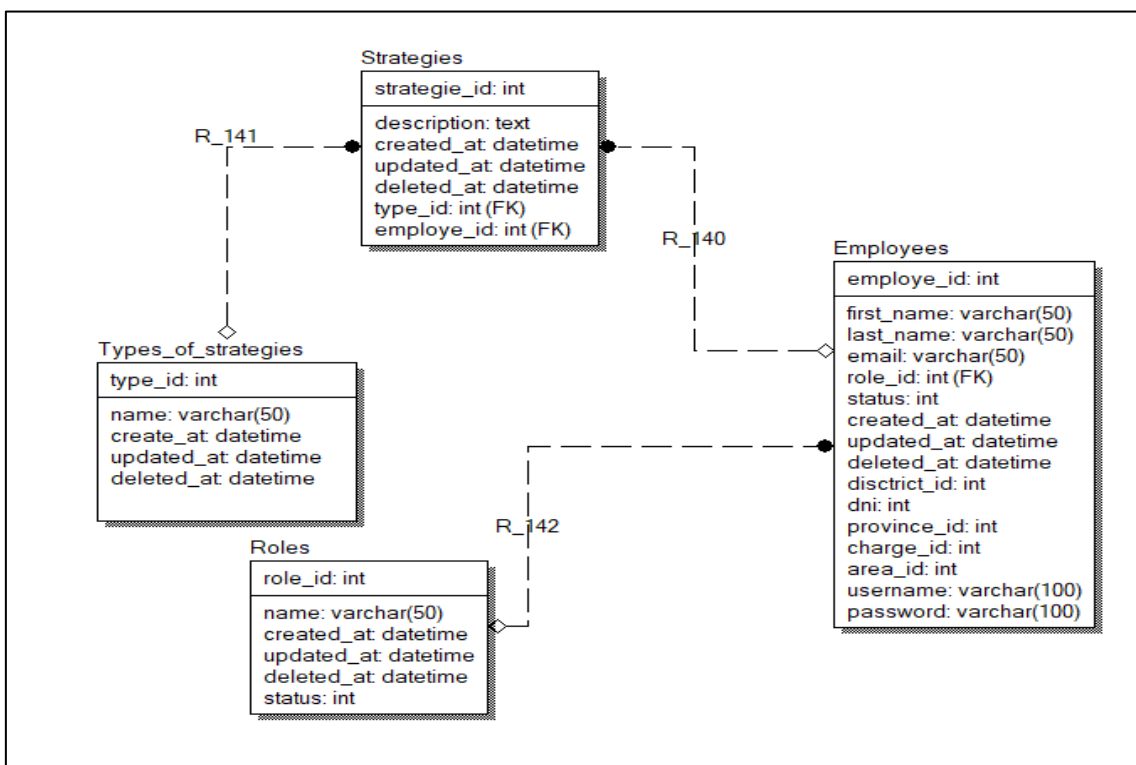


Figura 137. Diagrama físico de la base de datos RF26

Diseño

© Fuente: Elaboración propia

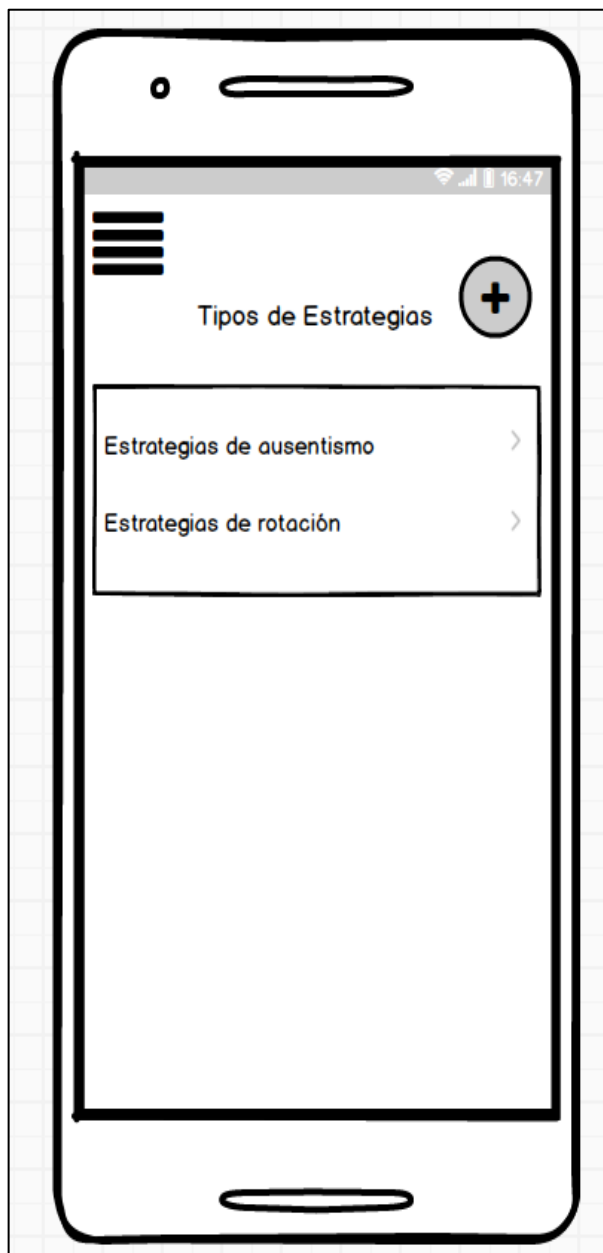


Figura 138. Prototipo de RF26

En la figura 138 se muestra el prototipo para la GUI de seleccionar tipos de estrategias del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

implementación

© Fuente: Elaboración propia

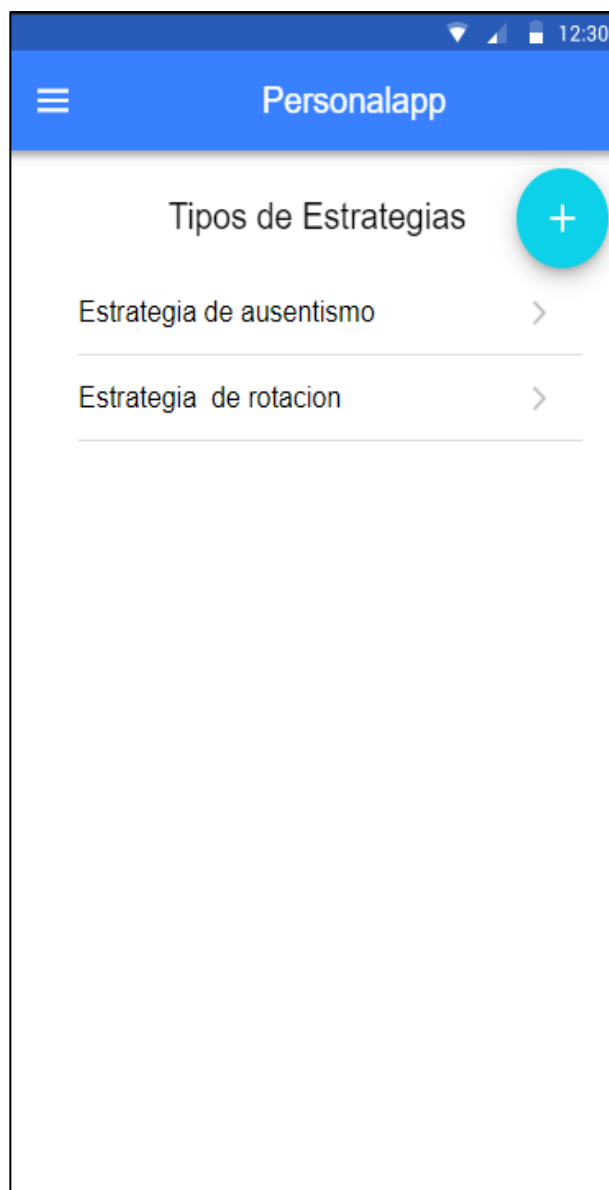


Figura 139. Interface de RF26

En la figura 139 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Listado de tipos de estrategias
public function loadTypesStrategies() {
    $types_strategies = types_of_strategies::all();
    return $types_strategies;
}
```

Figura 140 Código de RF26

En esta imagen 140 de la presente tesis se puede observar el código fuente del requerimiento funcional 26.

Requerimiento RF27 – Listar estrategias de ausentismo

El requerimiento 27, tiene como objetivo que en la aplicación se debe poder ver la lista de estrategias de ausentismo para que así el empleado pueda ser incentivado a no llegar tarde o faltar en la empresa, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 27).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

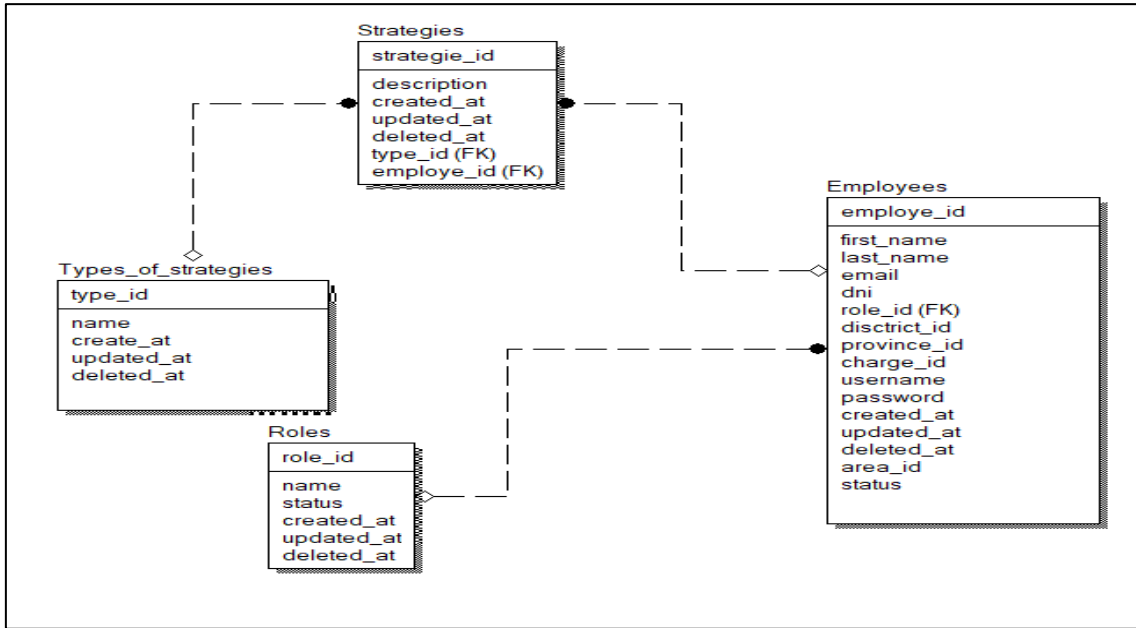


Figura 141. Diagrama lógico de la base de datos RF27

© Fuente: Elaboración propia

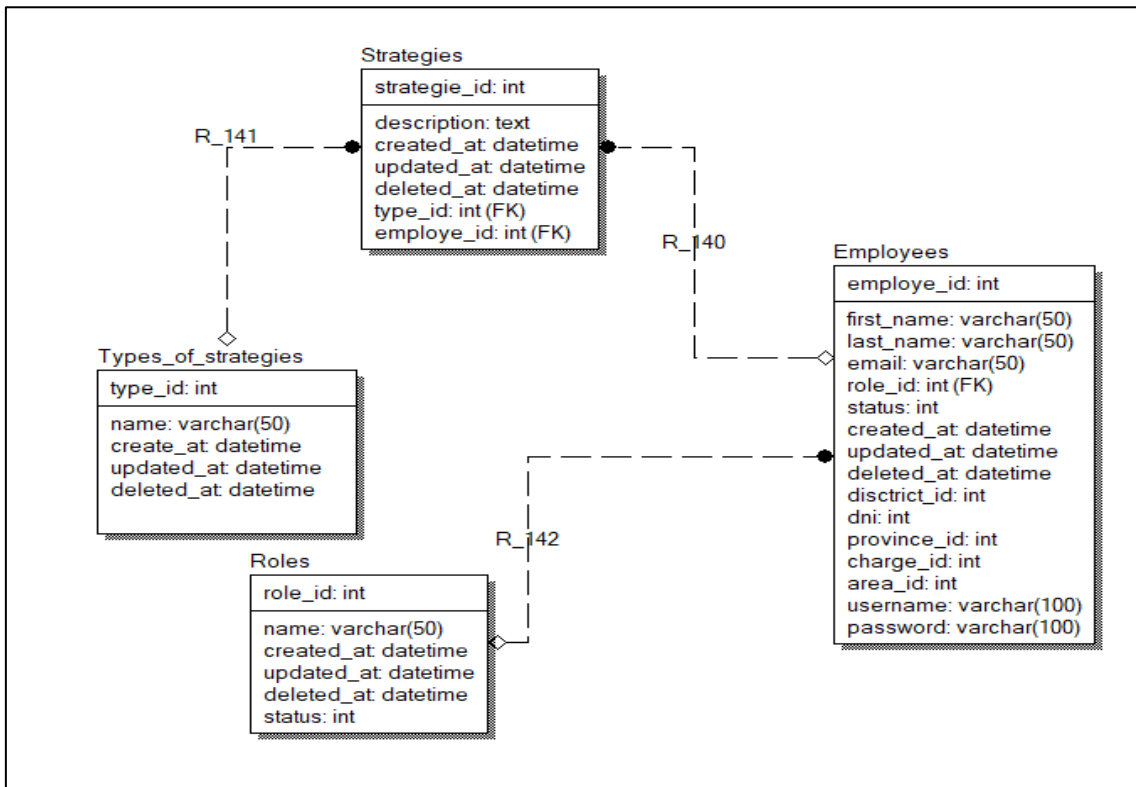


Figura 142. Diagrama físico de la base de datos RF27

Diseño

© Fuente: Elaboración propia

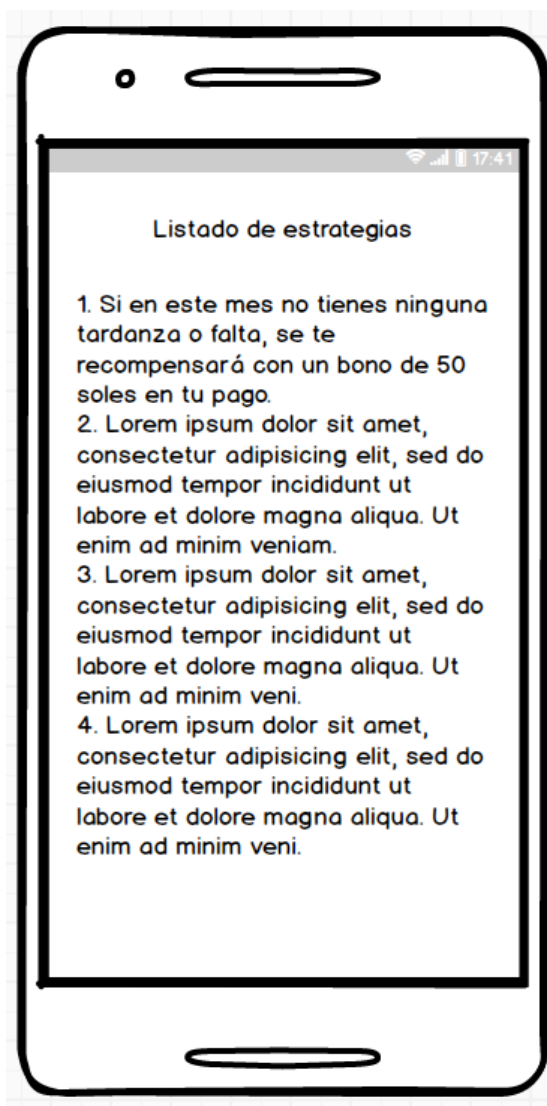


Figura 143. Prototipo de RF27

En la figura 143 se muestra el prototipo para la GUI de listar estrategias de ausentismo del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

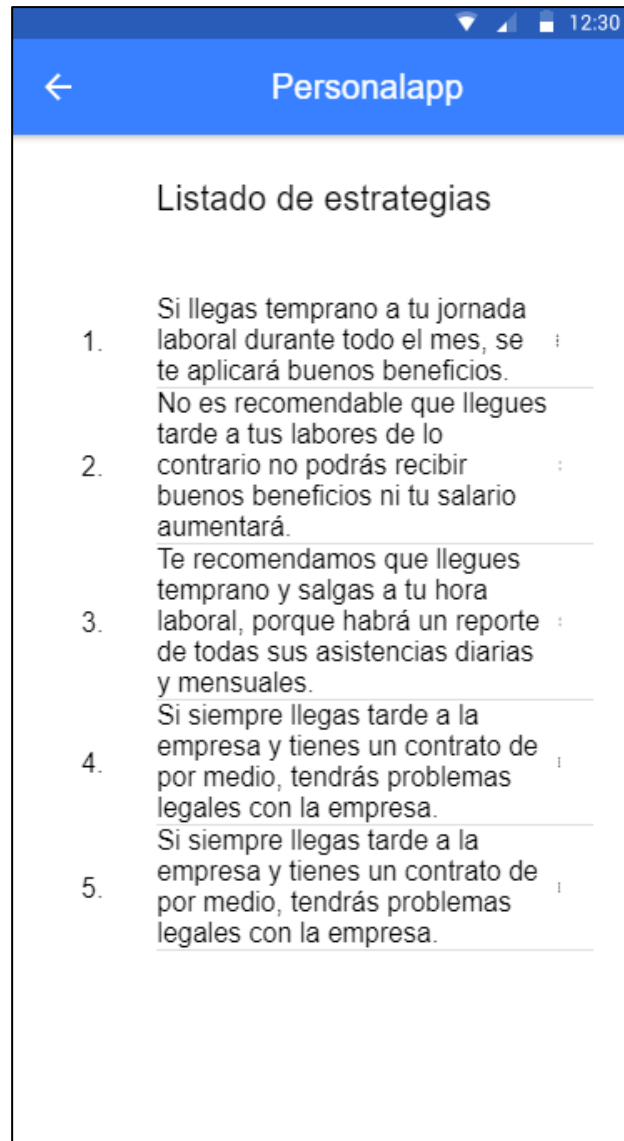


Figura 144. Interface de RF27

En la figura 144 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Lista de estrategias por tipo
public function listadobystrategies($id){
    $employees = Strategie::where('type_id',$id)->get();
    return $employees;
}
```

Figura 145. Código de RF27

En esta imagen 145 de la presente tesis se puede observar el código fuente del requerimiento funcional 27.

Requerimiento RF28 – Listar estrategias de rotación

El requerimiento 28, tiene como objetivo que en la aplicación se debe poder ver la lista de estrategias de rotación para que así el empleado pueda ser incentivado a no retirarse de la empresa, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 28).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

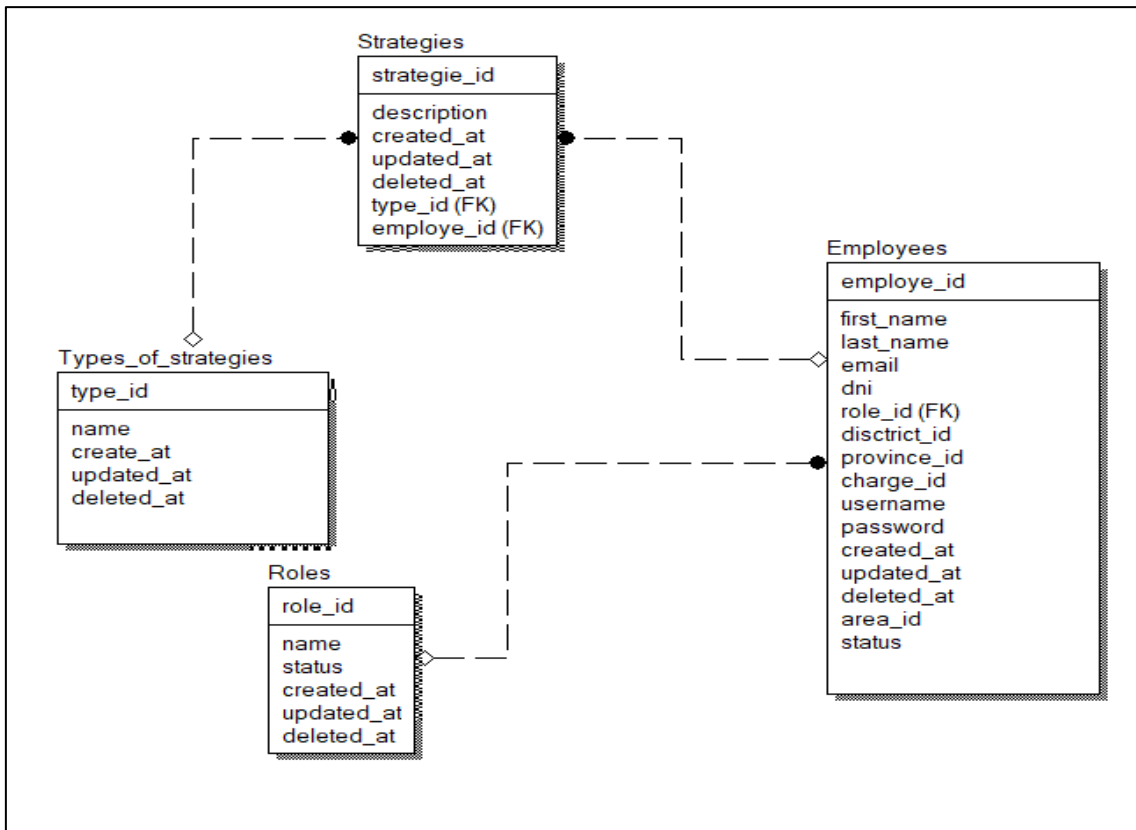


Figura 146. Diagrama lógico de la base de datos RF28

© Fuente: Elaboración propia

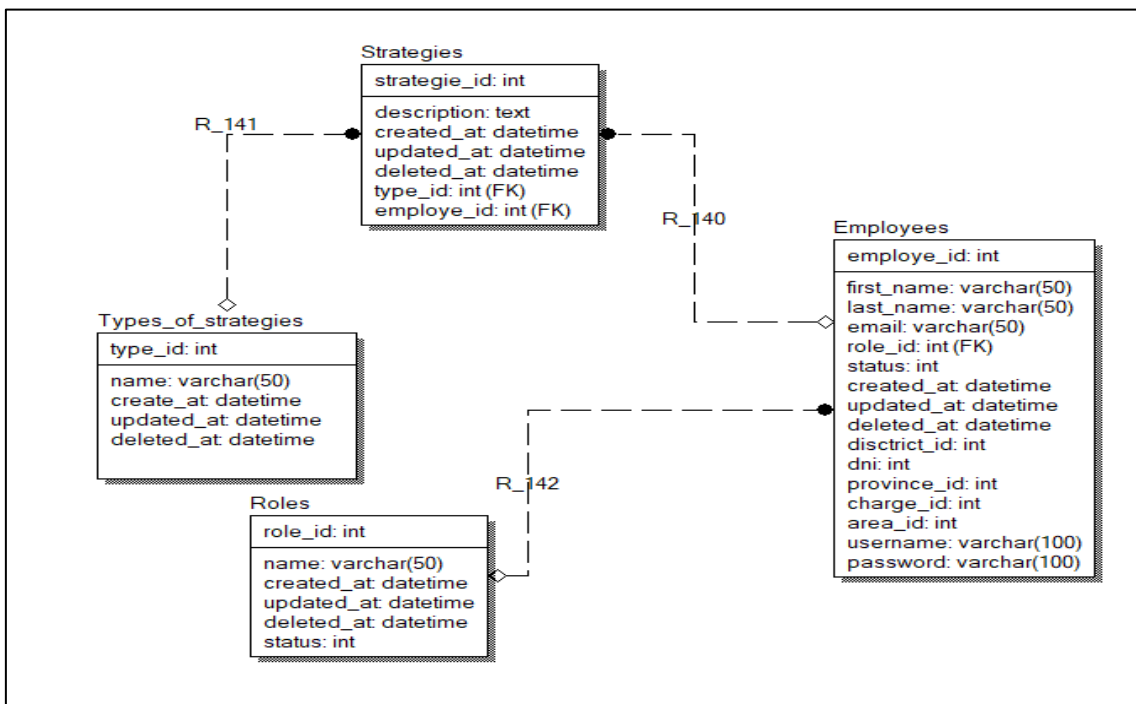


Figura 147. Diagrama físico de la base de datos RF28

Diseño

© Fuente: Elaboración propia

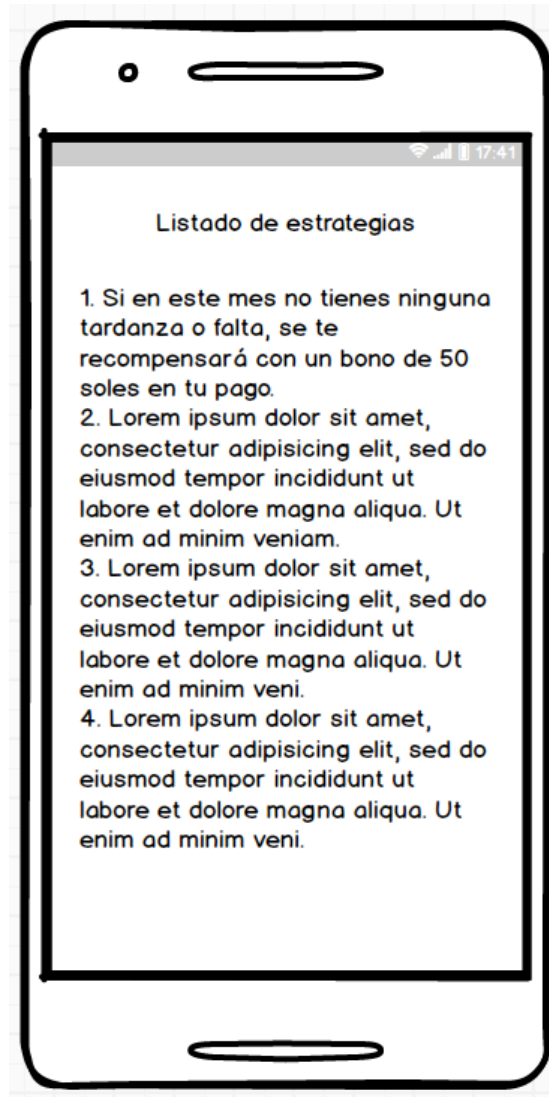


Figura 148. Prototipo de RF28

En la figura 148 se muestra el prototipo para la GUI de listar estrategias de rotación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

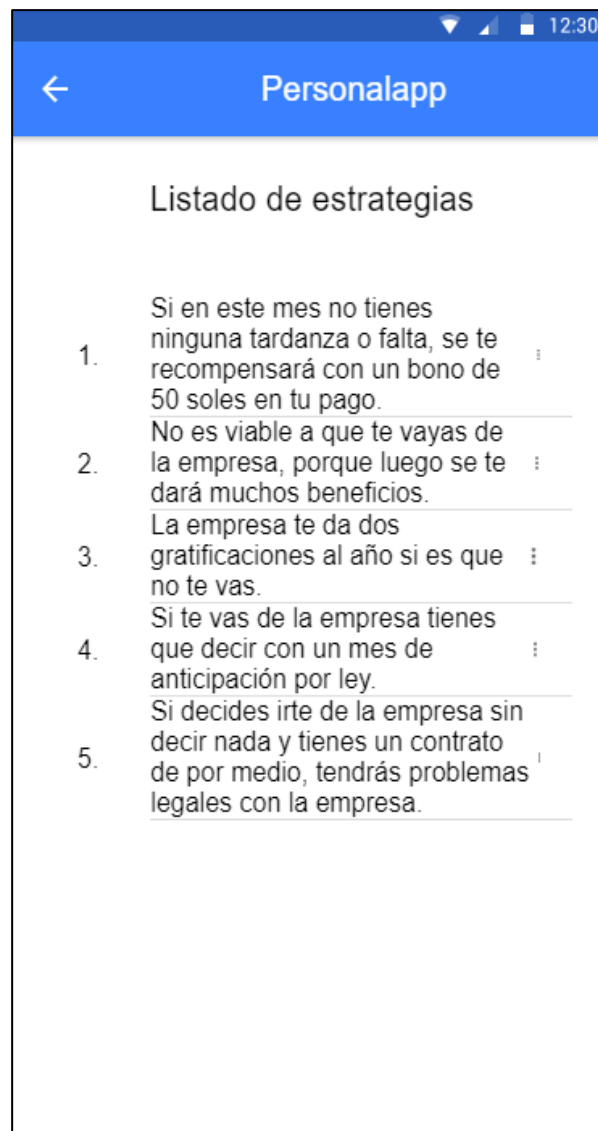


Figura 149. Interface de RF28

En la figura 149 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
// Lista de estrategias por tipo
public function listadobystrategies($id){
    $employees = Strategie::where('type_id',$id)->get();
    return $employees;
}
```

Figura 150. Código de RF28

En esta imagen 150 de la presente tesis se puede observar el código fuente del requerimiento funcional 28.

Requerimiento RF29 – Crear estrategias

El requerimiento 29, tiene como objetivo que en la aplicación se debe poder crear estrategias de negocio, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 29).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

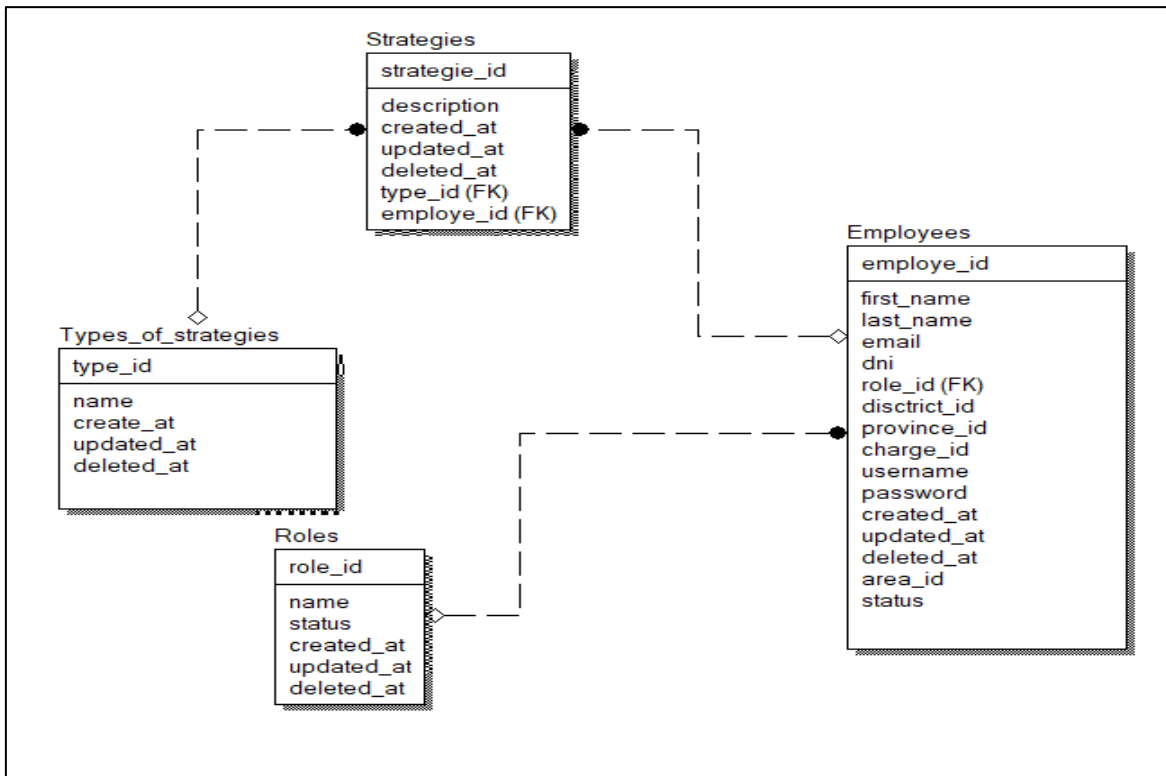


Figura 151. Diagrama lógico de la base de datos RF29

© Fuente: Elaboración propia

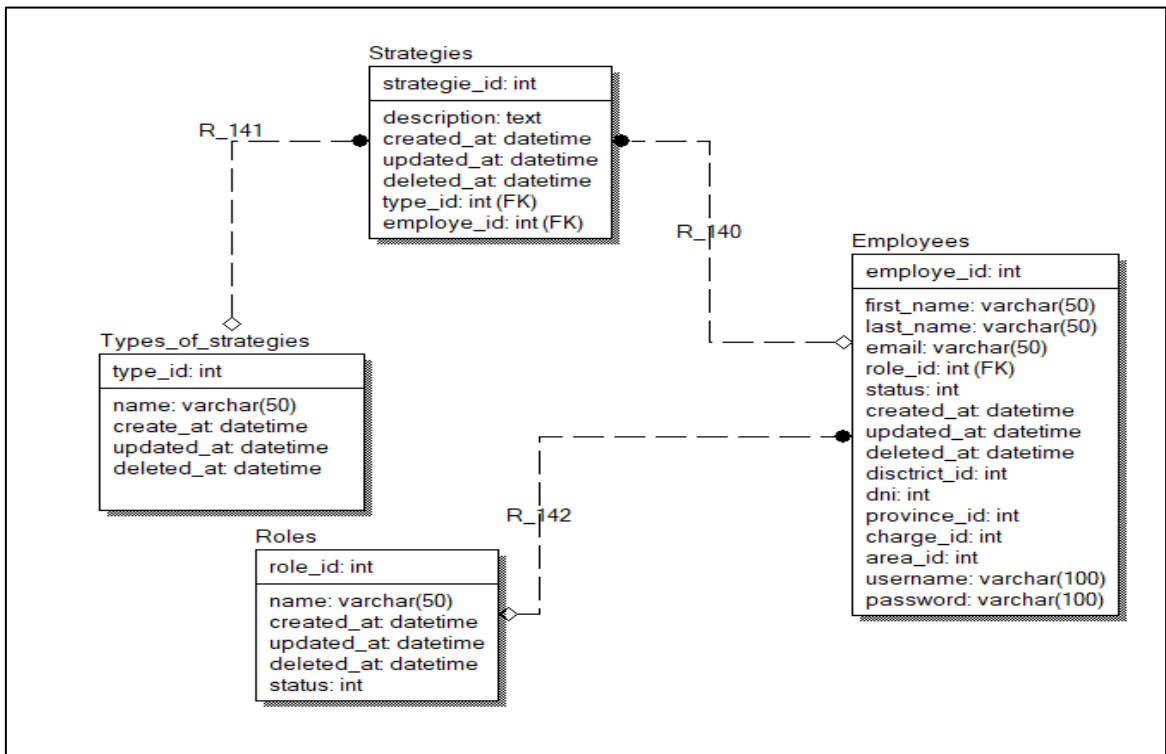


Figura 152. Diagrama físico de la base de datos RF29

Diseño

© Fuente: Elaboración propia

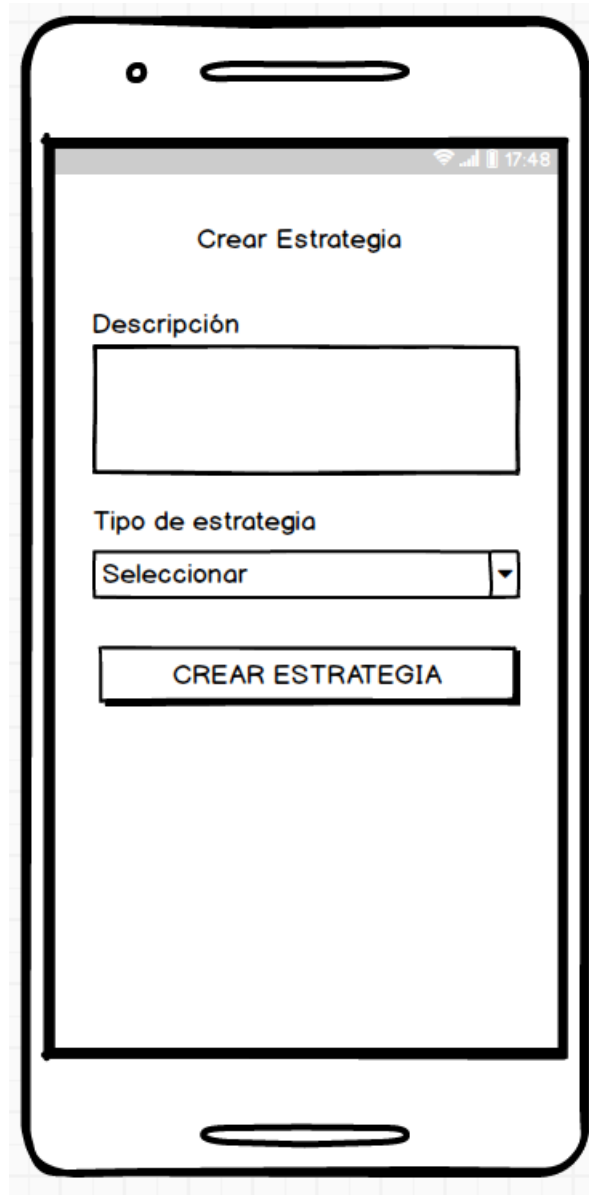
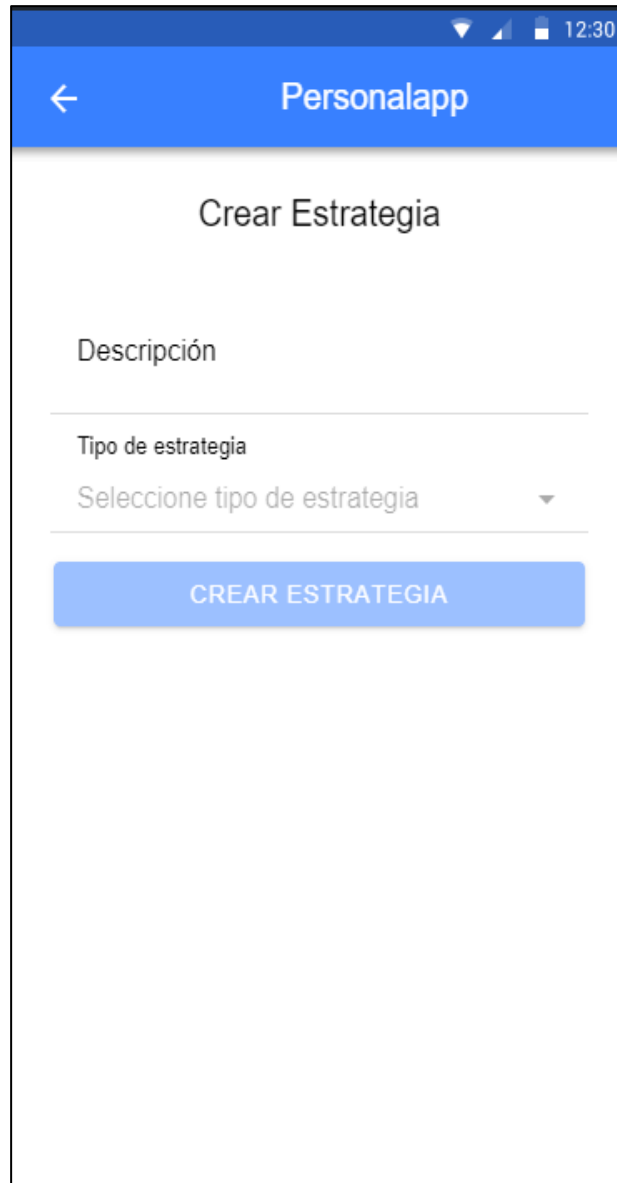


Figura 153. Prototipo de RF29

En la figura 153 se muestra el prototipo para la GUI de crear estrategias del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. The main heading is 'Crear Estrategia'. Below this, there is a text input field labeled 'Descripción'. Underneath is a dropdown menu labeled 'Tipo de estrategia' with the placeholder text 'Seleccione tipo de estrategia'. At the bottom of the form is a blue button labeled 'CREAR ESTRATEGIA'. The top of the screen shows a blue navigation bar with a back arrow and the app name 'Personalapp'. The status bar at the very top indicates the time as 12:30.

Figura 154. Interface de RF29

En la figura 154 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function save(Request $request) {
    $strategie = new Strategie();
    $strategie->description = $request->description;
    $strategie->type_id = $request->type_id;
    $strategie->save();
    return response()->json([
        'msg' => 'Estrategia registrado correctamente',
        'type_id' => $request->type_id,
        'exito' => '200'
    ]);
}
```

Figura 155. Código de RF29

En esta imagen 155 de la presente tesis se puede observar el código fuente del requerimiento funcional 29.

Requerimiento RF30 – Actualizar estrategias de ausentismo

El requerimiento 30, tiene como objetivo que en la aplicación se debe poder actualizar las estrategias de ausentismo, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 30).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

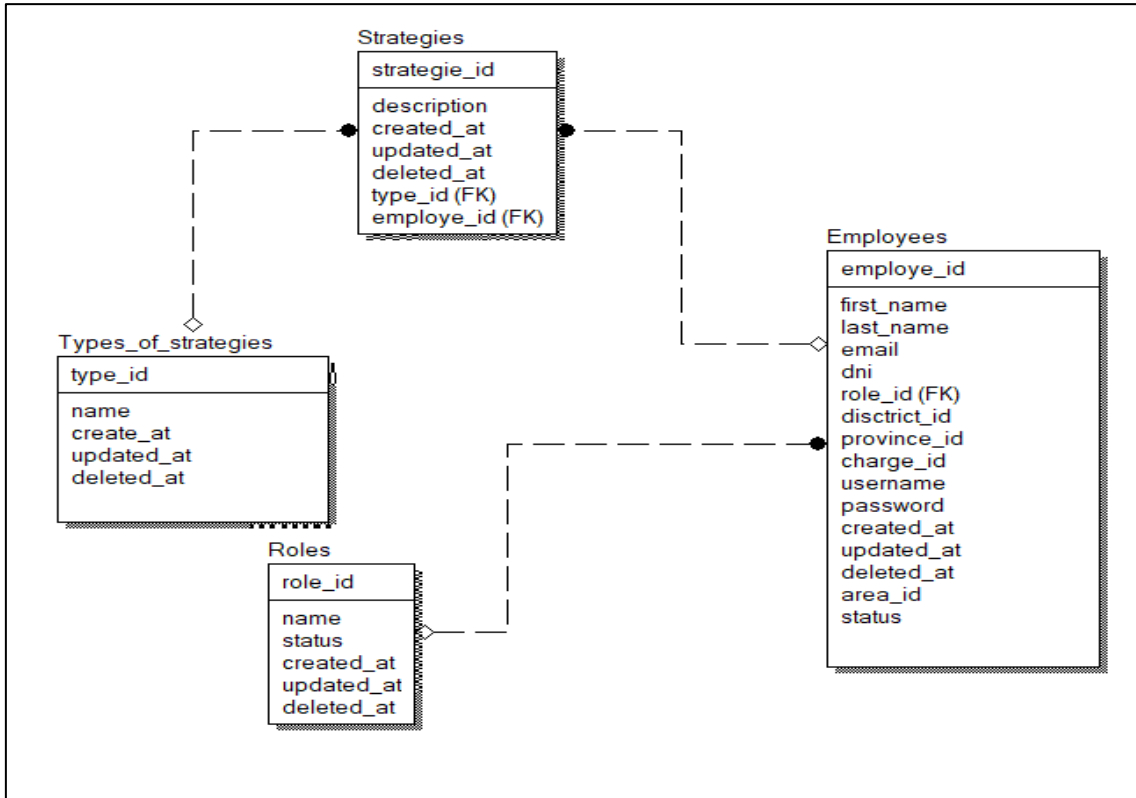


Figura 156. Diagrama lógico de la base de datos RF30

© Fuente: Elaboración propia

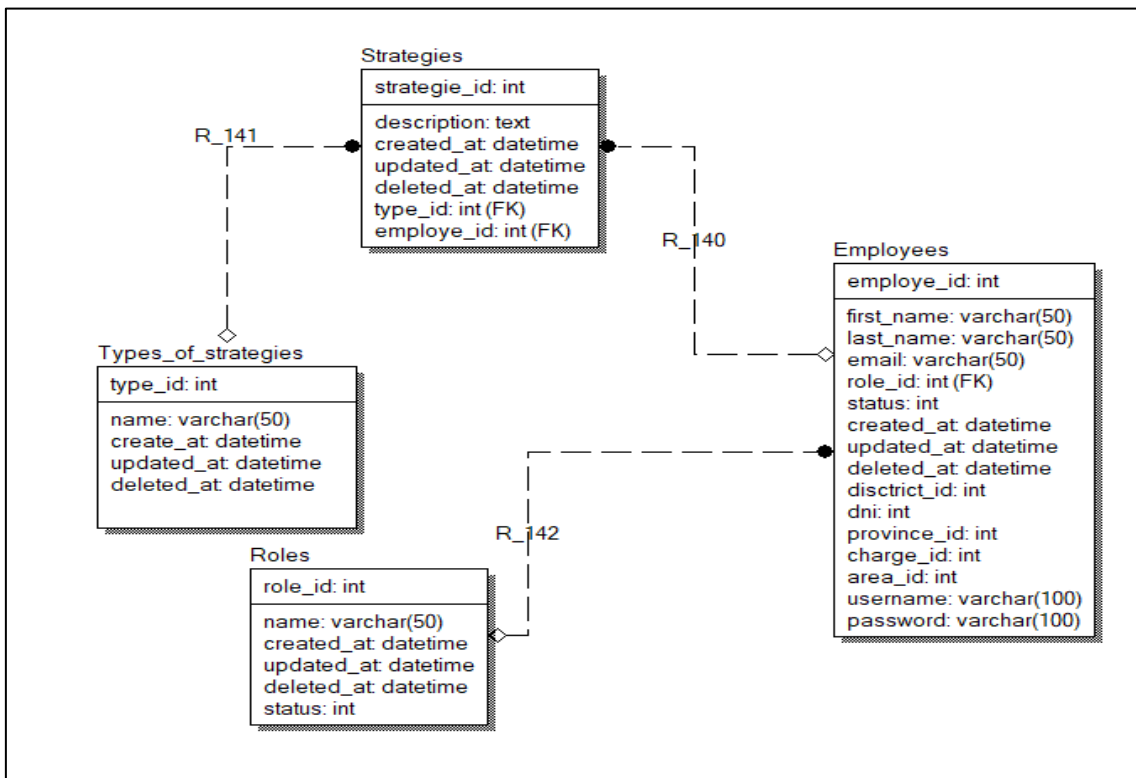


Figura 157. Diagrama físico de la base de datos RF30

Diseño

© Fuente: Elaboración propia

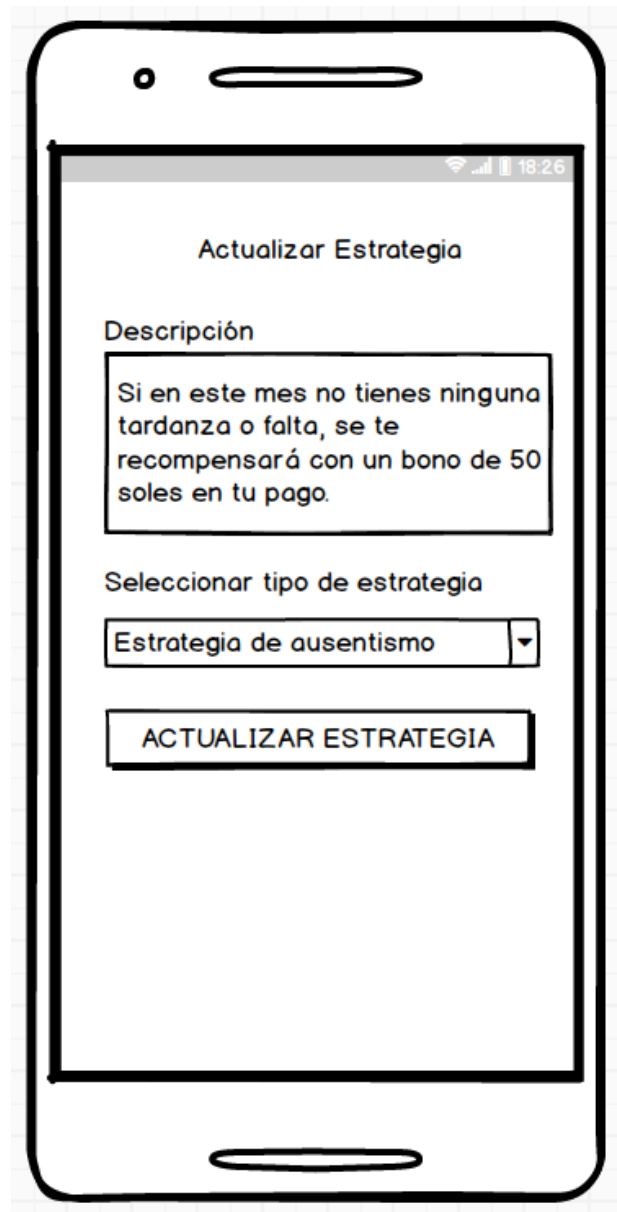


Figura 158. Prototipo de RF30

En la figura 158 se muestra el prototipo para la GUI de actualizar estrategias de ausentismo del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

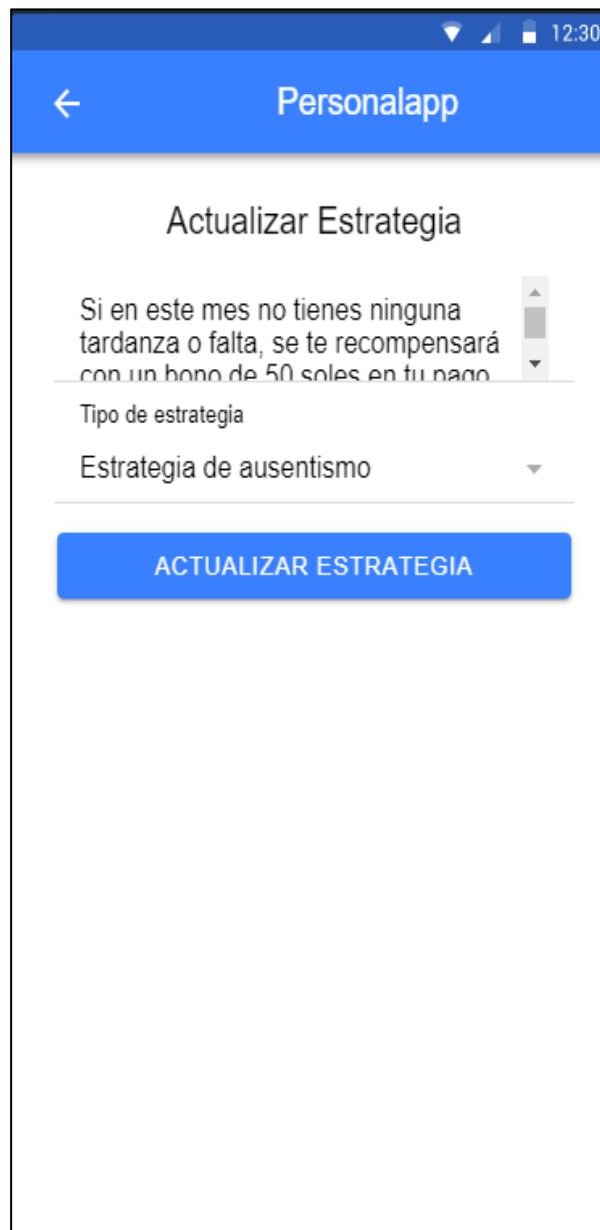


Figura 159. Interface de RF30

En la figura 159 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateStrategie(Request $request,$id) {  
    $strategie= Strategie::find($id);  
    $strategie->description = $request->description;  
    $strategie->type_id = $request->type_id;  
    $strategie->save();  
    return $strategie->type_id;  
}
```

Figura 160. Código de RF30

En esta imagen 160 de la presente tesis se puede observar el código fuente del requerimiento funcional 30.

Requerimiento RF31 – Eliminar estrategias de ausentismo

El requerimiento 31, tiene como objetivo que en la aplicación se debe poder eliminar las estrategias de ausentismo, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 31).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer

© Fuente: Elaboración propia

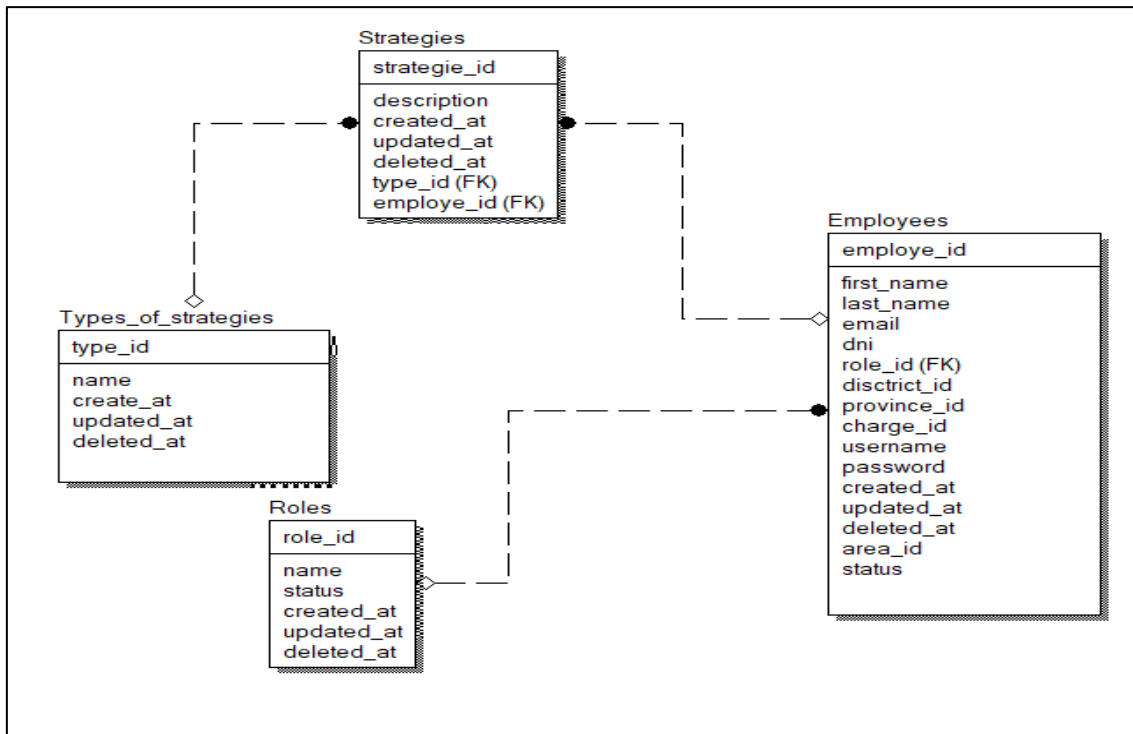


Figura 161. Diagrama lógico de la base de datos RF31

© Fuente: Elaboración propia

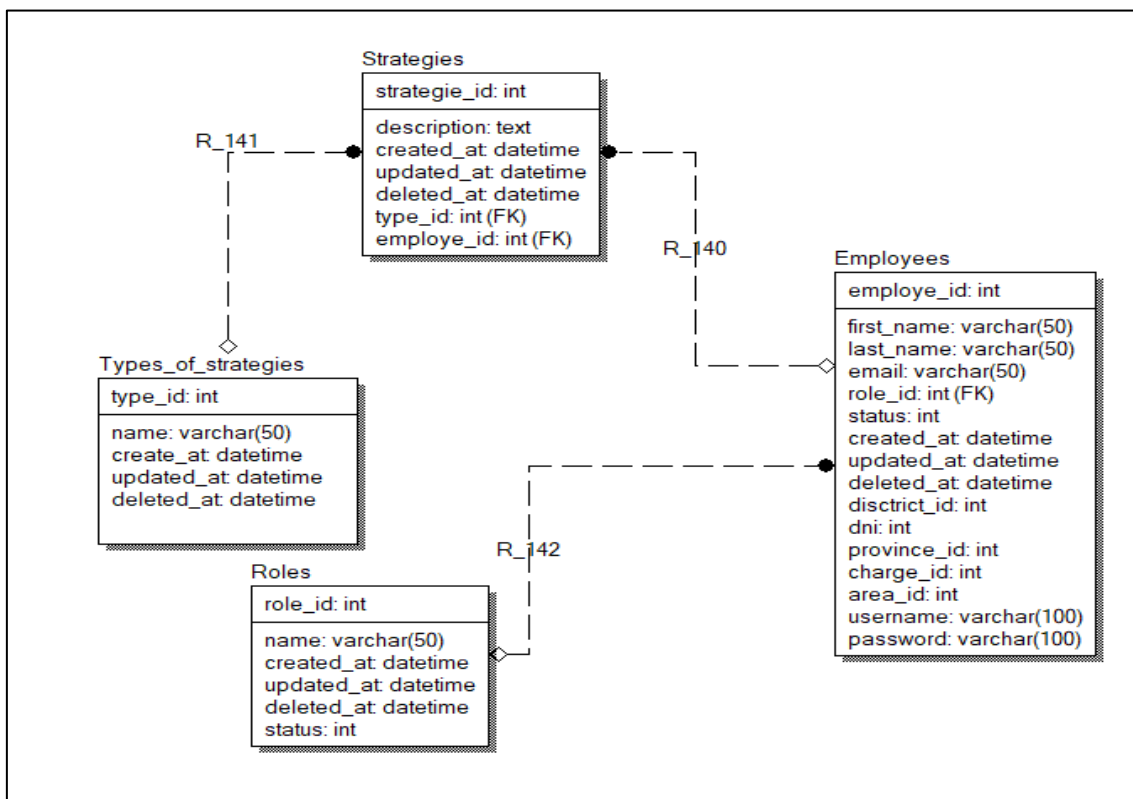


Figura 162. Diagrama físico de la base de datos RF31

Diseño

© Fuente: Elaboración propia

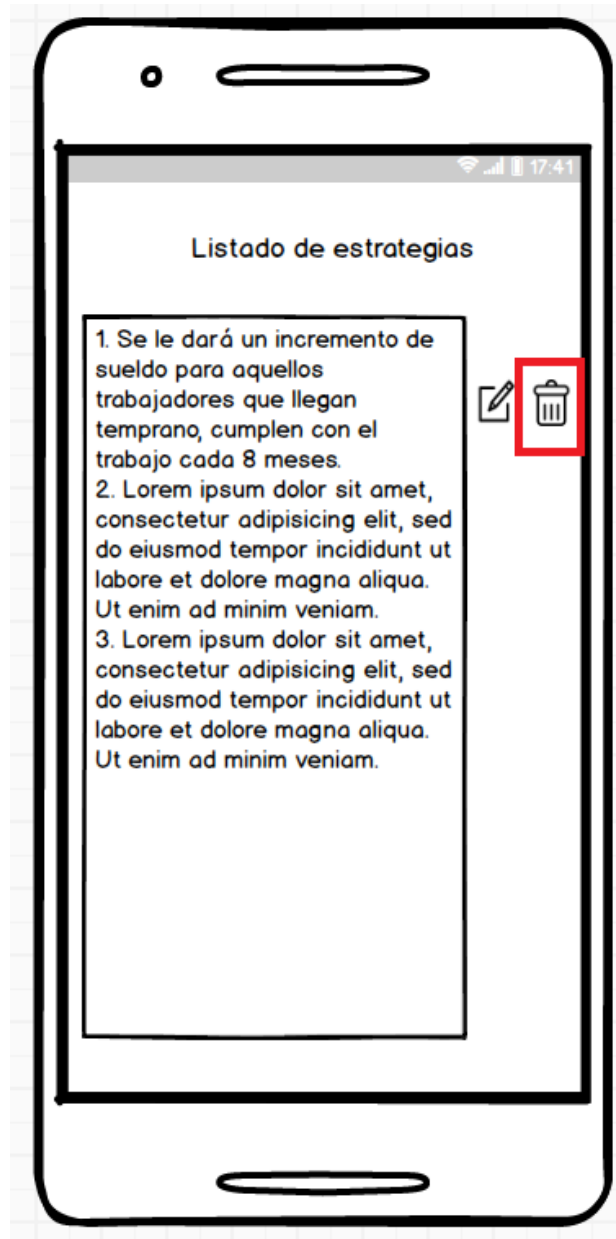


Figura 163. Prototipo de RF31

En la figura 163 se muestra el prototipo para la GUI de eliminar estrategias de ausentismo del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

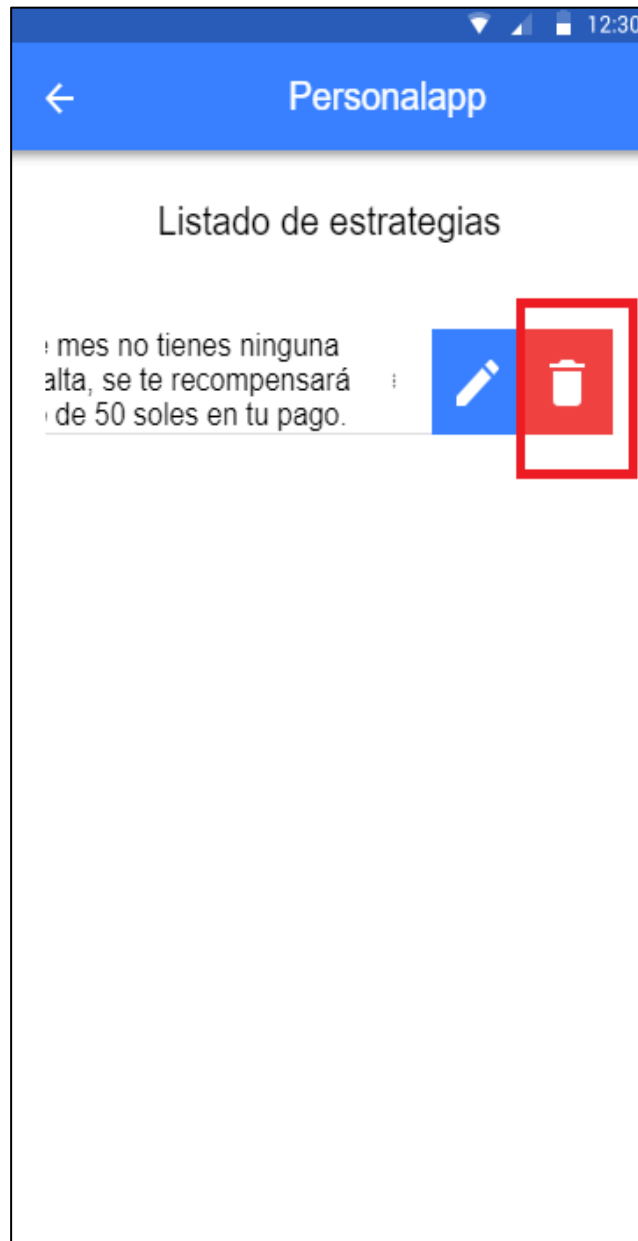


Figura 164. Interface de RF31

En la figura 164 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteStrategie(Request $request){
    $strategie = Strategie::find($this->request->input('key: id'));
    $strategie->delete();
    return response()->json([
        'msg' => 'exito',
        'strategie_type_id' => $strategie->type_id
    ], status: 200);
}
```

Figura 165. Código de RF31

En esta imagen 165 de la presente tesis se puede observar el código fuente del requerimiento funcional 31.

Requerimiento RF32 – Actualizar estrategias de rotación

El requerimiento 32, tiene como objetivo que en la aplicación se debe poder actualizar las estrategias de rotación por si se desea modificar las descripciones, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 32).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

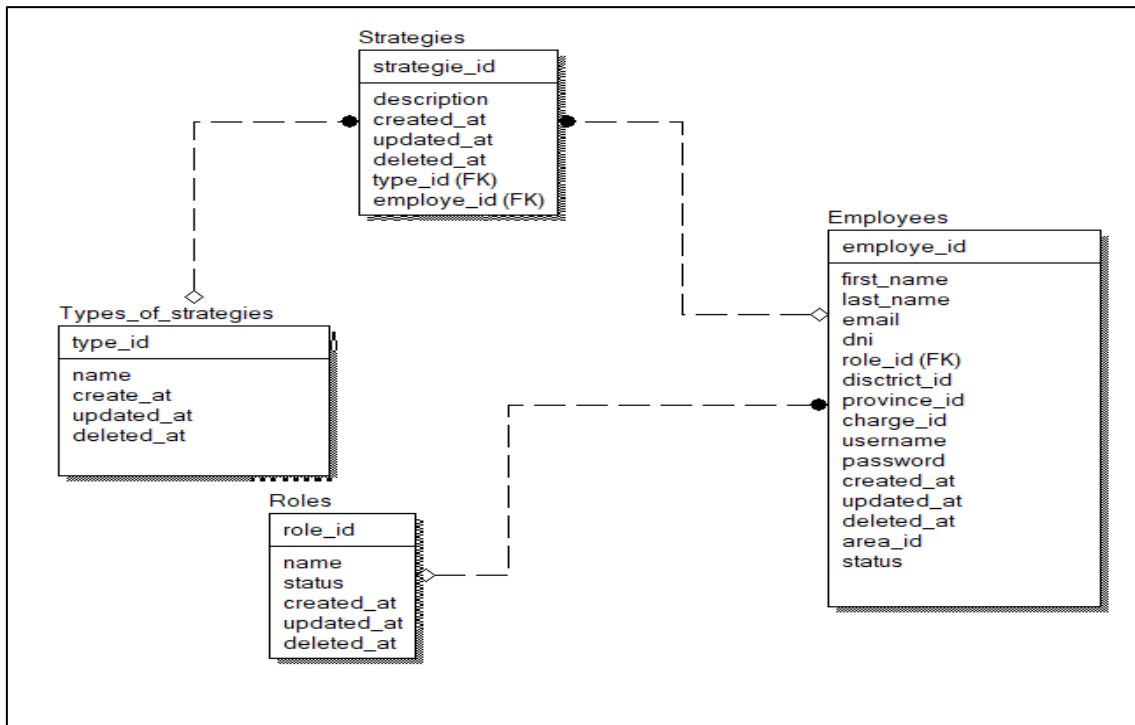


Figura 166. Diagrama lógico de la base de datos RF32

© Fuente: Elaboración propia

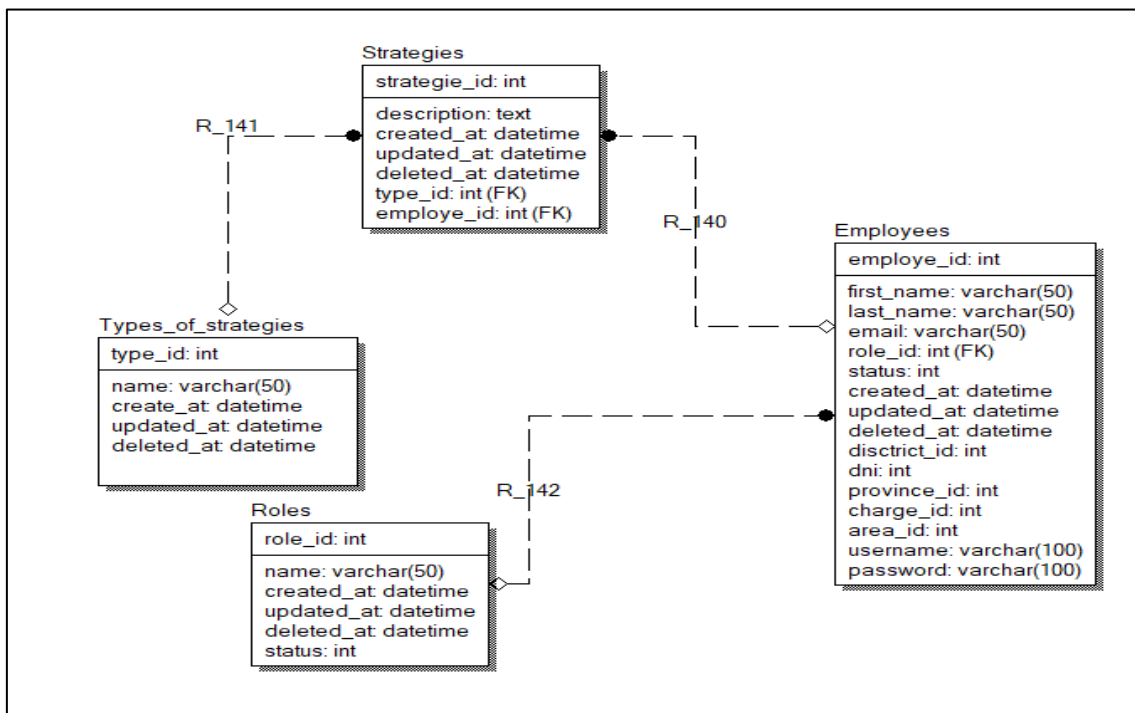


Figura 167. Diagrama físico de la base de datos RF32

Diseño

© Fuente: Elaboración propia

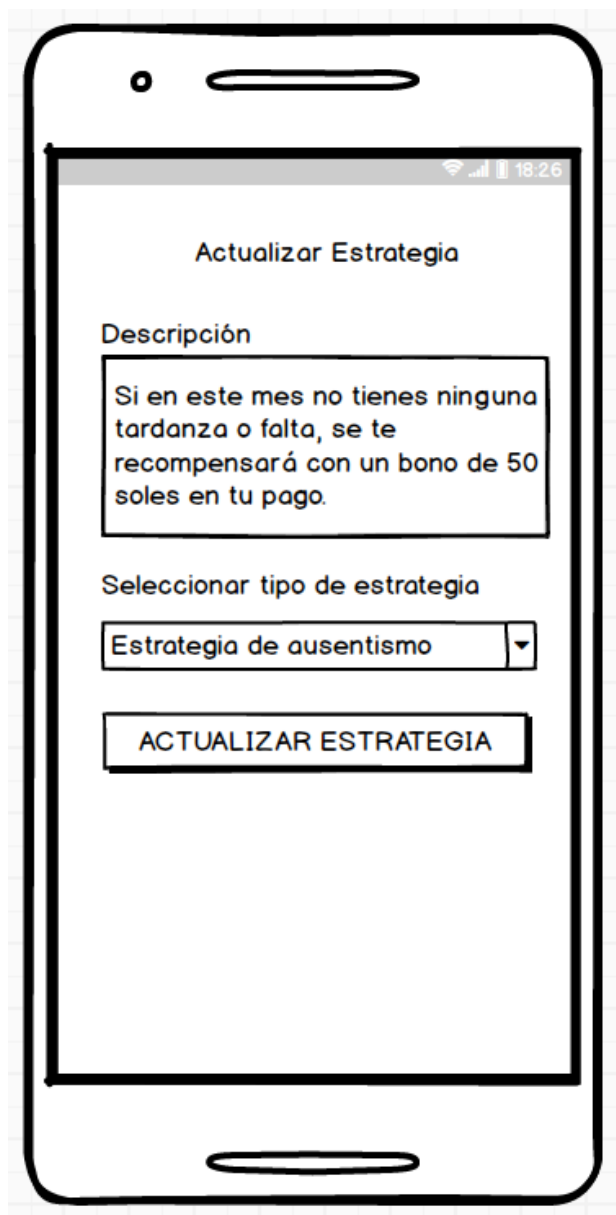


Figura 168. Prototipo de RF32

En la figura 168 se muestra el prototipo para la GUI de actualizar estrategias de rotación del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

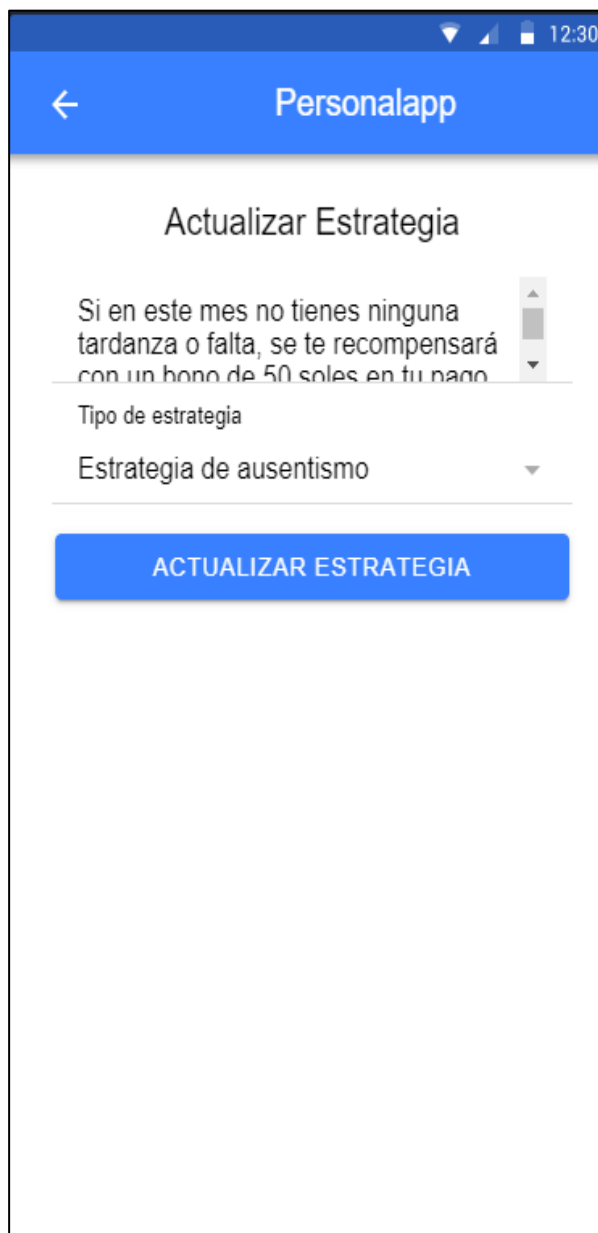


Figura 169. Interface de RF32

En la figura 169 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateStrategie(Request $request,$id){
    $strategie= Strategie::find($id);
    $strategie->description = $request->description;
    $strategie->type_id = $request->type_id;
    $strategie->save();
    return $strategie->type_id;
}
```

Figura 170. Código de RF32

En esta imagen 170 de la presente tesis se puede observar el código fuente del requerimiento funcional 32.

Requerimiento RF33 – Eliminar estrategias de rotación

El requerimiento 33, tiene como objetivo que en la aplicación se debe poder eliminar las estrategias de rotación, dicho requerimiento solo debe hacerlo el usuario “Administrador”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 33).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

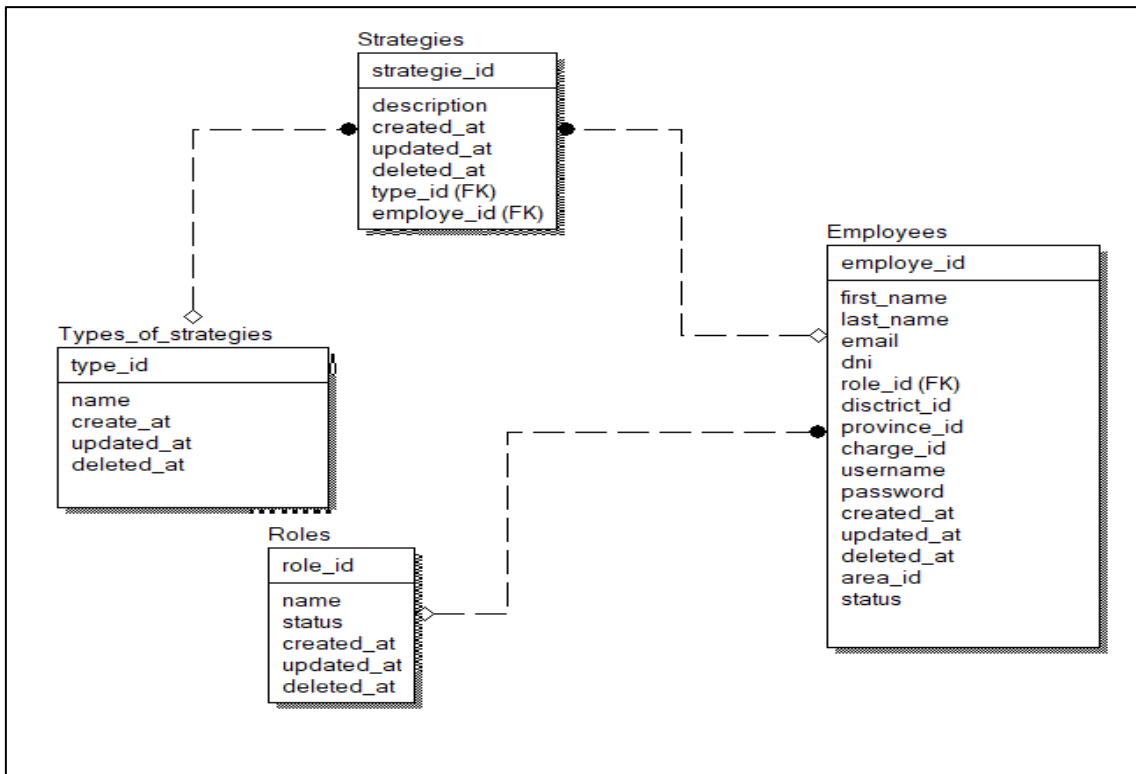


Figura 171. Diagrama lógico de la base de datos RF33

© Fuente: Elaboración propia

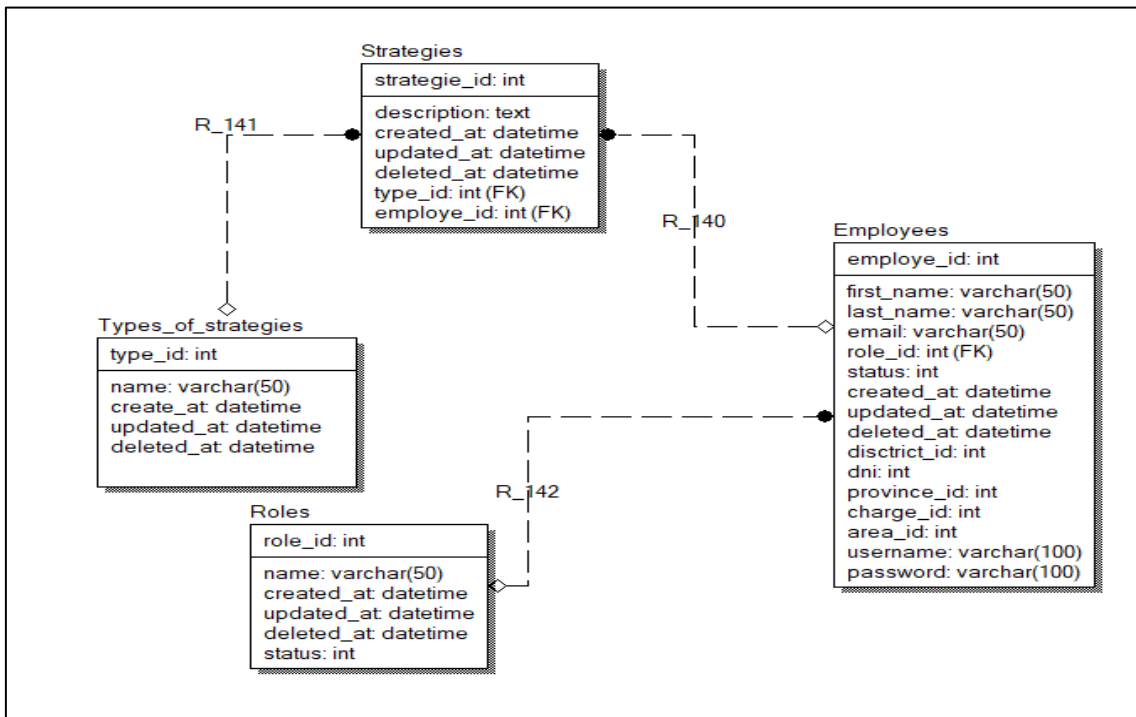


Figura 172. Diagrama físico de la base de datos RF33

Diseño

© Fuente: Elaboración propia

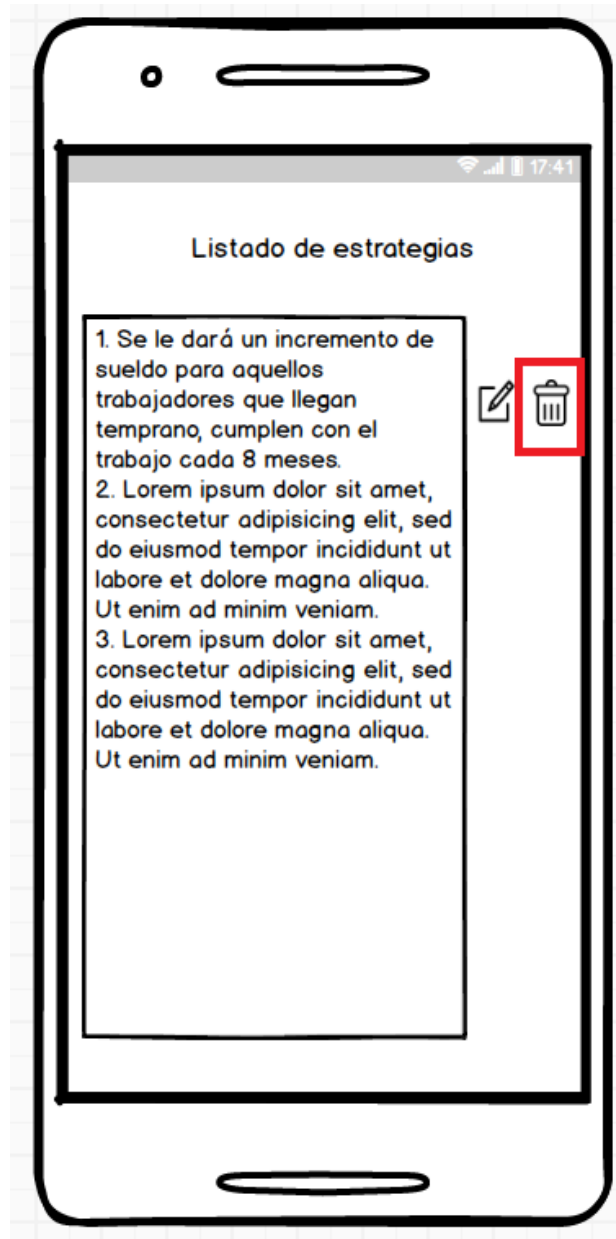


Figura 173. Prototipo de RF33

En la figura 173. se muestra el prototipo para la GUI de eliminar estrategias de rotación del usuario administrador que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

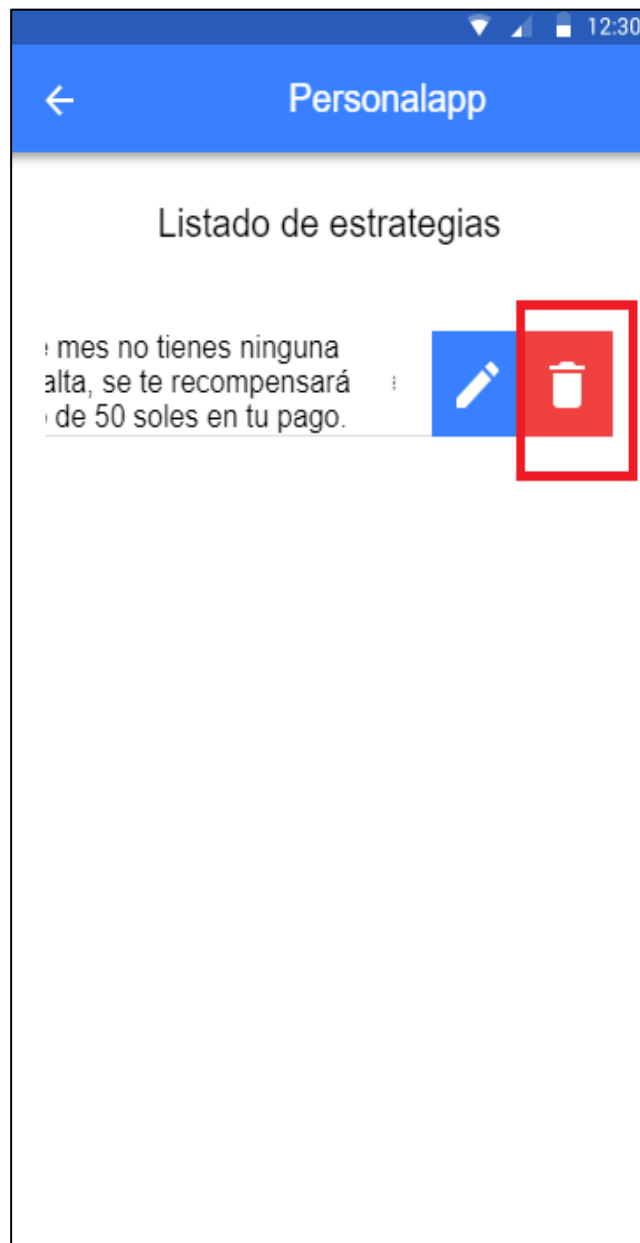


Figura 174. Interface de RF33

En la figura 174 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteStrategie(Request $request){
    $strategie = Strategie::find($this->request->input('key: id'));
    $strategie->delete();
    return response()->json([
        'msg' => 'exito',
        'strategie_type_id' => $strategie->type_id
    ], status: 200);
}
```

Figura 175. Código de RF33

En esta imagen 175 de la presente tesis se puede observar el código fuente del requerimiento funcional 33.

Requerimiento RF34 – Listar reglamentos

El requerimiento 34, tiene como objetivo que la aplicación debe tener un listado de reglamentos, en el cual los usuarios podrán visualizar para que así puedan saber las políticas de la empresa, dicho requerimiento solo deben hacerlo los usuarios “Administrador”, “Recursos humanos” y “Obrero”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

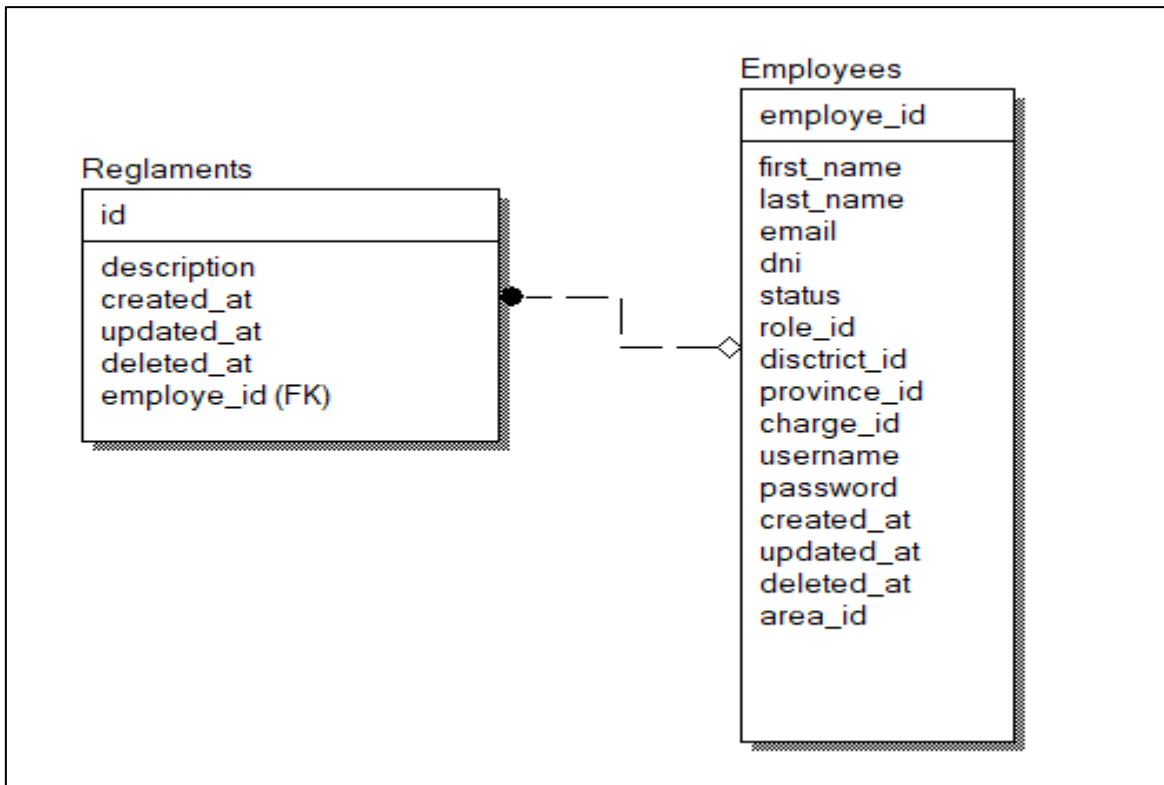


Figura 176. Diagrama lógico de la base de datos RF34

© Fuente: Elaboración propia

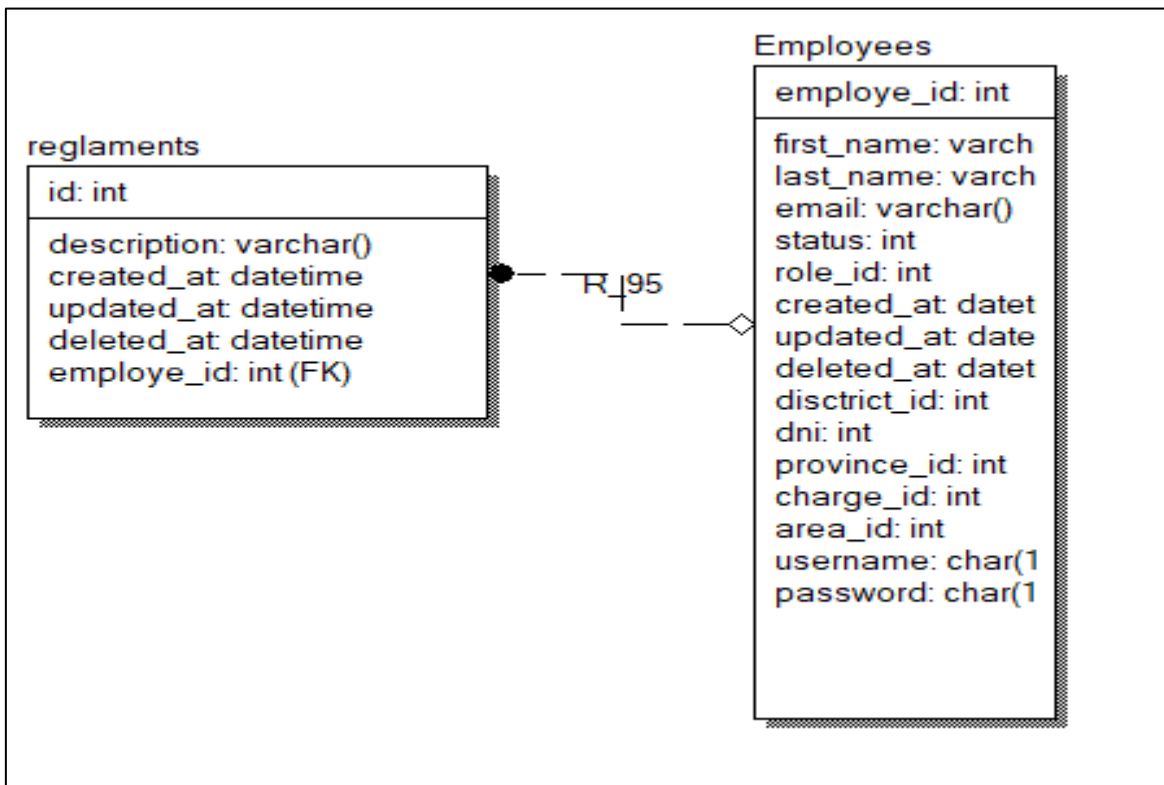


Figura 177. Diagrama físico de la base de datos RF34

Diseño

© Fuente: Elaboración propia

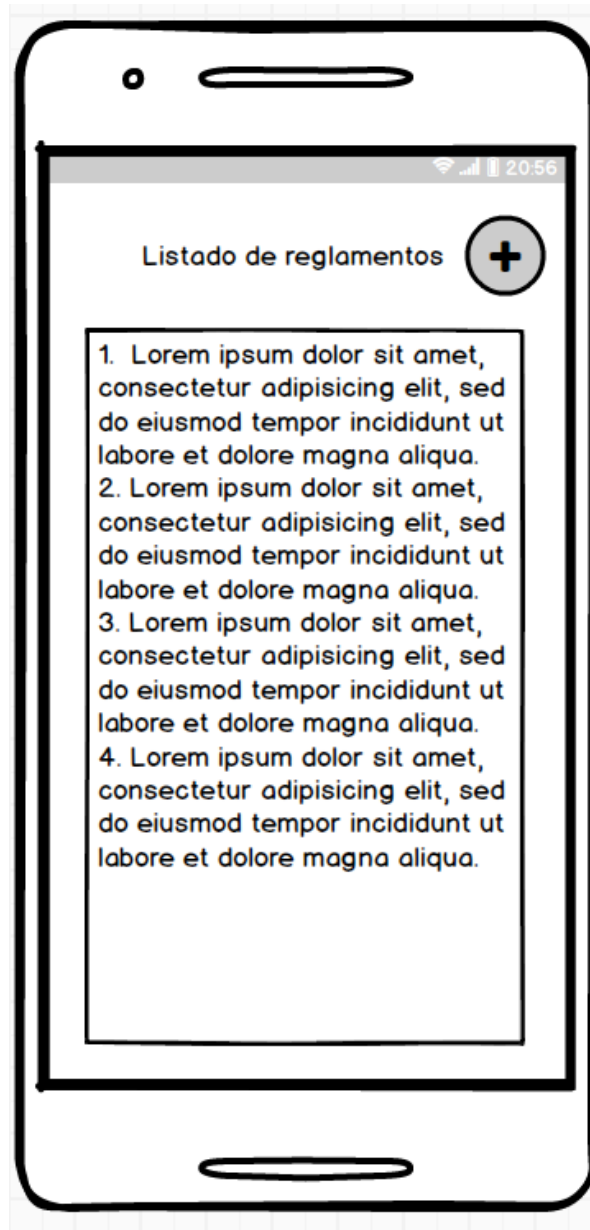


Figura 178. Prototipo de RF34

En la figura 178 se muestra el prototipo para la GUI de listar reglamentos del usuario administrador, recursos humanos y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

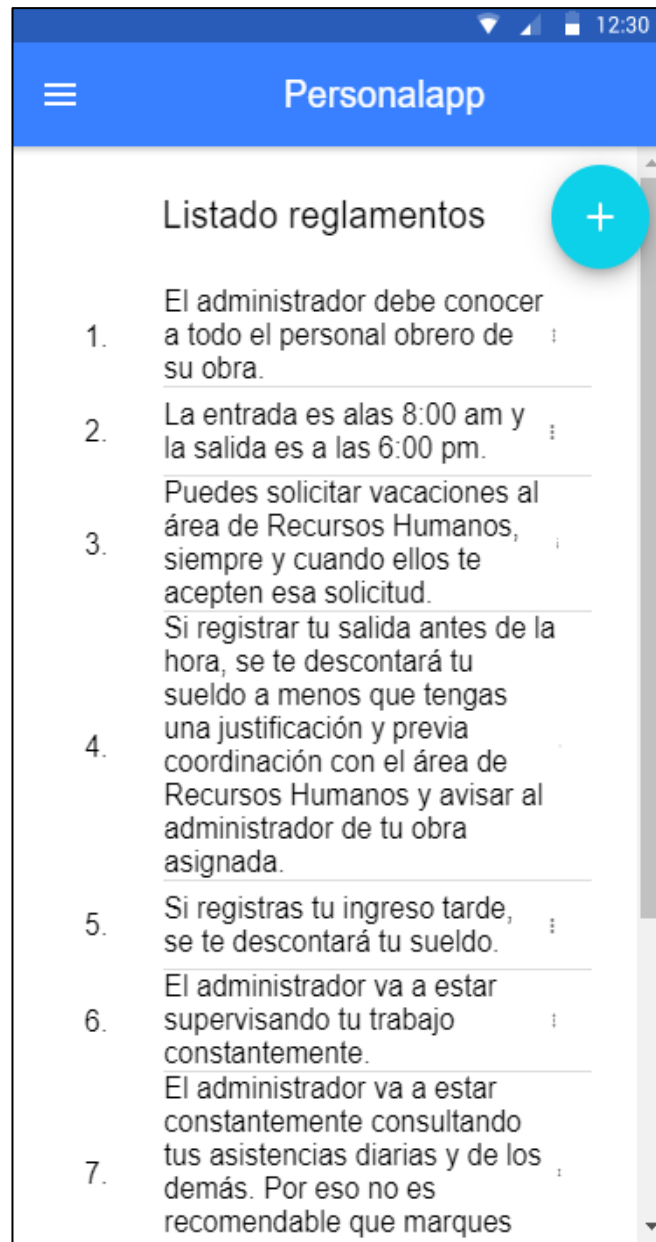


Figura 179. Interface de RF34

En la figura 179 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadReglaments() {  
    $reglaments = Reglament::all();  
    return $reglaments;  
}
```

Figura 180. Código de RF34

En esta imagen 180 de la presente tesis se puede observar el código fuente del requerimiento funcional 34.

Requerimiento RF35 – Crear reglamentos

El requerimiento 35, tiene como objetivo que en la aplicación se debe poder crear los reglamentos de la empresa para que puedan visualizar los usuarios, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 35).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

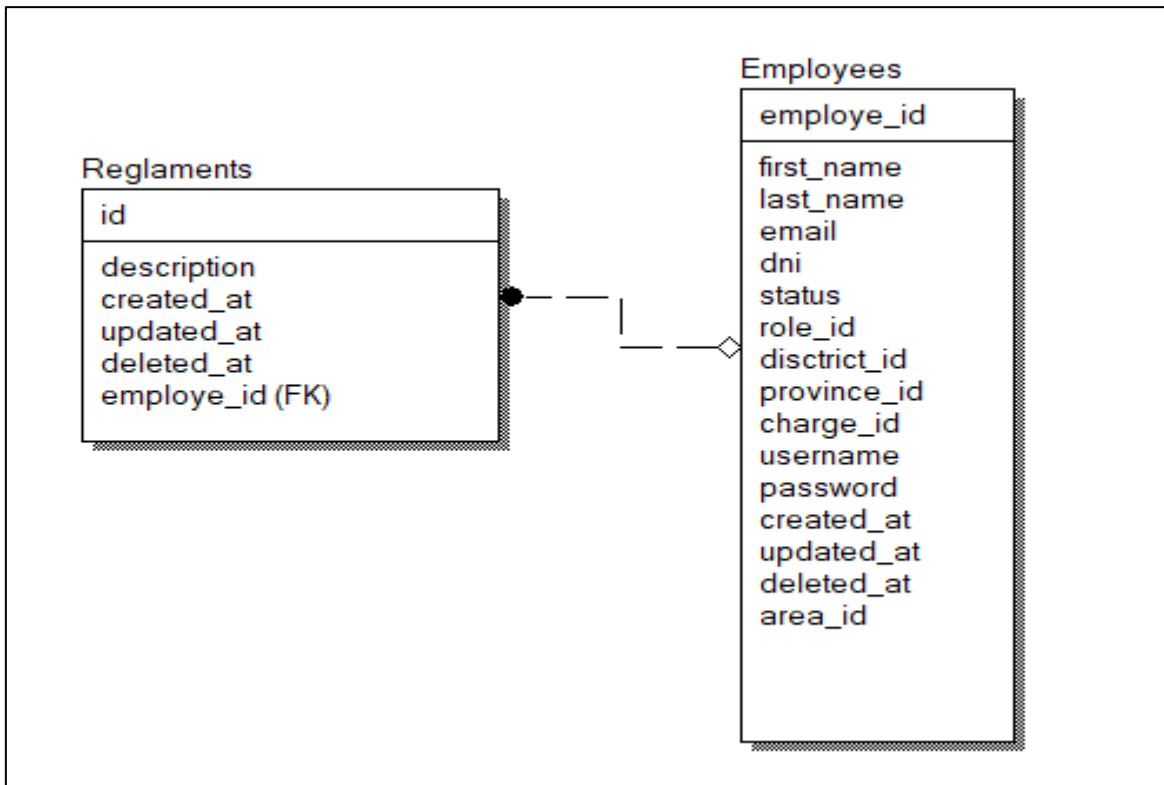


Figura 181. Diagrama lógico de la base de datos RF35

© Fuente: Elaboración propia

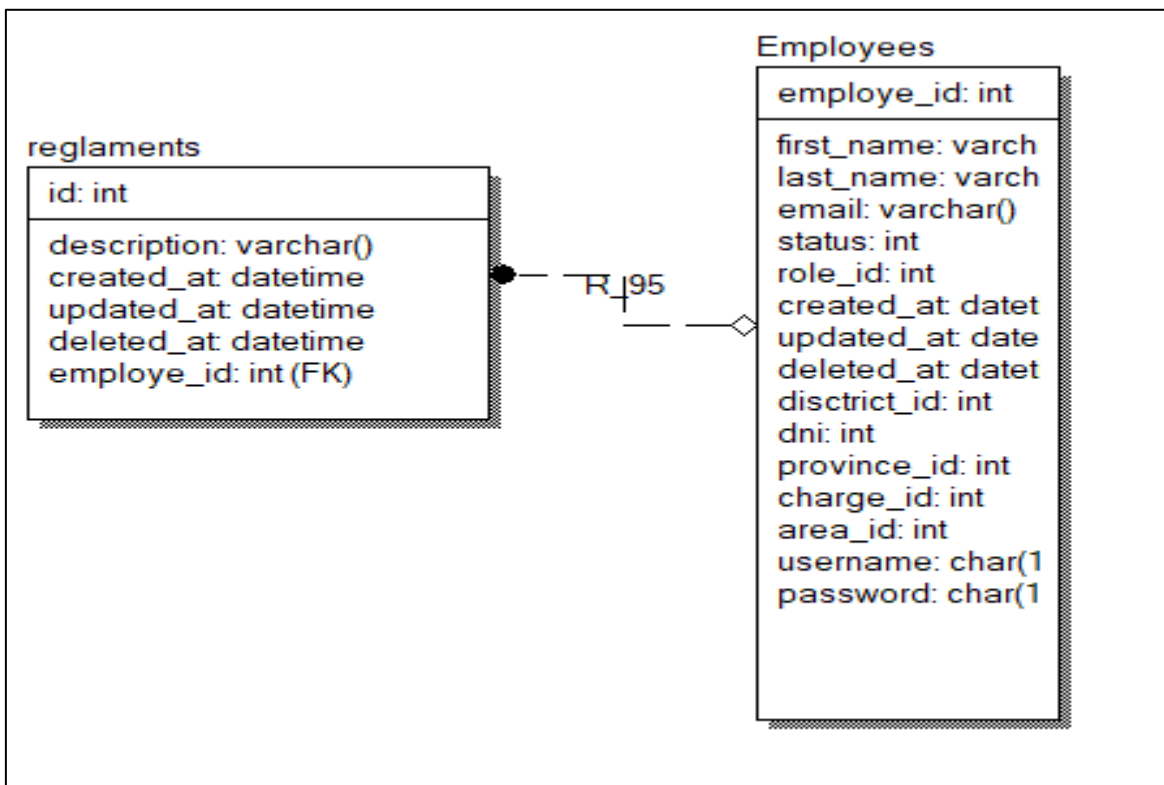


Figura 182. Diagrama físico de la base de datos RF35

Diseño

© Fuente: Elaboración propia

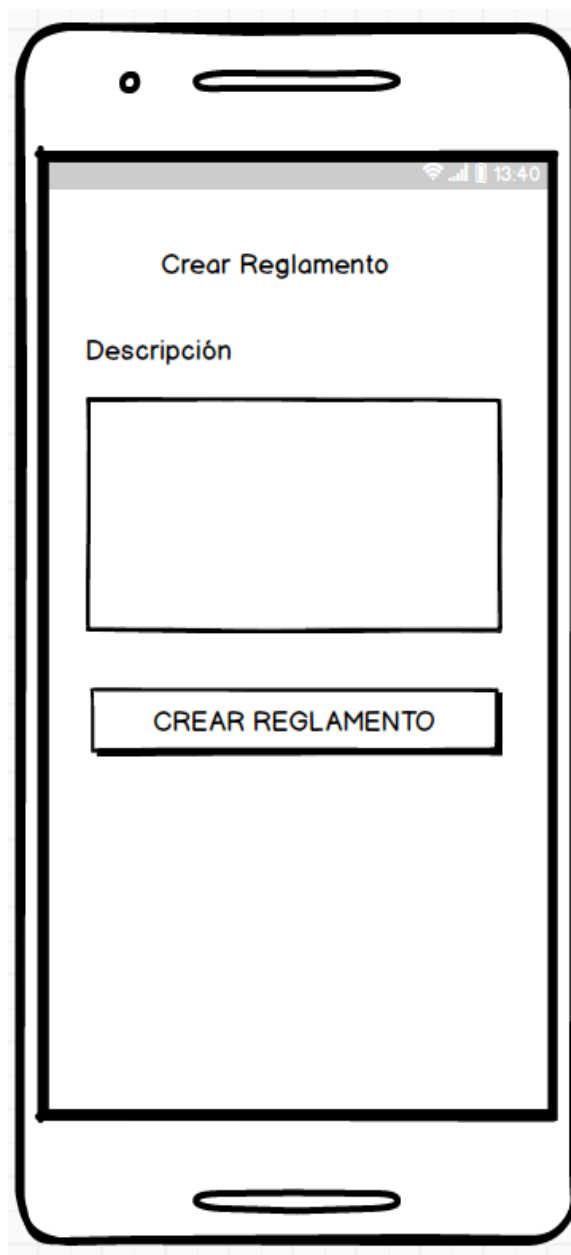


Figura 183. Prototipo de RF35

En la figura 183 se muestra el prototipo para la GUI de crear reglamentos del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

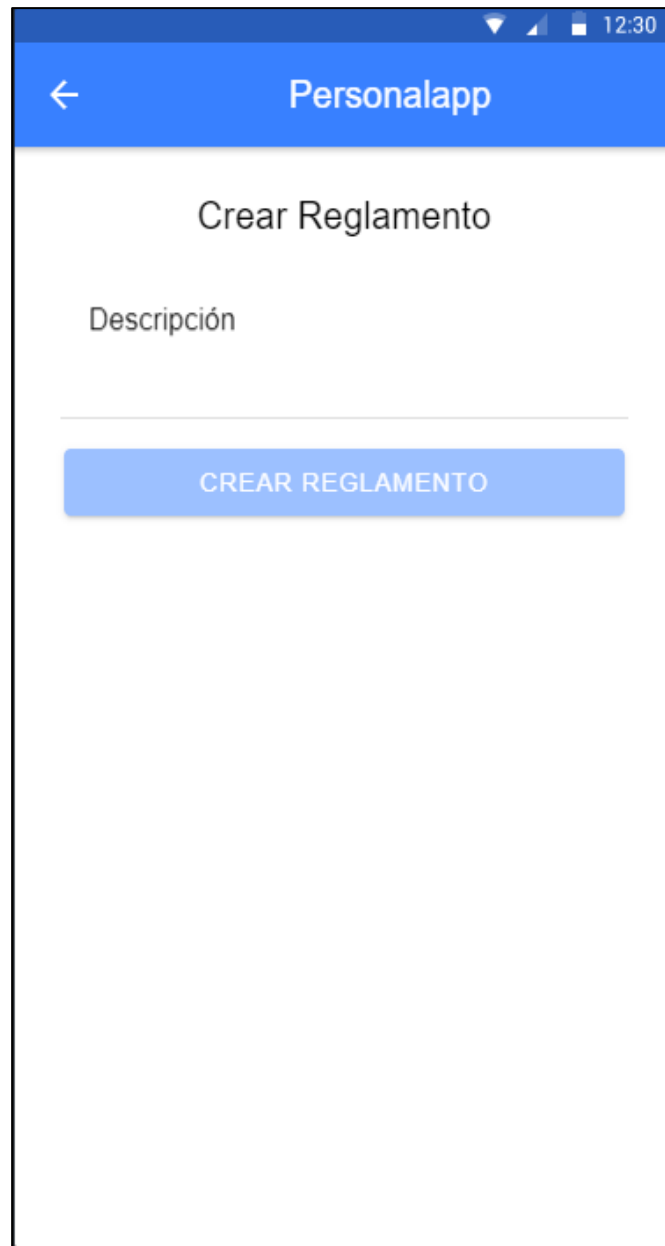


Figura 184. Interface de RF35

En la figura 184 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function save(Request $request){
    $reglament = new Reglament();
    $reglament->description = $request->description;
    $reglament->employee_id = $request->employee_id;
    $reglament->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 185. Código de RF35

En esta imagen 185 de la presente tesis se puede observar el código fuente del requerimiento funcional 35.

Requerimiento RF36 – Actualizar reglamentos

El requerimiento 36, tiene como objetivo que en la aplicación se debe poder actualizar las descripciones de los reglamentos de la empresa, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 36).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

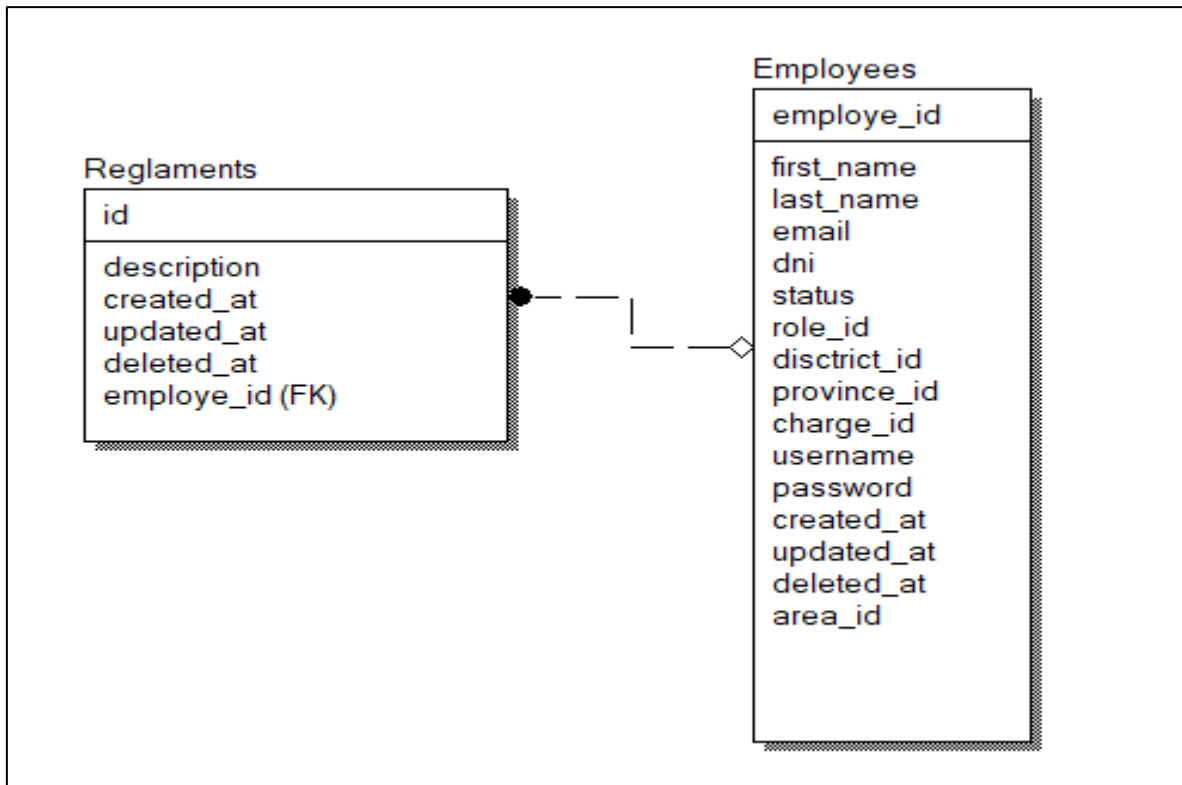


Figura 186. Diagrama lógico de la base de datos RF36

© Fuente: Elaboración propia

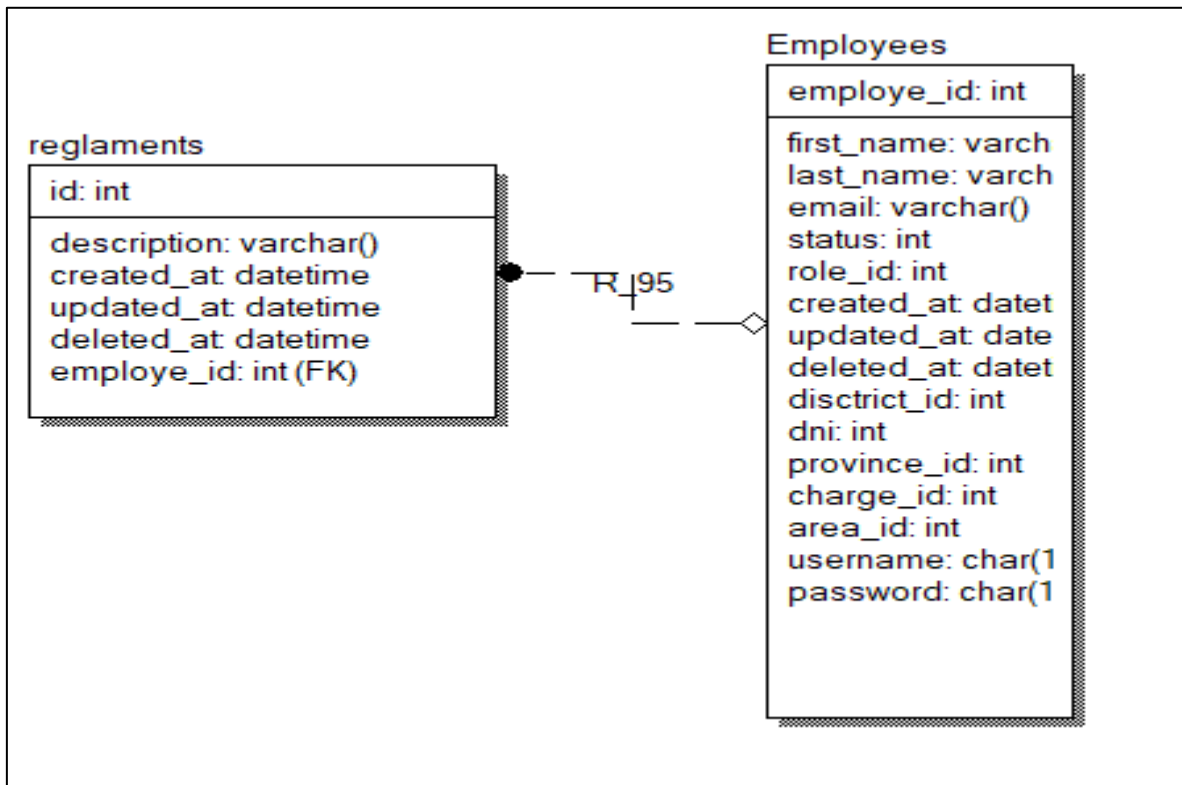


Figura 187. Diagrama físico de la base de datos RF36

Diseño

© Fuente: Elaboración propia

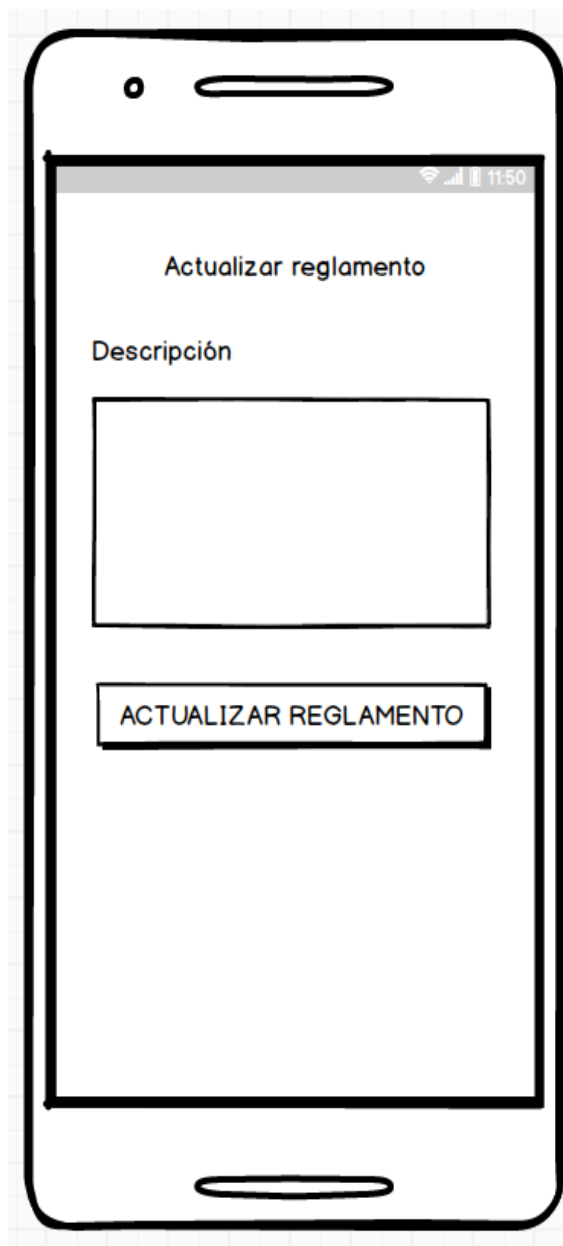


Figura 188. Prototipo de RF36

En la figura 188 se muestra el prototipo para la GUI de actualizar reglamentos del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

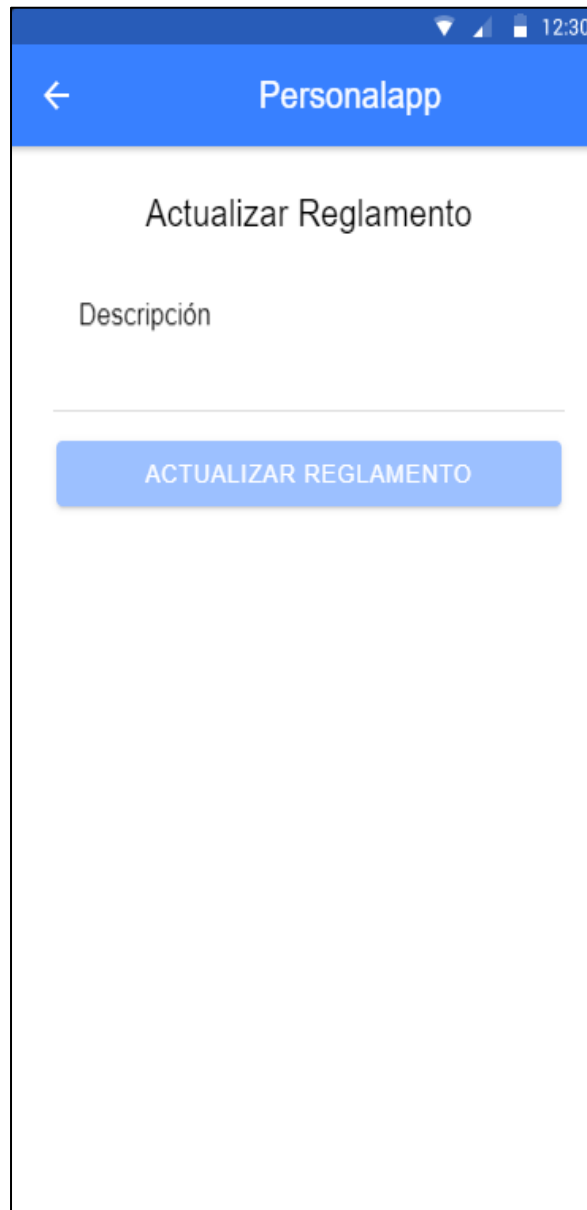


Figura 189. Interface de RF36

En la figura 189 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateReglament(Request $request,$id){
    $reglament = Reglament::find($id);
    $reglament->description = $request->description;
    $reglament->employee_id = $request->employee_id;
    $reglament->save();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 190. Código de RF36

En esta imagen 190 de la presente tesis se puede observar el código fuente del requerimiento funcional 36.

Requerimiento RF37 – Eliminar reglamentos

El requerimiento 37, tiene como objetivo que en la aplicación se debe poder eliminar los reglamentos, ya creados anteriormente, dicho requerimiento solo puede hacerlo el usuario “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 37).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

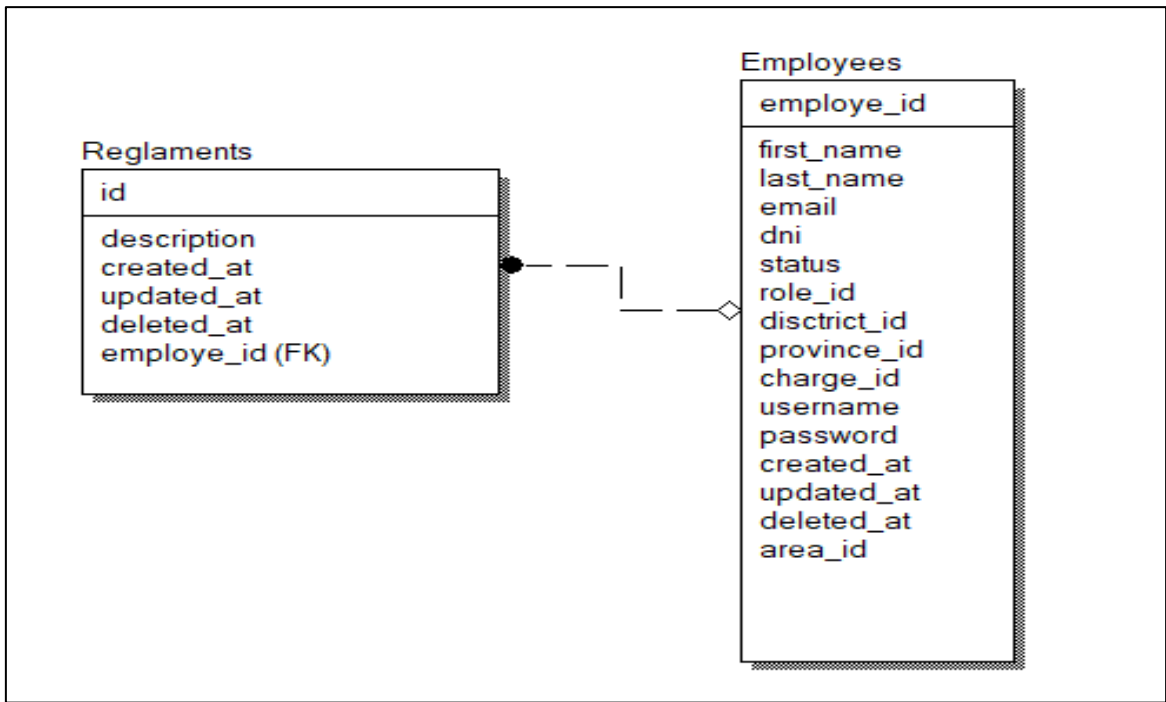


Figura 191. Diagrama lógico de la base de datos RF37

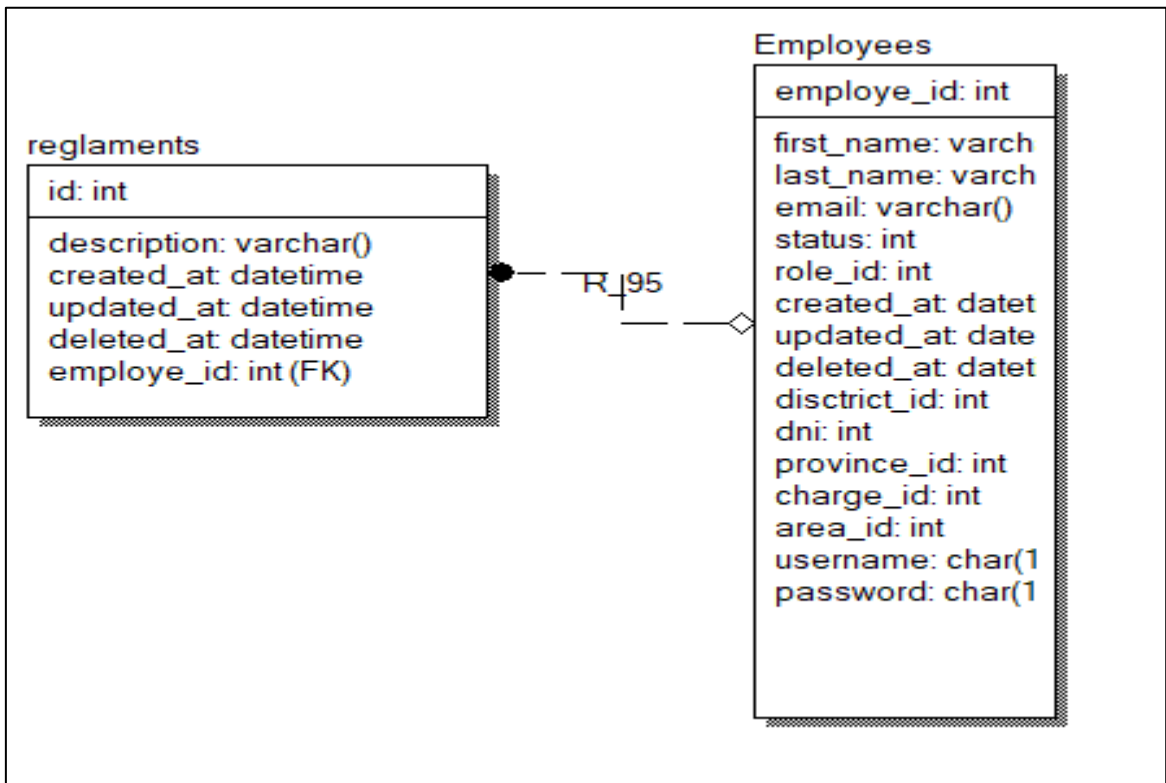


Figura 192. Diagrama físico de la base de datos RF37

Diseño

© Fuente: Elaboración propia

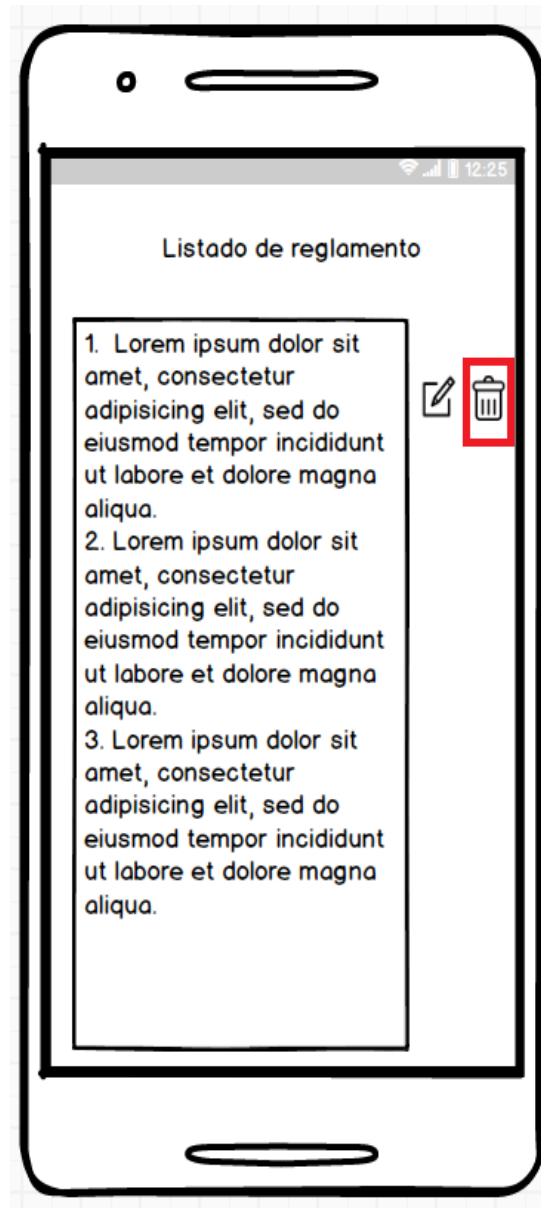


Figura 193. Prototipo de RF37

En la figura 193 se muestra el prototipo para la GUI de eliminar reglamentos del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

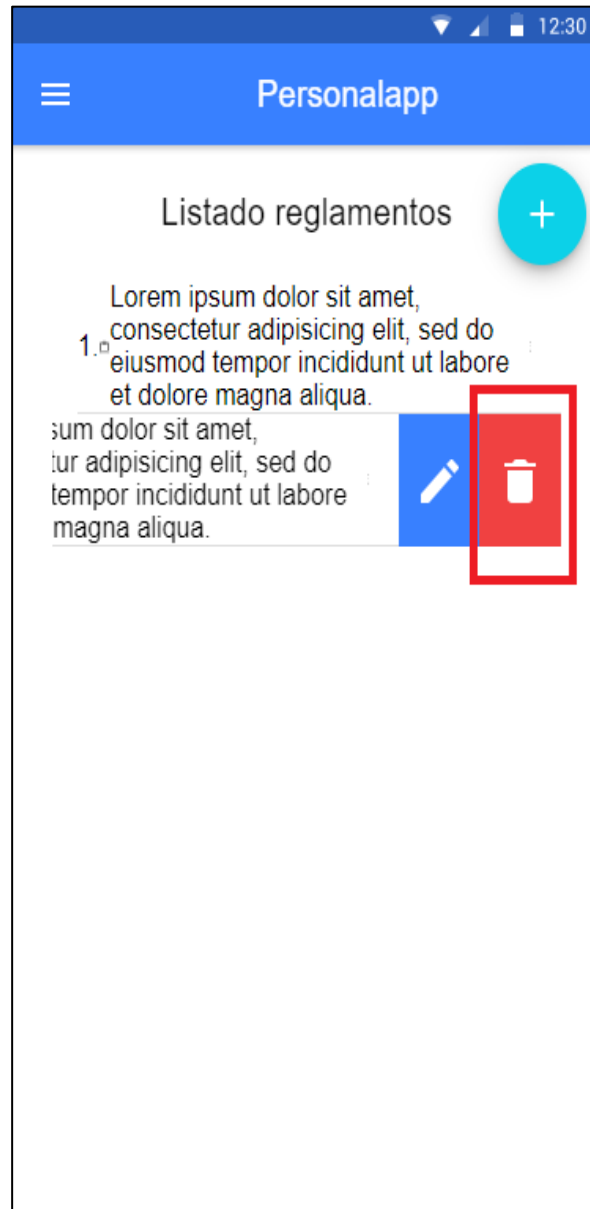


Figura 194. Interface de RF37

En la figura 194 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteReglament(Request $request){
    $employee = Reglament::find($this->request->input('key: id'));
    $employee->delete();
    return response()->json([
        'msg' => 'exito'
    ], status: 200);
}
```

Figura 195. Código de RF37

En esta imagen 195 de la presente tesis se puede observar el código fuente del requerimiento funcional 37.

Requerimiento RF38 – Listar gratificaciones

El requerimiento 38, tiene como objetivo que en la aplicación se debe tener un listado de las gratificaciones, en el cual cada usuario podrá visualizar dicha gratificación, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 38).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

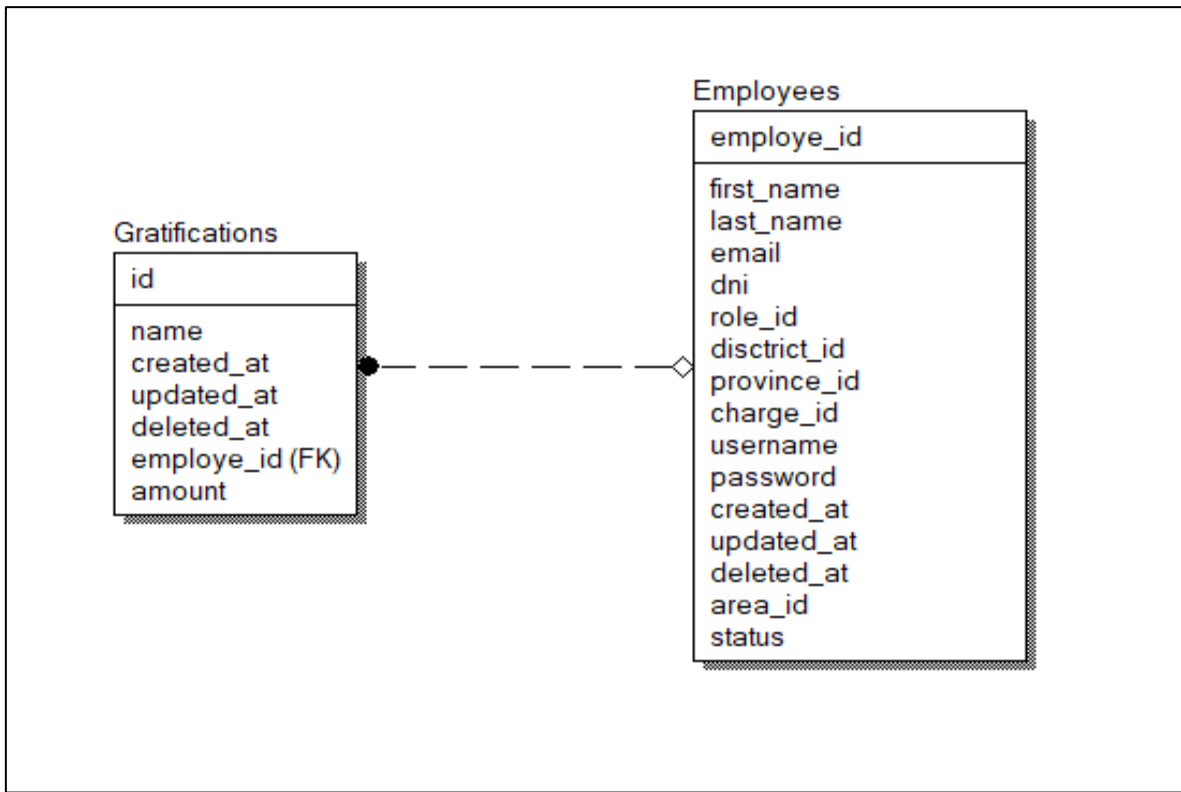


Figura 196. Diagrama lógico de la base de datos RF38

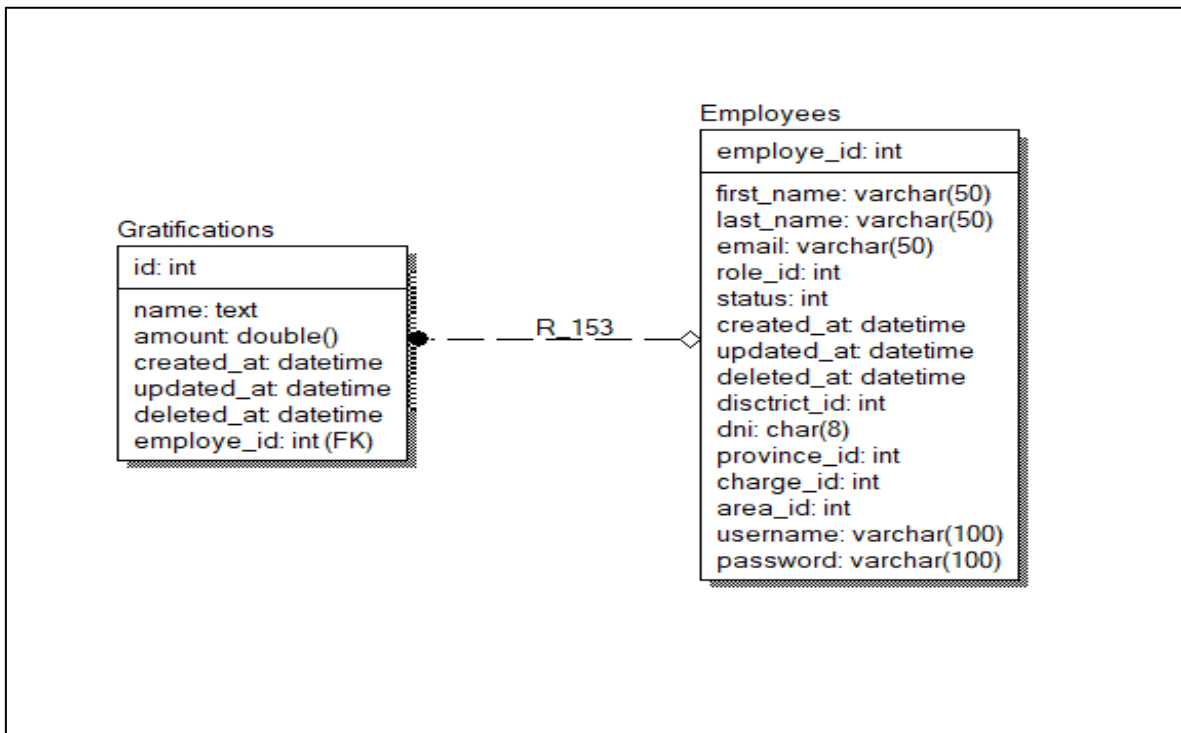


Figura 197. Diagrama físico de la base de datos RF38

Diseño

© Fuente: Elaboración propia

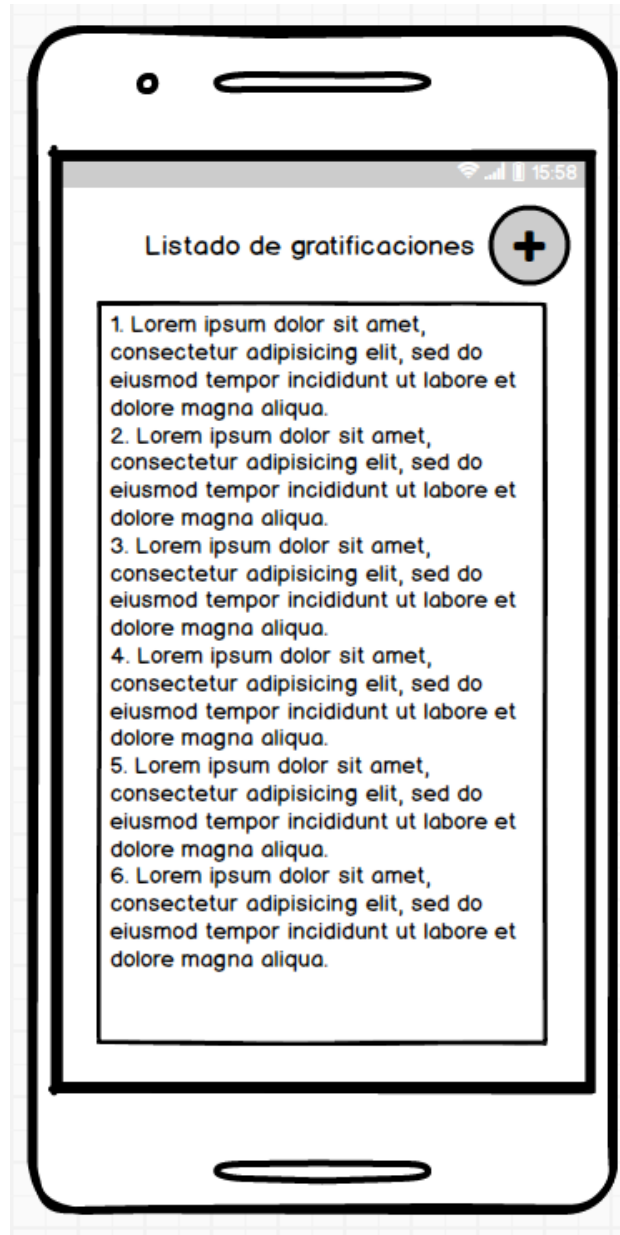


Figura 198. Prototipo de RF38

En la figura 198 se muestra el prototipo para la GUI de listar gratificaciones de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

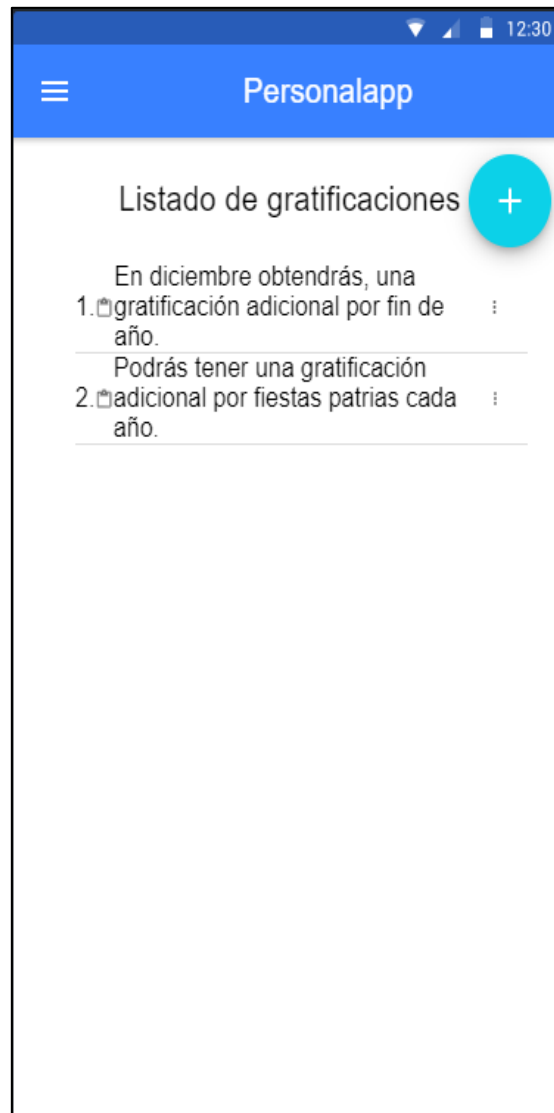


Figura 199. Interface de RF38

En la figura 199 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadGratifications() {  
    $gratifications = Gratification::all();  
    return $gratifications;  
}
```

Figura 200. Código de RF38

En esta imagen 200 de la presente tesis se puede observar el código fuente del requerimiento funcional 38.

Requerimiento RF39 – Crear gratificaciones

El requerimiento 39, tiene como objetivo que en la aplicación se debe poder crear gratificaciones para los usuarios, el cual podrán verlo después en la app, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 39).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

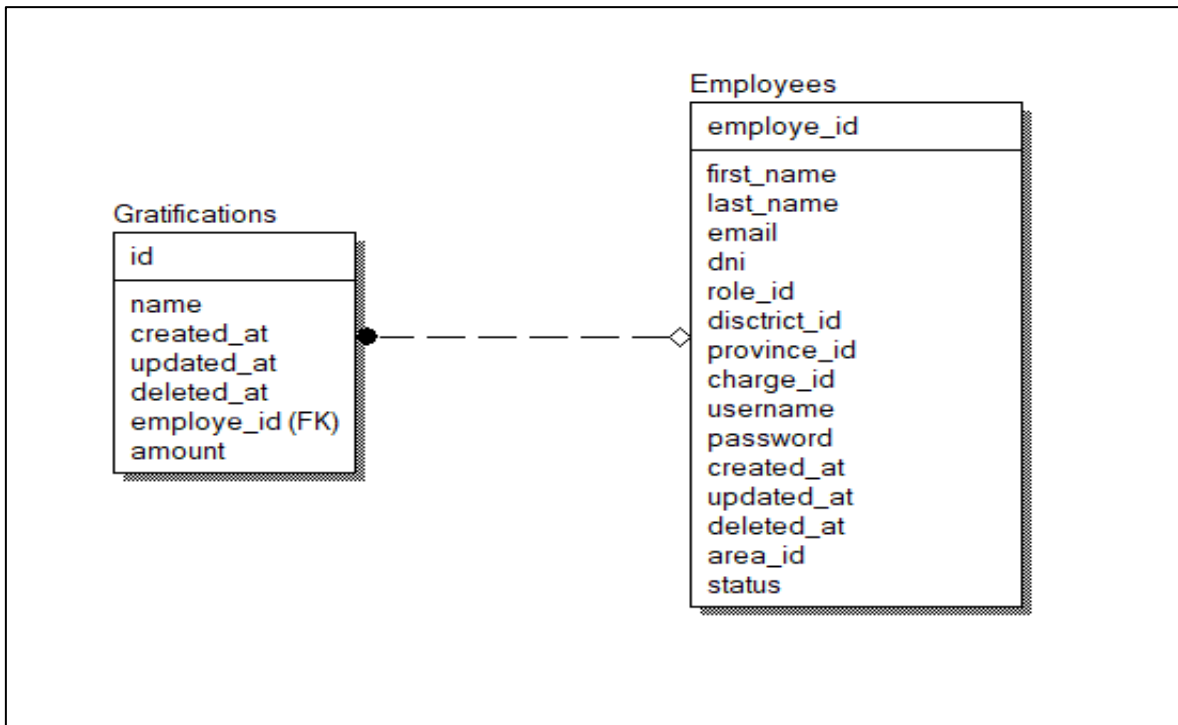


Figura 201. Diagrama lógico de la base de datos RF39

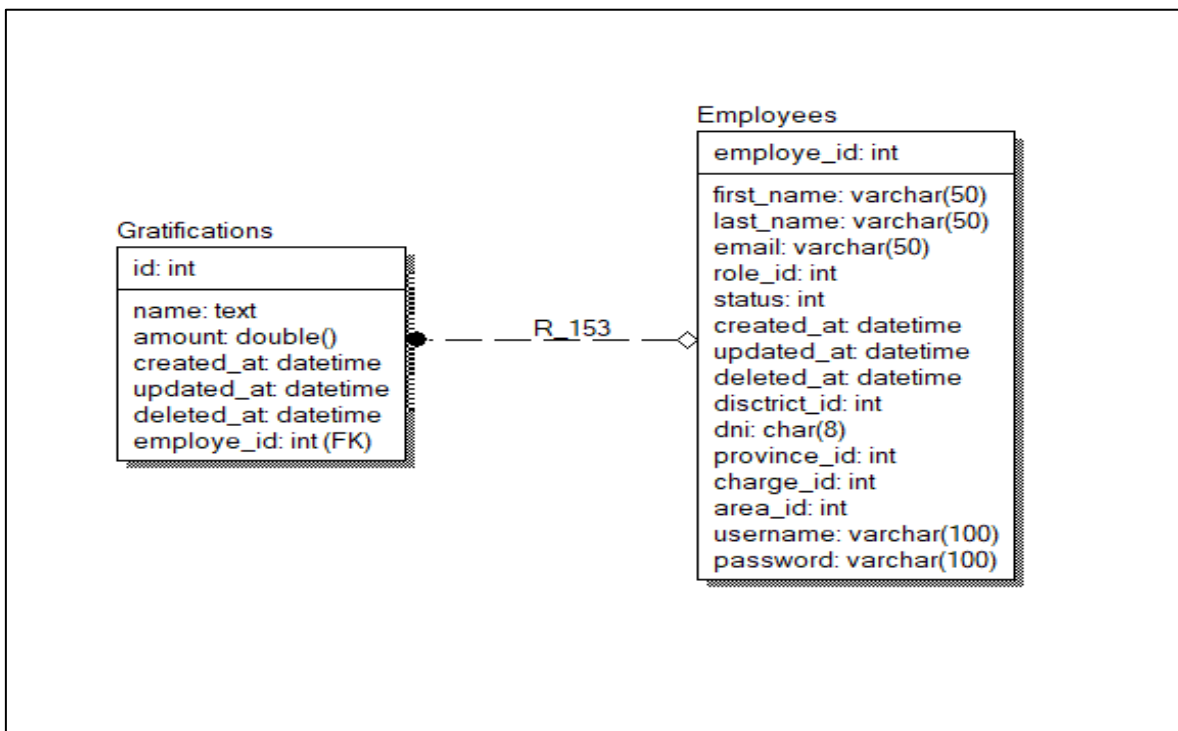


Figura 202. Diagrama físico de la base de datos RF39

Diseño

© Fuente: Elaboración propia



Figura 203. Prototipo de RF39

En la figura 203 se muestra el prototipo para la GUI de crear gratificaciones del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



Figura 204. Interface de RF39

En la figura 204 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function createGratification(Request $request) {
    // $miação = Carbon::now('America/Lima');
    $gratification = Gratification::all();
    if(count($gratification) == 2){
        return response()->json([
            'msg' => 'imposible',
        ], status: 200);
    }else{
        $gratification = new Gratification();
        $gratification->name = $request->name;
        $gratification->employee_id = $request->employee_id;
        $gratification->save();
        return response()->json([
            'msg' => 'exito',
        ], status: 200);
    }
}
```

Figura 205. Código de RF39

En esta imagen 205 de la presente tesis se puede observar el código fuente del requerimiento funcional 39.

Requerimiento RF40 – Actualizar gratificaciones

El requerimiento 40, tiene como objetivo que en la aplicación se debe poder actualizar las gratificaciones de los usuarios, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 40).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

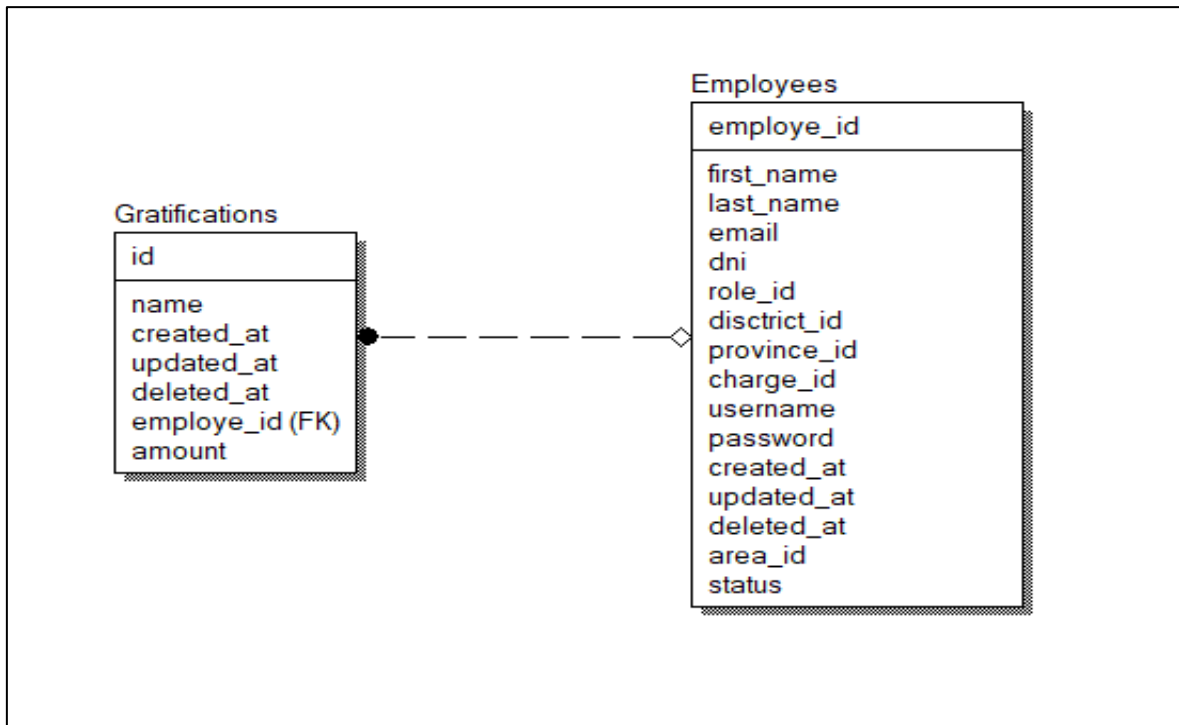


Figura 206. Diagrama lógico de la base de datos RF40

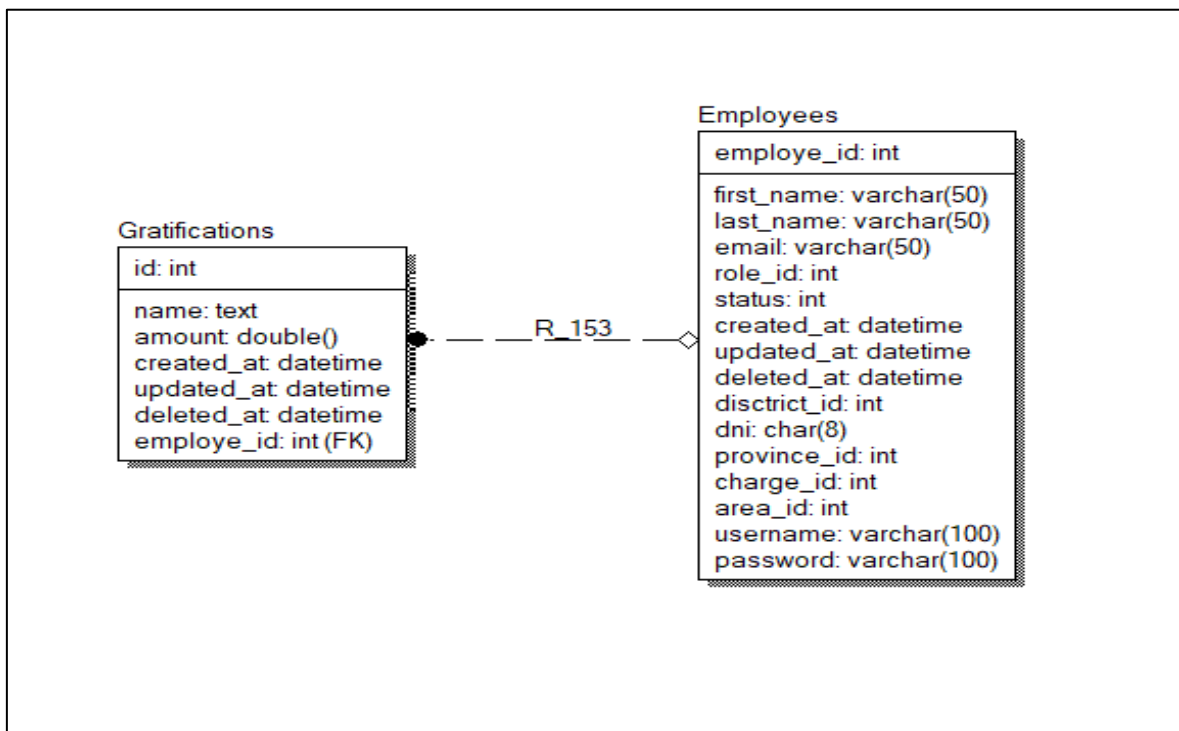


Figura 207. Diagrama físico de la base de datos RF40

© Fuente: Elaboración propia



Figura 208. Prototipo de RF40

En la figura 208 se muestra el prototipo para la GUI de actualizar gratificaciones del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

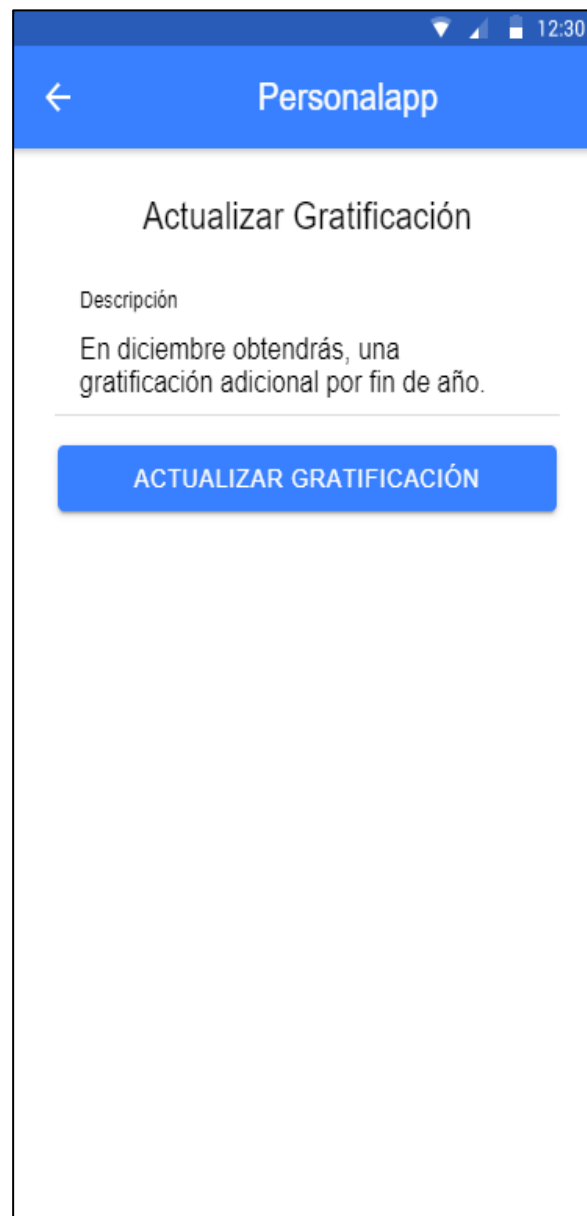


Figura 209. Interface de RF40

En la figura 209 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function editGratification(Request $request,$id){
    $gratification = Gratification::findOrFail($id);
    return $gratification;
}

public function updateGratification(Request $request,$id){
    $gratification = Gratification::findOrFail($id);
    $gratification->name = $request->name;
    $gratification->employee_id = $request->employee_id;
    $gratification->save();
    return response()->json([
        'msg' => 'exito',
    ], status: 200);
}
```

Figura 210. Código de RF40

En esta imagen 210 de la presente tesis se puede observar el código fuente del requerimiento funcional 40.

Requerimiento RF41 – Eliminar gratificaciones

El requerimiento 41, tiene como objetivo que en la aplicación se debe poder eliminar las gratificaciones de los usuarios, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 41).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

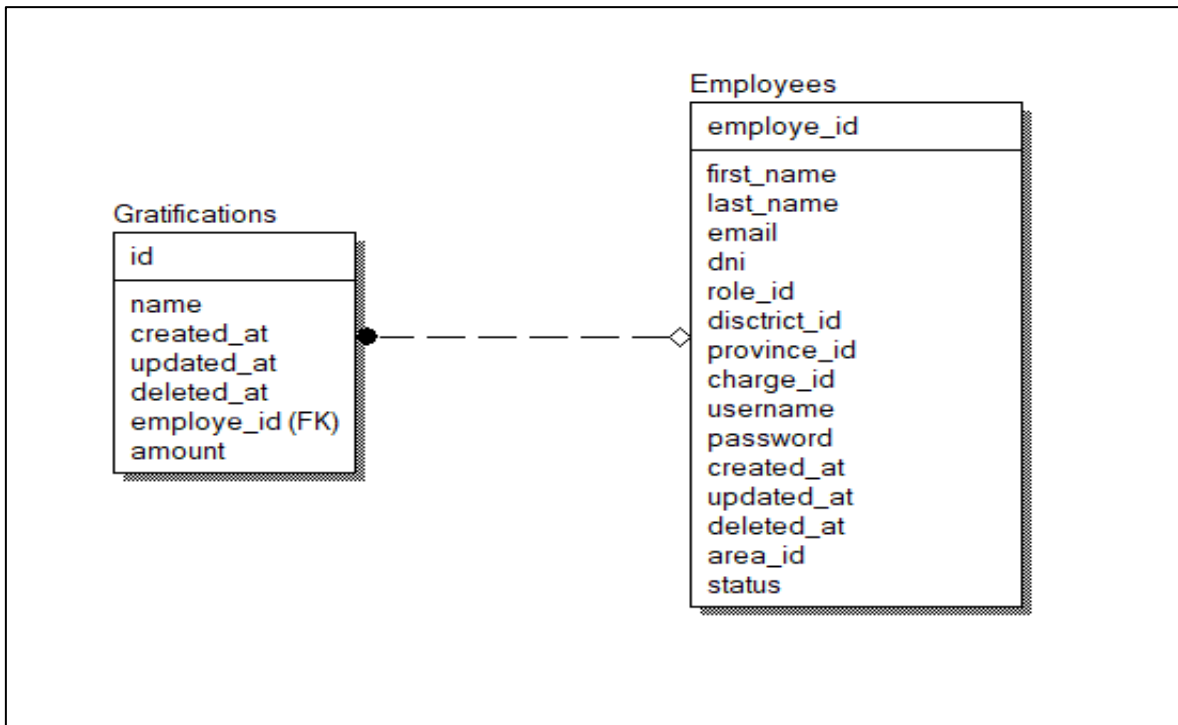


Figura 211. Diagrama lógico de la base de datos RF41

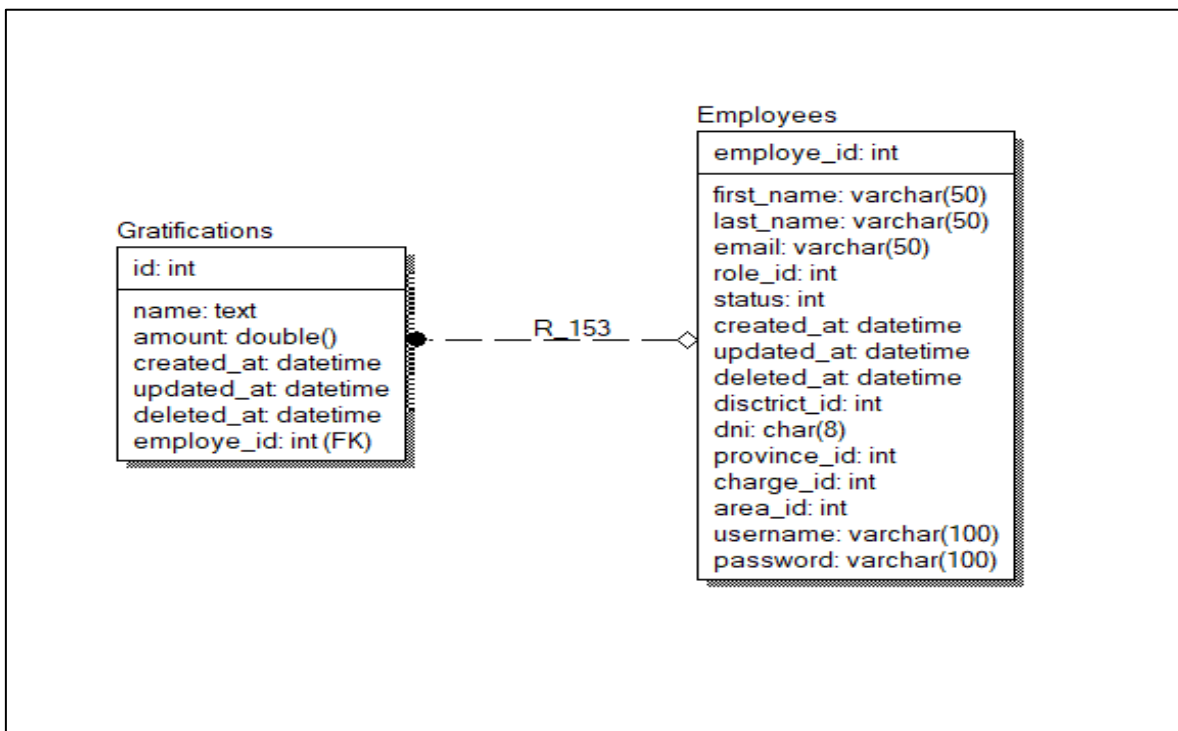


Figura 212. Diagrama físico de la base de datos RF41

Diseño

© Fuente: Elaboración propia

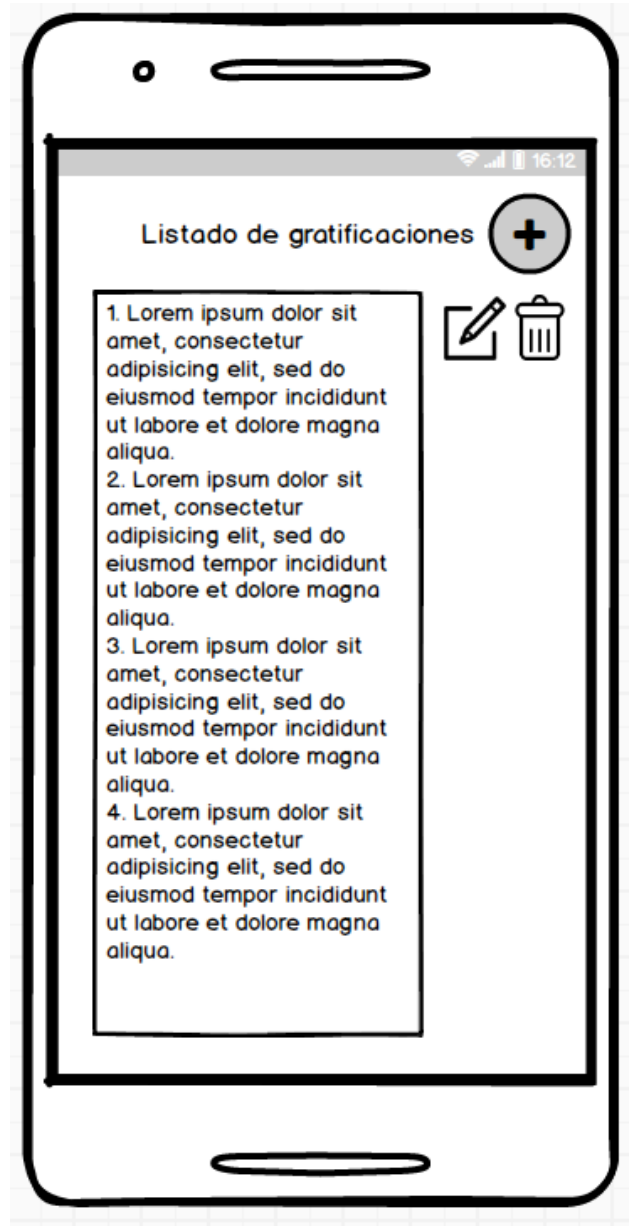


Figura 213. Prototipo de RF41

En la figura 213 se muestra el prototipo para la GUI de eliminar gratificaciones del usuario administrador y recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

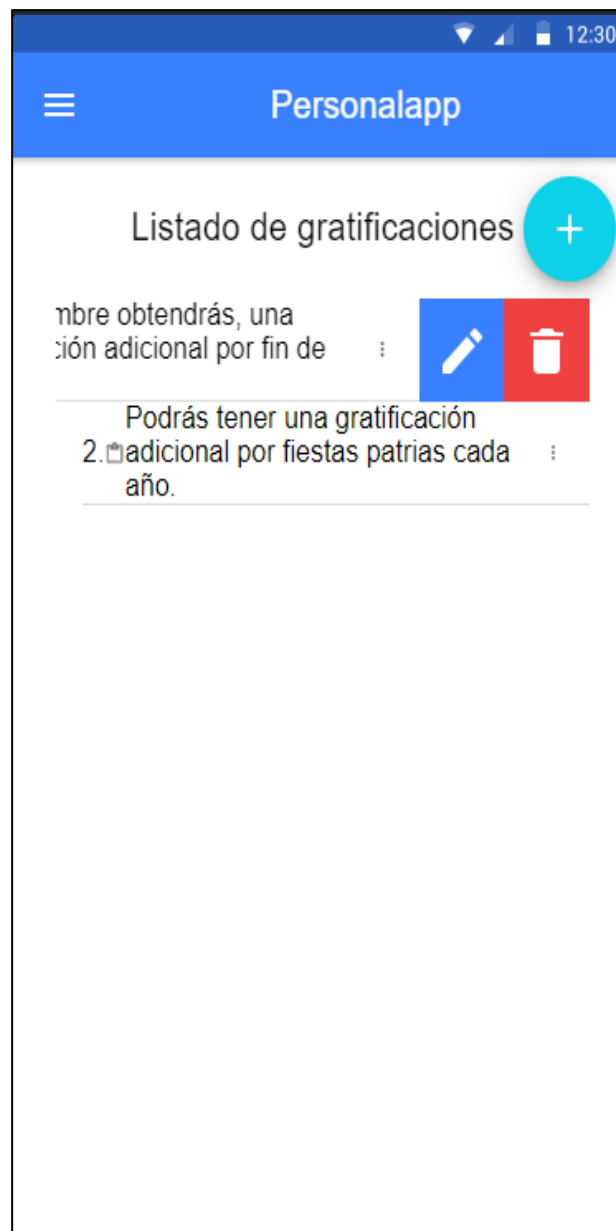


Figura 214. Interface de RF41

En la figura 214 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteJustification(Request $request) {  
    $justification = Justification::findOrFail($request->id);  
    $justification->delete();  
    return response()->json([  
        'msg' => 'exito',  
    ], status: 200);  
}
```

Figura 215. Código de RF41

En esta imagen 215 de la presente tesis se puede observar el código fuente del requerimiento funcional 41.

Requerimiento RF42 – Pulsar justificación

El requerimiento 42, tiene como objetivo que en la aplicación se debe ver un botón para entrar al listado de justificación, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 42).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

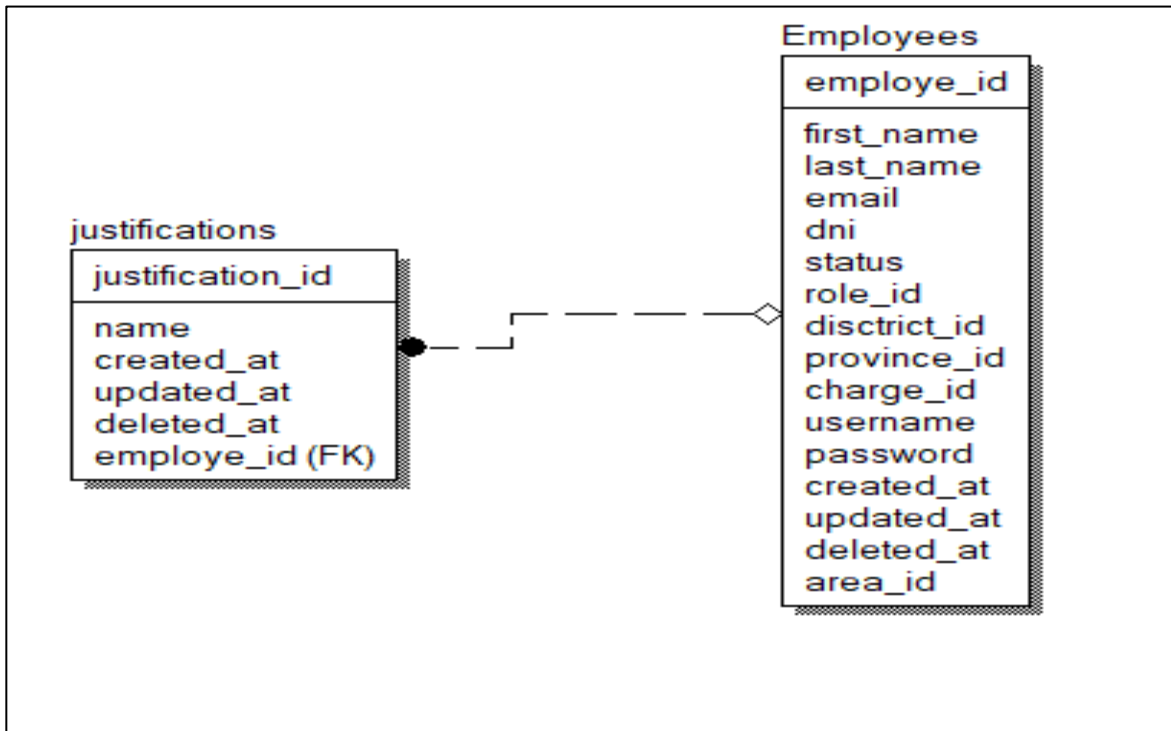


Figura 216. Diagrama lógico de la base de datos RF42

© Fuente: Elaboración propia

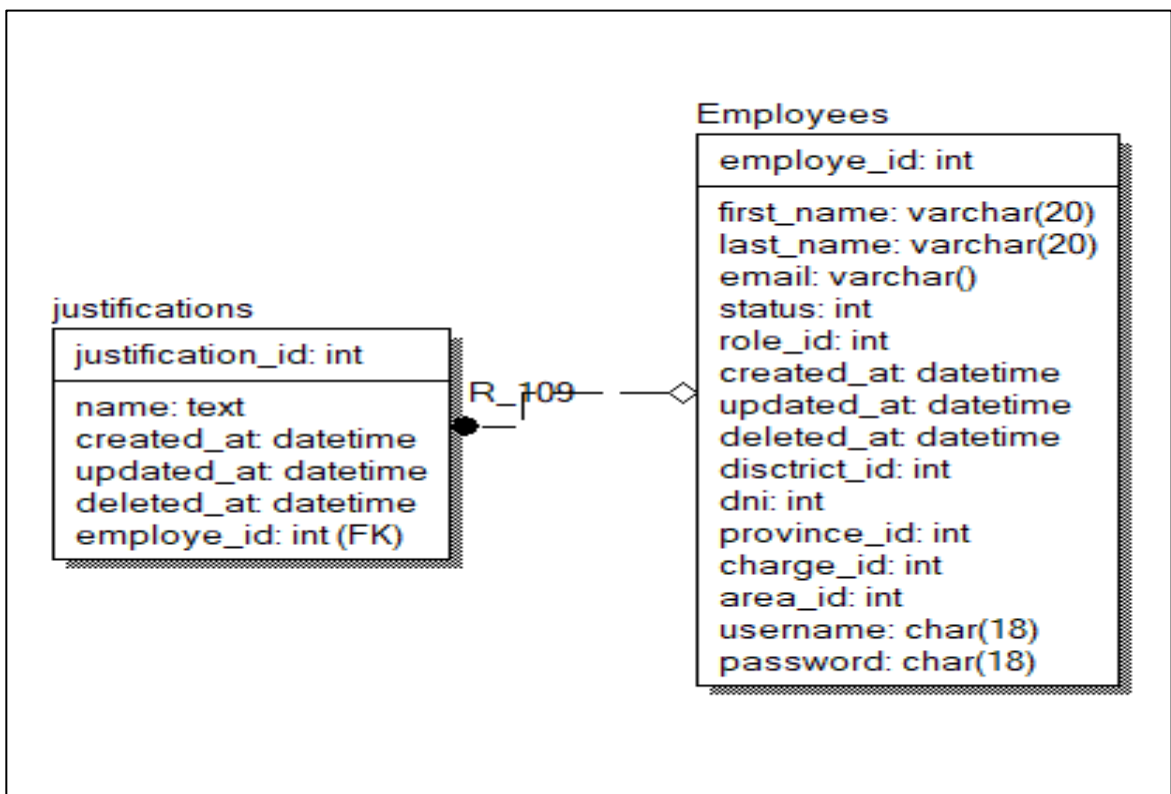


Figura 217. Diagrama físico de la base de datos RF42

Diseño

© Fuente: Elaboración propia



Figura 218. Prototipo de RF42

En la figura 218 se muestra el prototipo para la GUI de pulsar justificación de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

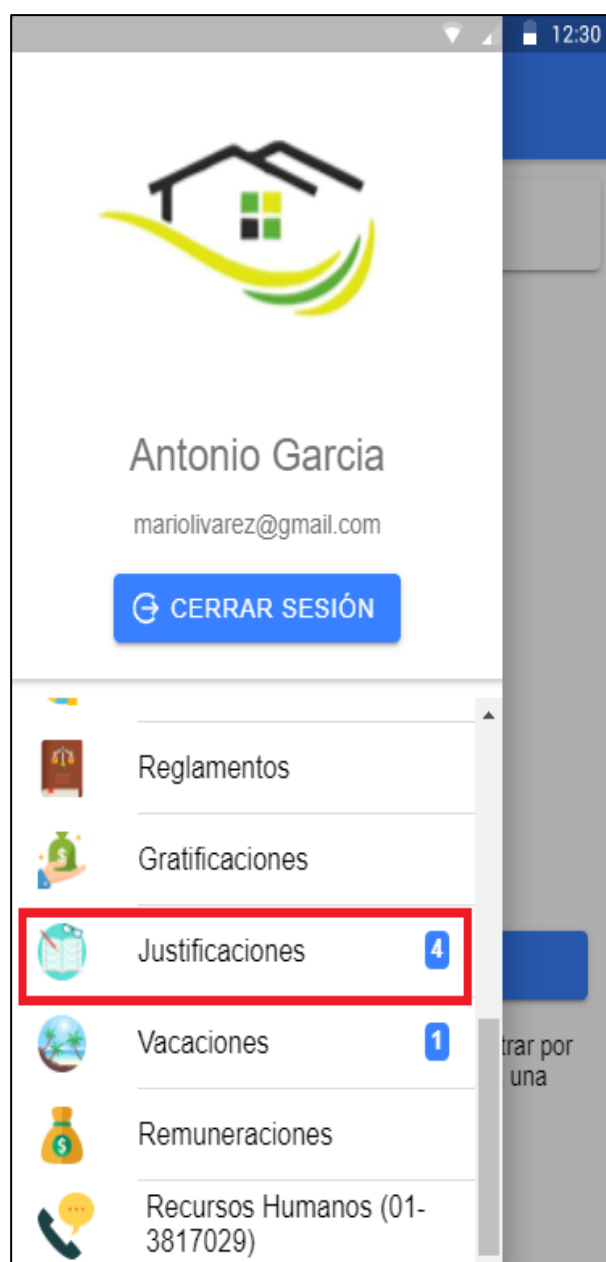


Figura 219. Interface de RF42

En la figura 219 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
if(res == 1){
  this.pages = [
    { title: 'Listado de obras', page: '/workcontroller', icon:'git-network' },
    { title: 'Listado de empleados', page: '/employeecontroller', icon:'git-network' },
    { title: 'Asignar Obra', page: '/assignwork', icon:'git-network' },
    { title: 'Empleados Desvinculados', page: '/dismissal', icon:'contacts' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Visualizar estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 2){
  this.pages = [
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 3){
  this.pages = [
    { title: 'Visualizar Obra Asignada', page: '/workassign', icon:'briefcase' },
    { title: 'Consultar Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Visualizar estrategias de negocio', page: '/viewnegotiations', icon:'briefcase' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}
```

Figura 220. Código de RF42

En esta imagen 220 de la presente tesis se puede observar el código fuente del requerimiento funcional 42.

Requerimiento RF43 – Listar empleado justificación

El requerimiento 43, tiene como objetivo que en la aplicación se debe poder ver un listado de los empleados que tengan o no una justificación, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 43).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

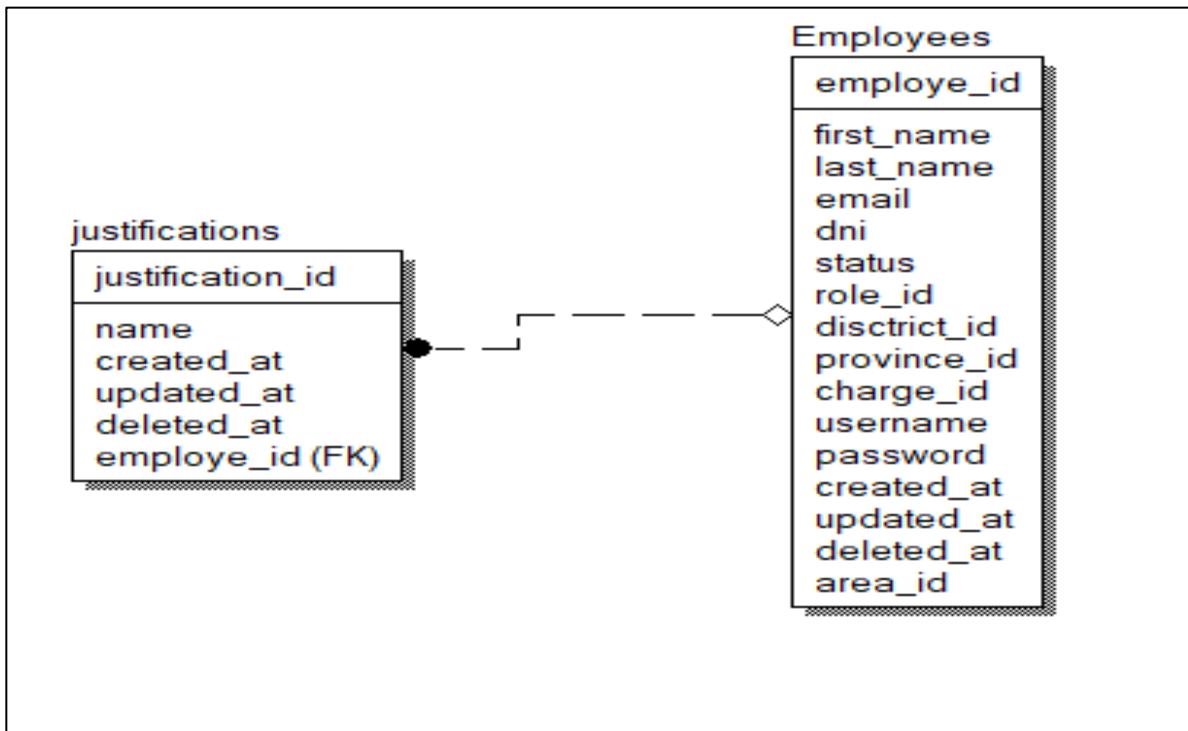


Figura 221. Diagrama lógico de la base de datos RF43

© Fuente: Elaboración propia

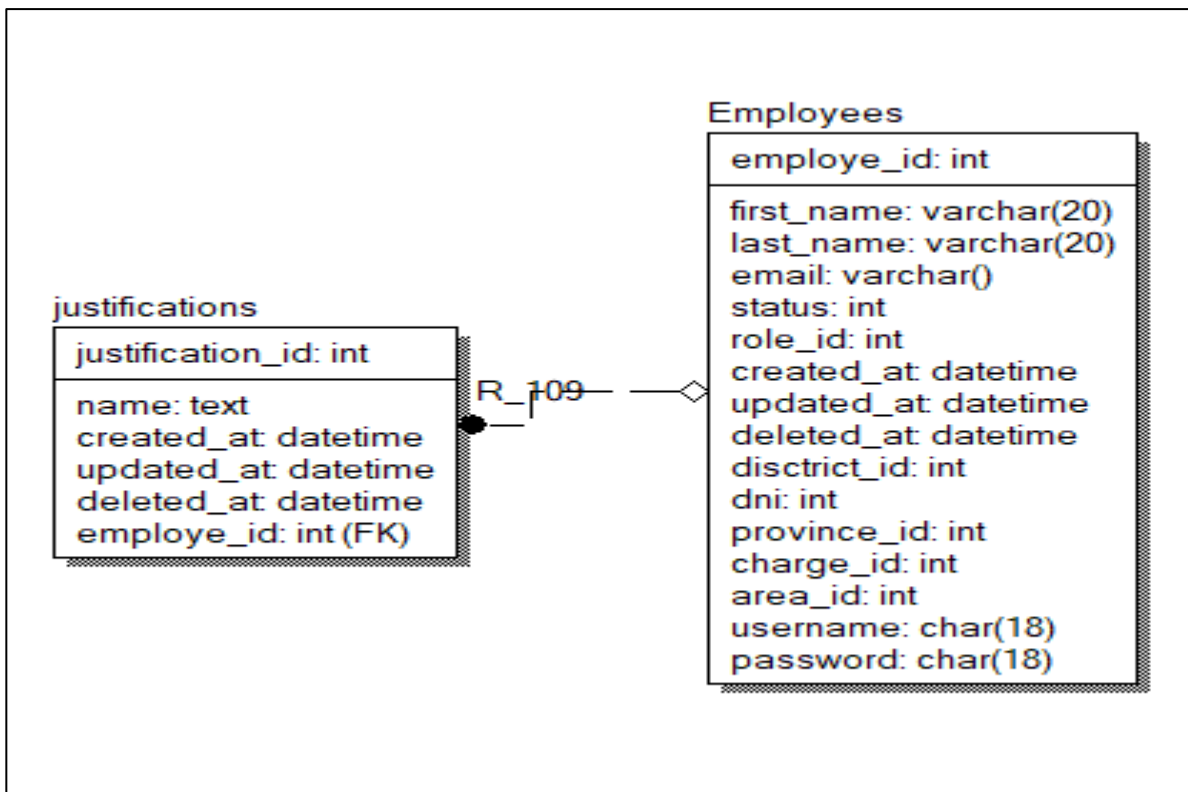


Figura 222. Diagrama físico de la base de datos RF43

Diseño

© Fuente: Elaboración propia



Figura 223. Prototipo de RF43

En la figura 223 se muestra el prototipo para la GUI de listar empleado justificación del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

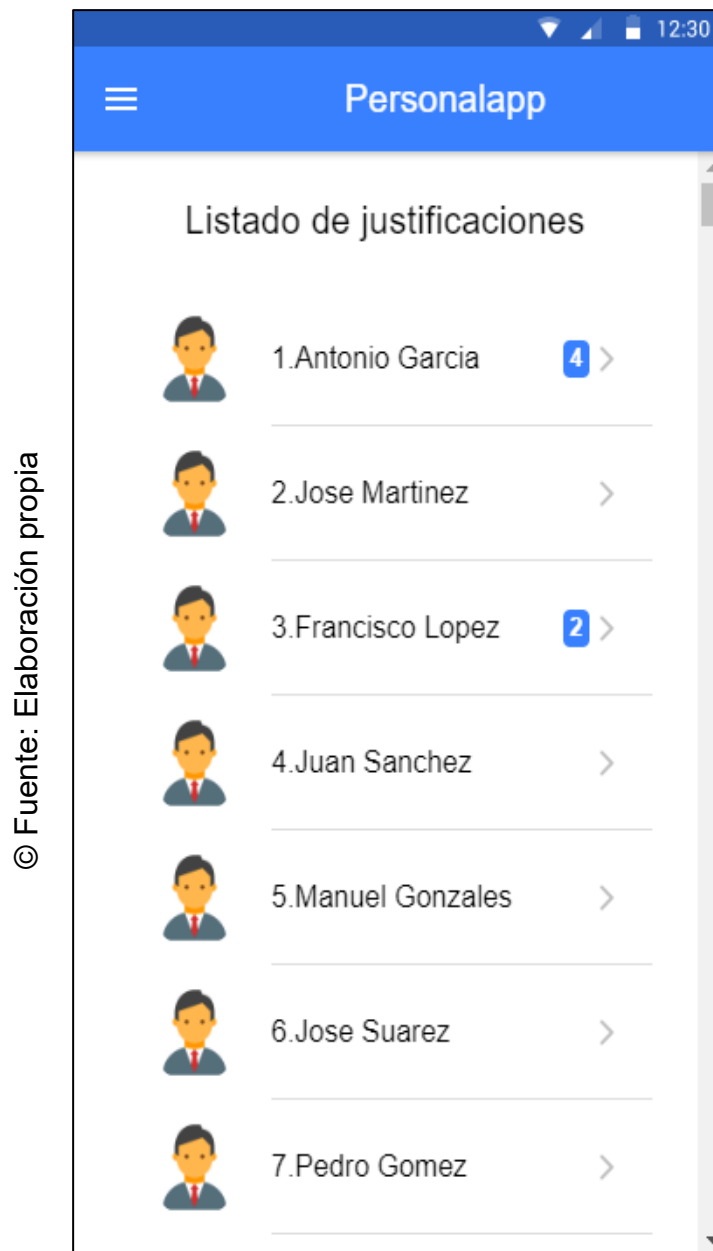


Figura 224 Interface de RF43

En la figura 224 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadJustificationFull(Request $request){
    $employees = Employee::with( relations: 'justifications')->whereIn( column: 'role_id', [1,3])->get();
    //return $employees;
    foreach ($employees as $employee){
        $contador = 0;
        foreach($employee->justifications as $justification){
            if($justification->status == 0){
                $contador = $contador + 1;
                $employee->total_justifications = $contador;
            }
        }
    }
    return $employees;
}
```

Figura 225. Código de RF43

En esta imagen 225 de la presente tesis se puede observar el código fuente del requerimiento funcional 43.

Requerimiento RF44 – Seleccionar empleado justificación

El requerimiento 44, tiene como objetivo que en la aplicación se debe poder seleccionar al empleado que tenga o no una justificación, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

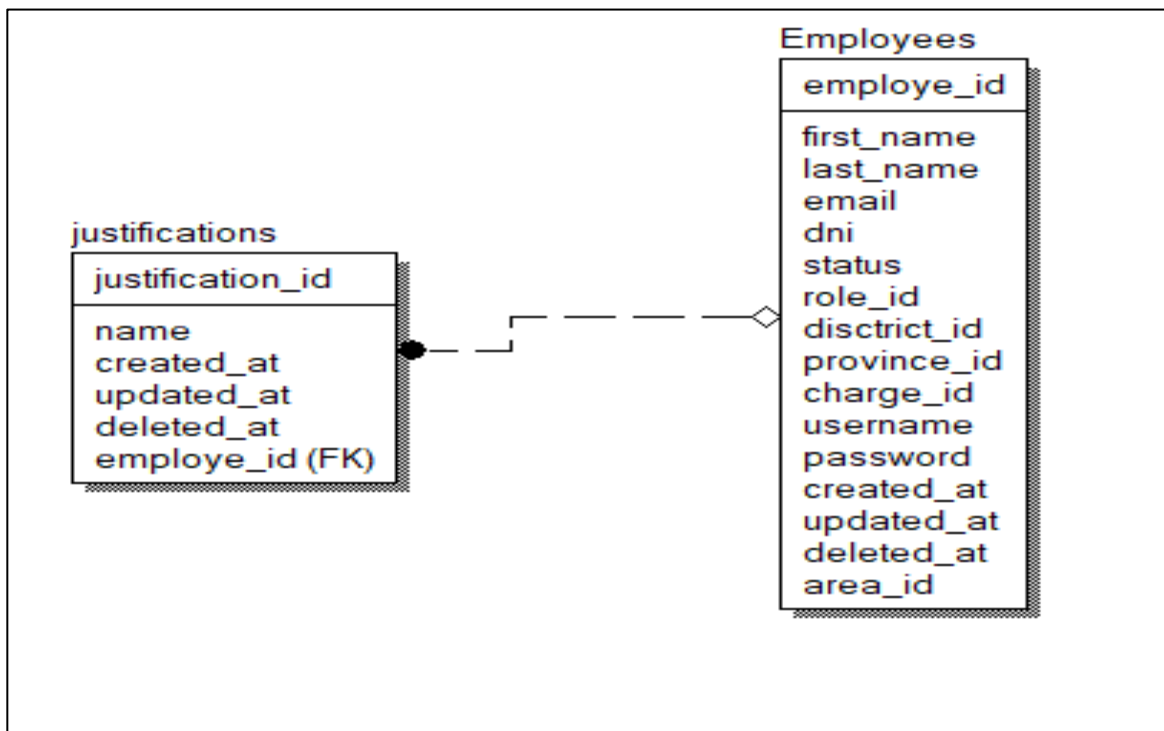


Figura 226. Diagrama lógico de la base de datos RF44

© Fuente: Elaboración propia

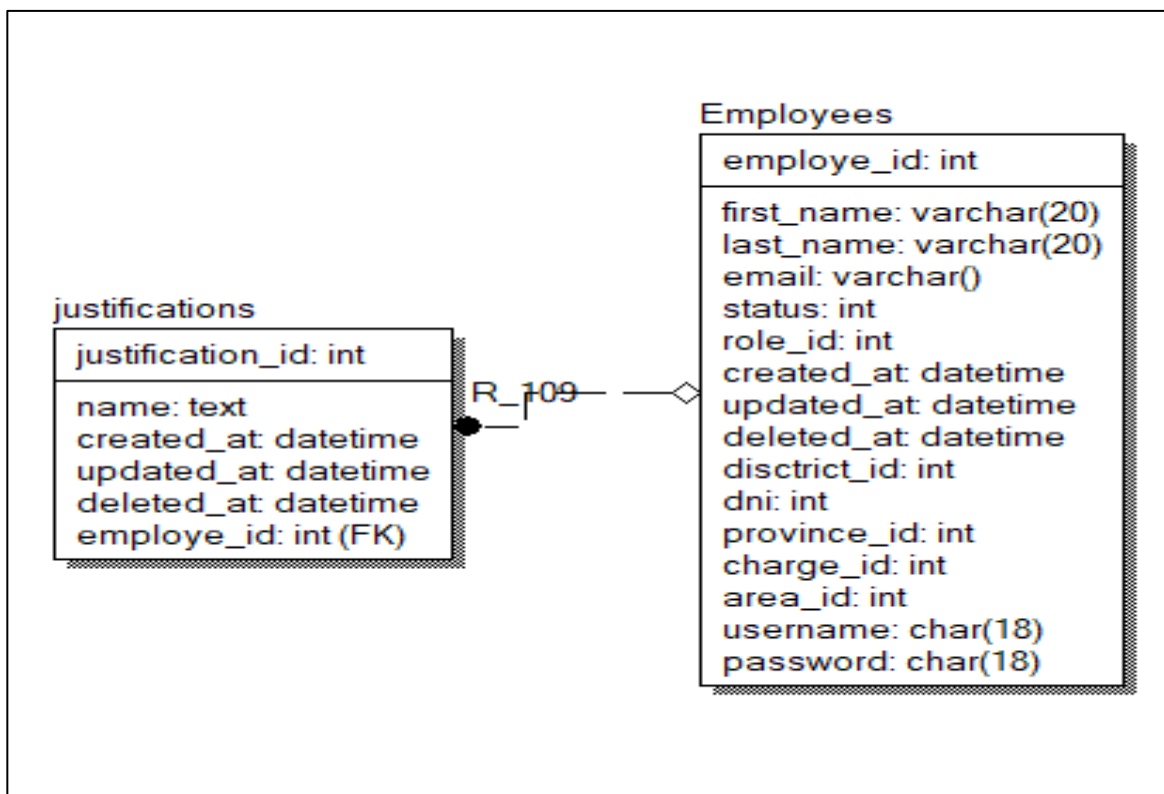


Figura 227. Diagrama físico de la base de datos RF44

Diseño

© Fuente: Elaboración propia

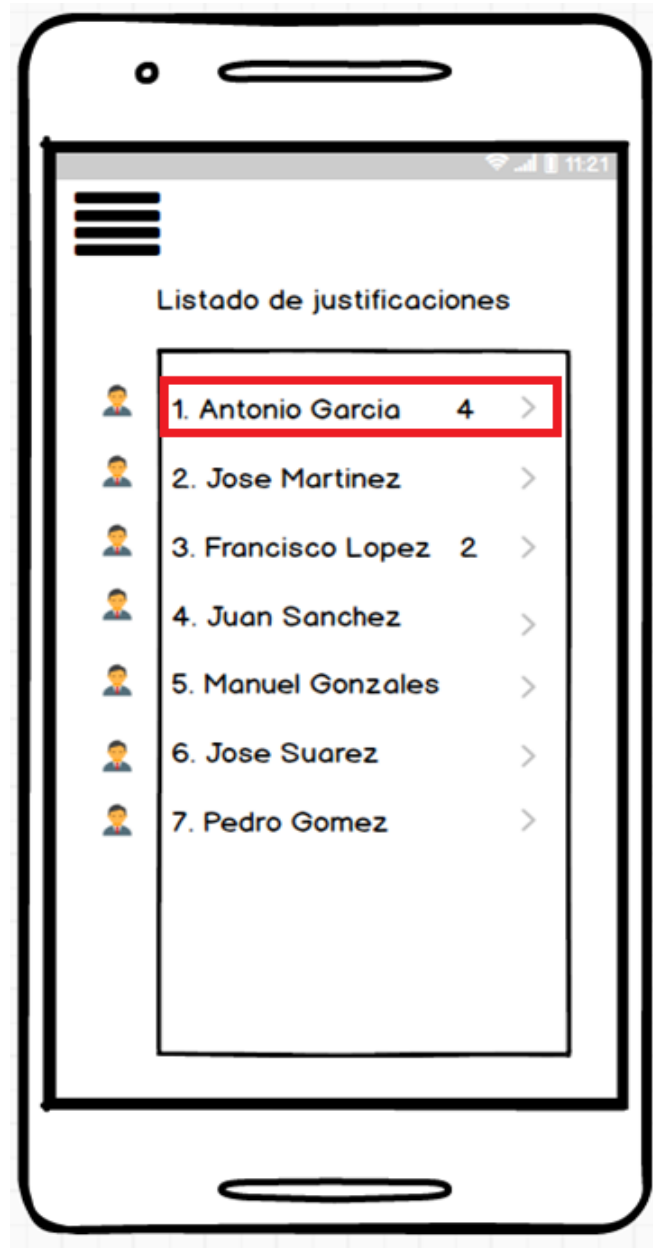


Figura 228. Prototipo de RF44

En la figura 228 se muestra el prototipo para la GUI de seleccionar empleado justificación del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

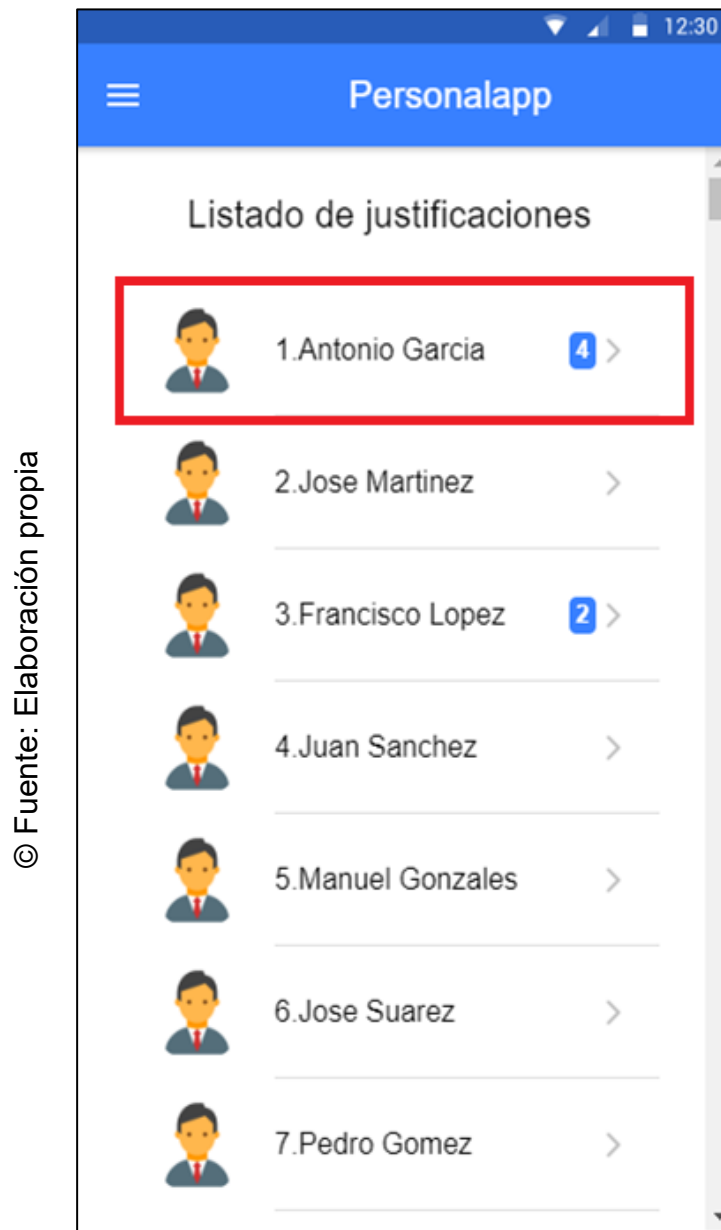


Figura 229. Interface de RF44

En la figura 229 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function justificationByEmployee($id) {  
    $justifications = Justification::where('employee_id',$id)->get();  
    return $justifications;  
}
```

Figura 230. Código de RF44

En esta imagen 230 de la presente tesis se puede observar el código fuente del requerimiento funcional 44.

Requerimiento RF45 – Listar justificación

El requerimiento 45, tiene como objetivo que en la aplicación se debe poder ver el listado de las justificaciones de los empleados de la organización, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero”, y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 45).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

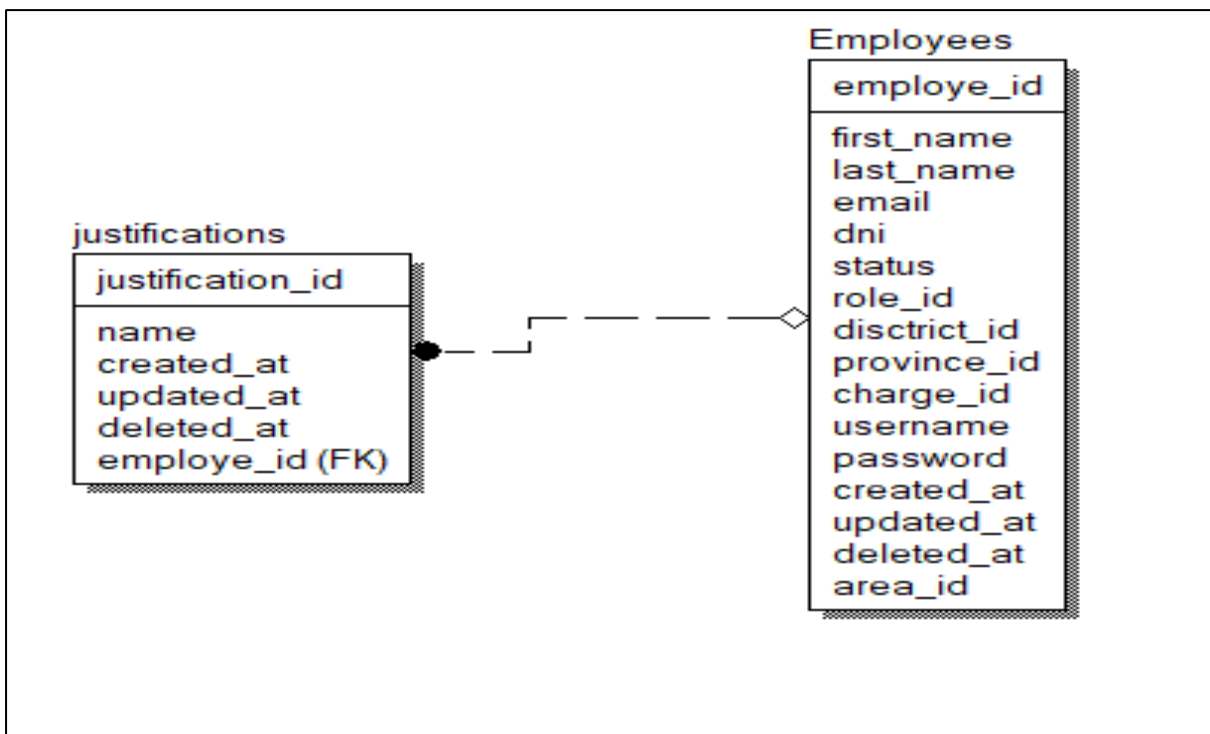


Figura 231. Diagrama lógico de la base de datos RF45

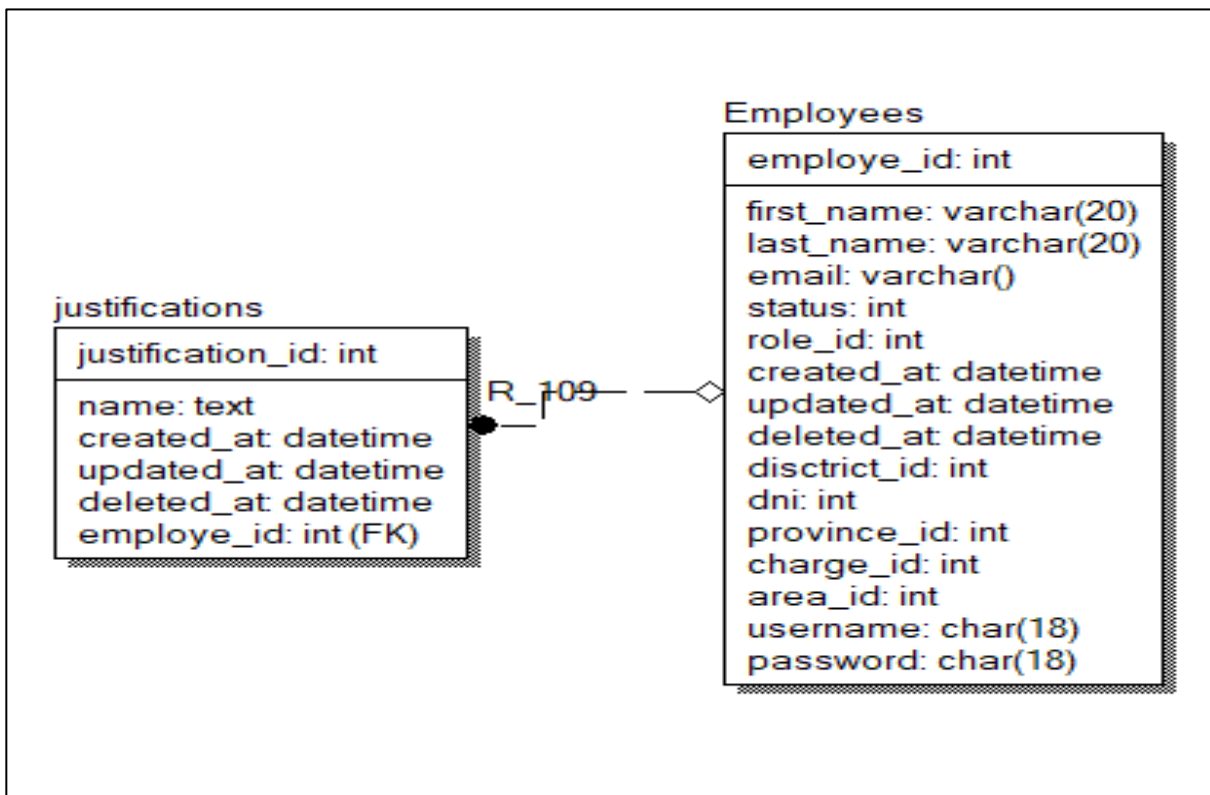


Figura 232. Diagrama físico de la base de datos RF45

Diseño

© Fuente: Elaboración propia

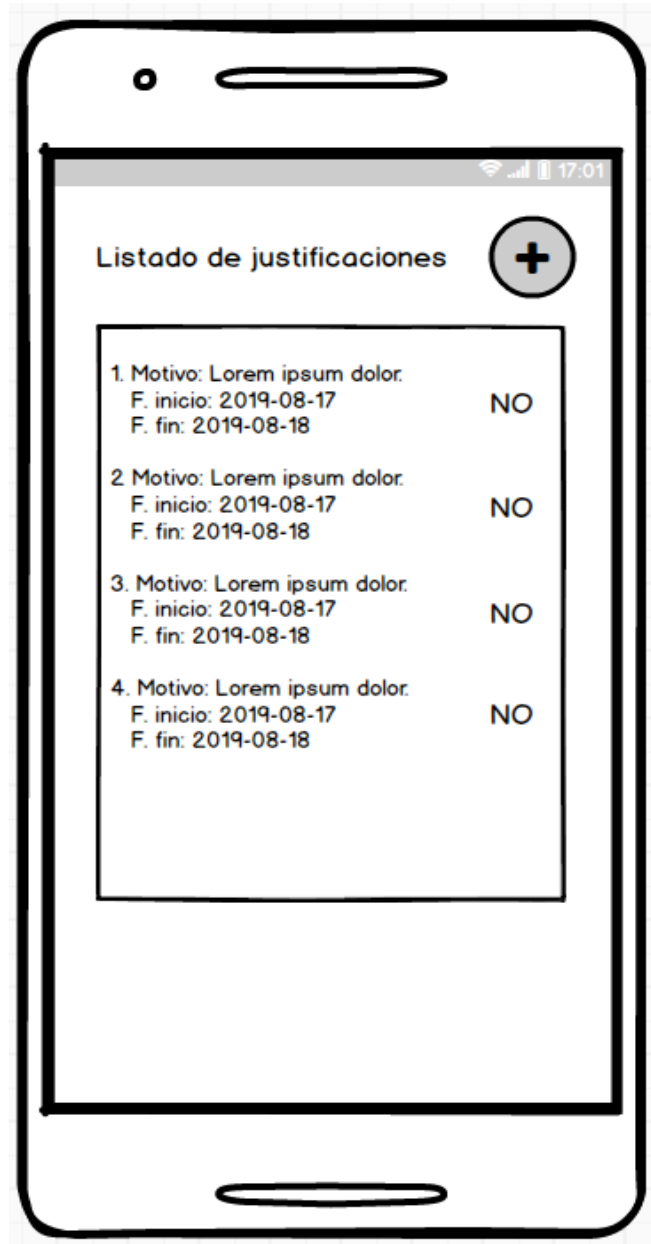


Figura 233. Prototipo de RF45

En la figura 233 se muestra el prototipo para la GUI de listar justificación de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

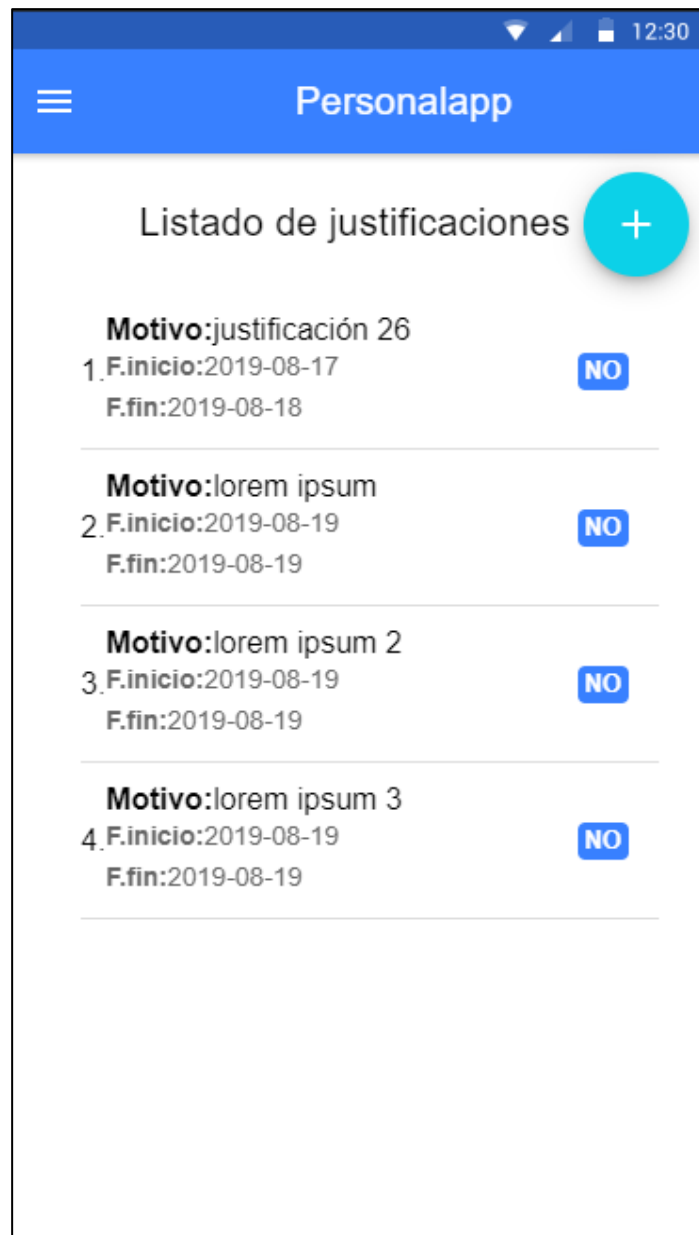


Figura 234. Interface de RF45

En la figura 234 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadJustificationsByEmployee(Request $request) {  
    $justification = Justification::where('employee_id', $request->all())->get();  
    return $justification;  
}
```

Figura 235. Código de RF45

En esta imagen 235 de la presente tesis se puede observar el código fuente del requerimiento funcional 45.

Requerimiento RF46 – Aprobar justificación

El requerimiento 46, tiene como objetivo que en la aplicación se debe poder aprobar las justificaciones de los empleados, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero”, y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 46).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

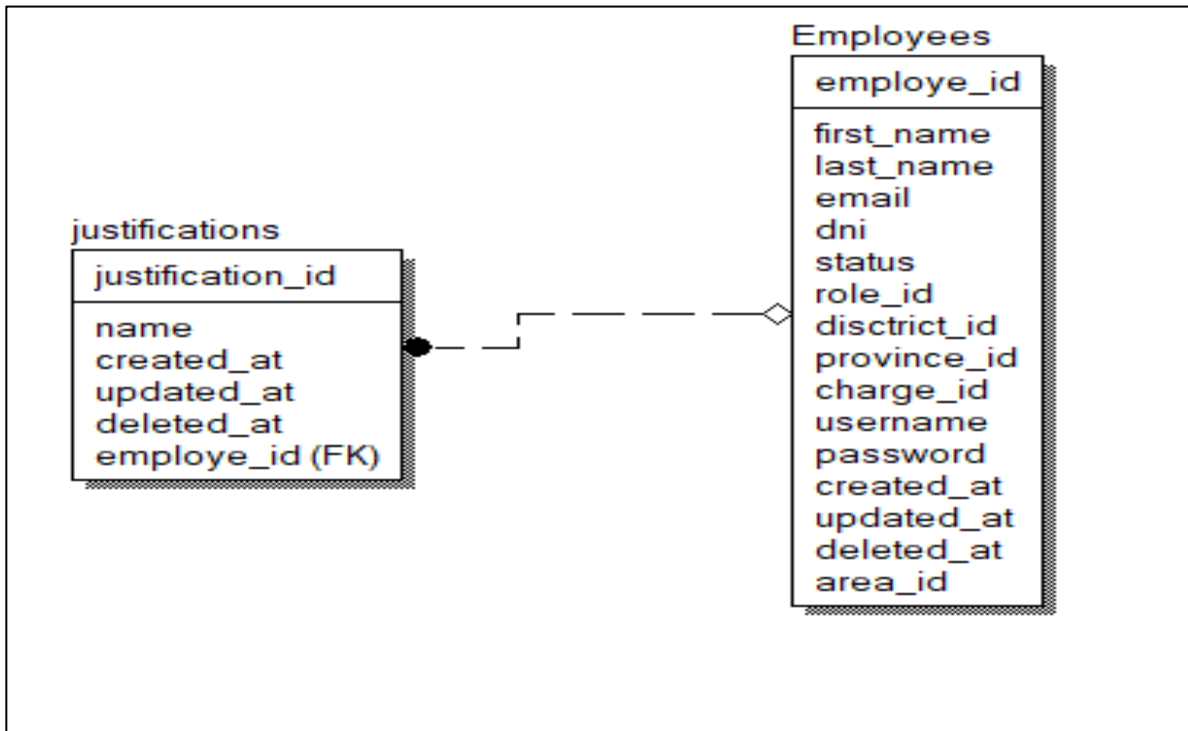


Figura 236. Diagrama lógico de la base de datos RF46

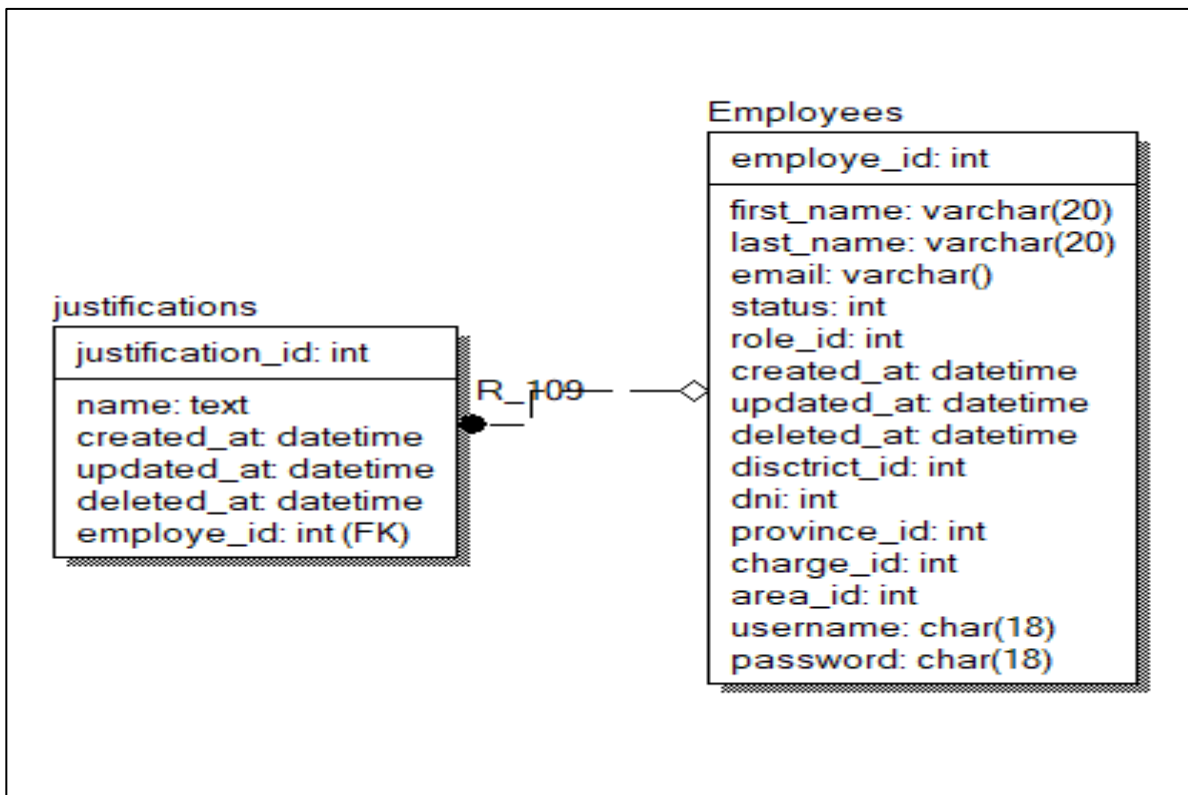


Figura 237. Diagrama físico de la base de datos RF46

Diseño

© Fuente: Elaboración propia

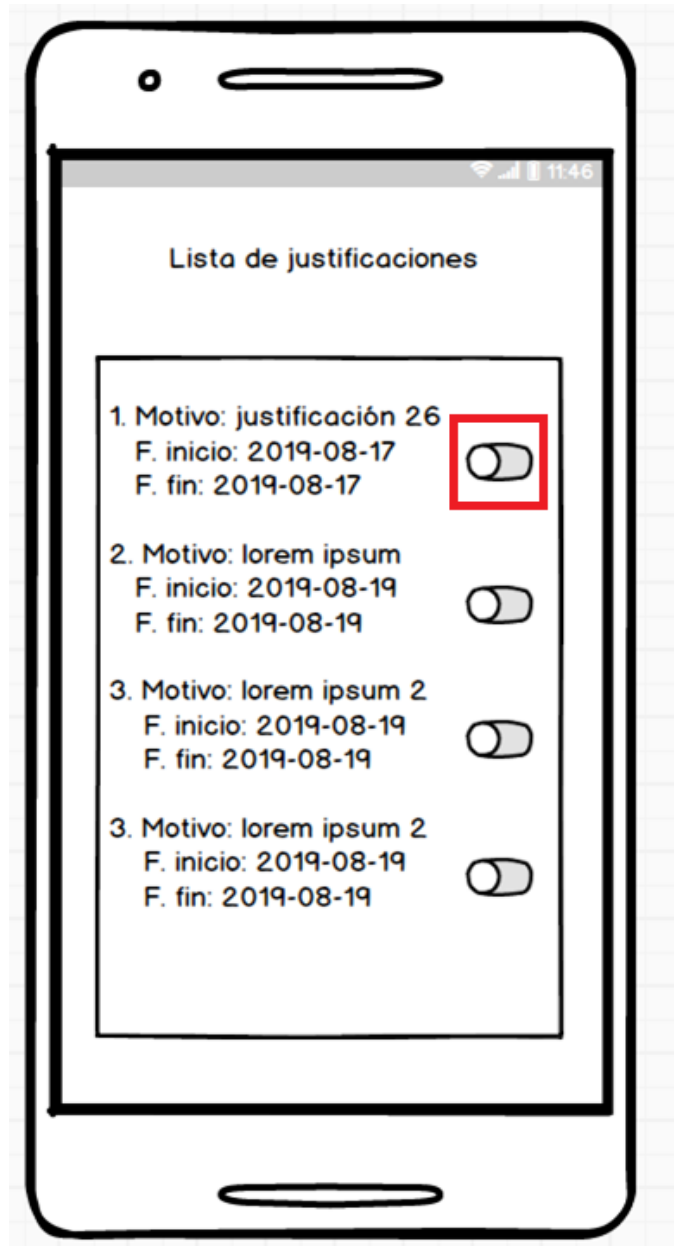


Figura 238. Prototipo de RF46

En la figura 238 se muestra el prototipo para la GUI de aprobar justificación del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

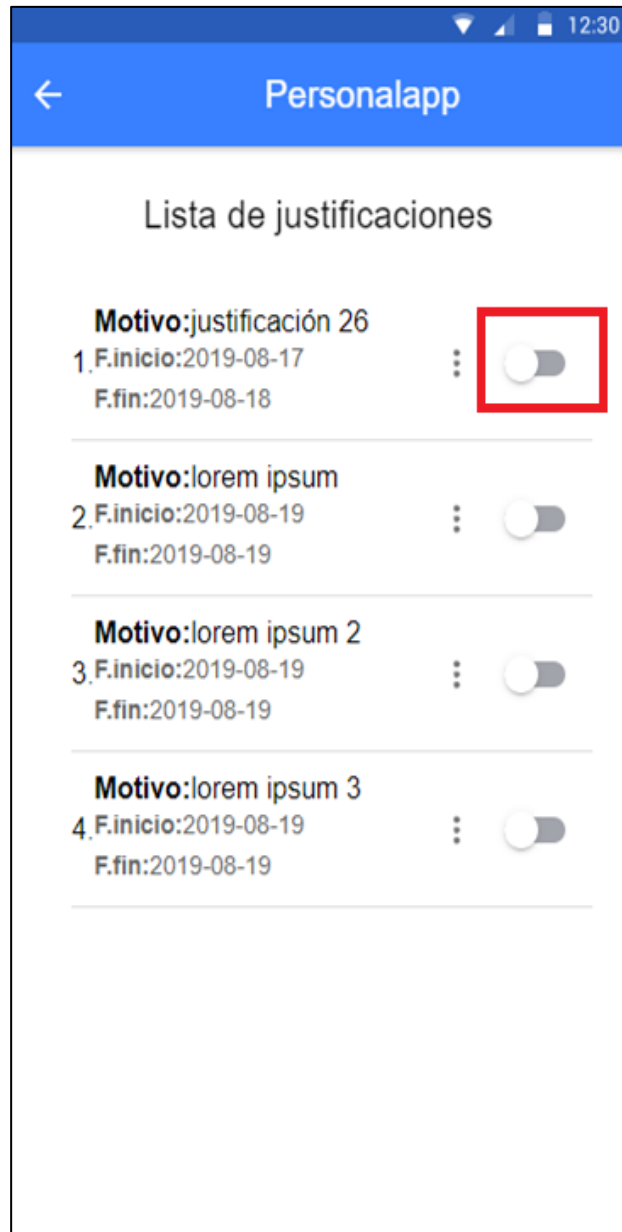


Figura 239. Interface de RF46

En la figura 239 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function statusJustification(Request $request) {  
    $justification = Justification::findOrFail($request->id);  
    $justification->status = $request->status;  
    $justification->save();  
    return $justification;  
}
```

Figura 240. Código de RF46

En esta imagen 240 de la presente tesis se puede observar el código fuente del requerimiento funcional 46.

Requerimiento RF47 – Crear justificación

El requerimiento 47, tiene como objetivo que en la aplicación se debe poder crear una justificación, para que el área de recursos humanos pueda visualizarlo, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 47).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

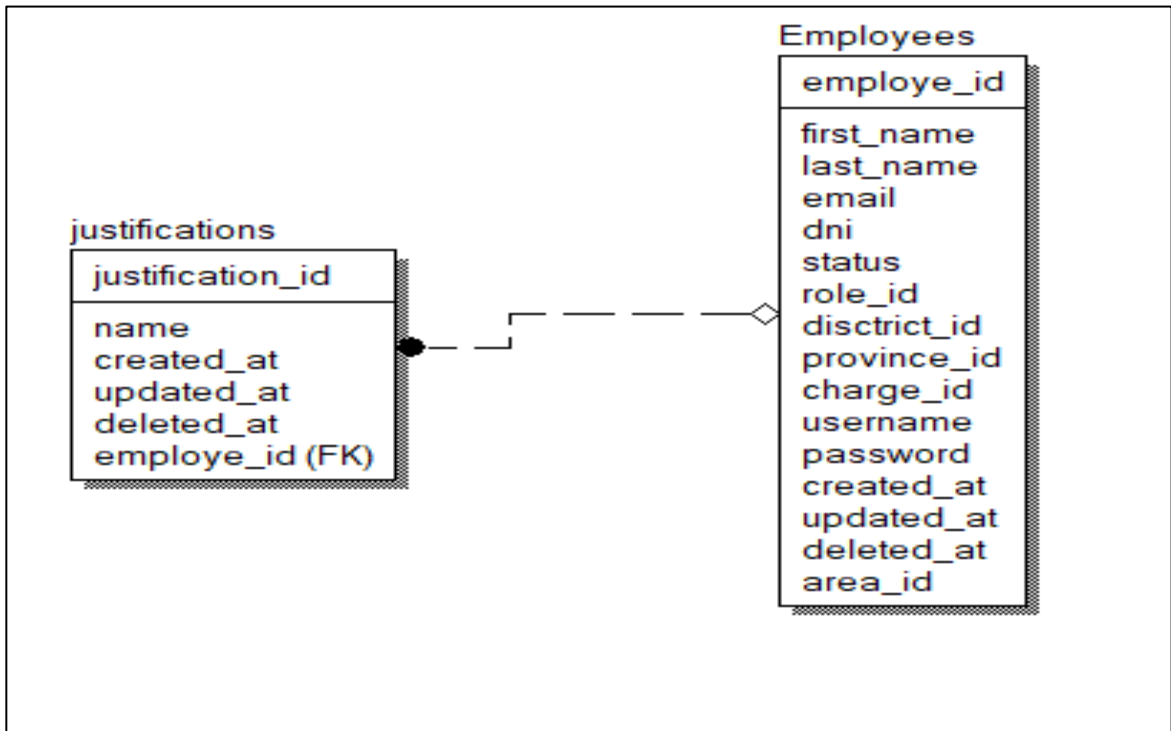


Figura 241. Diagrama lógico de la base de datos RF47

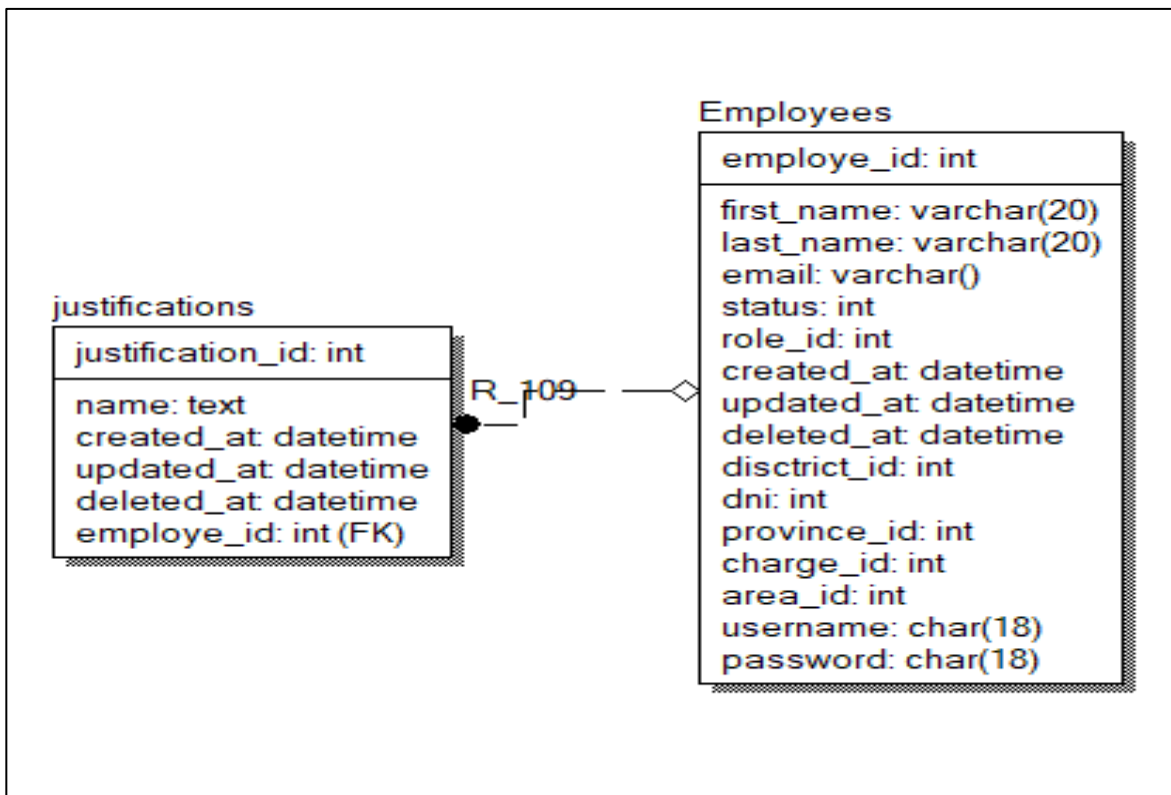


Figura 242. Diagrama físico de la base de datos RF47

Diseño

© Fuente: Elaboración propia

El prototipo muestra una pantalla de un teléfono móvil con el título "Crear justificación". Debajo del título hay un campo de texto etiquetado "Descripción". A continuación, se encuentran dos campos de fecha: "Fecha de inicio" con el valor "2019/08/19" y "Fecha de fin" con el valor "2019/08/25", cada uno acompañado de un ícono de calendario. Al final de la pantalla hay un botón que dice "CREAR JUSTIFICACIÓN".

Figura 243. Prototipo de RF47

En la figura 243 se muestra el prototipo para la GUI de crear justificación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. At the top, there is a blue header bar with a white back arrow on the left and the text 'Personalapp' in the center. Below the header, the main content area is white and contains the following elements: the title 'Crear justificacion' centered; a text input field labeled 'Descripción'; a text input field labeled 'Fecha inicio'; a text input field labeled 'Fecha fin'; and a large blue button at the bottom with the text 'CREAR JUSTIFICACIÓN' in white capital letters. The status bar at the top right of the phone shows the time as 12:30.

Figura 244. Interface de RF47

En la figura 244 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function createJustification(Request $request) {
    $date_init = $this->request->input( key: 'date_init');
    $date_end = $this->request->input( key: 'date_end');
    if($date_init > $date_end){
        return response()->json([
            'msg' => 'La fecha de inicio no puede ser menor a la fecha de fin.',
            'status: 200);
    }else{
        $mytime = Carbon::now( tz: 'America/Lima');
        $justification = new Justification();
        $justification->description = $request->description;
        $justification->status = 0;
        $justification->employee_id = $request->employee_id;
        $justification->date_init = $request->date_init;
        $justification->date_end = $request->date_end;
        $justification->save();
        return response()->json([
            'msg' => 'exito',
            'status: 200);
    }
}
```

Figura 245. Código de RF47

En esta imagen 245. de la presente tesis se puede observar el código fuente del requerimiento funcional 47.

Requerimiento RF48 – Actualizar justificación

El requerimiento 48, tiene como objetivo que en la aplicación se debe poder actualizar sus justificaciones, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 48).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

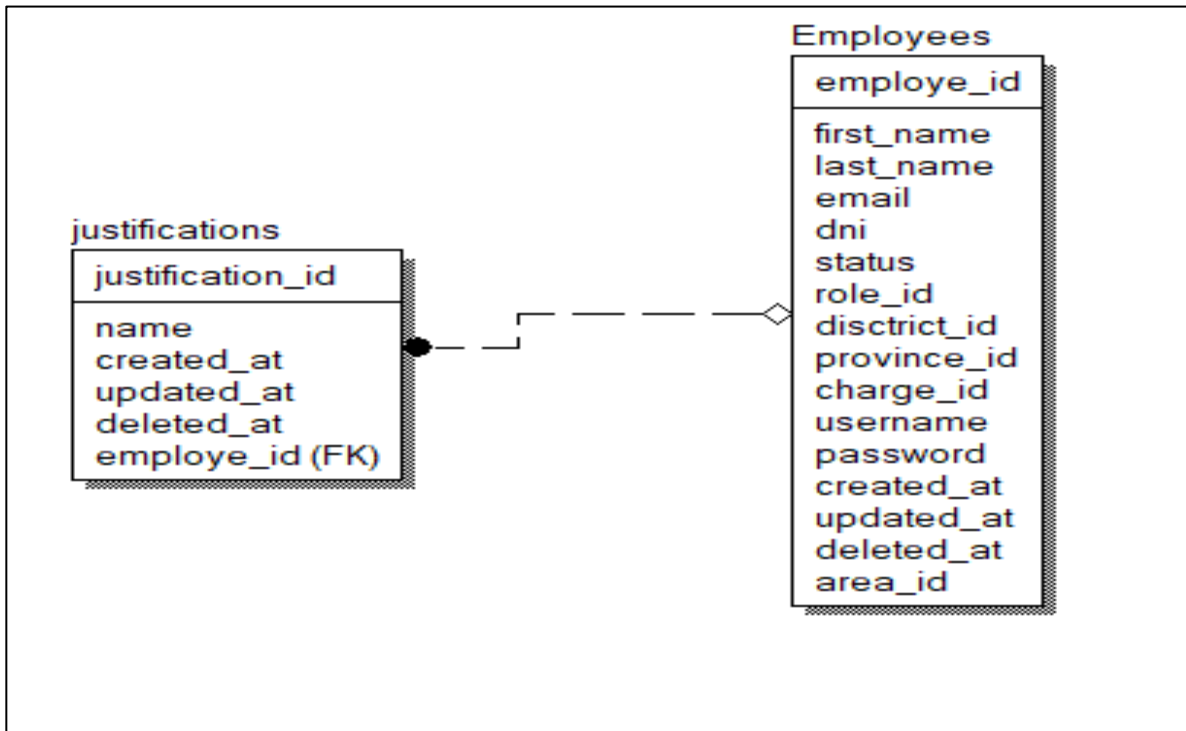


Figura 246. Diagrama lógico de la base de datos RF48

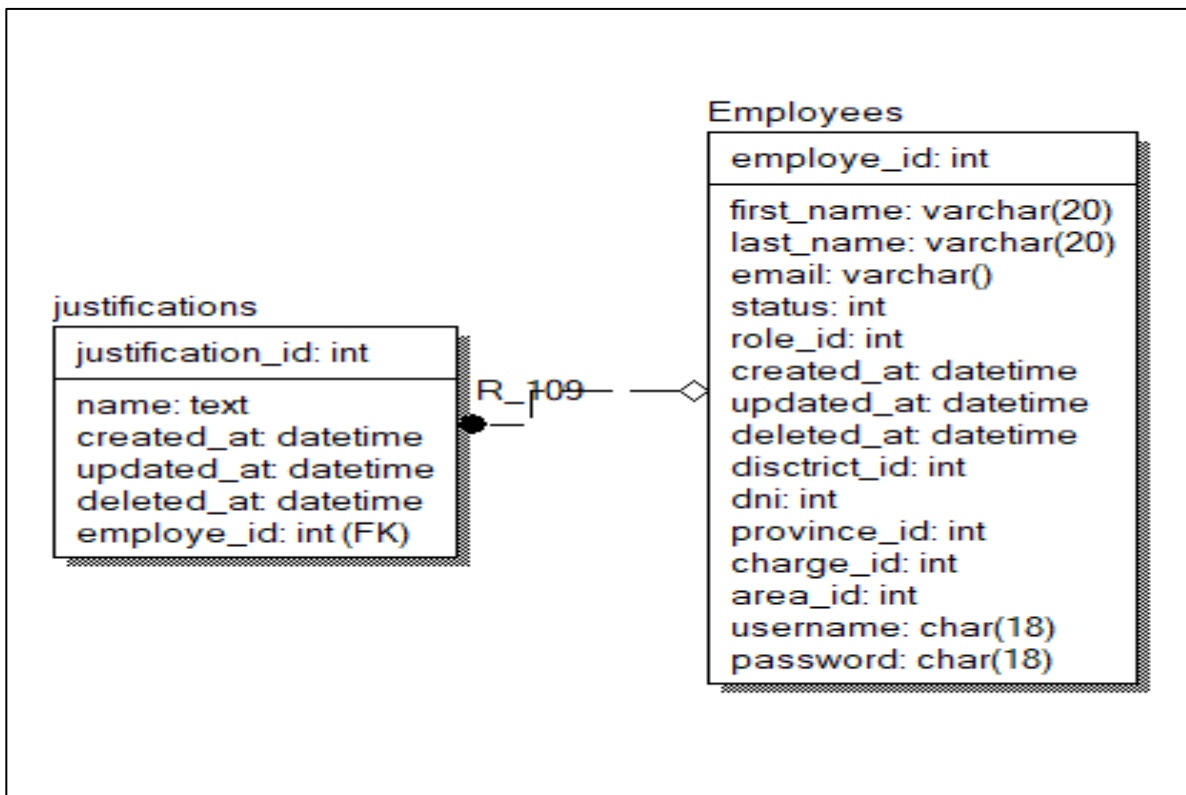


Figura 247. Diagrama físico de la base de datos RF48

Diseño

© Fuente: Elaboración propia

Actualizar justificación

Descripción

lorem ipsum

Fecha de inicio

2019/08/17

Fecha de fin

2019/08/19

ACTUALIZAR JUSTIFICACIÓN

Detailed description: The image shows a mobile application prototype for updating a justification. The screen is titled 'Actualizar justificación'. It features a text input field for 'Descripción' containing 'lorem ipsum'. Below this are two date selection fields: 'Fecha de inicio' with the value '2019/08/17' and 'Fecha de fin' with the value '2019/08/19'. Each date field has a calendar icon to its right. At the bottom of the form is a large button labeled 'ACTUALIZAR JUSTIFICACIÓN'. The entire form is enclosed in a rounded rectangular frame representing a smartphone screen, with a status bar at the top showing signal strength, Wi-Fi, and the time '10:41'.

Figura 248. Prototipo de RF48

En la figura 248 se muestra el prototipo para la GUI de actualizar justificación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación



Figura 249. Interface de RF48

En la figura 249 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateJustification(Request $request,$id) {
    $justification= Justification::find($id);
    $justification->description = $request->description;
    $justification->date_init = $request->date_init;
    $justification->date_end = $request->date_end;
    $justification->save();
    return response()->json([
        'msg' => 'exito',
    ], status: 200);
}
```

Figura 250. Código de RF48

En esta imagen 250 de la presente tesis se puede observar el código fuente del requerimiento funcional 48.

Requerimiento RF49 – Eliminar justificación

El requerimiento 49, tiene como objetivo que en la aplicación se debe poder eliminar sus justificaciones, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 49).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

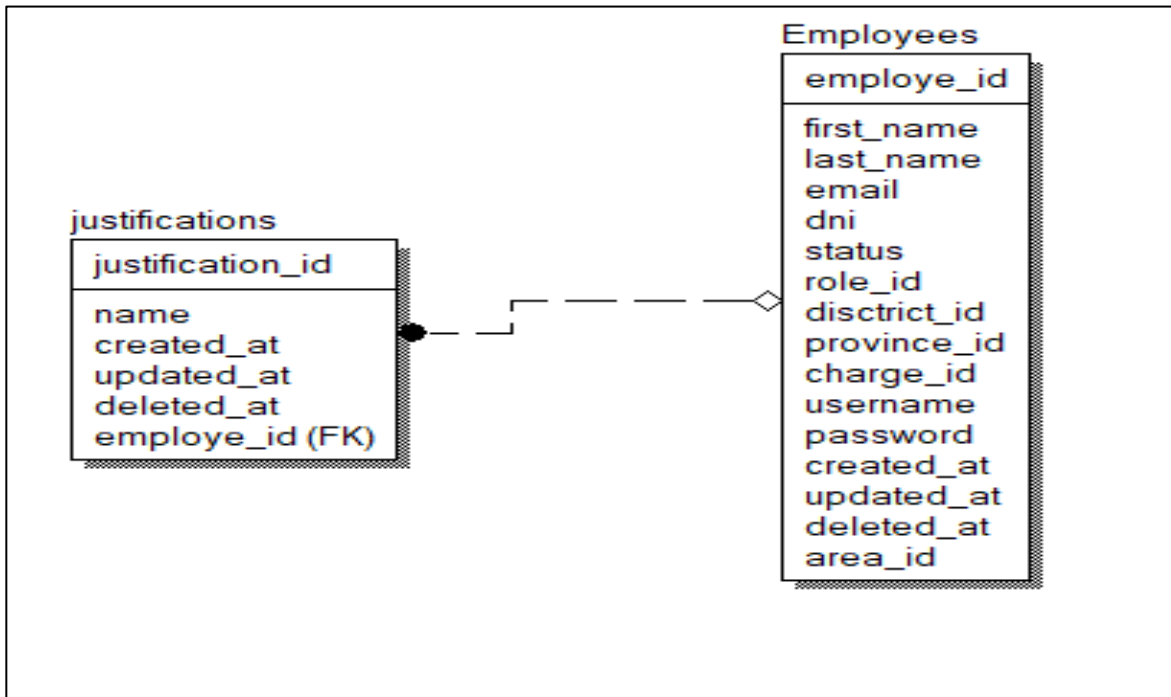


Figura 251. Diagrama lógico de la base de datos RF49

© Fuente: Elaboración propia

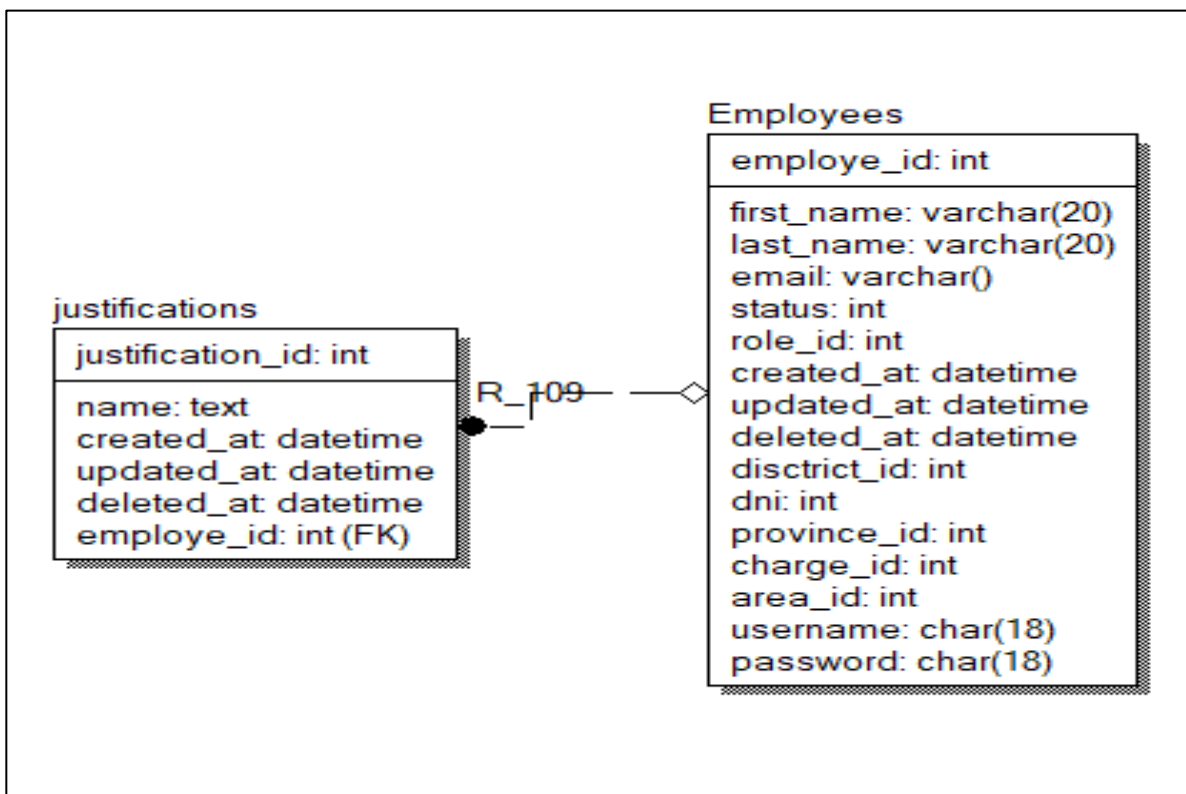


Figura 252. Diagrama físico de la base de datos RF49

Diseño

© Fuente: Elaboración propia

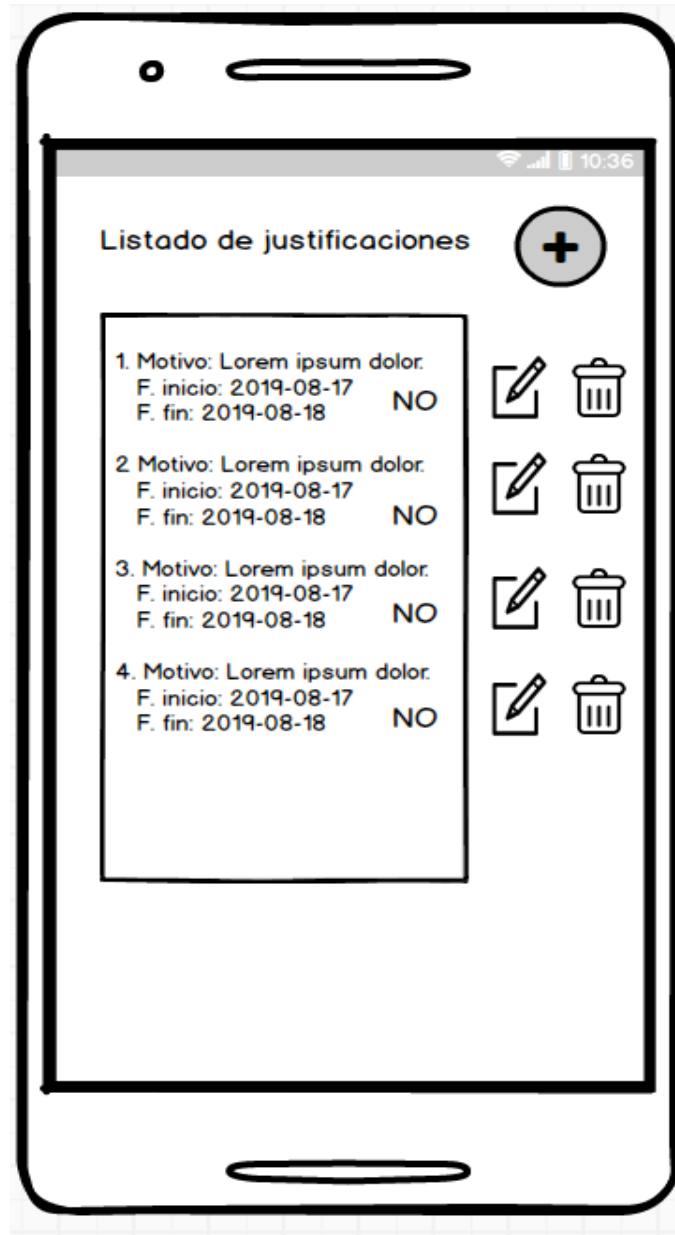


Figura 253. Prototipo de RF49

En la figura 253 se muestra el prototipo para la GUI de eliminar justificación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

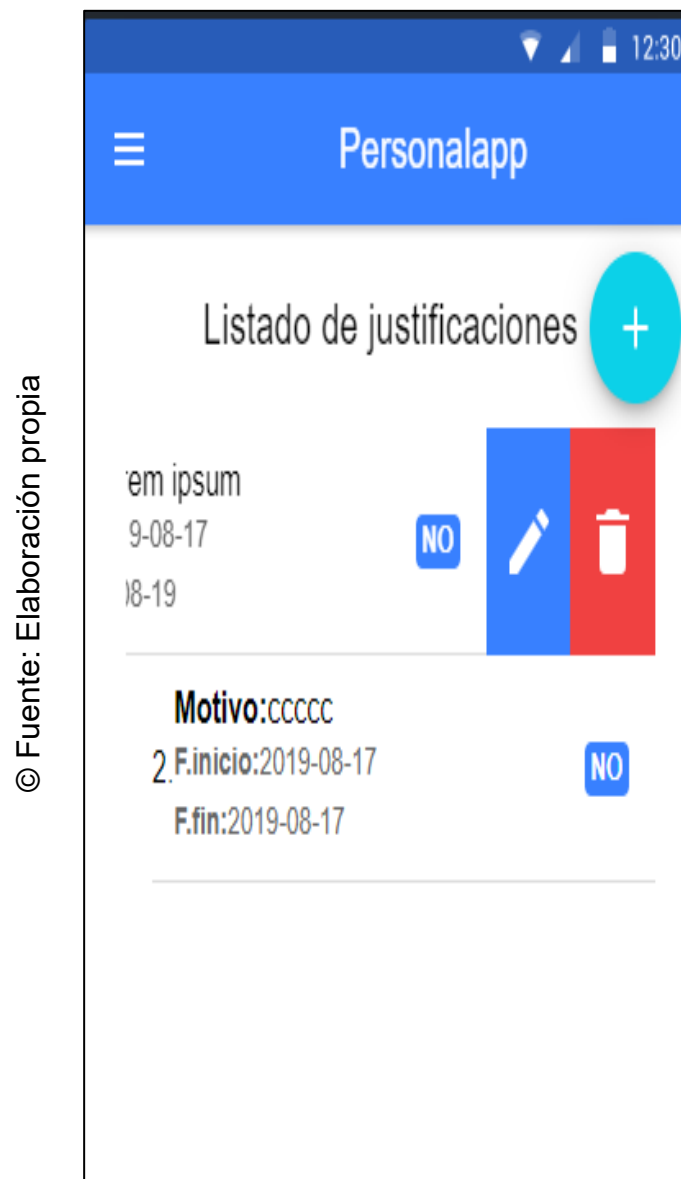


Figura 254. Interface de RF49

En la figura 254 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteJustification(Request $request) {  
    $justification = Justification::findOrFail($request->id);  
    $justification->delete();  
    return response()->json([  
        'msg' => 'exito',  
    ], status: 200);  
}
```

Figura 255. Código de RF49

En esta imagen 255 de la presente tesis se puede observar el código fuente del requerimiento funcional 49.

Requerimiento RF50 – Pulsar vacación

El requerimiento 50, tiene como objetivo que en la aplicación se debe poder pulsar la vacación en la barra lateral, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero” y “Recursos Humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 50).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

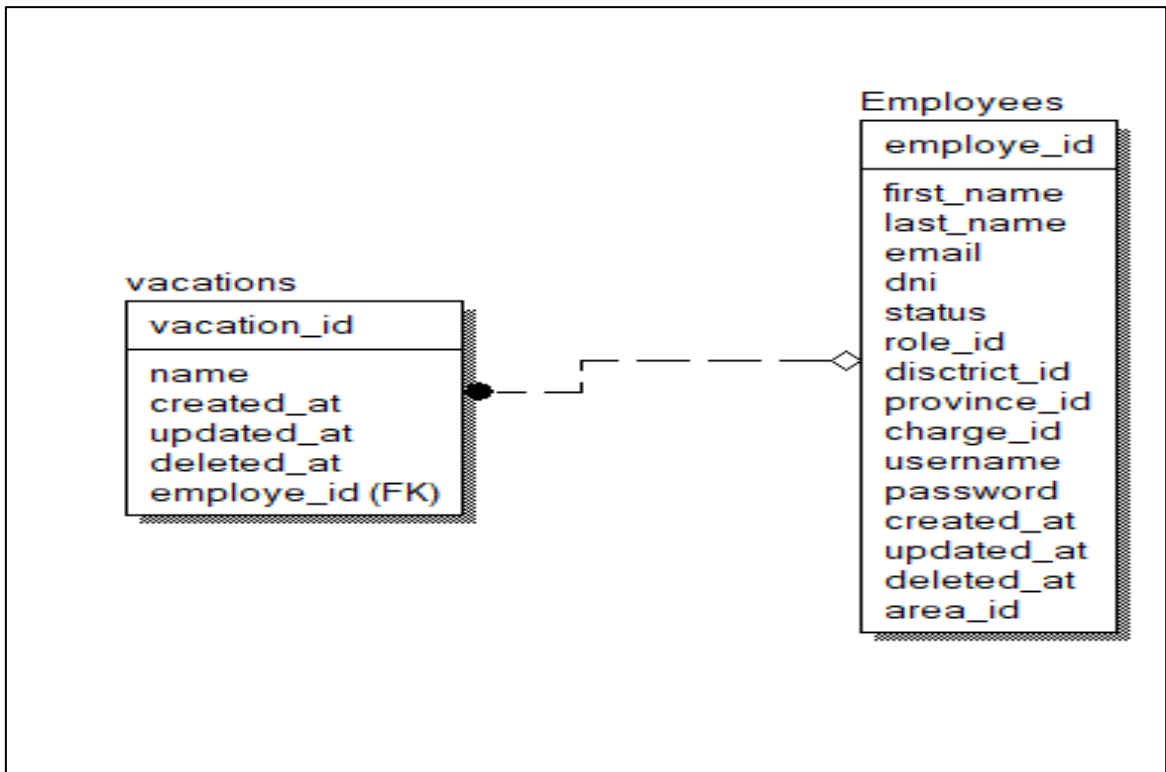


Figura 256. Diagrama lógico de la base de datos RF50

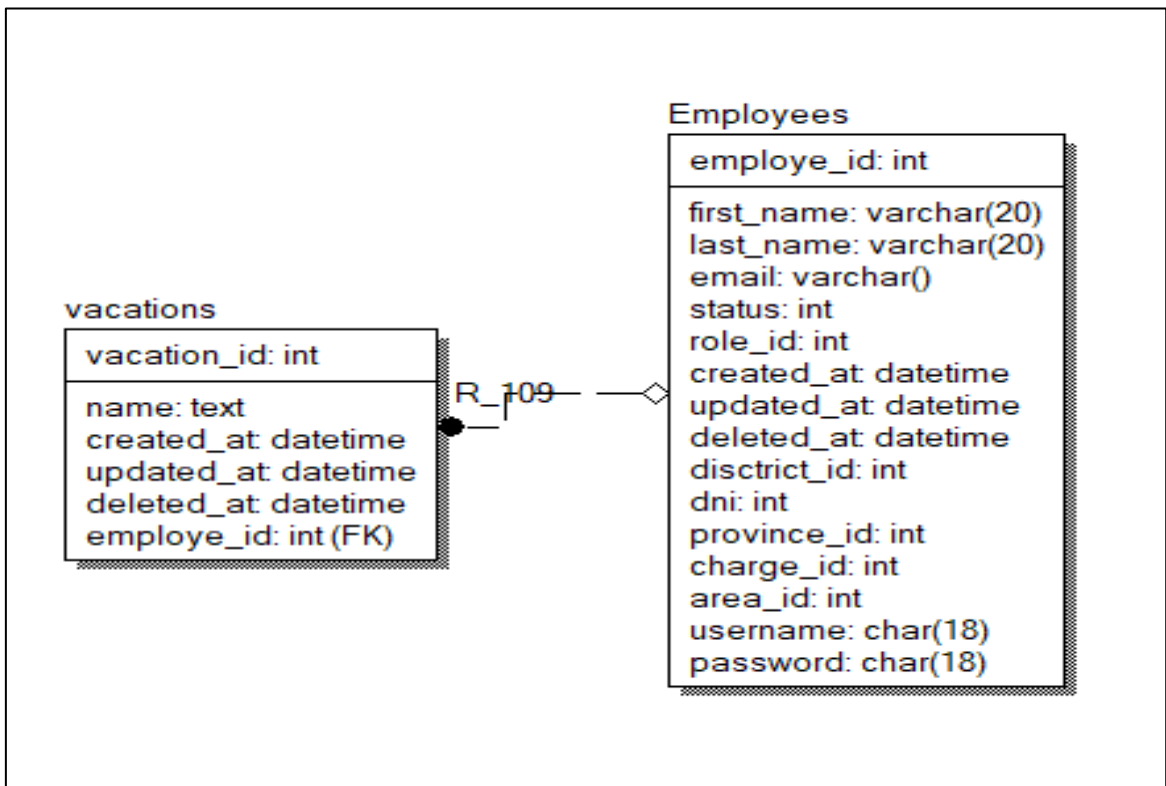


Figura 257. Diagrama físico de la base de datos RF50

Diseño

© Fuente: Elaboración propia

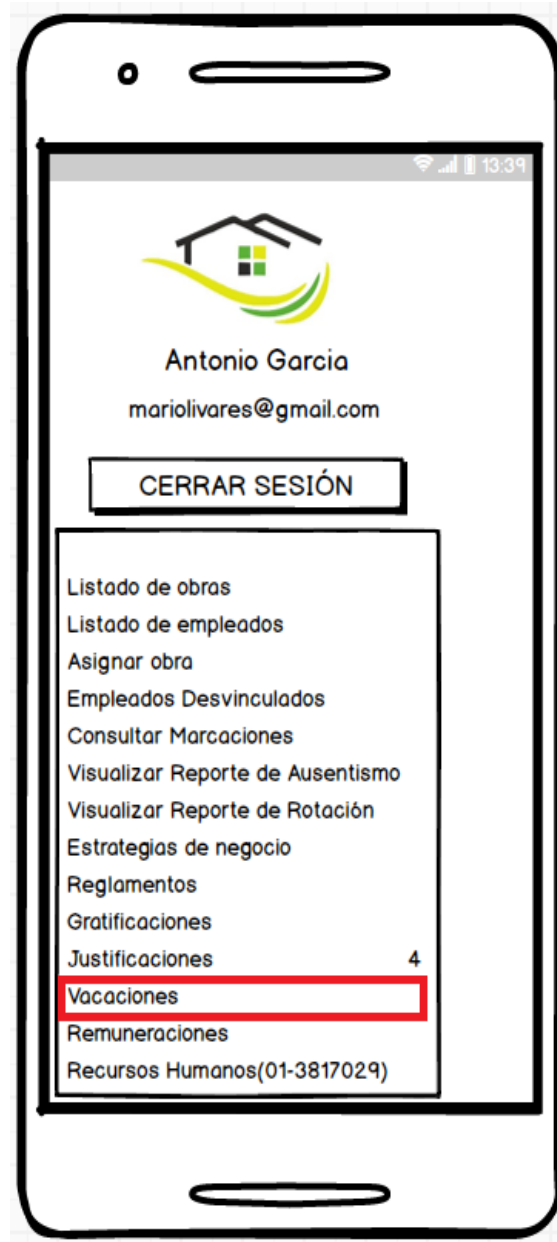


Figura 258. Prototipo de RF50

En la figura 258 se muestra el prototipo para la GUI de pulsar vacación de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

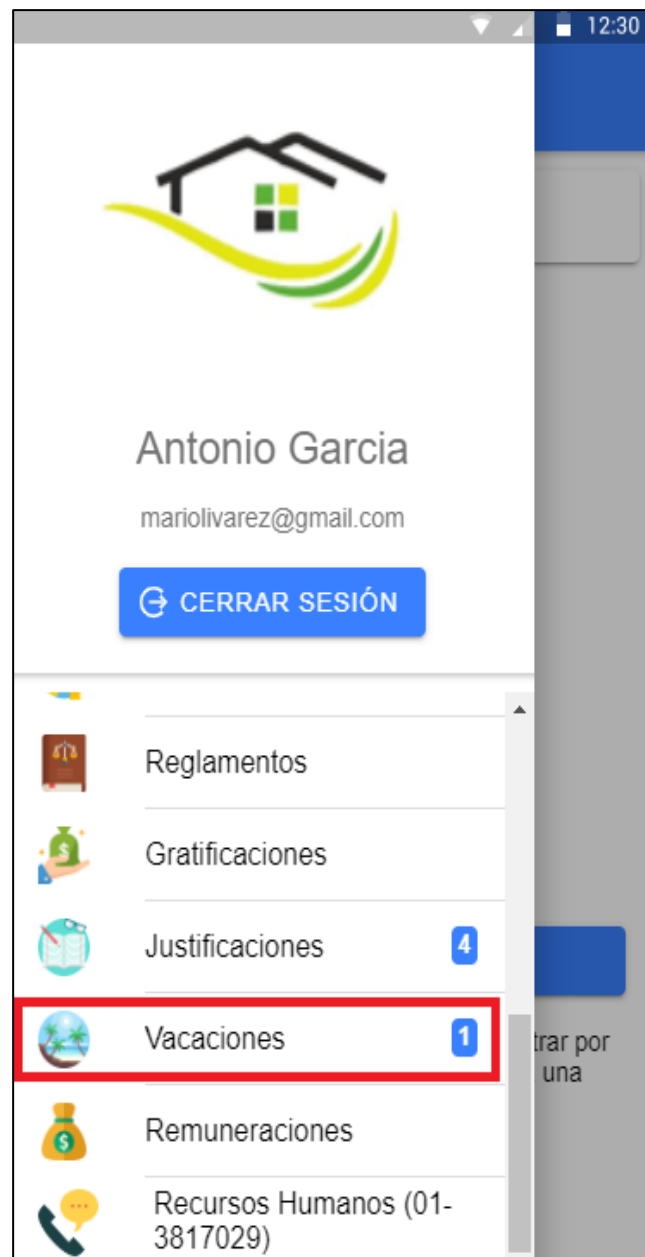


Figura 259. Interface de RF50

En la figura 259 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
if(res == 1){
  this.pages = [
    { title: 'Listado de obras', page: '/workcontroller', icon:'git-network' },
    { title: 'Listado de empleados', page: '/employeecontroller', icon:'git-network' },
    { title: 'Asignar Obra', page: '/assignwork', icon:'git-network' },
    { title: 'Empleados Desvinculados', page: '/dismissal', icon:'contacts' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Visualizar estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 2){
  this.pages = [
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 3){
  this.pages = [
    { title: 'Visualizar Obra Asignada', page: '/workassign', icon:'briefcase' },
    { title: 'Consultar Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Visualizar estrategias de negocio', page: '/viewnegotiations', icon:'briefcase' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}
```

Figura 260. Código de RF50

En esta imagen 260 de la presente tesis se puede observar el código fuente del requerimiento funcional 50.

Requerimiento RF51 – Listar empleado vacación

El requerimiento 51, tiene como objetivo que en la aplicación se debe poder ver el listado de los empleados que tengan o no una vacación, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 51).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

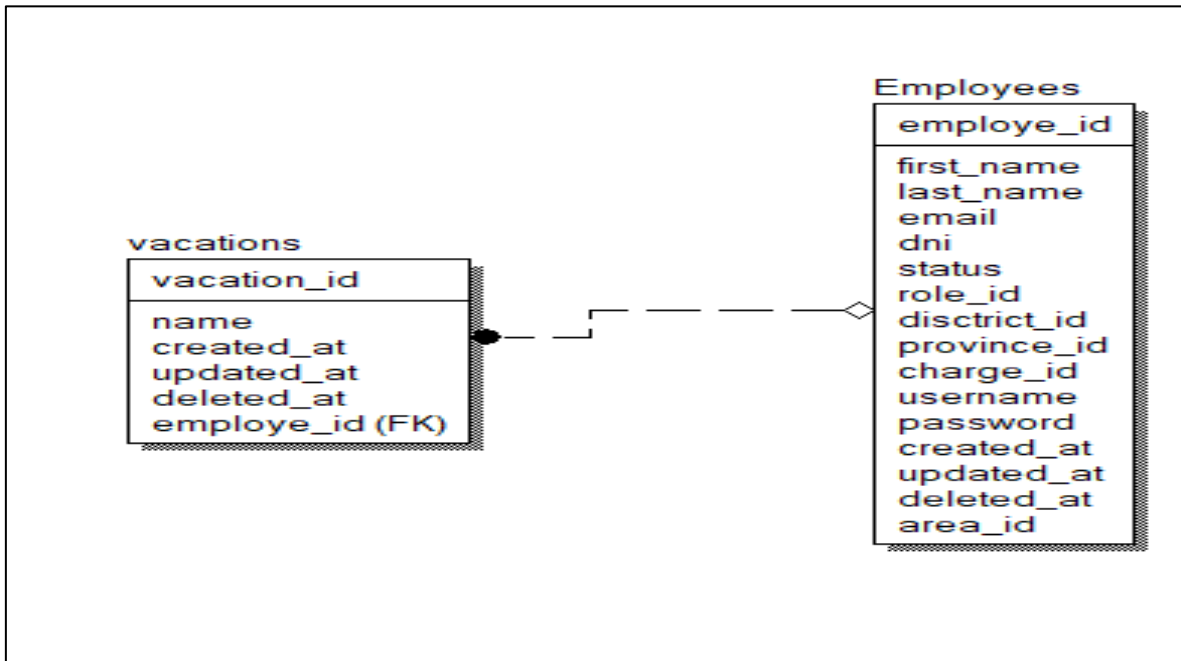


Figura 261. Diagrama lógico de la base de datos RF51

© Fuente: Elaboración propia

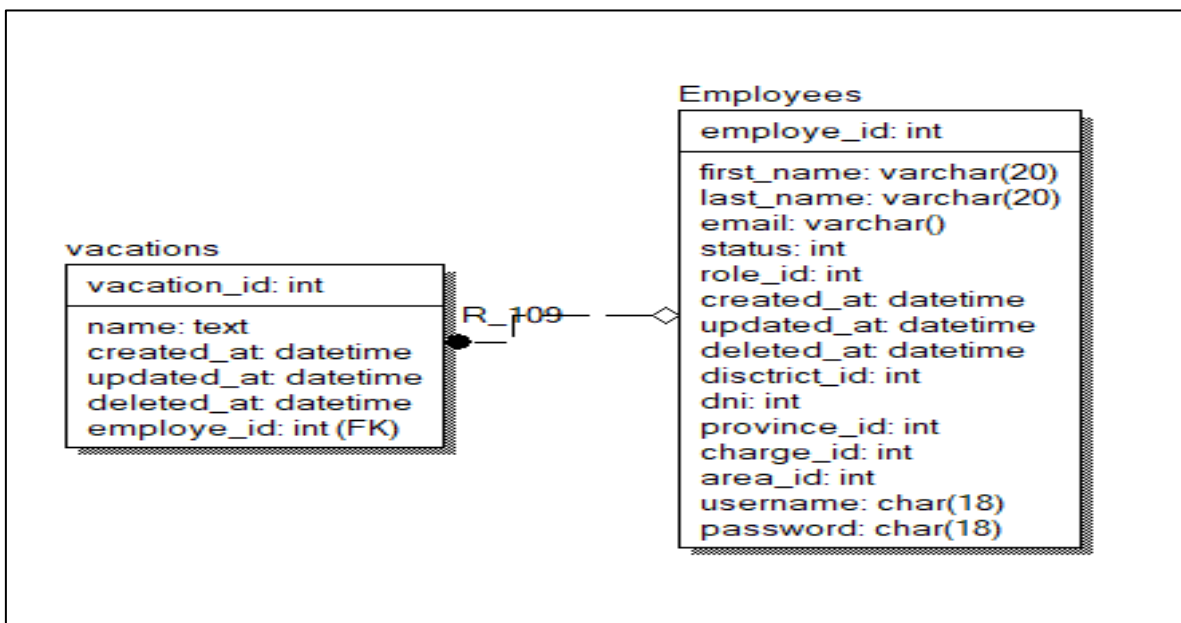


Figura 262. Diagrama físico de la base de datos RF51

Diseño

© Fuente: Elaboración propia



Figura 263. Prototipo de RF51

En la figura 263 se muestra el prototipo para la GUI de listar empleado vacación del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

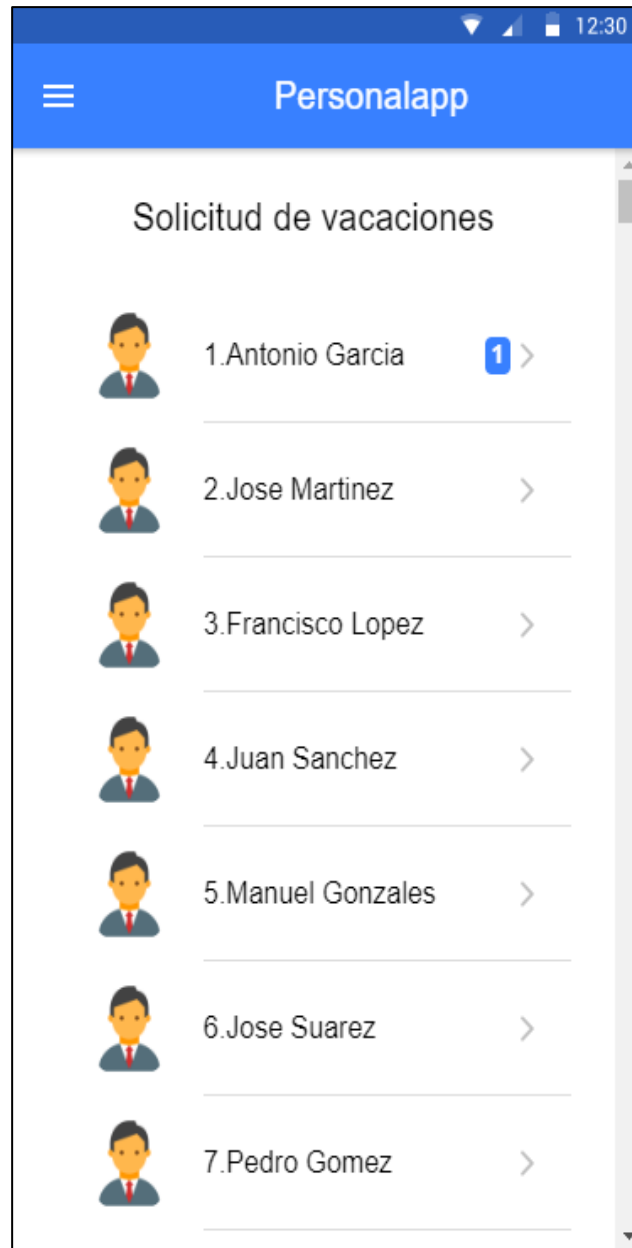


Figura 264. Interface de RF51

En la figura 264 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadDataListVacations(Request $request){
    $role_id = $request->role_id;
    $user_id = $request->user_id;
    if($role_id == 1 or $role_id == 3){
        $vacations = Vacation::where('employee_id',$user_id)->get();
        return $vacations;
    }else{
        $employees = Employee::with( relations: 'vacations')->whereIn( column: 'role_id',[1,3])->get();
        //return $employees;
        foreach ($employees as $employee){
            $contador = 0;
            foreach($employee->vacations as $vacation){
                if($vacation->status == 0){
                    $contador = $contador + 1;
                    $employee->total_vacations = $contador;
                }
            }
        }
        return $employees;
    }
}
```

Figura 265. Código de RF51

En esta imagen 265 de la presente tesis se puede observar el código fuente del requerimiento funcional 51.

Requerimiento RF52 – Seleccionar empleado vacación

El requerimiento 52, tiene como objetivo que en la aplicación se debe poder ver y seleccionar a los empleados que tengan o no una vacación, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 52).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

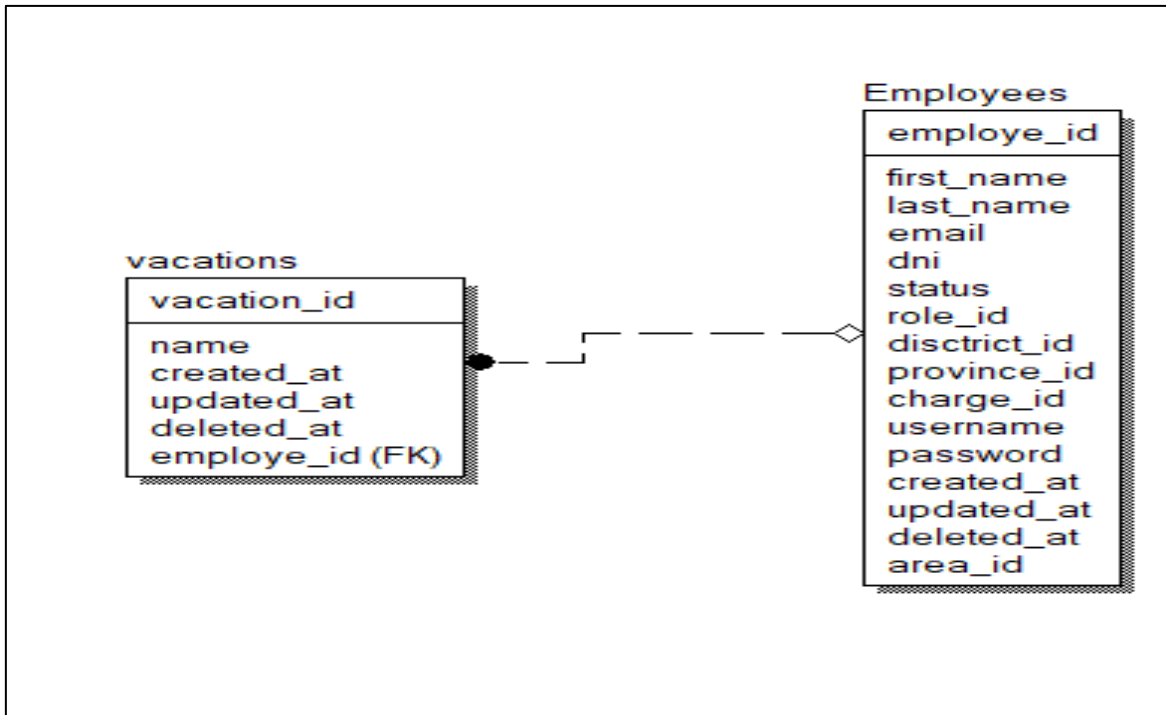


Figura 266. Diagrama lógico de la base de datos RF52

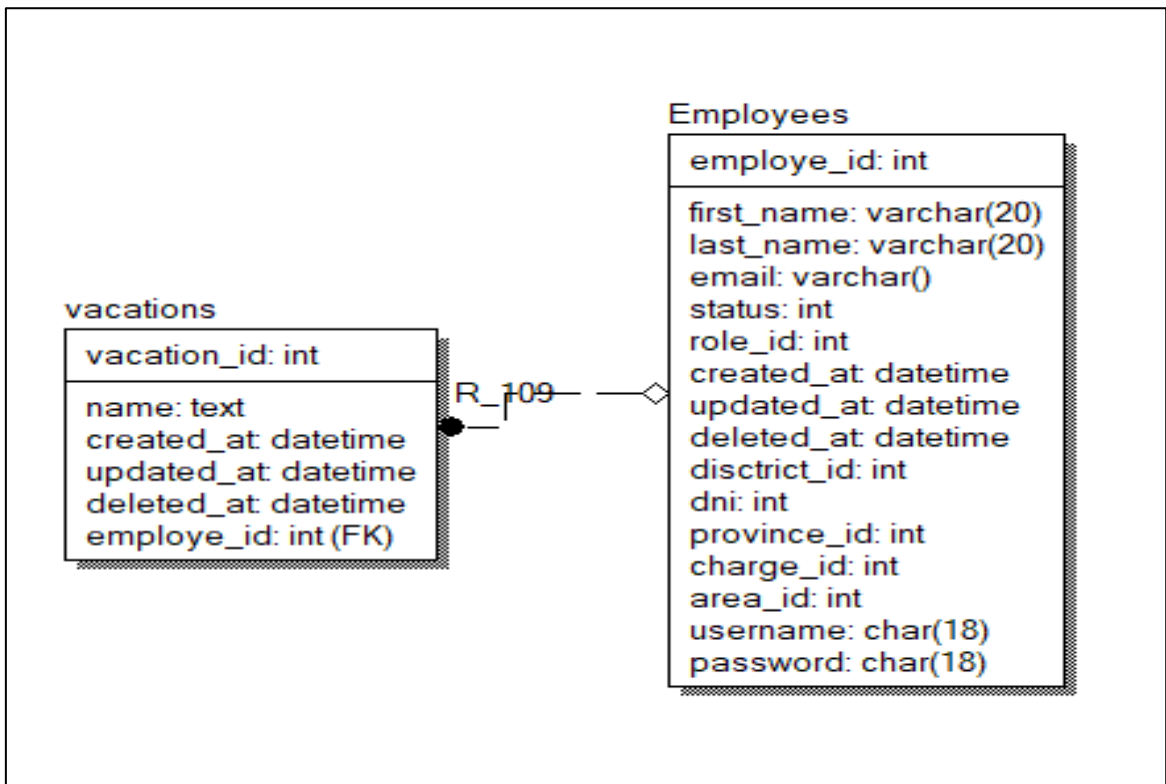


Figura 267. Diagrama físico de la base de datos RF52

Diseño

© Fuente: Elaboración propia



Figura 268. Prototipo de RF52

En la figura 268 se muestra el prototipo para la GUI de seleccionar empleado vacaciones del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia .

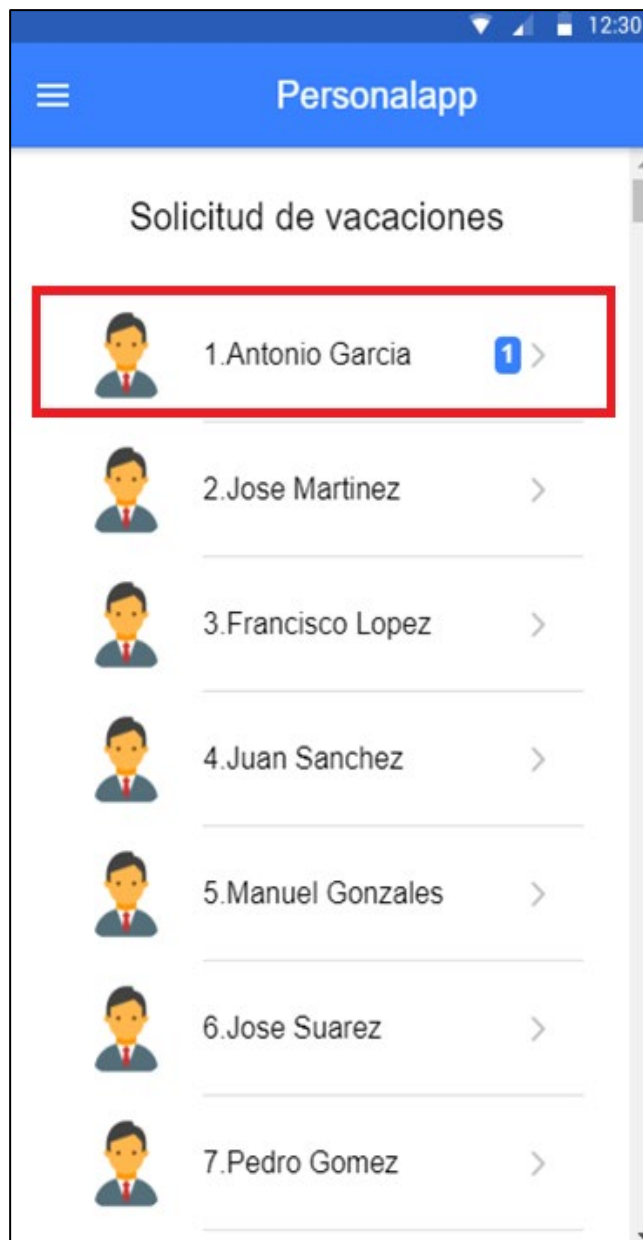


Figura 269. Interface de RF52

En la figura 269 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function vacationByEmployee($id) {  
    $vacations = Vacation::where('employee_id',$id)->get();  
    return $vacations;  
}
```

Figura 270. Código de RF52

En esta imagen 270 de la presente tesis se puede observar el código fuente del requerimiento funcional 52.

Requerimiento RF53 – Listar vacación

El requerimiento 53, tiene como objetivo que en la aplicación se debe poder ver el listado de las vacaciones de los empleados de la empresa, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero” y “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 53).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

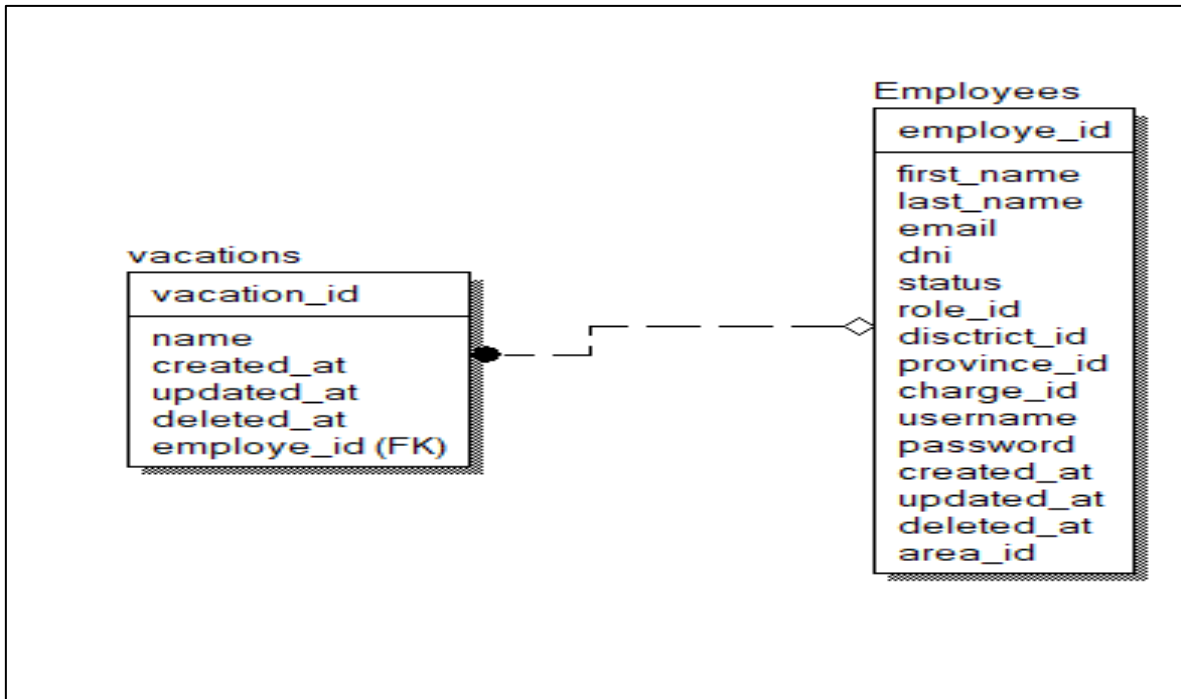


Figura 271. Diagrama lógico de la base de datos RF53

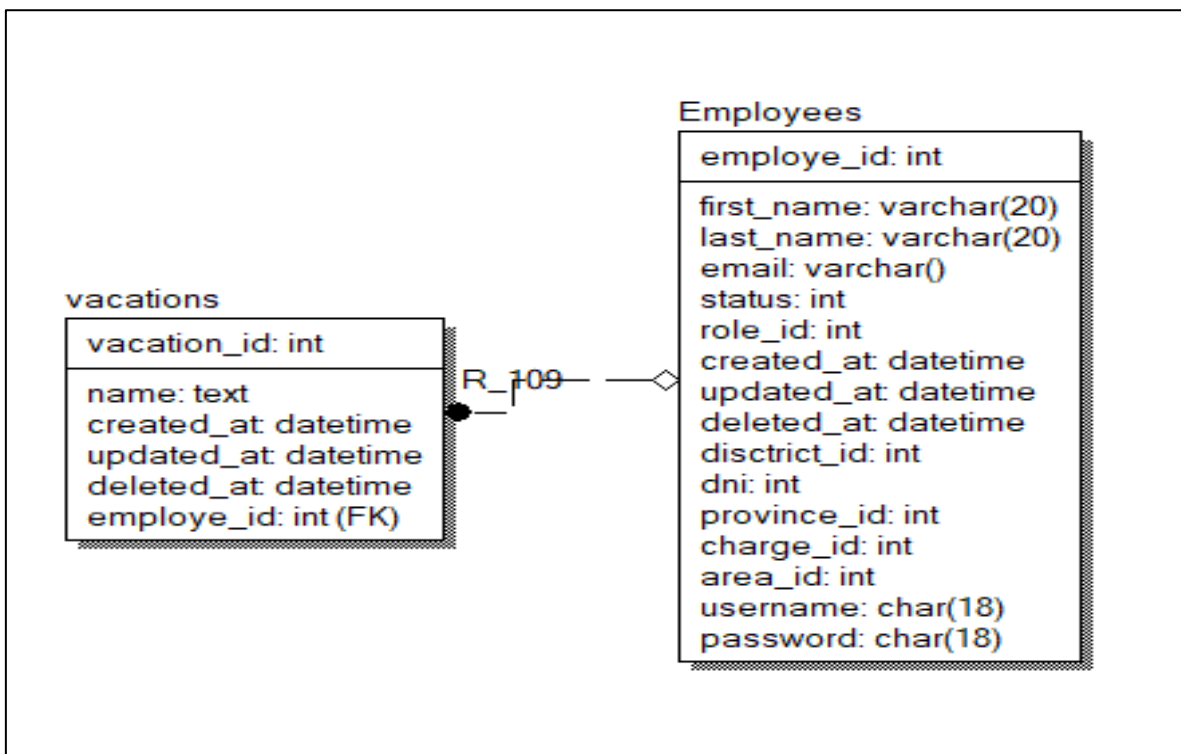


Figura 272. Diagrama físico de la base de datos RF53

Diseño

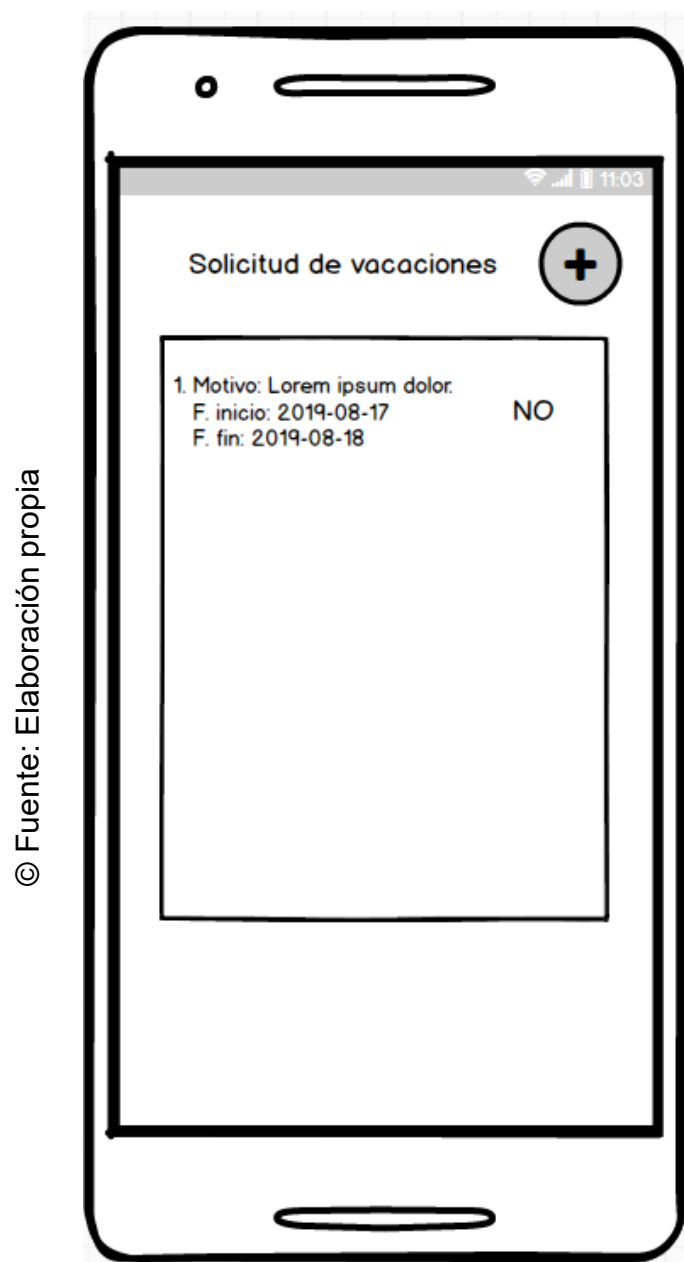


Figura 273. Prototipo de RF53

En la figura 273 se muestra el prototipo para la GUI de listar vacación de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

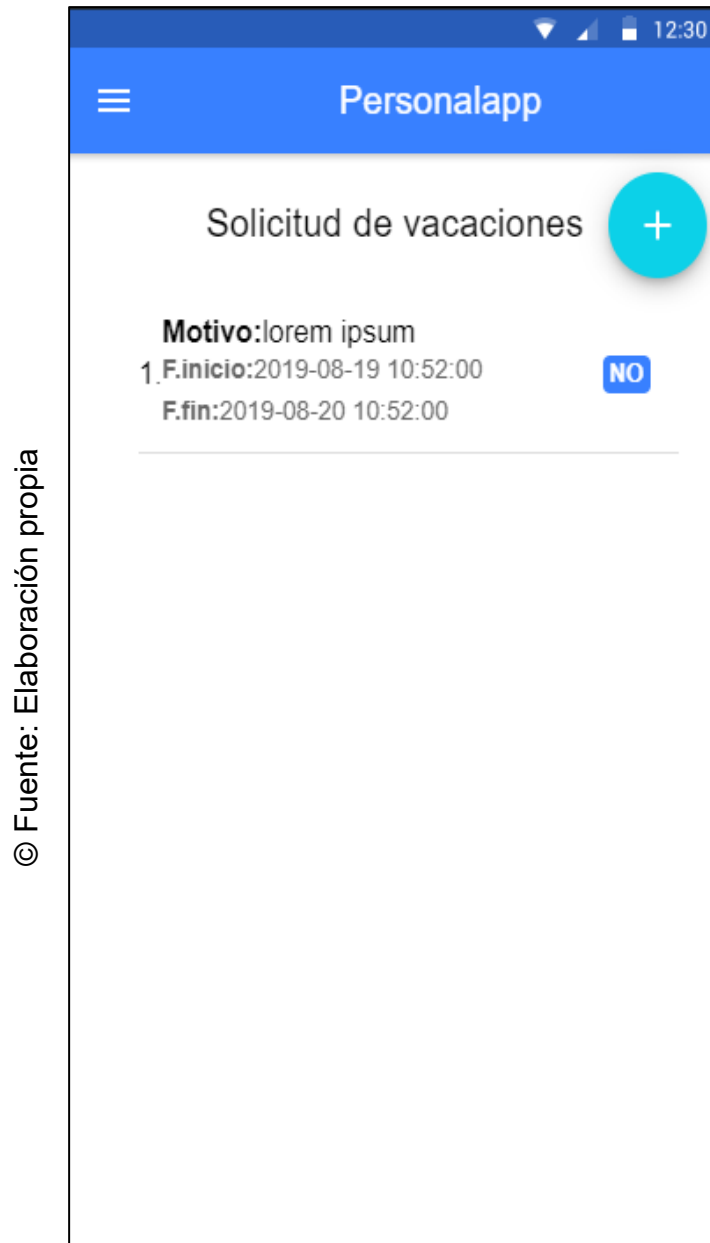


Figura 274. Interface de RF53

En la figura 274 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadDataListVacations(Request $request){
    $role_id = $request->role_id;
    $user_id = $request->user_id;
    if($role_id == 1 or $role_id == 3){
        $vacations = Vacation::where('employee_id',$user_id)->get();
        return $vacations;
    }else{
        $employees = Employee::with( relations: 'vacations')->whereIn( column: 'role_id',[1,3])->get();
        //return $employees;
        foreach ($employees as $employee){
            $contador = 0;
            foreach($employee->vacations as $vacation){
                if($vacation->status == 0){
                    $contador = $contador + 1;
                    $employee->total_vacations = $contador;
                }
            }
        }
        return $employees;
    }
}
```

Figura 275. Código de RF53

En esta imagen 275 de la presente tesis se puede observar el código fuente del requerimiento funcional 53.

Requerimiento RF54 – Aprobar vacación

El requerimiento 54, tiene como objetivo que en la aplicación se debe poder aprobar las vacaciones de los empleados de la empresa, dicho requerimiento solo debe hacerlo los usuarios “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 54).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

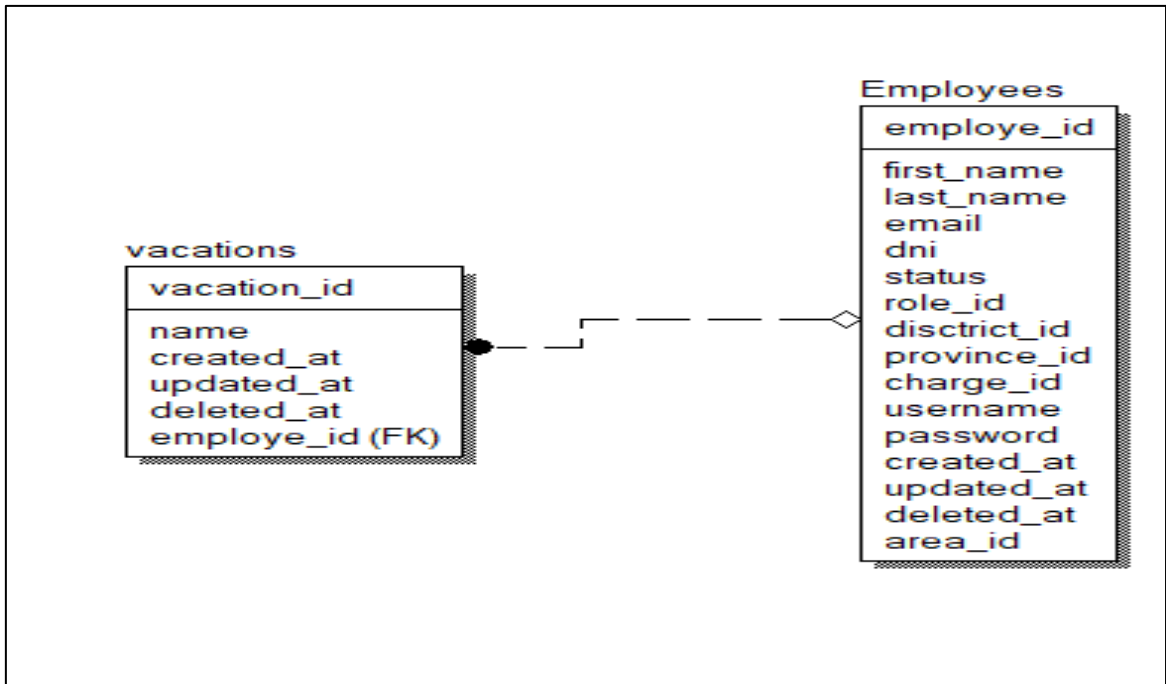


Figura 276. Diagrama lógico de la base de datos RF54

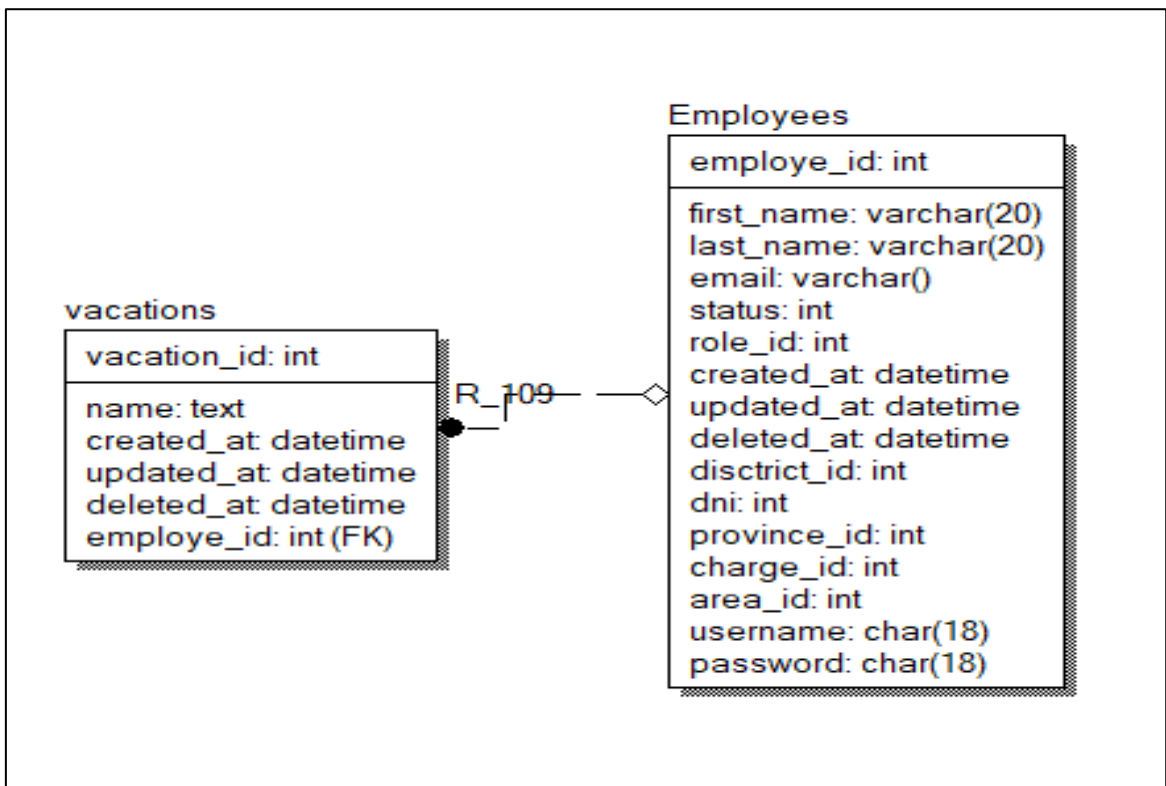


Figura 277. Diagrama físico de la base de datos RF54

Diseño

© Fuente: Elaboración propia

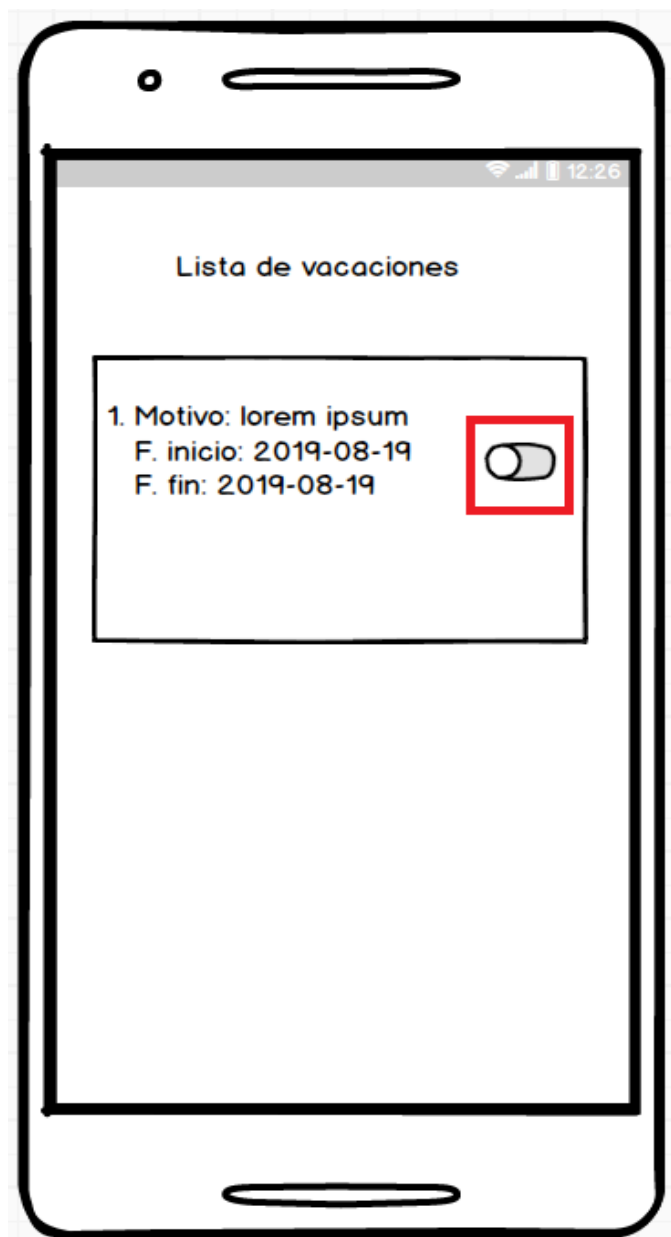


Figura 278. Prototipo de RF54

En la figura 278 se muestra el prototipo para la GUI de aprobar vacación del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

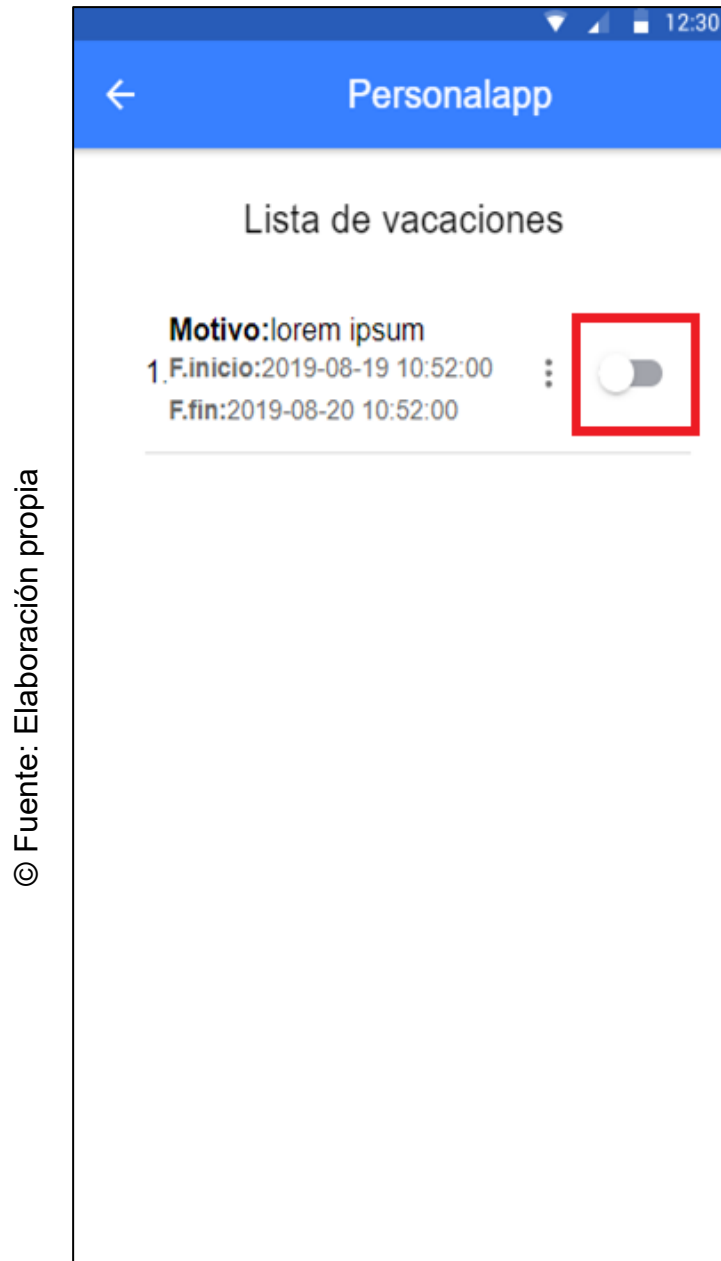


Figura 279. Interface de RF54

En la figura 279 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function statusVacation(Request $request){
    $vacation = Vacation::findOrFail($request->id);
    $vacation->status = $request->status;
    $vacation->save();
    return $vacation;
}
```

Figura 280. Código de RF54

En esta imagen 280 de la presente tesis se puede observar el código fuente del requerimiento funcional 54.

Requerimiento RF55 – Crear vacación

El requerimiento 55, tiene como objetivo que en la aplicación se debe poder crear una solicitud de vacación, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 55).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

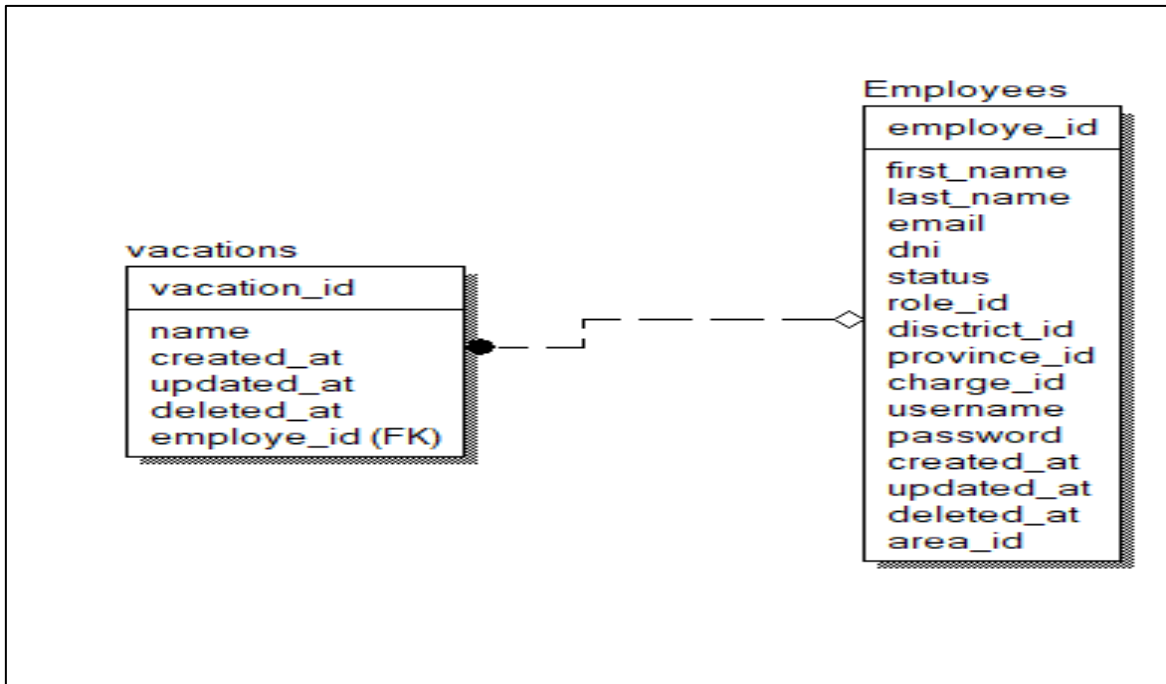


Figura 281. Diagrama lógico de la base de datos RF55

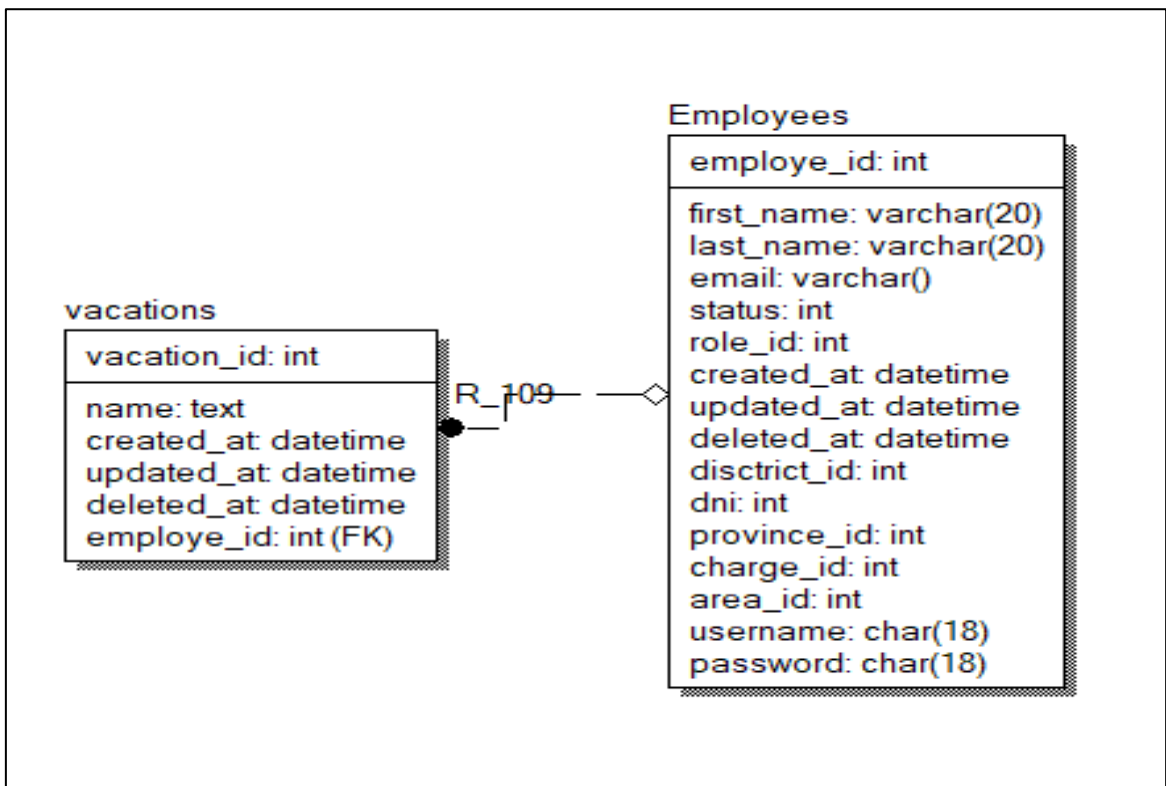


Figura 282. Diagrama físico de la base de datos RF55

Diseño

© Fuente: Elaboración propia

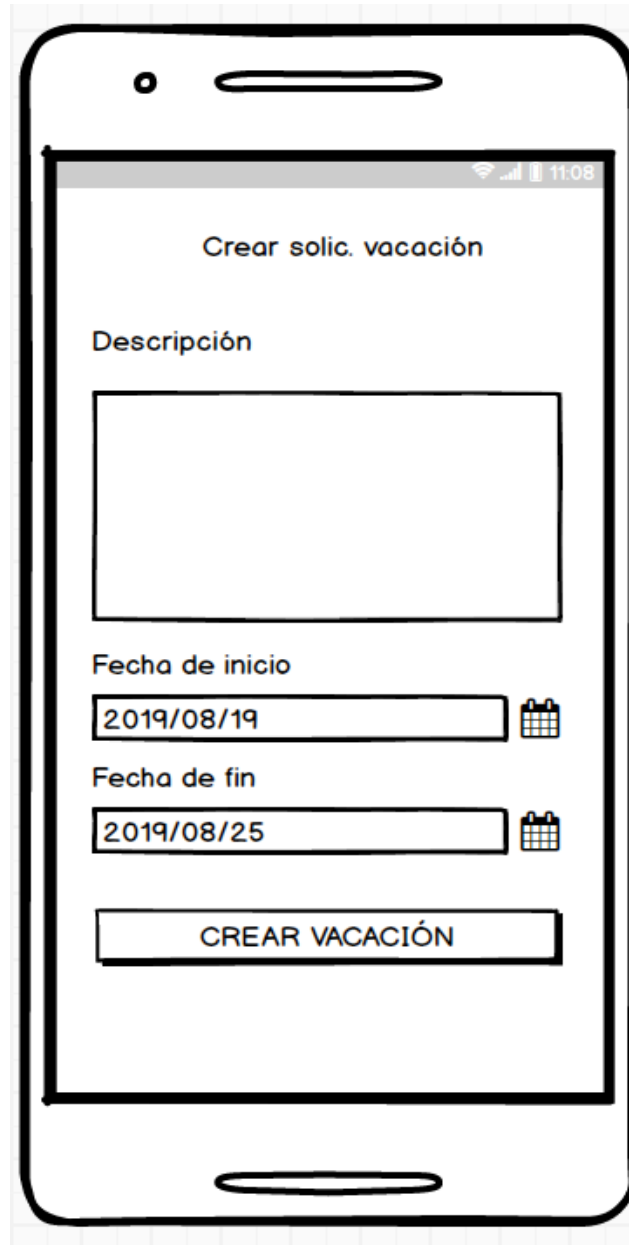
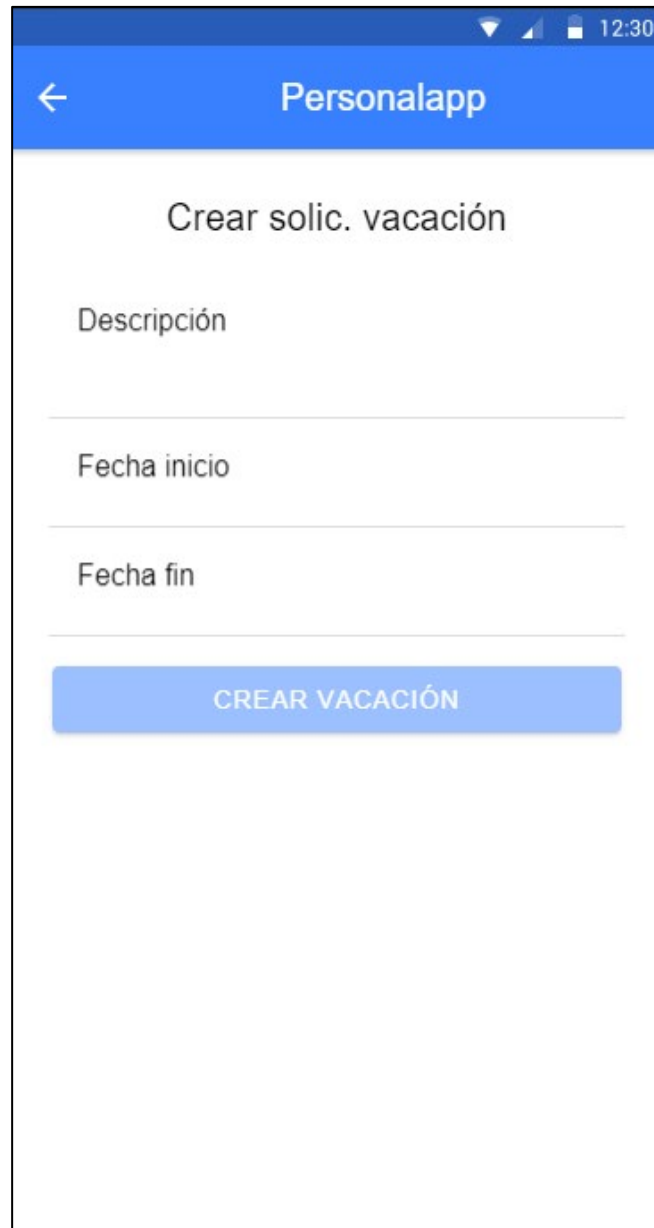


Figura 283. Prototipo de RF55

En la figura 283 se muestra el prototipo para la GUI de crear vacación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. At the top, there is a blue header bar with a back arrow on the left and the title 'Personalapp' in the center. Below the header, the main content area is white and contains the following elements: a title 'Crear solíc. vacación', a text input field labeled 'Descripción', a text input field labeled 'Fecha inicio', a text input field labeled 'Fecha fin', and a large blue button at the bottom with the text 'CREAR VACACIÓN'. The status bar at the top right shows the time as 12:30 and icons for signal strength, Wi-Fi, and battery.

Figura 284. Interface de RF55

En la figura 284 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function createVacation(Request $request){
    $date_init = $this->request->input( key: 'date_init');
    $date_end = $this->request->input( key: 'date_end');
    if($date_init > $date_end){
        return response()->json([
            'msg' => 'La fecha de inicio no puede ser menor a la fecha de fin.',
        ], status: 200);
    }else{
        $mytime = Carbon::now( tz: 'America/Lima');
        $responses = Vacation::where('employee_id',$request->employee_id)->get();
        if(count($responses) > 0){
            foreach ($responses as $response){
                $yearbd = Carbon::parse($response->date_init)->format( format: 'Y');
                $yearnow = $mytime->year;
                if($yearnow > $yearbd and $response->status == 1){
                    $vacation = new Vacation();
                    $vacation->description = $request->description;
                    $vacation->status = 0;
                    $vacation->employee_id = $request->employee_id;
                    $vacation->date_init = $request->date_init;
                    $vacation->date_end = $request->date_end;
                    $vacation->save();
                    return response()->json([
                        'msg' => 'exito',
                    ], status: 200);
                }else if($yearnow == $yearbd and $response->status == 1){
                    return response()->json([
                        'msg' => 'Solo puede realizar una solíc. de vacación por año.',
                    ], status: 200);
                }else{
                    return response()->json([
                        'msg' => 'Ya tiene una solíc. de vacación creada y aún no ha sido aprobada.',
                    ], status: 200);
                }
            }
        }else{
            $vacation = new Vacation();
            $vacation->description = $request->description;
            $vacation->status = 0;
            $vacation->employee_id = $request->employee_id;
            $vacation->date_init = $request->date_init;
            $vacation->date_end = $request->date_end;
            $vacation->save();
            return response()->json([
                'msg' => 'exito',
            ], status: 200);
        }
    }
}
```

Figura 285. Código de RF55

En esta imagen 285 de la presente tesis se puede observar el código fuente del requerimiento funcional 55.

Requerimiento RF56 – Actualizar vacación

El requerimiento 56, tiene como objetivo que en la aplicación se debe poder actualizar su solicitud de vacación y recursos humanos lo visualizará actualizado, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 56).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

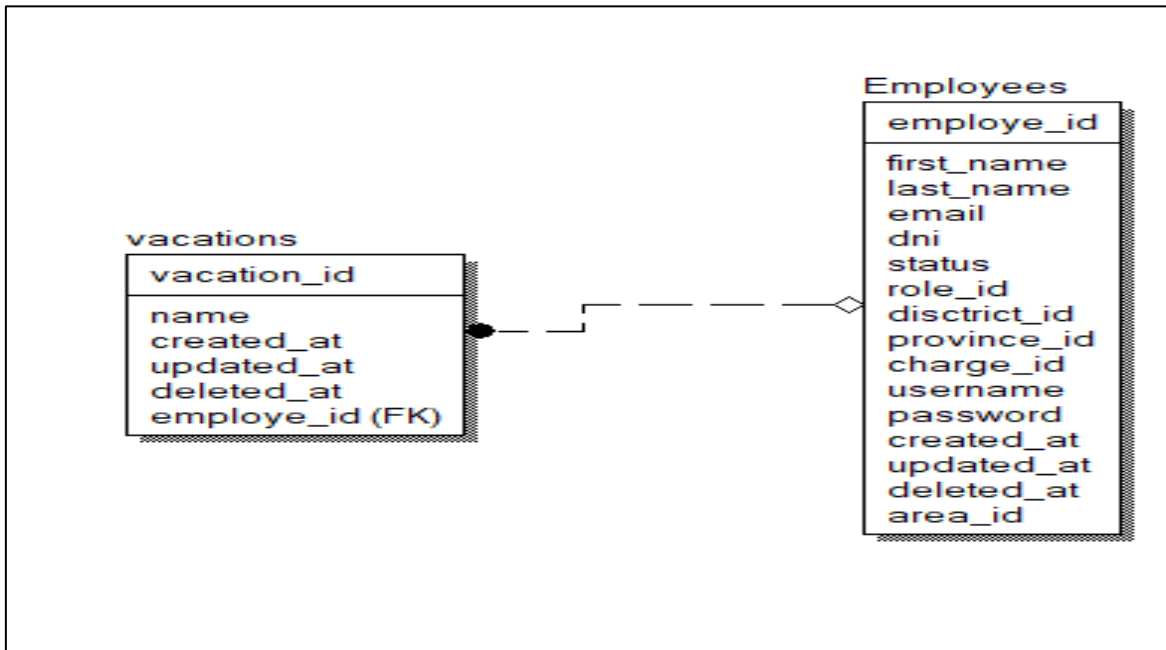


Figura 286. Diagrama lógico de la base de datos RF56

© Fuente: Elaboración propia

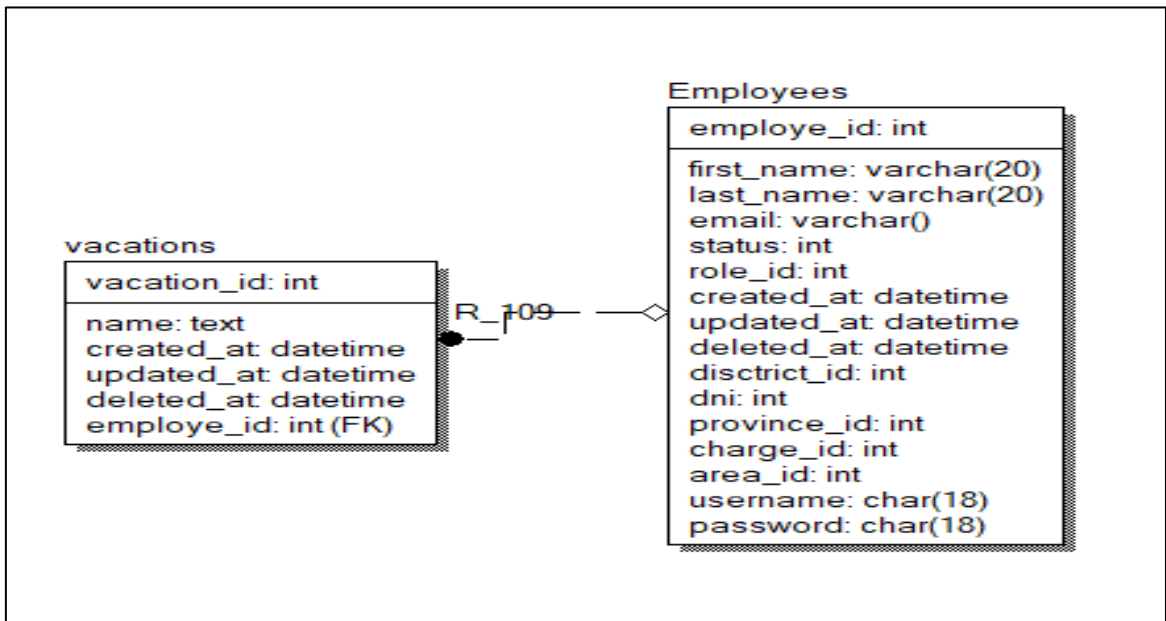


Figura 287. Diagrama físico de la base de datos RF56

Diseño

© Fuente: Elaboración propia

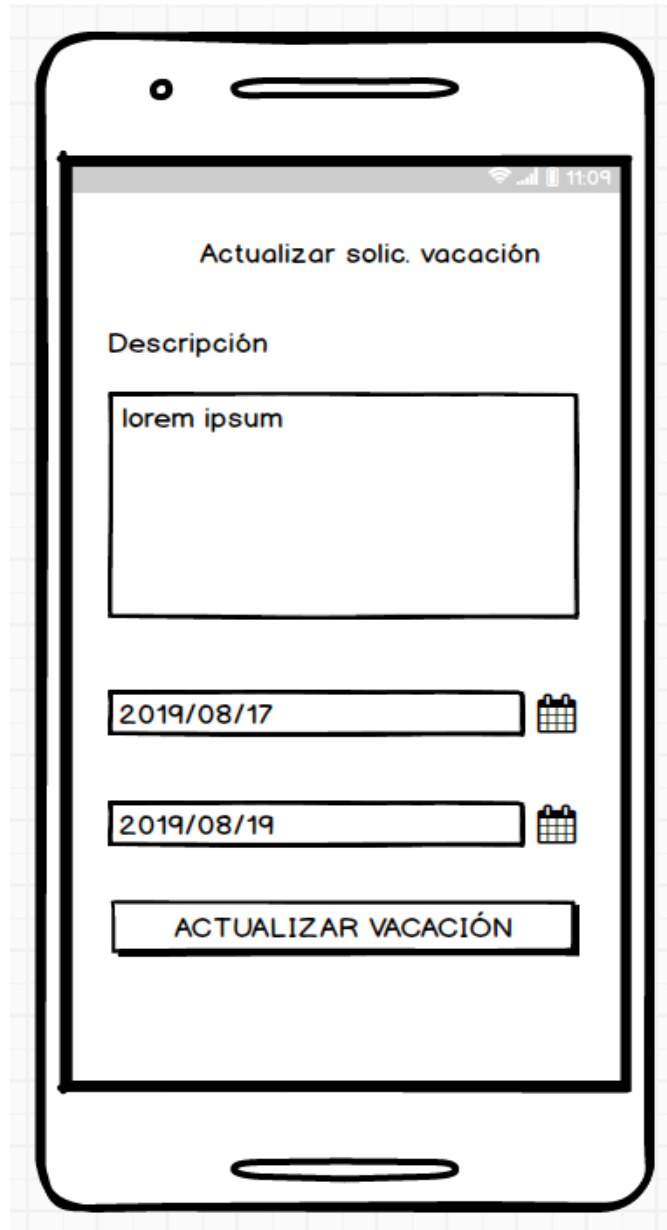


Figura 288. Prototipo de RF56

En la figura 288 se muestra el prototipo para la GUI de actualizar vacación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

The screenshot shows a mobile application interface with a blue header bar containing a back arrow and the text 'Personalapp'. The main content area is white and features the title 'Actualizar solíc. Vacación'. Below the title, there are three form fields: 'Descripción' with the placeholder text 'lorem ipsum', 'Fecha inicio' with the value '2019-08-19', and 'Fecha fin' with the value '2019-08-20'. At the bottom of the form is a blue button with the text 'ACTUALIZAR VACACION'. The status bar at the top right shows the time '12:30' and icons for signal strength and battery.

Figura 289. Interface de RF56

En la figura 289 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateVacation(Request $request,$id){
    $vacation = Vacation::find($id);
    $vacation->description = $request->description;
    $vacation->date_init = $request->date_init;
    $vacation->date_end = $request->date_end;
    $vacation->save();
    return response()->json([
        'msg' => 'exito',
    ], status: 200);
}
```

Figura 290. Código de RF56

En esta imagen 290 de la presente tesis se puede observar el código fuente del requerimiento funcional 56.

Requerimiento RF57 – Eliminar vacación

El requerimiento 57, tiene como objetivo que en la aplicación se debe poder eliminar su solicitud de vacación, dicho requerimiento solo debe hacerlo los usuarios “Administrador” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 57).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

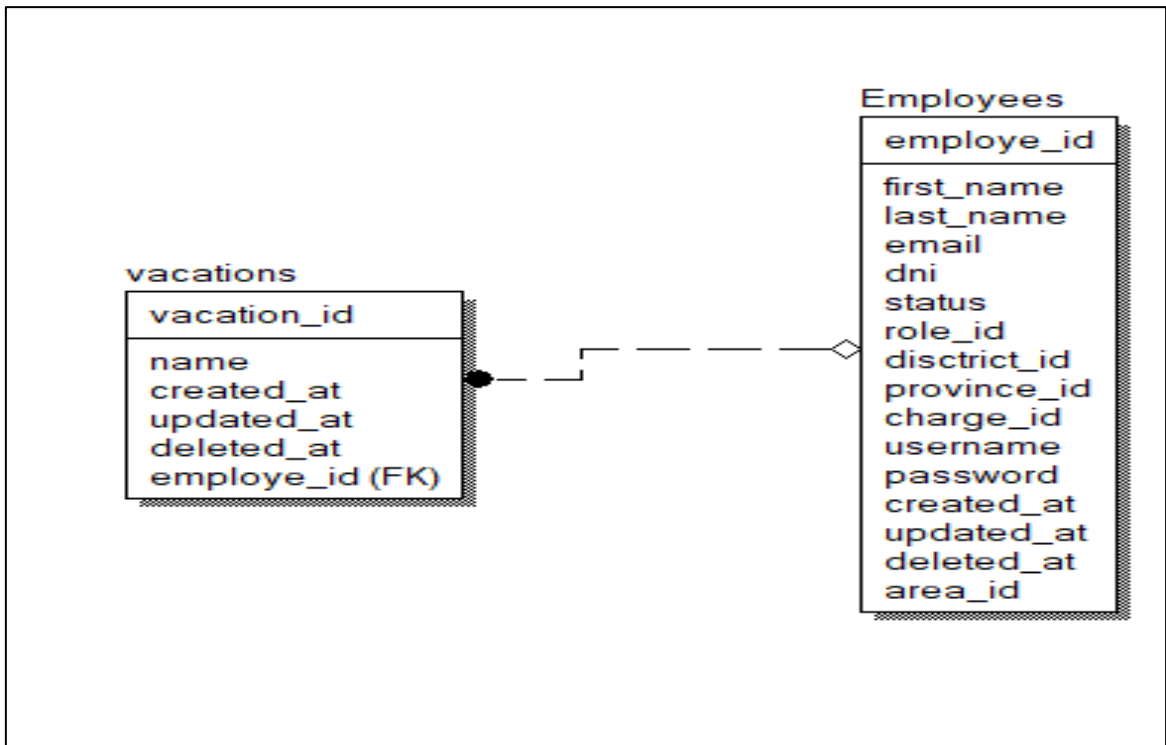


Figura 291. Diagrama lógico de la base de datos RF57

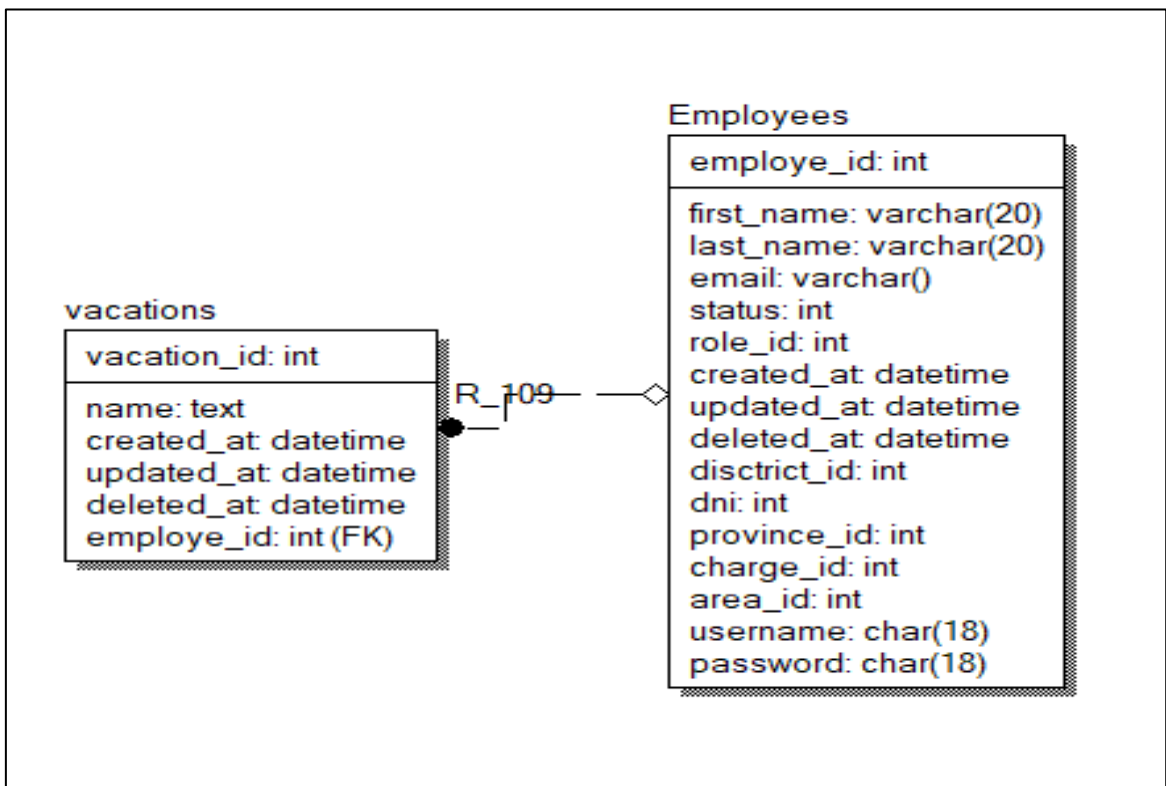


Figura 292. Diagrama físico de la base de datos RF57

Diseño

© Fuente: Elaboración propia

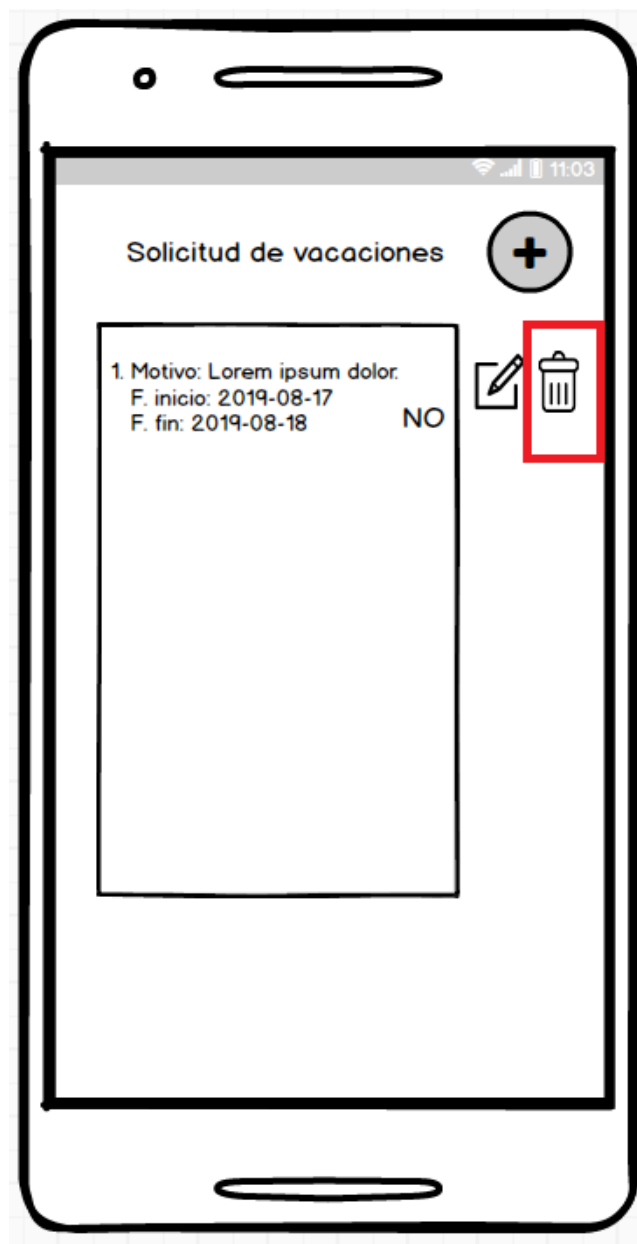


Figura 293. Prototipo de RF57

En la figura 293 se muestra el prototipo para la GUI de eliminar vacación del usuario administrador y obrero que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación



Figura 294. Interface de RF57

En la figura 294 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteVacation(Request $request) {  
    $vacation = Vacation::findOrFail($request->id);  
    $vacation->delete();  
    return response()->json([  
        'msg' => 'exito',  
    ], status: 200);  
}
```

Figura 295. Código de RF57

En esta imagen 295 de la presente tesis se puede observar el código fuente del requerimiento funcional 57.

Requerimiento RF58 – Pulsar remuneración

El requerimiento 58, tiene como objetivo que en la aplicación se debe poder ver un botón llamado “Remuneraciones” en la barra lateral para ver la remuneración de los empleados, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Obrero” y “Recursos humanos”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

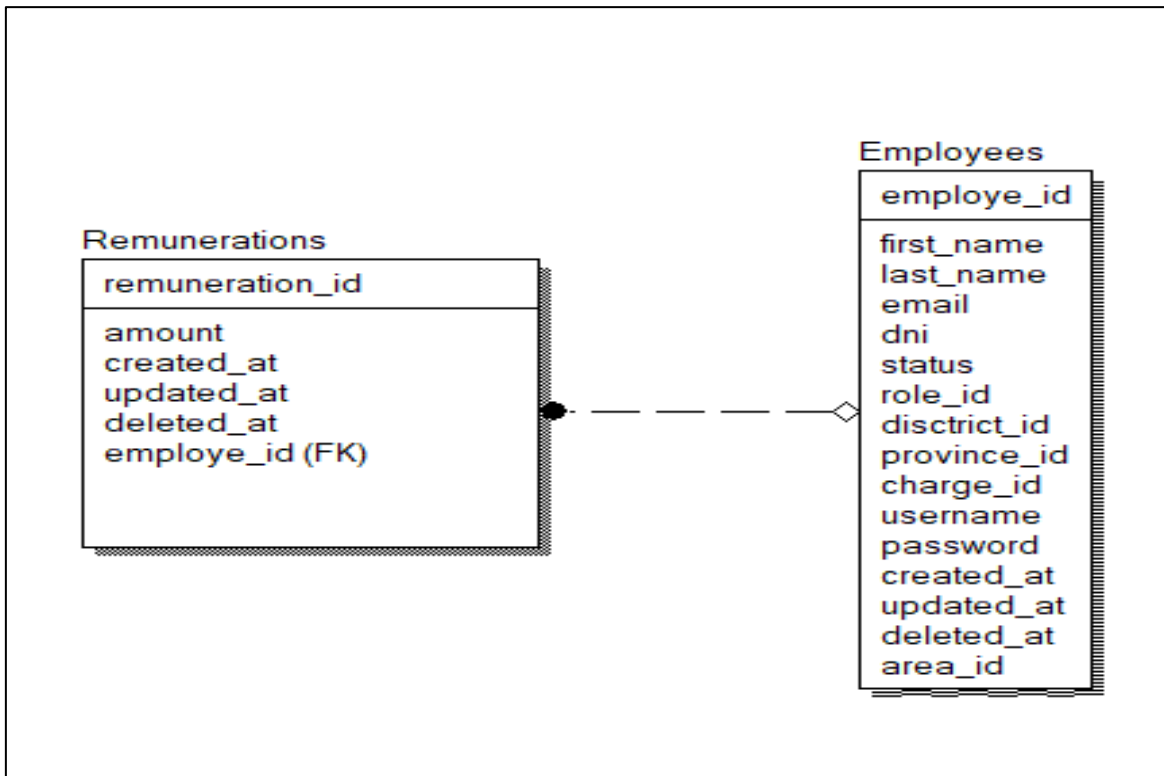


Figura 296. Diagrama lógico de la base de datos RF58

© Fuente: Elaboración propia

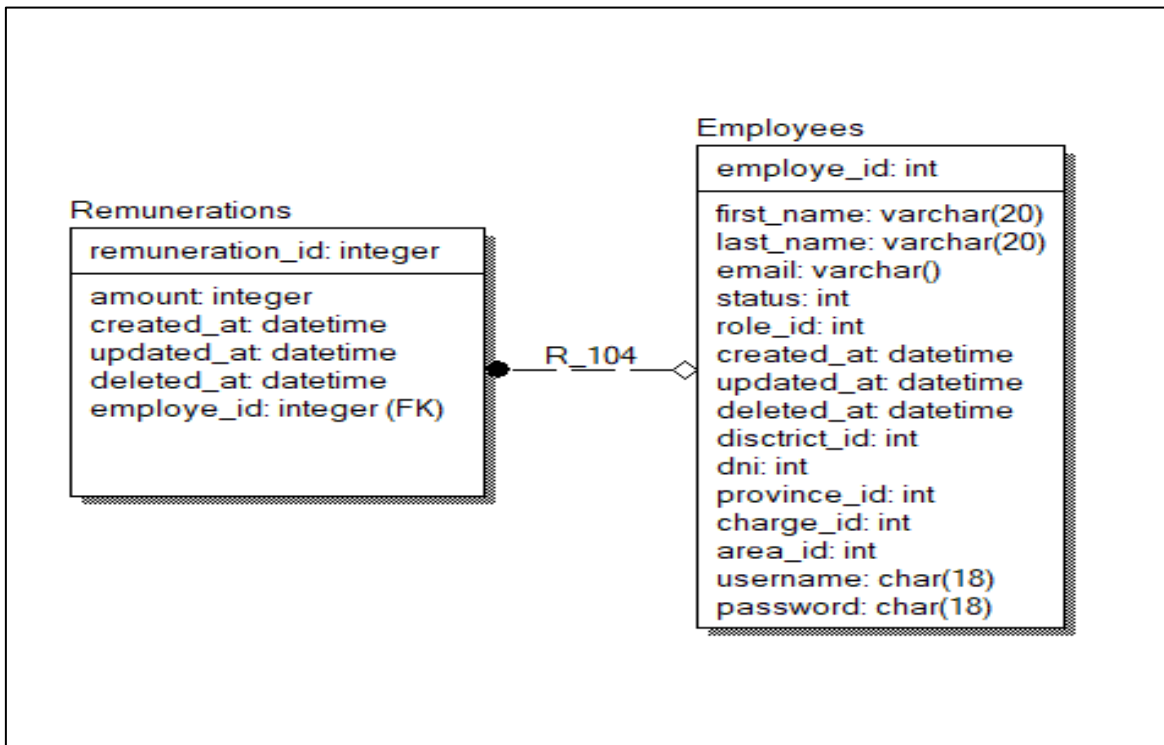


Figura 297. Diagrama físico de la base de datos RF58

Diseño

© Fuente: Elaboración propia

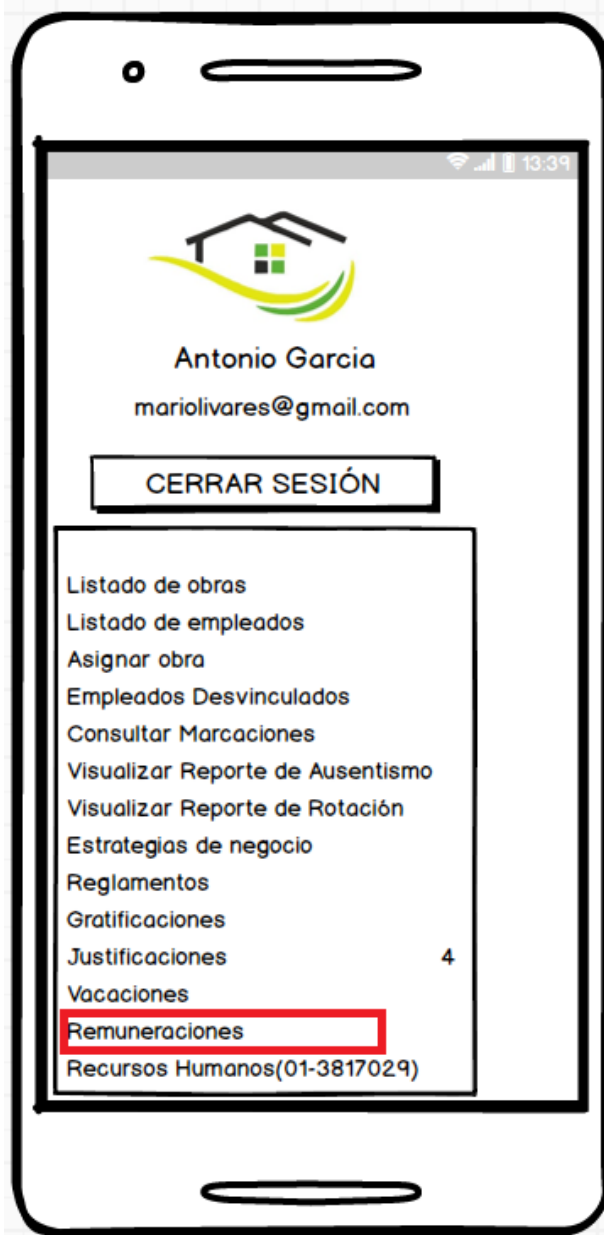


Figura 298. Prototipo de RF58

En la figura 298 se muestra el prototipo para la GUI de pulsar remuneración de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

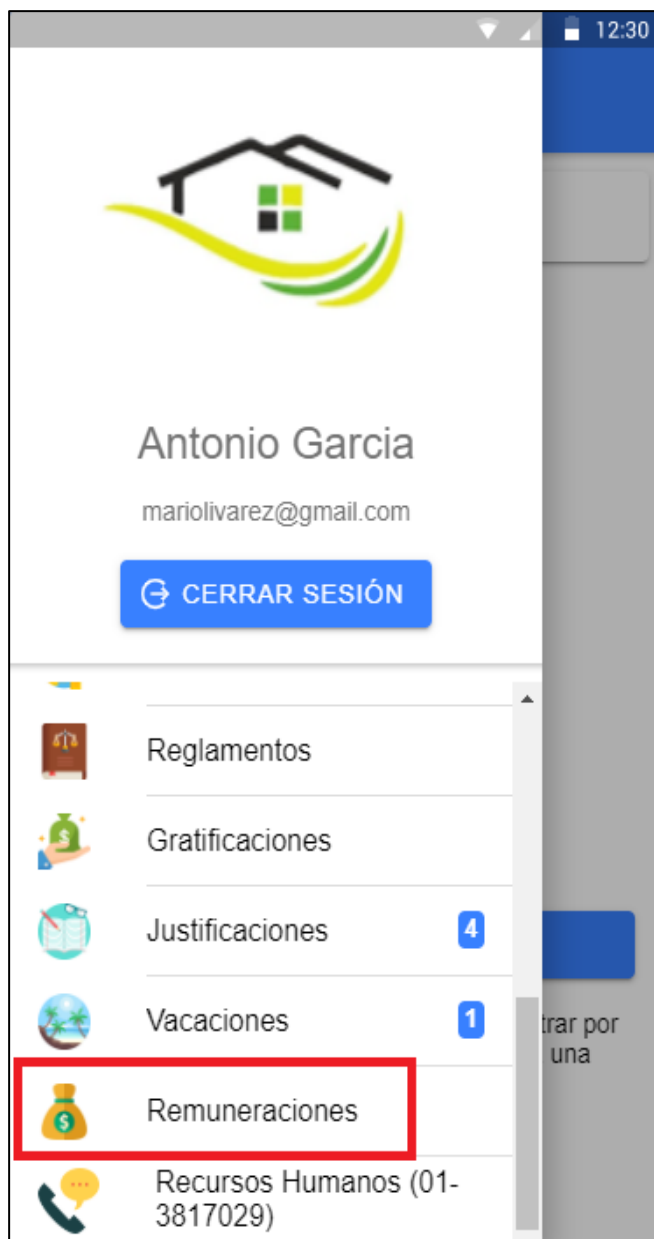


Figura 299. Interface de RF58

En la figura 299 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
if(res == 1){
  this.pages = [
    { title: 'Listado de obras', page: '/workcontroller', icon:'git-network' },
    { title: 'Listado de empleados', page: '/employeecontroller', icon:'git-network' },
    { title: 'Asignar Obra', page: '/assignwork', icon:'git-network' },
    { title: 'Empleados Desvinculados', page: '/dismissal', icon:'contacts' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 2){
  this.pages = [
    { title: 'Visualizar reporte de ausentismo', page: '/absenteeism', icon:'briefcase' },
    { title: 'Visualizar reporte de rotación', page: '/rotation', icon:'briefcase' },
    { title: 'Constular Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Reglamentos', page: '/listarreglaments', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}else if(res == 3){
  this.pages = [
    { title: 'Visualizar Obra Asignada', page: '/workassign', icon:'briefcase' },
    { title: 'Consultar Marcaciones', page: '/marcations', icon:'briefcase' },
    { title: 'Estrategias de negocio', page: '/strategies', icon:'briefcase' },
    { title: 'Gratificaciones', page: '/list', icon:'briefcase' },
    { title: 'Justificaciones', page: '/listjustifications', icon:'briefcase' },
    { title: 'Vacaciones', page: '/listvacations', icon:'briefcase' },
    { title: 'Remuneraciones', page: '/listremunerations', icon:'briefcase' }
  ]
}
```

Figura 300. Código de RF58

En esta imagen 300 de la presente tesis se puede observar el código fuente del requerimiento funcional 58.

Requerimiento RF59 – Listar empleado remuneración

El requerimiento 59, tiene como objetivo que en la aplicación se debe poder visualizar el listado de los empleados que tengan o no una remuneración, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

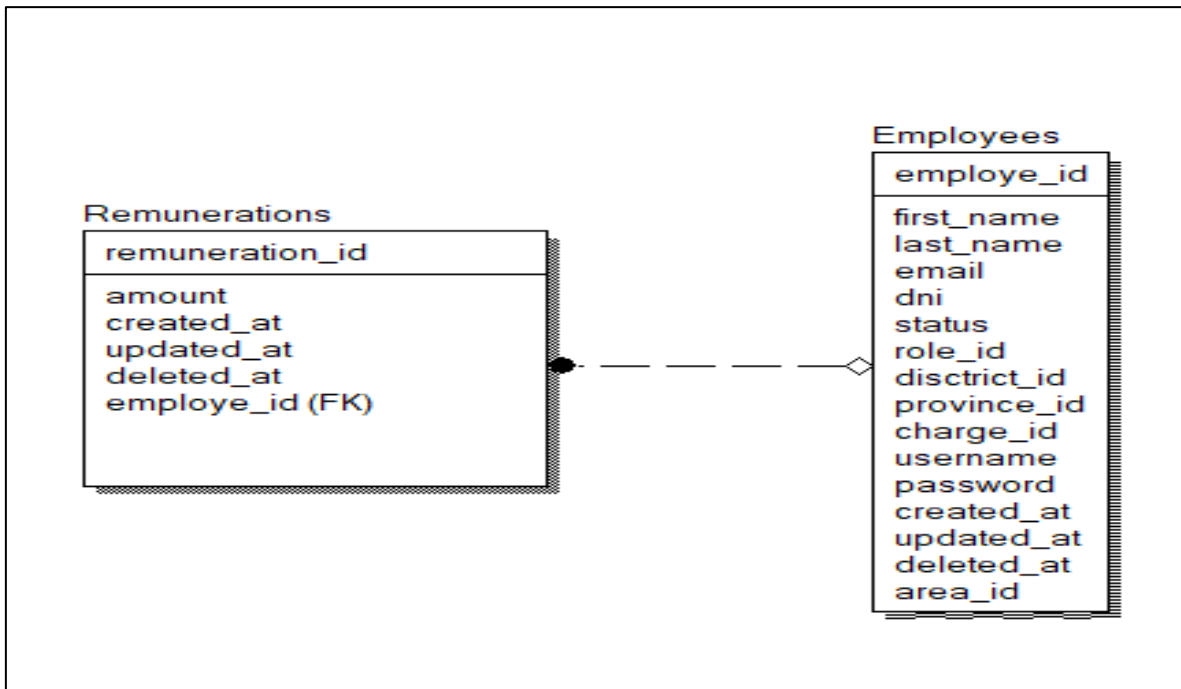


Figura 301. Diagrama lógico de la base de datos RF59

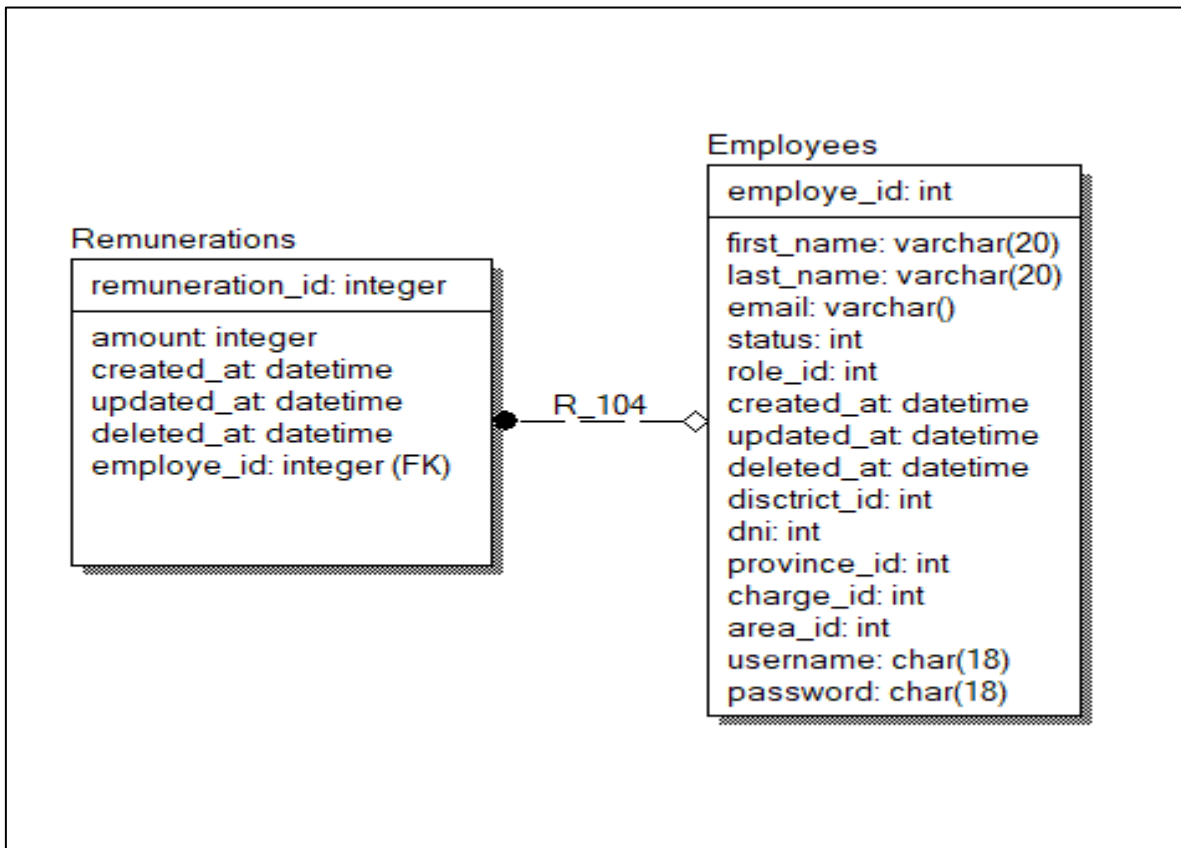


Figura 302. Diagrama físico de la base de datos RF59

Diseño

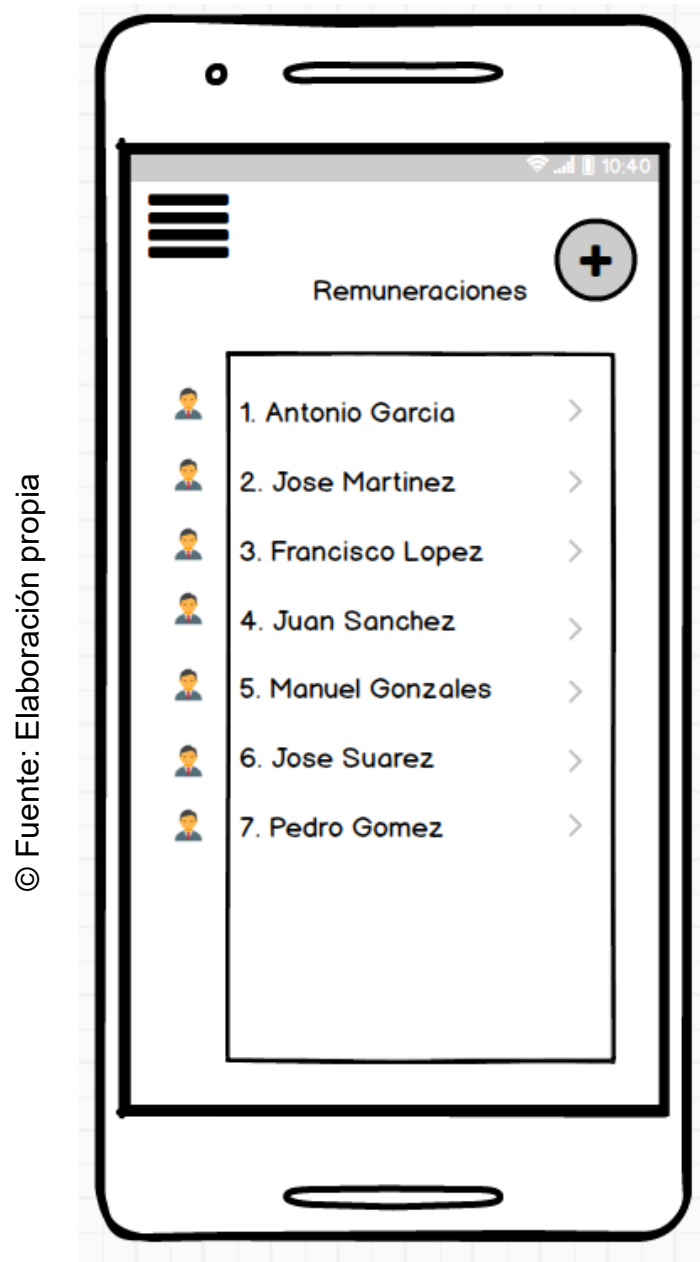


Figura 303. Prototipo de RF59

En la figura 303 se muestra el prototipo para la GUI de listar empleado remuneración del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

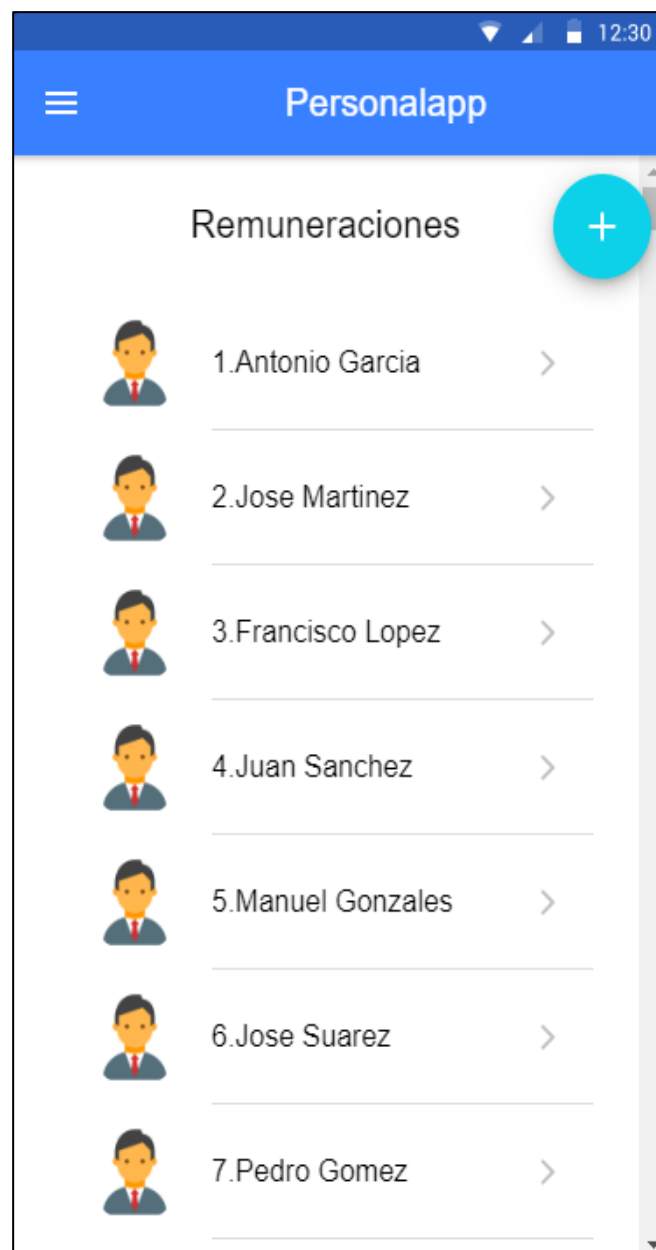


Figura 304. Interface de RF59

En la figura 304 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function loadDataListRemunerations(Request $request){
    $role_id = $request->role_id;
    $user_id = $request->user_id;
    if($role_id == 1 or $role_id == 3){
        $remunerations = Remuneration::where('employee_id',$user_id)->get();
        return $remunerations;
    }else{
        $employees = Employee::with( relations: 'remunerations' )->whereIn( column: 'role_id',[1,3])->get();
        //return $employees;
        return $employees;
    }
}
```

Figura 305. Código de RF59

En esta imagen 305 de la presente tesis se puede observar el código fuente del requerimiento funcional 59.

Requerimiento RF60 – Seleccionar empleado remuneración

El requerimiento 60, tiene como objetivo que en la aplicación se debe poder seleccionar a los empleados que tengan o no una remuneración, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

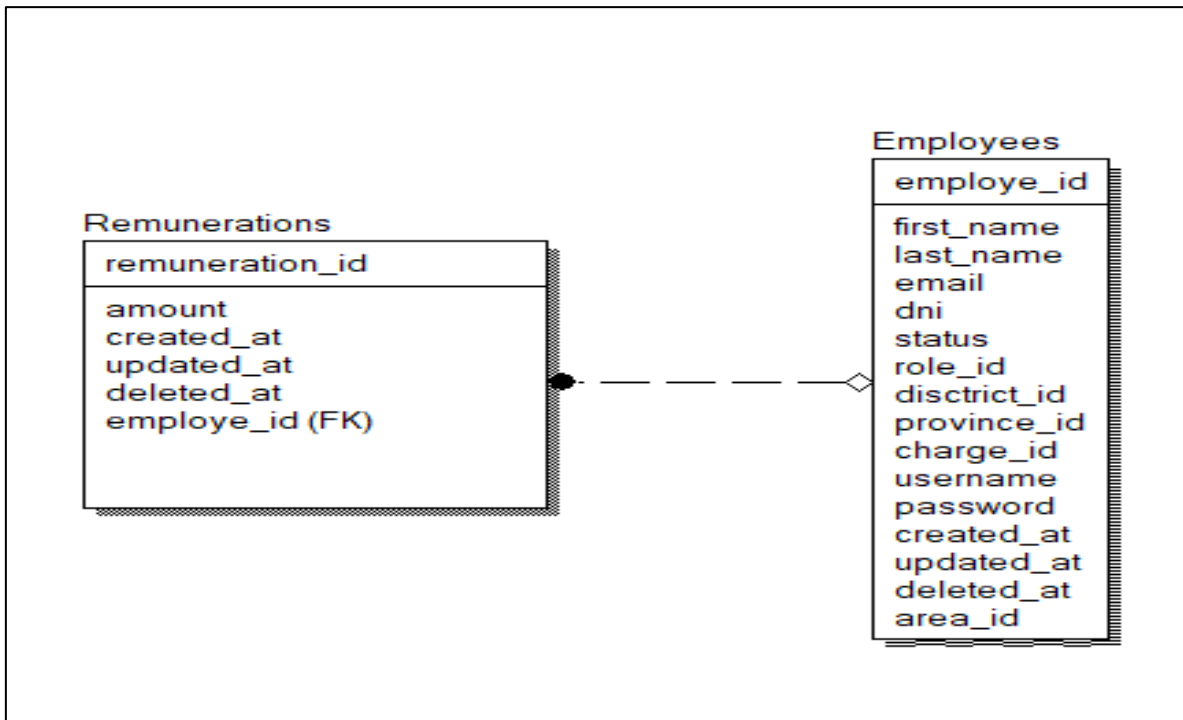


Figura 306. Diagrama lógico de la base de datos RF60

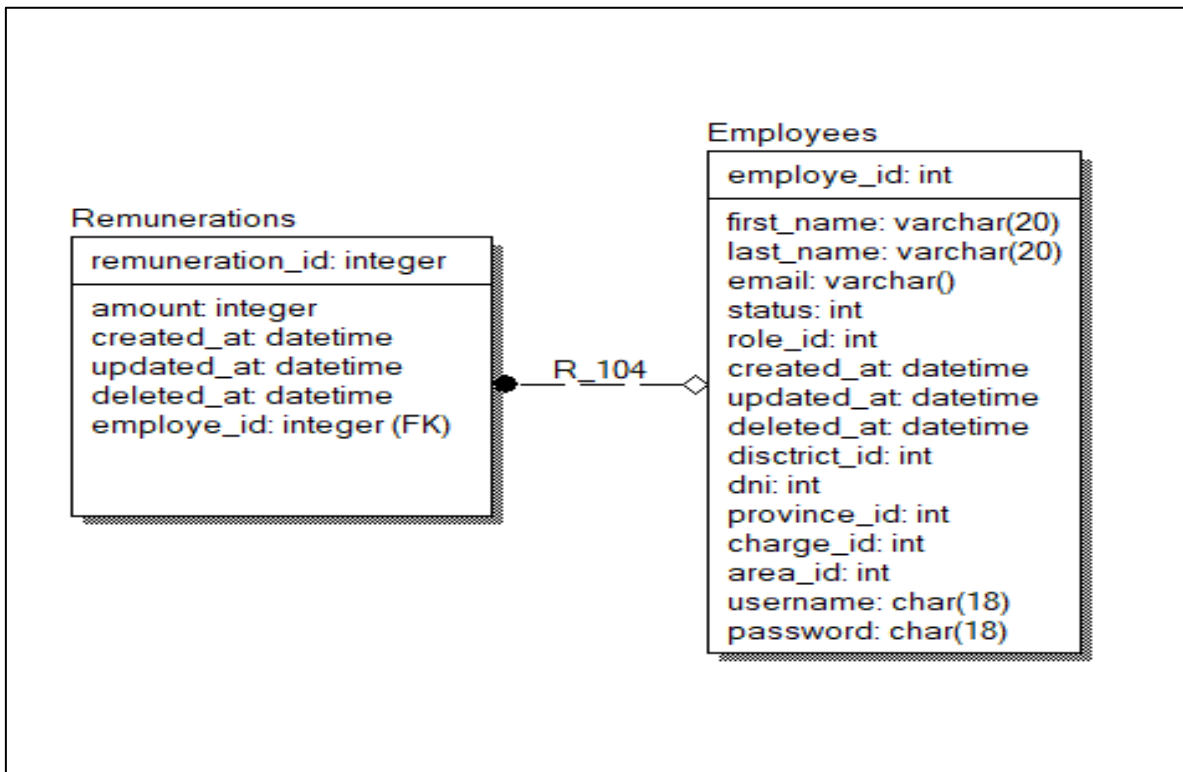


Figura 307. Diagrama físico de la base de datos RF60

Diseño

© Fuente: Elaboración propia

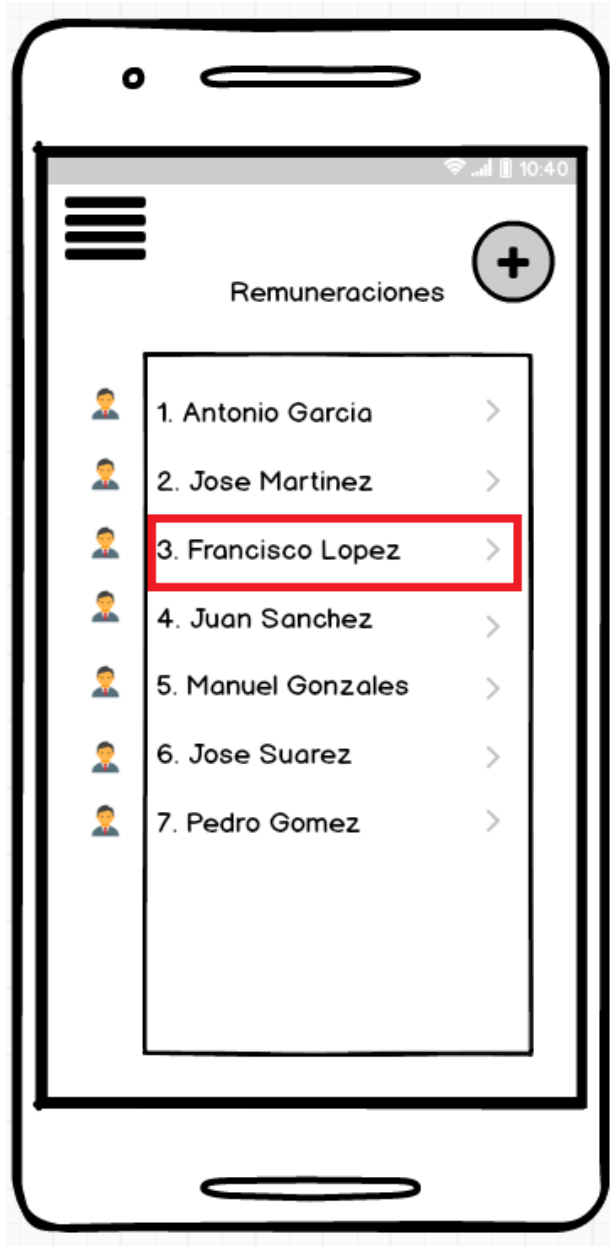


Figura 308. Prototipo de RF60

En la figura 308 se muestra el prototipo para la GUI de seleccionar empleado remuneración del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

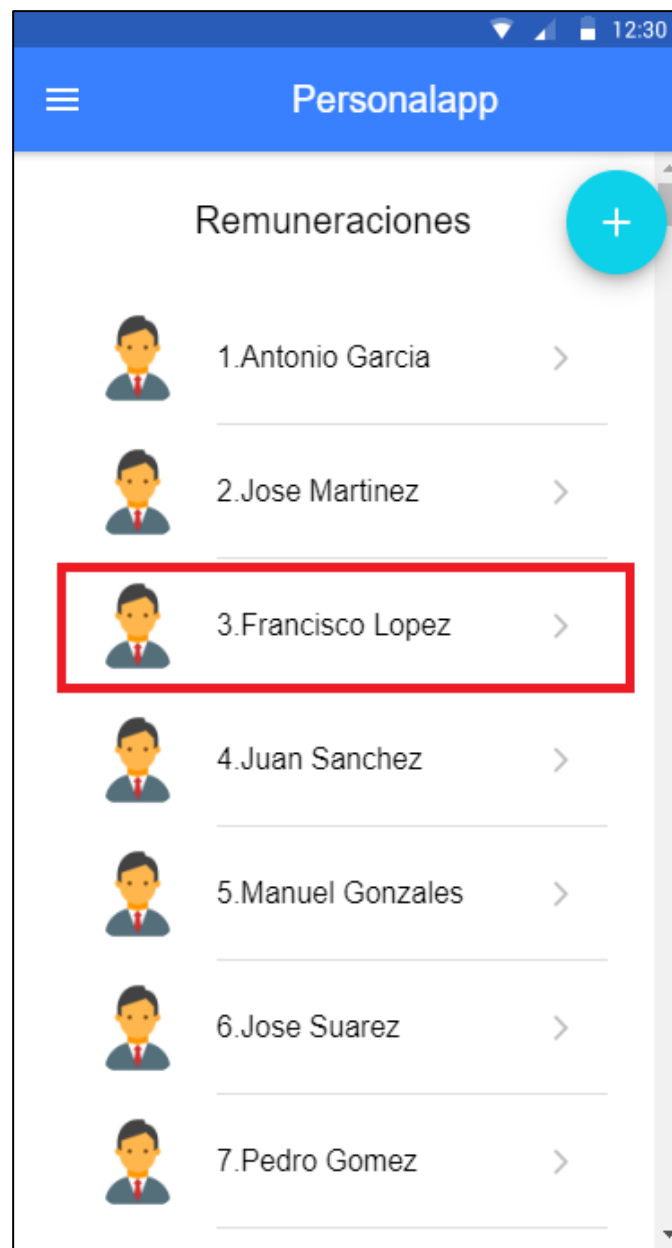


Figura 309. Interface de RF60

En la figura 309 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
<!-- List of Sliding Items -->
<ion-list *ngIf="role_id_menu == 2" #lista>
  <ion-item detail routerLink="/remunerationbyemployee/{{data.id}}" *ngFor="let data of datarest ; let i = index">
    {{i + 1}}.
    <ion-thumbnail slot="start">
      
    </ion-thumbnail>
    <ion-label>{{data.first_name}} {{data.last_name}}</ion-label>
  </ion-item>
</ion-list>
<ion-list *ngIf="role_id_menu == 1 || role_id_menu == 3" #lista>
  <ion-item-sliding *ngFor="let data of datarest; let i = index">
    <ion-item>
      {{i + 1}}.<ion-label><strong>Monto: </strong>{{data.amount}}</ion-label>
    </ion-item>
  </ion-item-sliding>
  <div *ngIf="datarest == ''">
    No hay remuneración para este empleado.
  </div>
</ion-list>
```

Figura 310. Código de RF60

En esta imagen 310 de la presente tesis se puede observar el código fuente del requerimiento funcional 60.

Requerimiento RF61 – Listar remuneración

El requerimiento 61, tiene como objetivo que en la aplicación se debe poder ver el listado de las remuneraciones de los empleados de la empresa, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Recursos humanos” y “Obrero”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 61).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

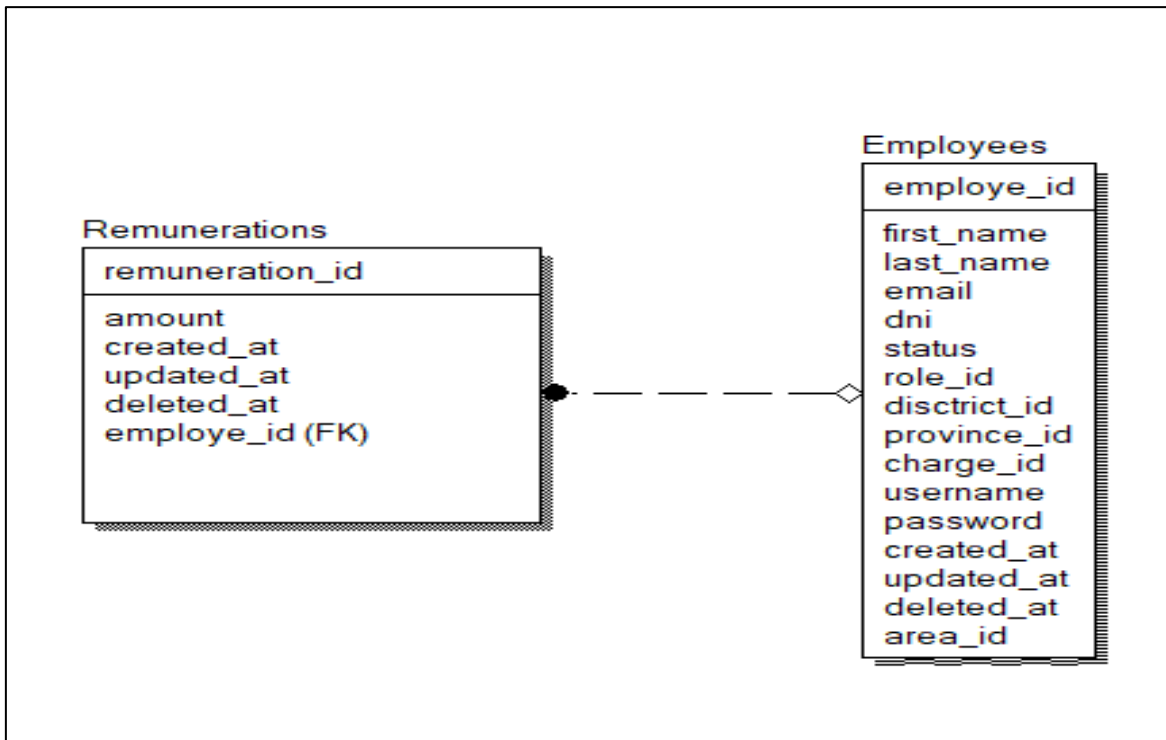


Figura 311. Diagrama lógico de la base de datos RF61

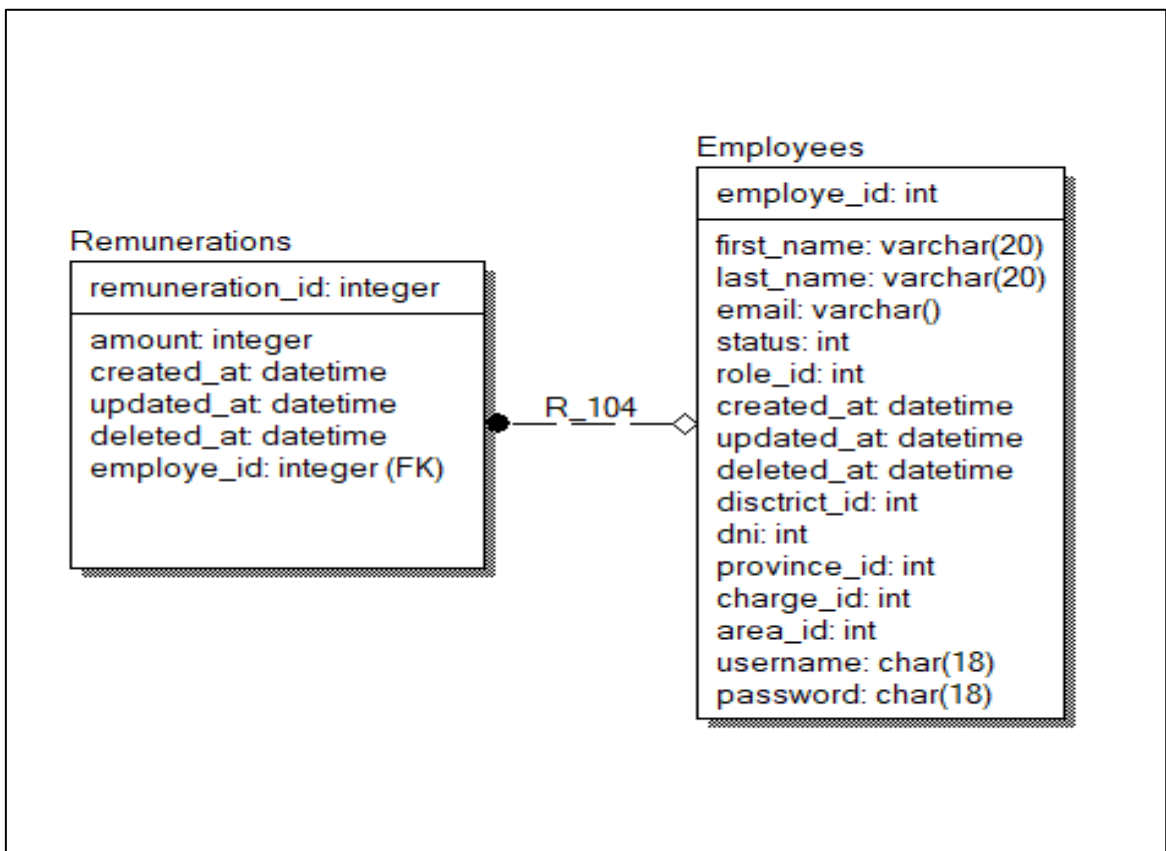


Figura 312. Diagrama físico de la base de datos RF61

Diseño

© Fuente: Elaboración propia

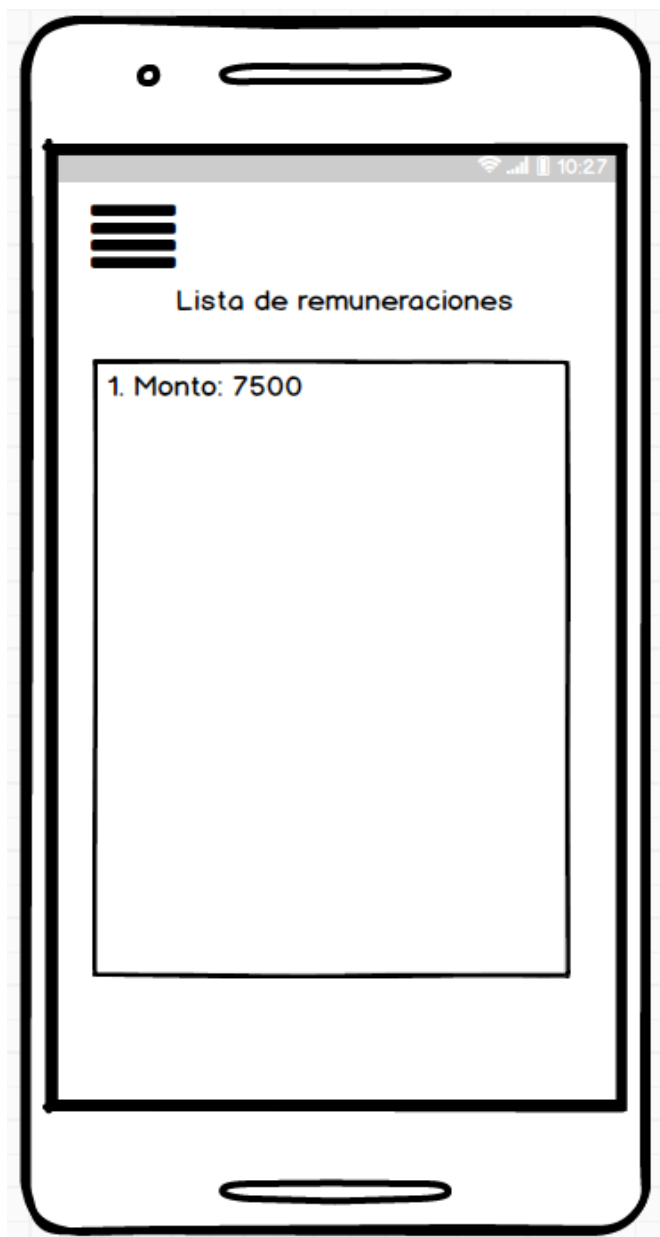


Figura 313. Prototipo de RF61

En la figura 313 se muestra el prototipo para la GUI de listar remuneración de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

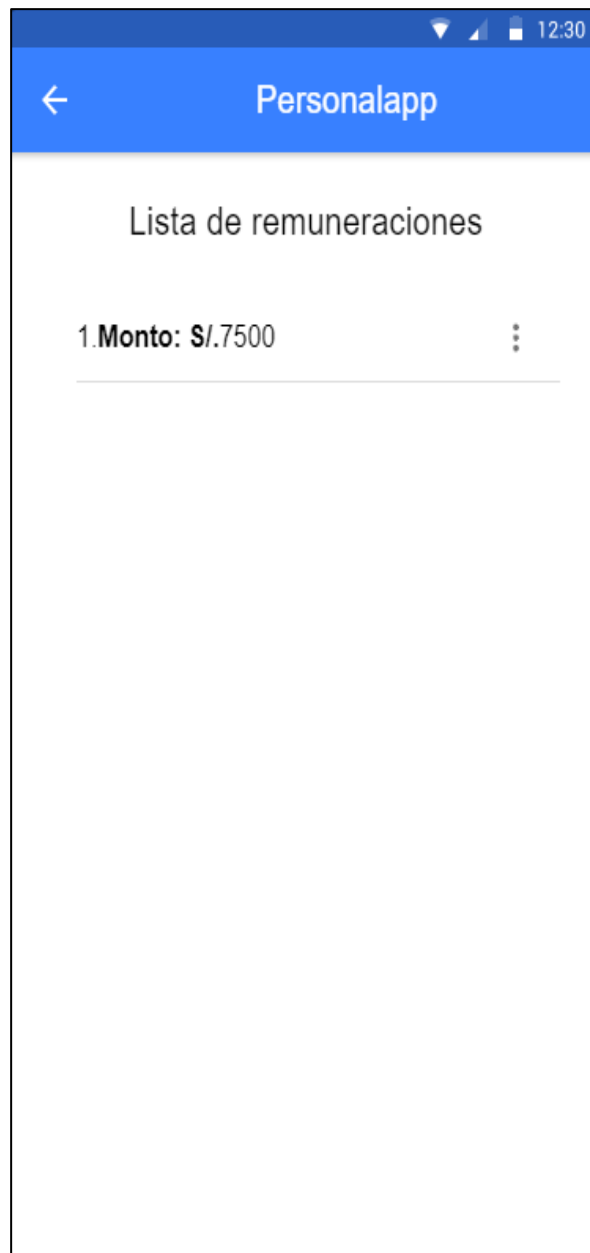


Figura 314. Interface de RF61

En la figura 314 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function remunerationByEmployee($id){  
    $remunerations = Remuneration::where('employee_id',$id)->get();  
    return $remunerations;  
}
```

Figura 315. Código de RF61

En esta imagen 315 de la presente tesis se puede observar el código fuente del requerimiento funcional 61.

Requerimiento RF62 – Crear remuneración

El requerimiento 62, tiene como objetivo que en la aplicación se debe poder crear las remuneraciones para los empleados, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 62).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

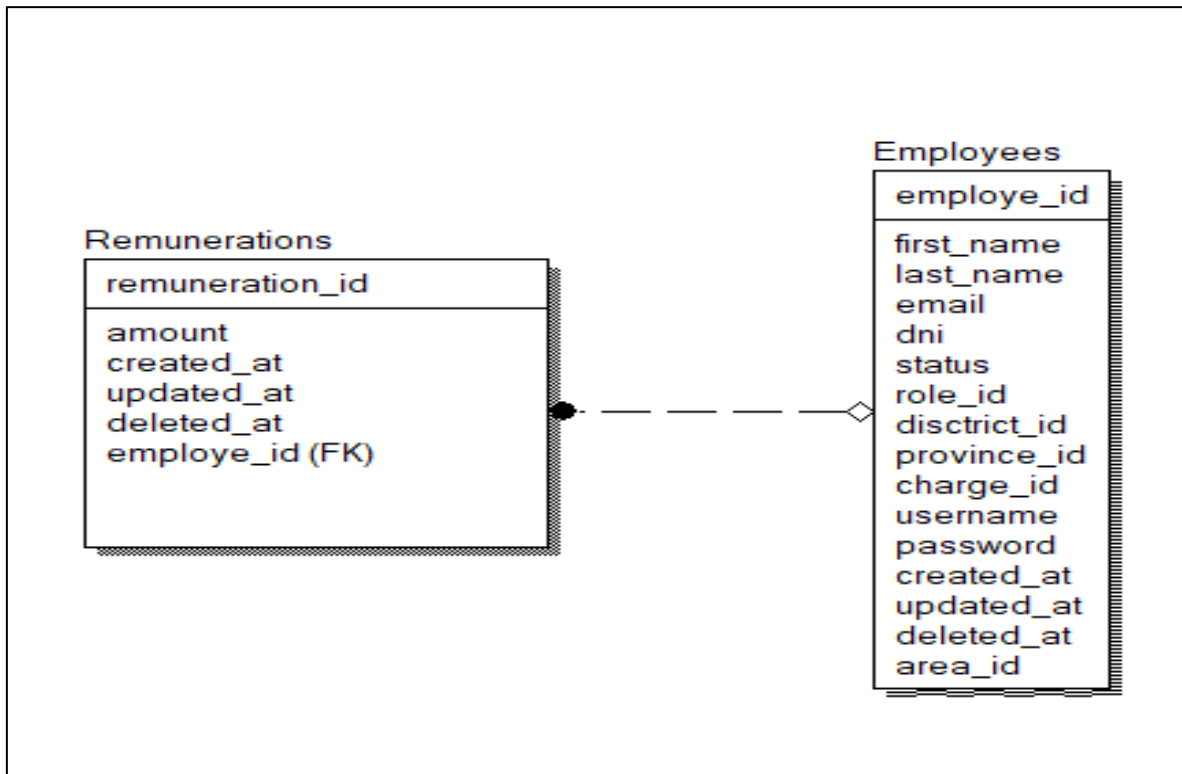


Figura 316. Diagrama lógico de la base de datos RF62

© Fuente: Elaboración propia

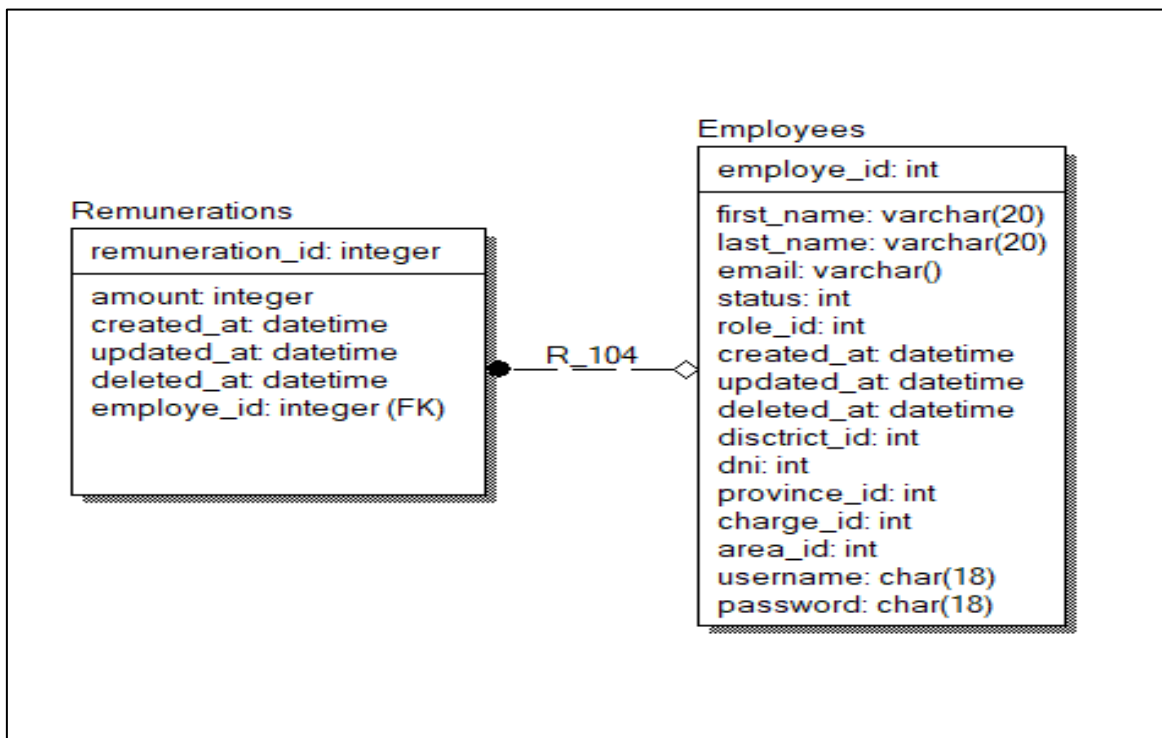


Figura 317. Diagrama físico de la base de datos RF62

Diseño

© Fuente: Elaboración propia

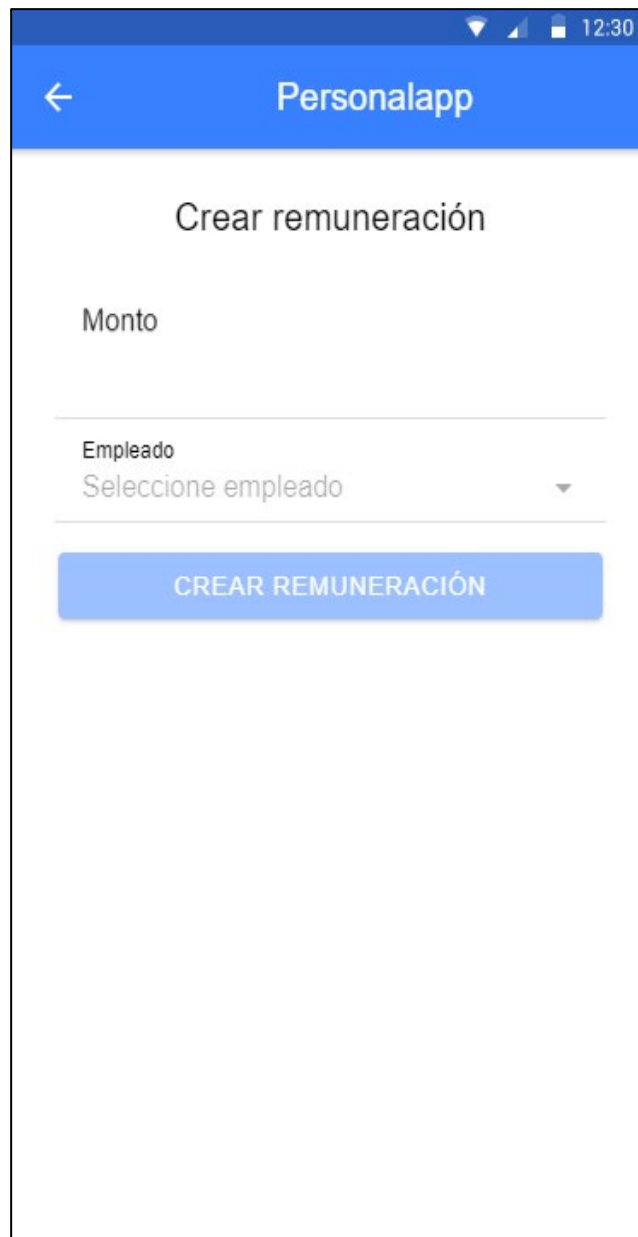


Figura 318. Prototipo de RF62

En la figura 318 se muestra el prototipo para la GUI de crear remuneración del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia



The screenshot shows a mobile application interface titled 'Personalapp'. At the top, there is a blue header bar with a back arrow on the left and the text 'Personalapp' in the center. Below the header, the main content area is white and contains the title 'Crear remuneración'. Underneath the title, there is a text input field labeled 'Monto'. Below the 'Monto' field is a horizontal line, followed by a dropdown menu labeled 'Empleado' with the text 'Seleccione empleado' and a downward arrow. At the bottom of the form is a blue button with the text 'CREAR REMUNERACIÓN' in white capital letters. The top status bar of the phone shows the time as 12:30 and various icons for signal, Wi-Fi, and battery.

Figura 319. Interface de RF62

En la figura 319 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

© Fuente: Elaboración propia

```
public function createRemuneration(Request $request){
    $employee_id = $request->employee_id;
    $response = Remuneration::where('employee_id',$employee_id)->get();
    if(count($response) >= 1 ){
        return response()->json([
            'msg' => 'exceso',
        ], status: 200);
    }else{
        $remuneration = new Remuneration();
        $remuneration->amount = $request->amount;
        $remuneration->employee_id = $request->employee_id;
        $remuneration->save();
        return response()->json([
            'msg' => 'exito',
        ], status: 200);
    }
}
```

Figura 320. Código de RF62

En esta imagen 320 de la presente tesis se puede observar el código fuente del requerimiento funcional 62.

Requerimiento RF63 – Actualizar remuneración

El requerimiento 63, tiene como objetivo que en la aplicación se debe poder actualizar el monto de remuneración de los empleados, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 63).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

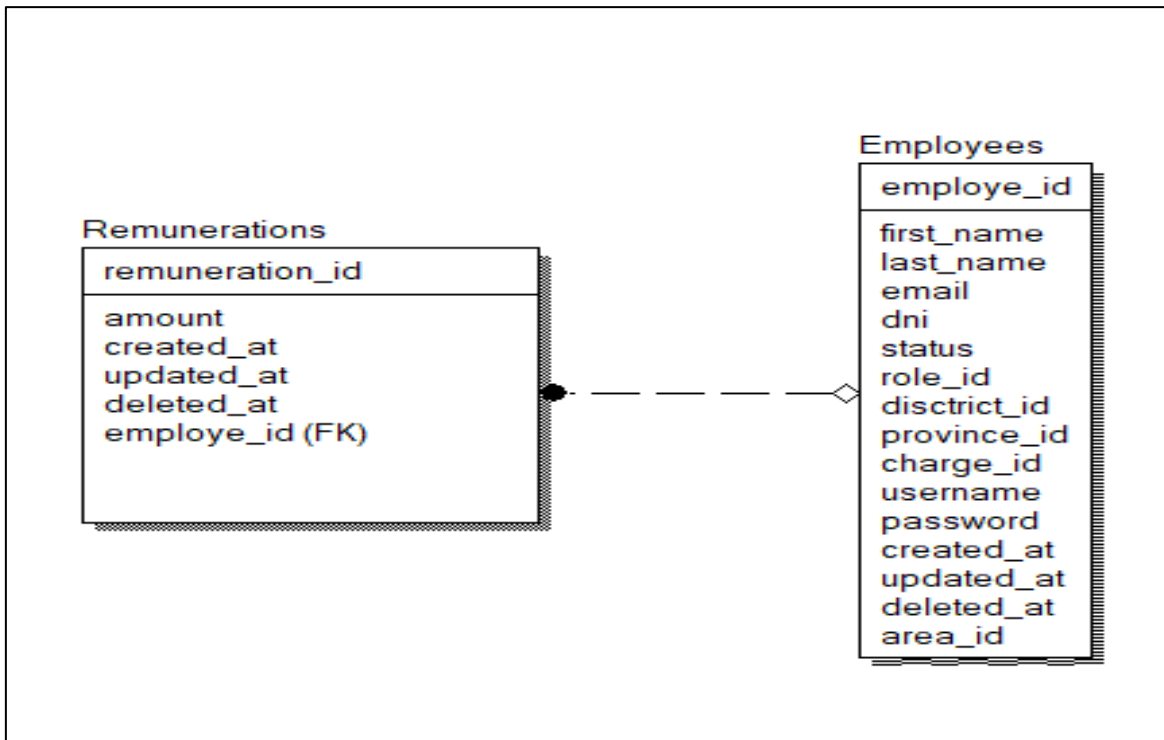


Figura 321. Diagrama lógico de la base de datos RF63

© Fuente: Elaboración propia

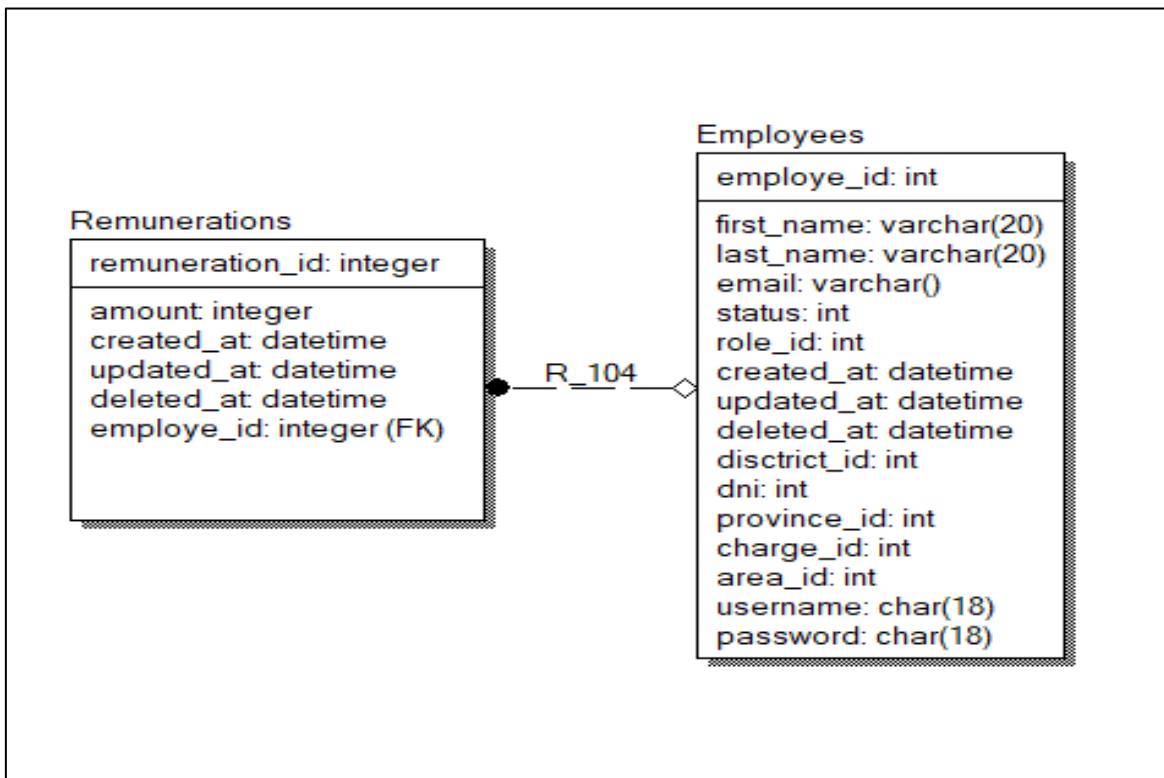
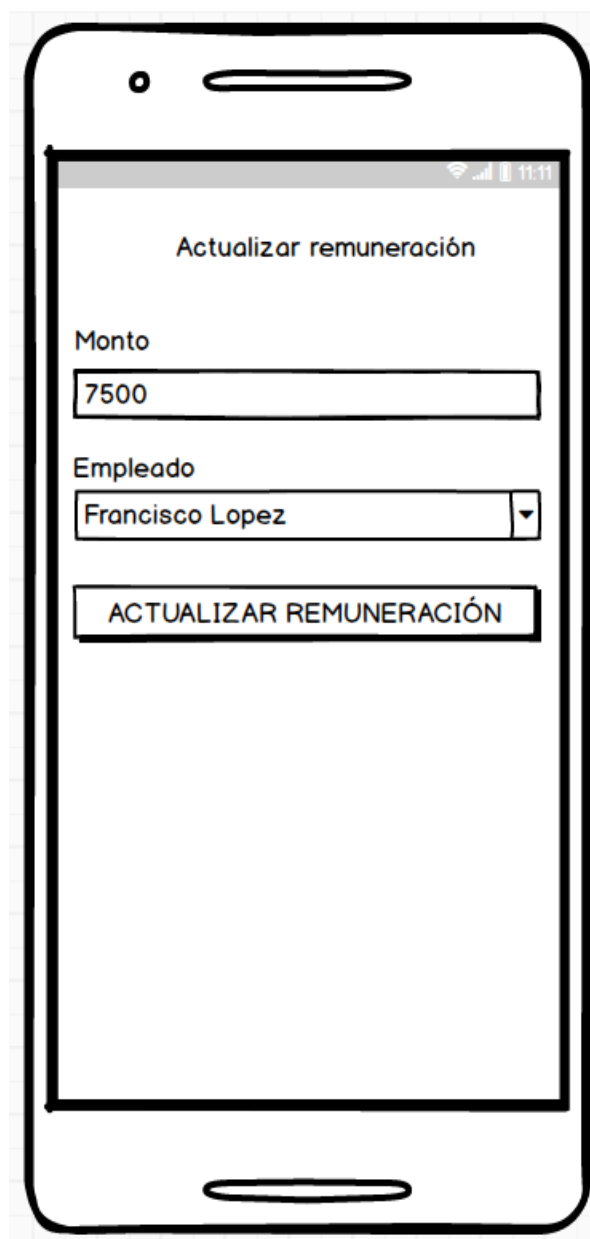


Figura 322. Diagrama físico de la base de datos RF63

Diseño

© Fuente: Elaboración propia



Actualizar remuneración

Monto

Empleado

ACTUALIZAR REMUNERACIÓN

Detailed description: The image shows a mobile application prototype on a smartphone. The screen displays a form titled 'Actualizar remuneración'. It contains three main input fields: a text box for 'Monto' with the value '7500', a dropdown menu for 'Empleado' with 'Francisco Lopez' selected, and a large button labeled 'ACTUALIZAR REMUNERACIÓN'. The status bar at the top shows signal strength, Wi-Fi, and the time 11:11. The phone's home indicator is visible at the bottom.

Figura 323. Prototipo de RF63

En la figura 323 se muestra el prototipo para la GUI de actualizar remuneración del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

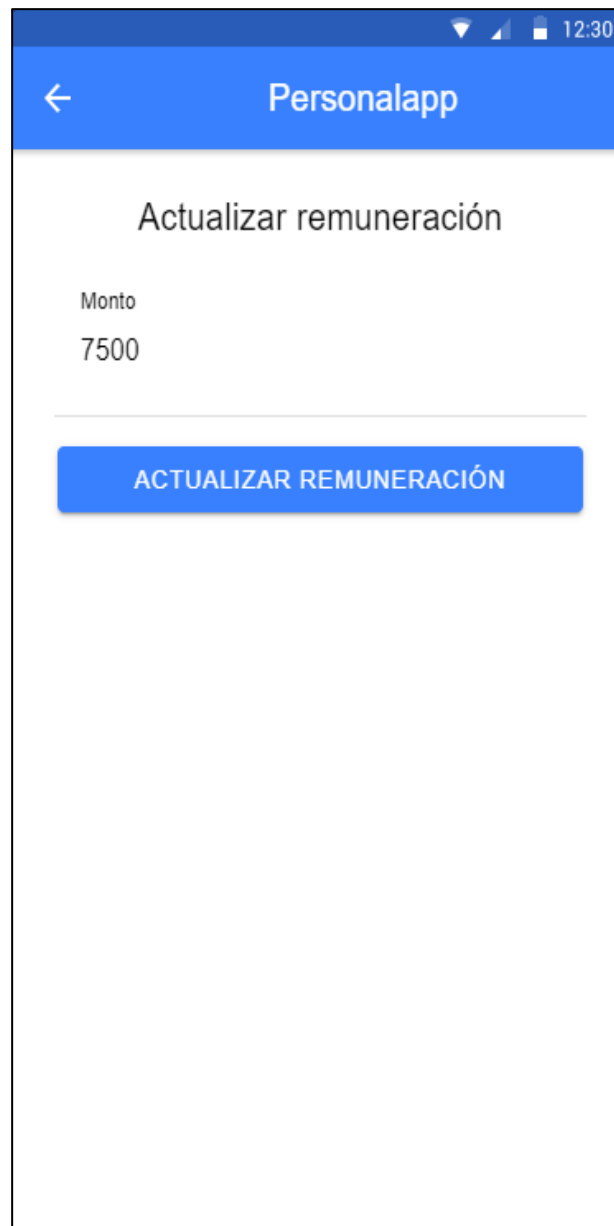


Figura 324. Interface de RF63

En la figura 324 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function updateRemuneration(Request $request,$id) {
    $remuneration = Remuneration::find($id);
    $remuneration->amount = $request->amount;
    $remuneration->save();
    return response()->json([
        'msg' => 'exito',
    ], status: 200);
}
```

Figura 325. Código de RF63

En esta imagen 325 de la presente tesis se puede observar el código fuente del requerimiento funcional 63.

Requerimiento RF64 – Eliminar remuneración

El requerimiento 64, tiene como objetivo que en la aplicación se va poder eliminar la remuneración de los usuarios, dicho requerimiento solo debe hacerlo el usuario “Recursos humanos”. Asimismo, se debe tener en cuenta las restricciones (Para un mayor entendimiento ver las restricciones de la historia de usuario 64).

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

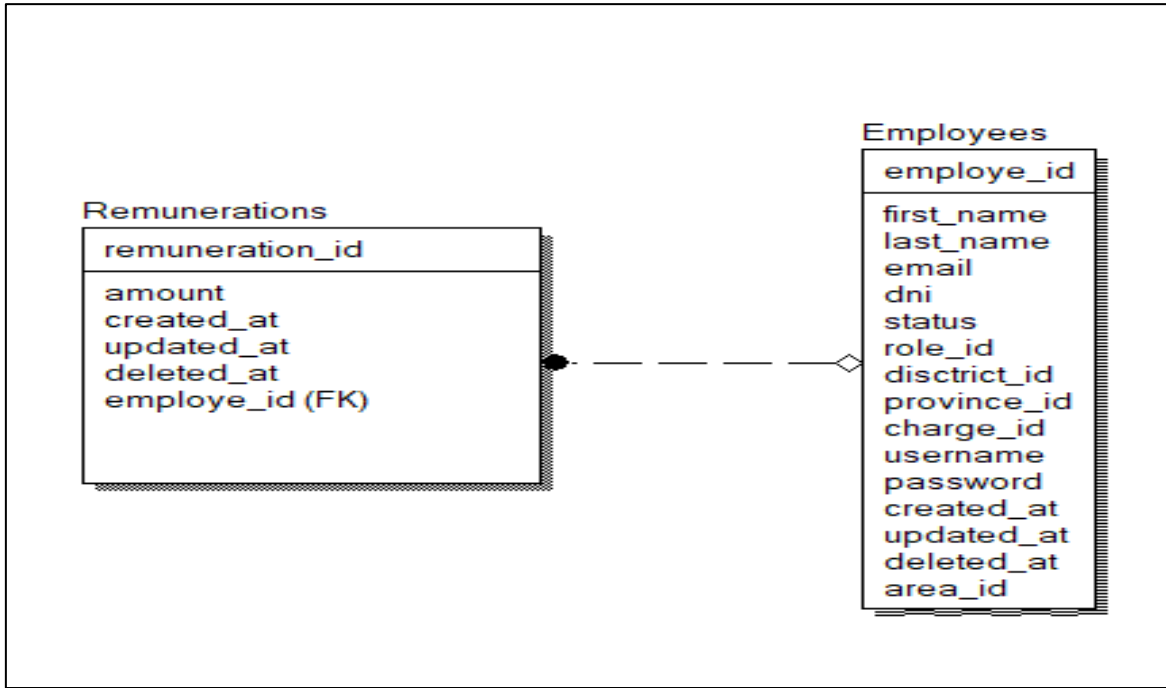


Figura 326. Diagrama lógico de la base de datos RF64

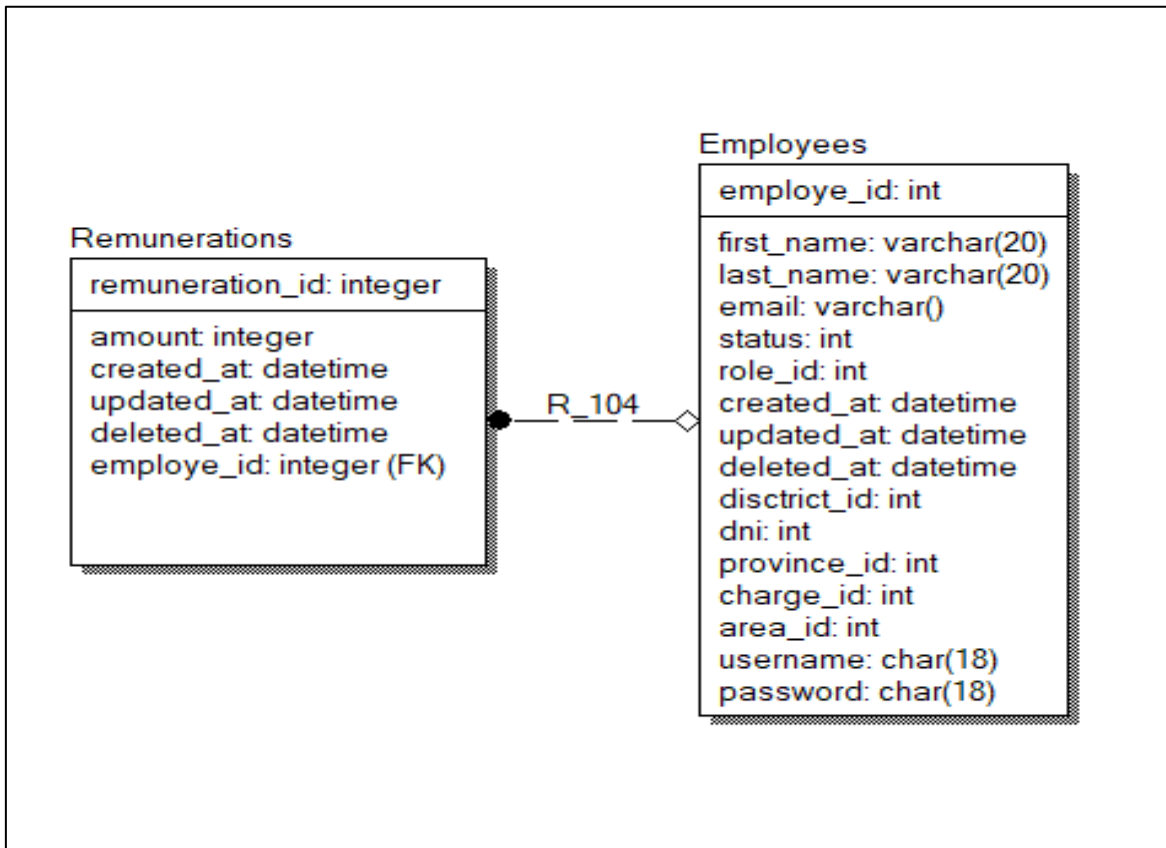


Figura 327. Diagrama físico de la base de datos RF64

Diseño

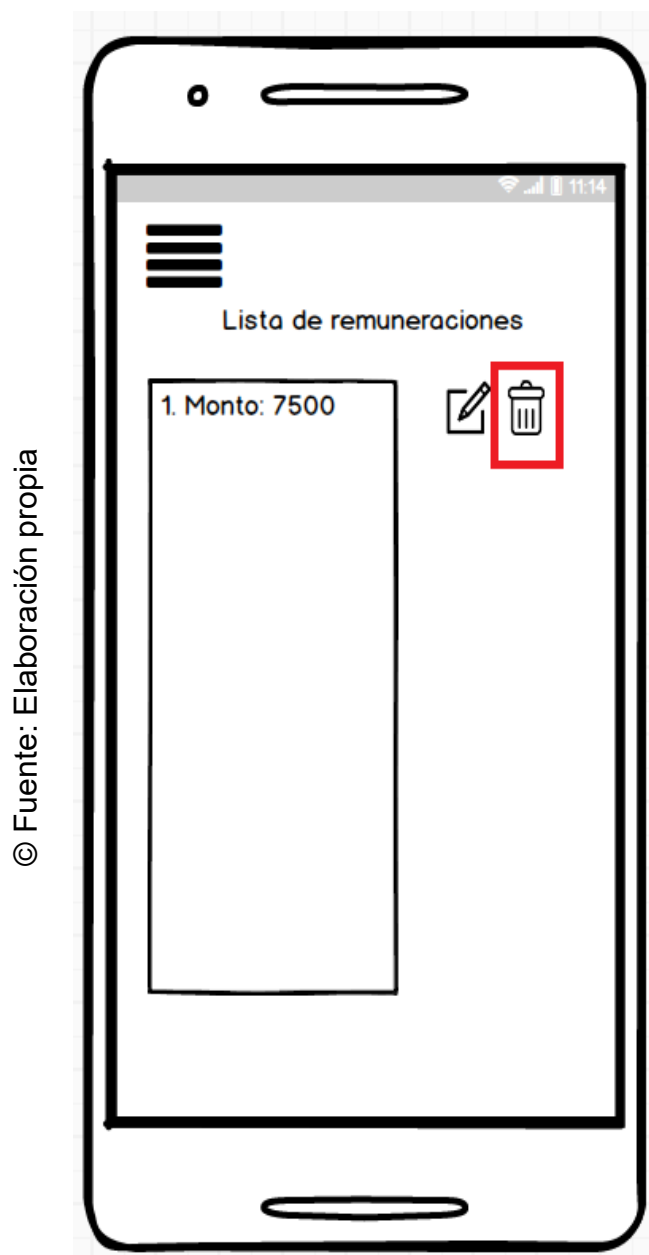


Figura 328. Prototipo de RF64

En la figura 328 se muestra el prototipo para la GUI de eliminar remuneración del usuario recursos humanos que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

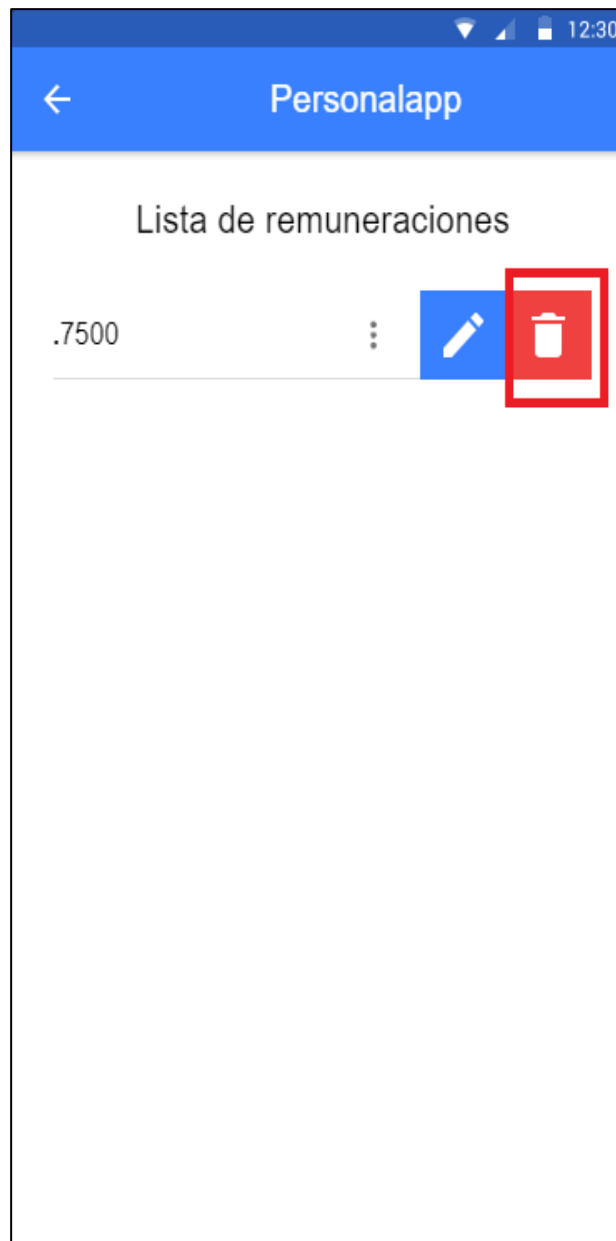


Figura 329. Interface de RF64

En la figura 329 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
public function deleteRemuneration(Request $request){
    $remuneration = Remuneration::findOrFail($request->id);
    $remuneration->delete();
    return response()->json([
        'msg' => 'exito',
    ], status: 200);
}
```

Figura 330. Código de RF64

En esta imagen 330 de la presente tesis se puede observar el código fuente del requerimiento funcional 64.

Requerimiento RF65 – Redirigir al home

El requerimiento 65, tiene como objetivo que en la aplicación se debe poder redirigir al Home de la app haciendo clic en Personal App, es decir redireccionará a la página de marcar asistencia, dicho requerimiento solo debe hacerlo los usuarios “Administrador”, “Recursos humanos” y “Obrero”.

A continuación, en las dos imágenes siguientes podrán observar el diseño de la base de datos lógico y físico de tablas involucradas de este requerimiento funcional. Para mayor información de cómo se llegó a realizar la base de datos, por favor dirigirse a la página 260 de la presente tesis ya que ahí hay un texto que deben de leer.

© Fuente: Elaboración propia

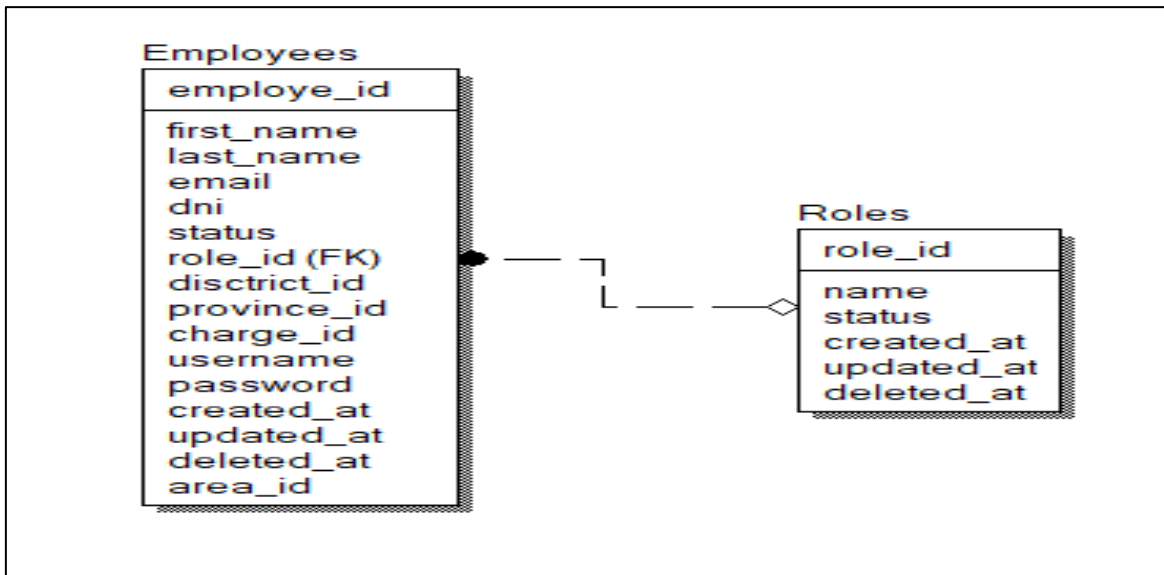


Figura 331. Diagrama lógico de la base de datos RF65

© Fuente: Elaboración propia

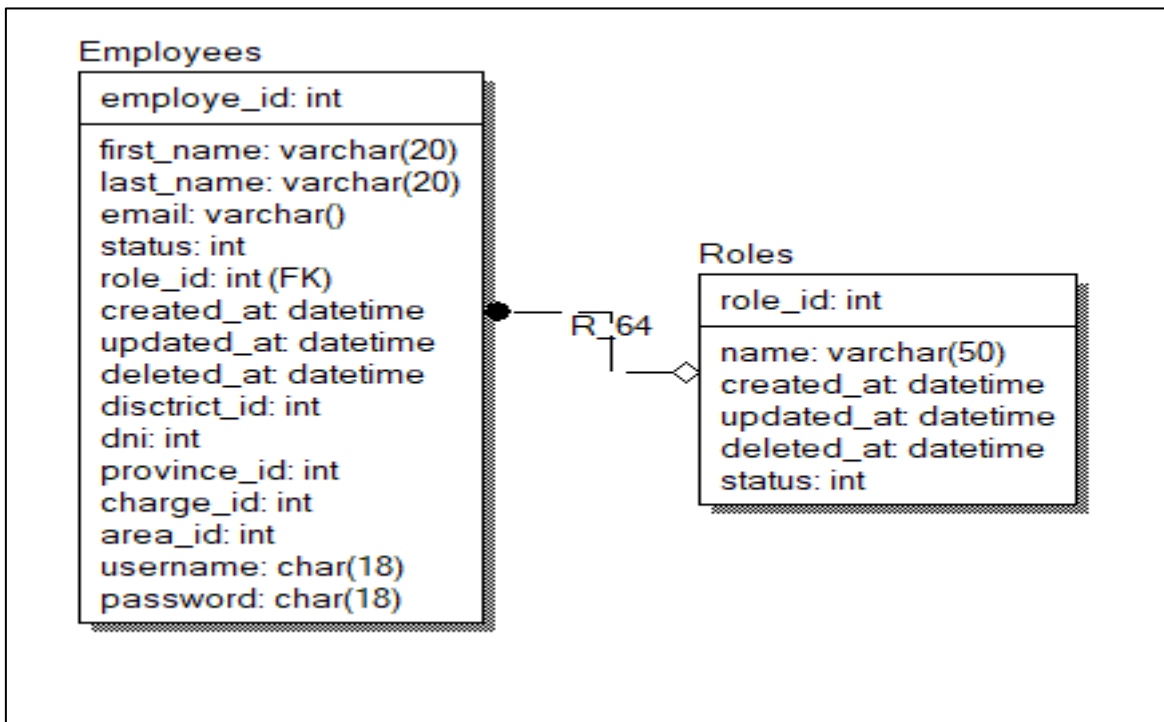


Figura 332. Diagrama físico de la base de datos RF65

Diseño

© Fuente: Elaboración propia



Figura 333. Prototipo de RF65

En la figura 333 se muestra el prototipo para la GUI redirigir al home de los usuarios del sistema que fue aprobada por el dueño del producto, este se realizó con la aplicación Balsamic Mockup.

Implementación

© Fuente: Elaboración propia

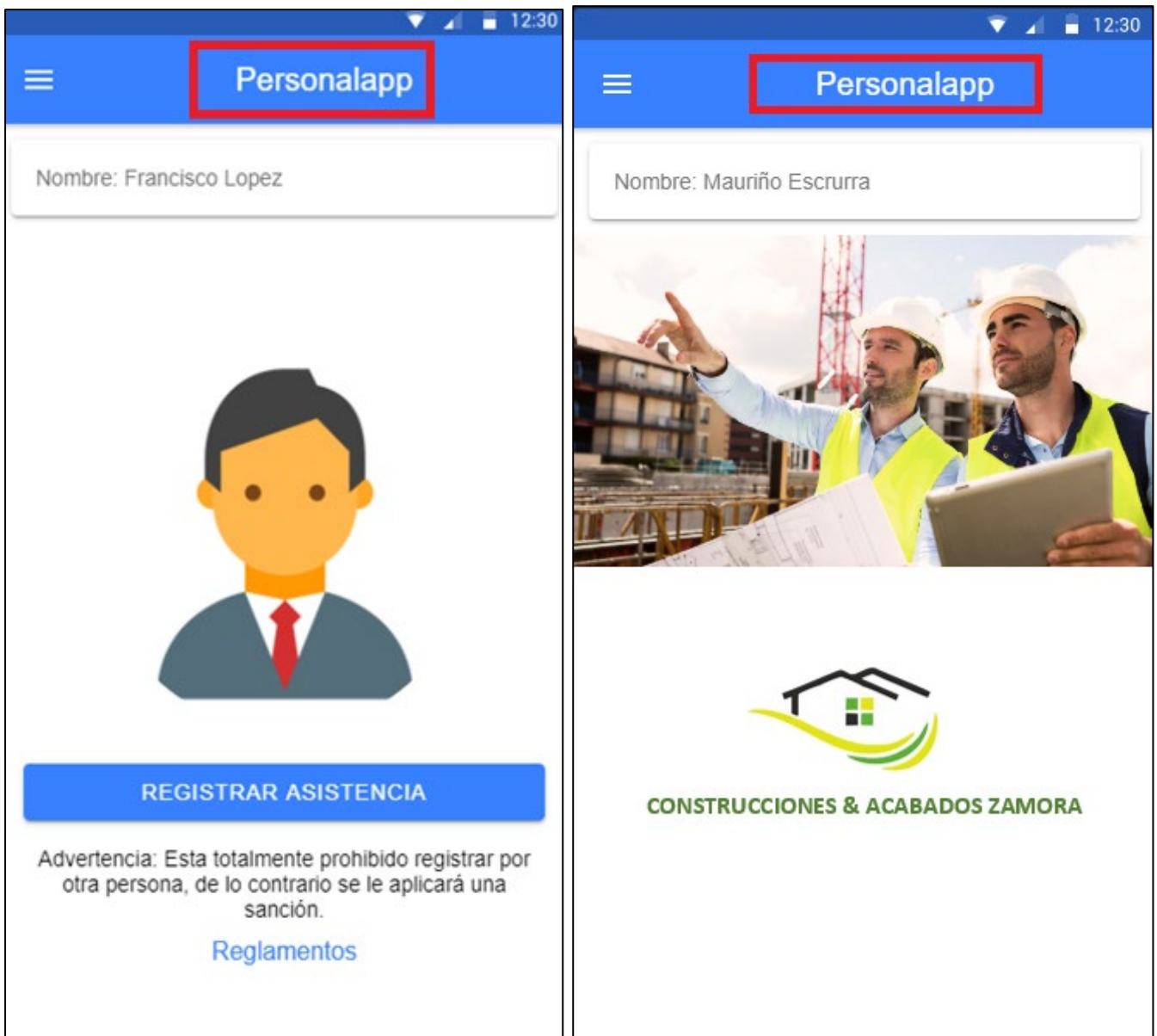


Figura 334. Interface de RF65

En la figura 334 se muestra la interface desarrollada por el equipo de trabajo y aprobada por el dueño del producto.

Código

```
async redirectHome(){  
  this.navCtrl.navigateRoot('/home');  
}
```

Figura 335. Código de RF65

En esta imagen 335 de la presente tesis se puede observar el código fuente del requerimiento funcional 65.

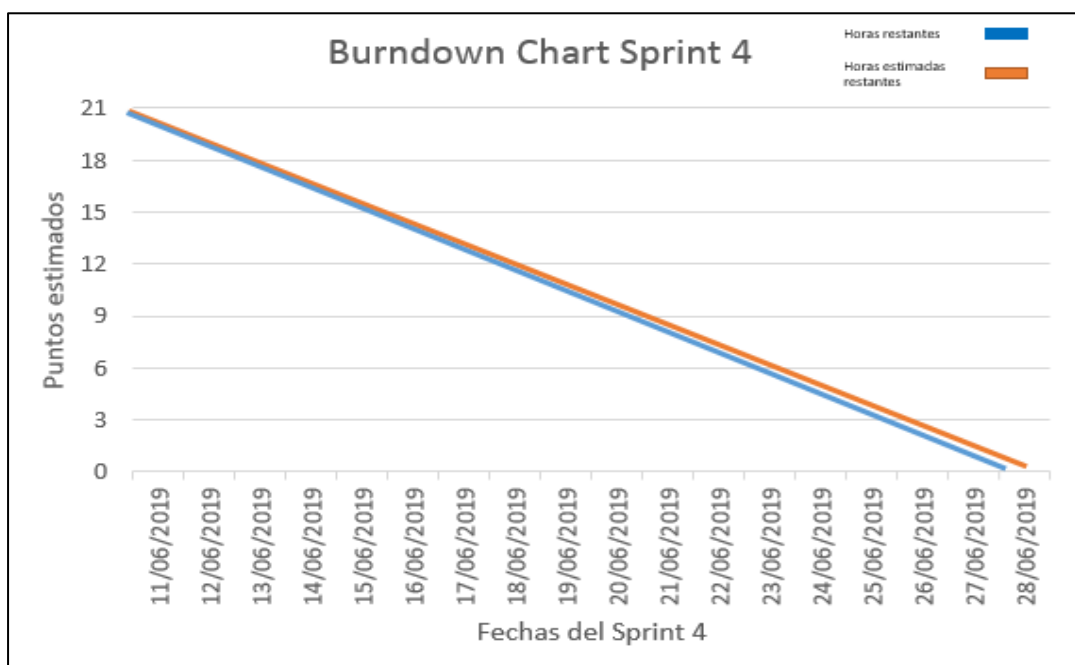



Figura 336. Burndown Chart Sprint 4

Para este sprint, se puede apreciar en esta figura 336, que el tiempo real de Scrum (línea naranja, ya que así dice en su documentación y nosotros que estamos laborando en consultoría también utilizamos este Burndown Chart ya que trabajamos con Scrum y sabemos cómo se usa) va acorde al tiempo planificado (línea azul, en el cual es el tiempo que se demora el equipo de desarrollo en acabar sus actividades), e incluso se terminó mucho antes de lo planificado. Bajo este escenario se puede determinar que se ha terminado todo correctamente.

Acta de reunión de entrega del Sprint 4

Acta de reunión de entrega del Sprint 4		
Fecha	27/06/2019	
Hora	3:00 pm	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Responsable	Lorenzo Palacios	Scrum Master
Participantes	Javier Zamora	Product Owner
	Maurihño Ecurra	Analista
	Luis Zamora	Programador
	Julio Sosa	Calidad
Tema	Presentación del Sprint 4	
Resumen de la reunión		
<p>En esta reunión se mostraron las pantallas de la aplicación a nivel funcional de acuerdo a las historias pactadas en la Planificación del Sprint 4.</p> <p>El cliente quedo satisfecho con el trabajo realizado y dio el visto bueno para el acta de implementación de la aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.</p>		

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

© Fuente: Elaboración propia

Figura 337. Acta de reunión de entrega del Sprint 4

El tema de reunión de esta acta es sobre la presentación de entrega del sprint 4.

Resumen del Sprint 4

Tabla 78: *Resumen del Sprint 4*

Total, de historias	53
Historias terminadas	53
Historias por terminar	0
Avance	100%

Fuente: Elaboración propia

Las pruebas realizadas a las funcionalidades durante este sprint fueron exitosas sin ningún error de diseño ni de código fuente.

Retrospectiva del Sprint 4

Al final del Sprint, el equipo Scrum se reunió para recibir la respuesta del Scrum Master, para saber cómo le fue en la reunión con el Product Owner, resulta que el producto se entregó sin problemas y con el cliente satisfecho.

Cosas Positivas:

- Lograron cumplir los resultados
- El apoyo del equipo en todo momento

Cosas Negativas:

- Ninguna

A continuación, se verán en las dos imágenes 338 – 339 el diagrama lógico y físico general de la base de datos de la app móvil respectivamente con todas las entidades incluidos sus atributos y relaciones correspondientes.

© Fuente: Elaboración propia

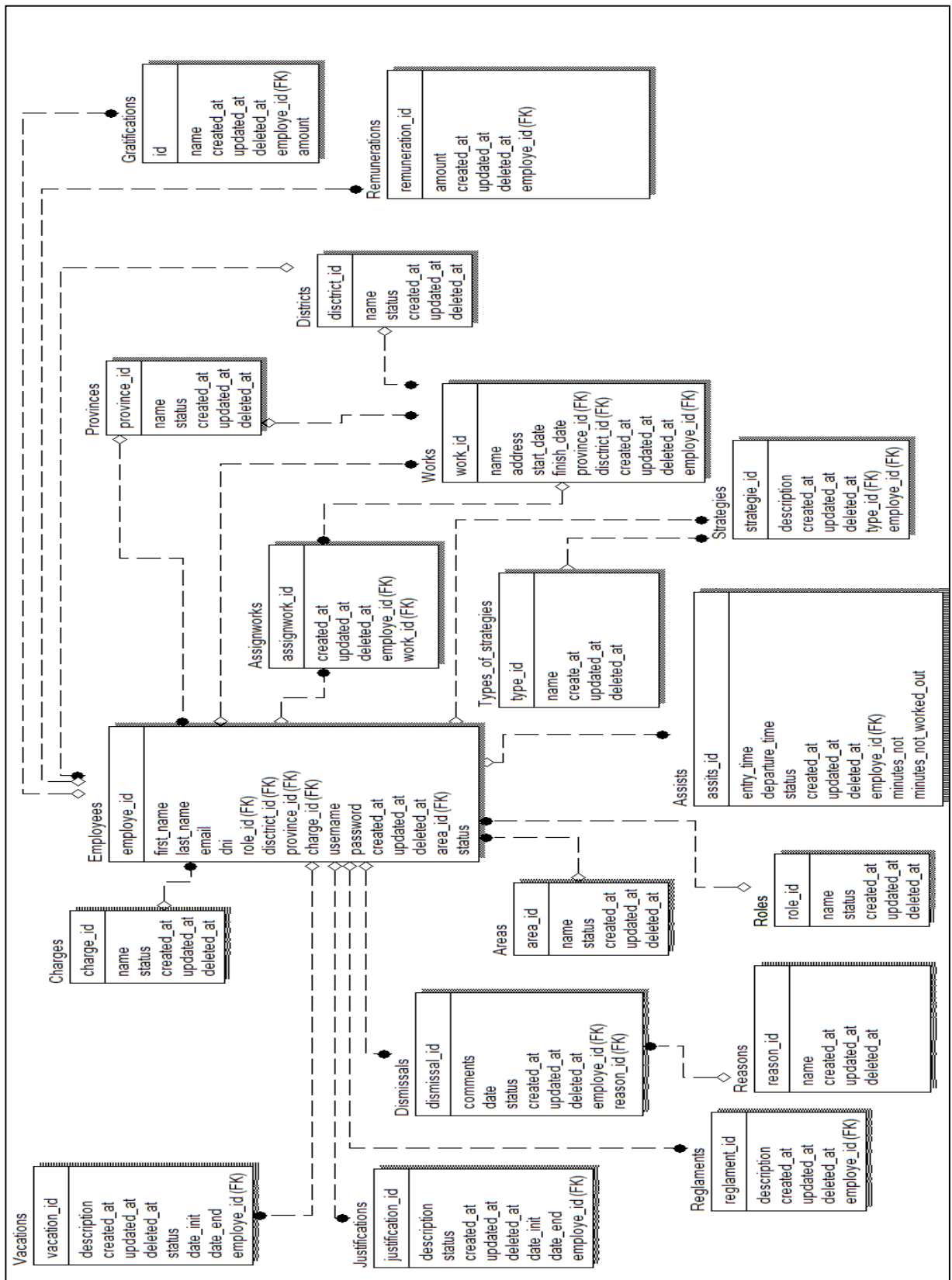


Figura 338. Diagrama lógico de la base de datos – General

© Fuente: Elaboración propia

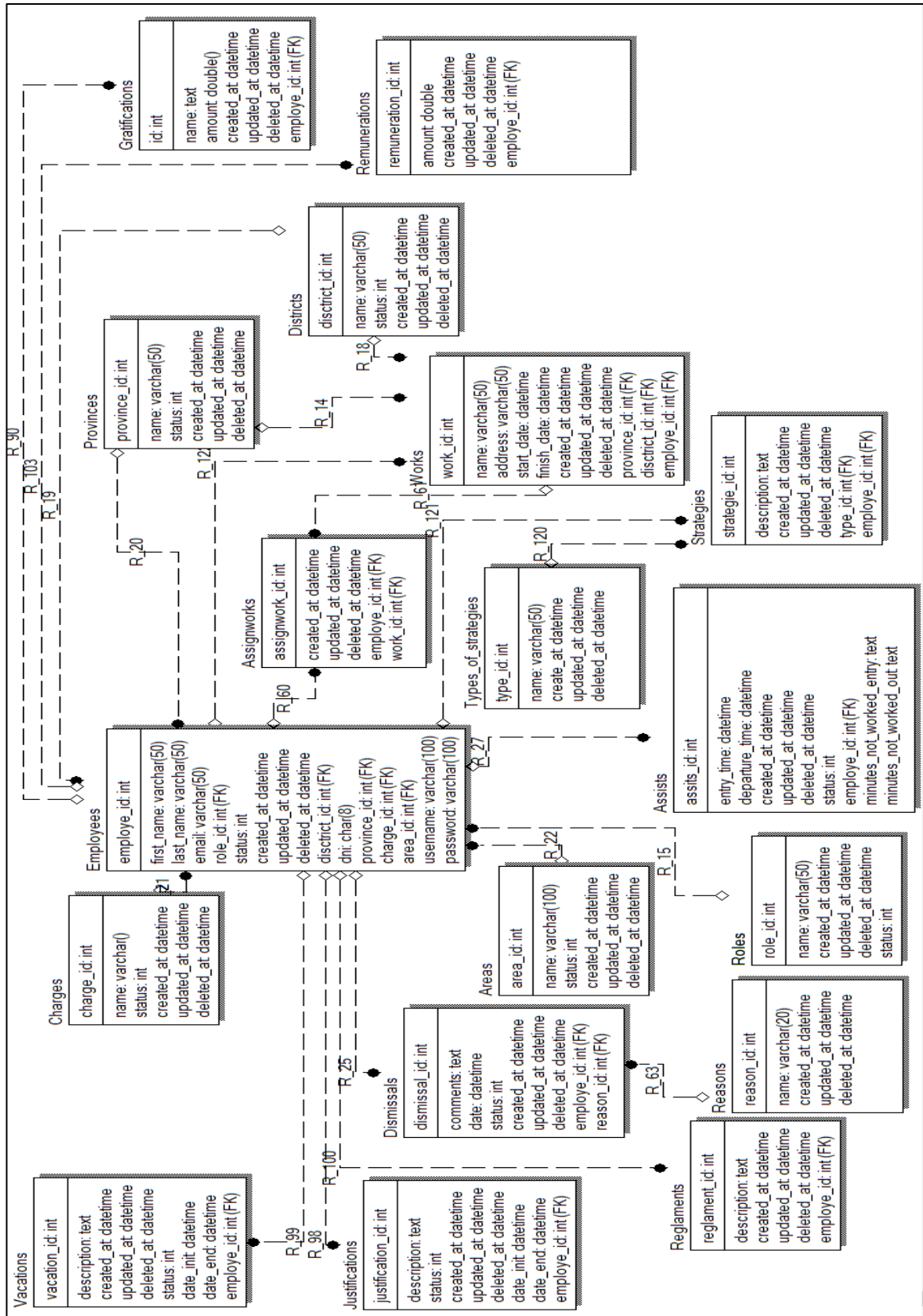


Figura 339. Diagrama físico de la base de datos – General

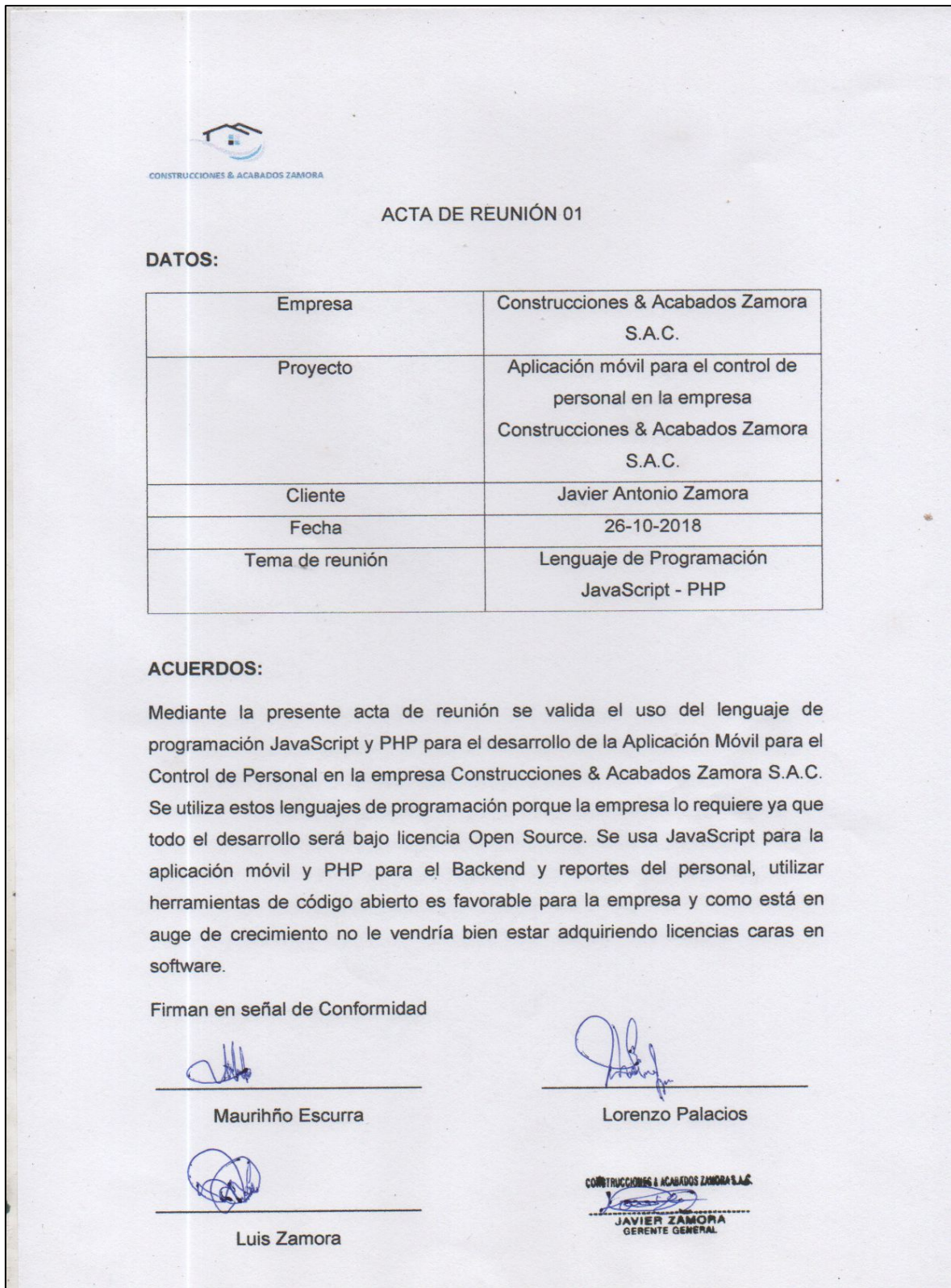



Figura 340. Acta de reunión 1

El tema de reunión de esta acta es sobre el lenguaje de programación a utilizar que son JavaScript y PHP.


CONSTRUCCIONES & ACABADOS ZAMORA

ACTA DE REUNIÓN 02

DATOS:

Empresa	Construcciones & Acabados Zamora S.A.C.
Proyecto	Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.
Cliente	Javier Antonio Zamora
Fecha	26-10-2018
Tema de reunión	Base de datos MySQL

ACUERDOS:

Mediante la presente acta de reunión se valida el uso del gestor de base de datos MySQL para el desarrollo de la Aplicación Móvil para el Control de Personal en la empresa Construcciones & Acabados Zamora S.A.C. Se utiliza dicho gestor porque la empresa lo requiere ya que todo el desarrollo será bajo licencia Open Source, utilizar herramientas de código abierto es favorable para la empresa y como está en auge de crecimiento no le vendría bien estar adquiriendo licencias caras en software.

Firman en señal de Conformidad




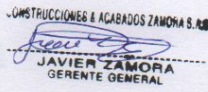
 Maurihño Ecurra	 Lorenzo Palacios
 Luis Zamora	 Gerente

Figura 341. Acta de reunión 2

El tema de reunión de esta acta es sobre la base de datos a utilizar que es MySQL.

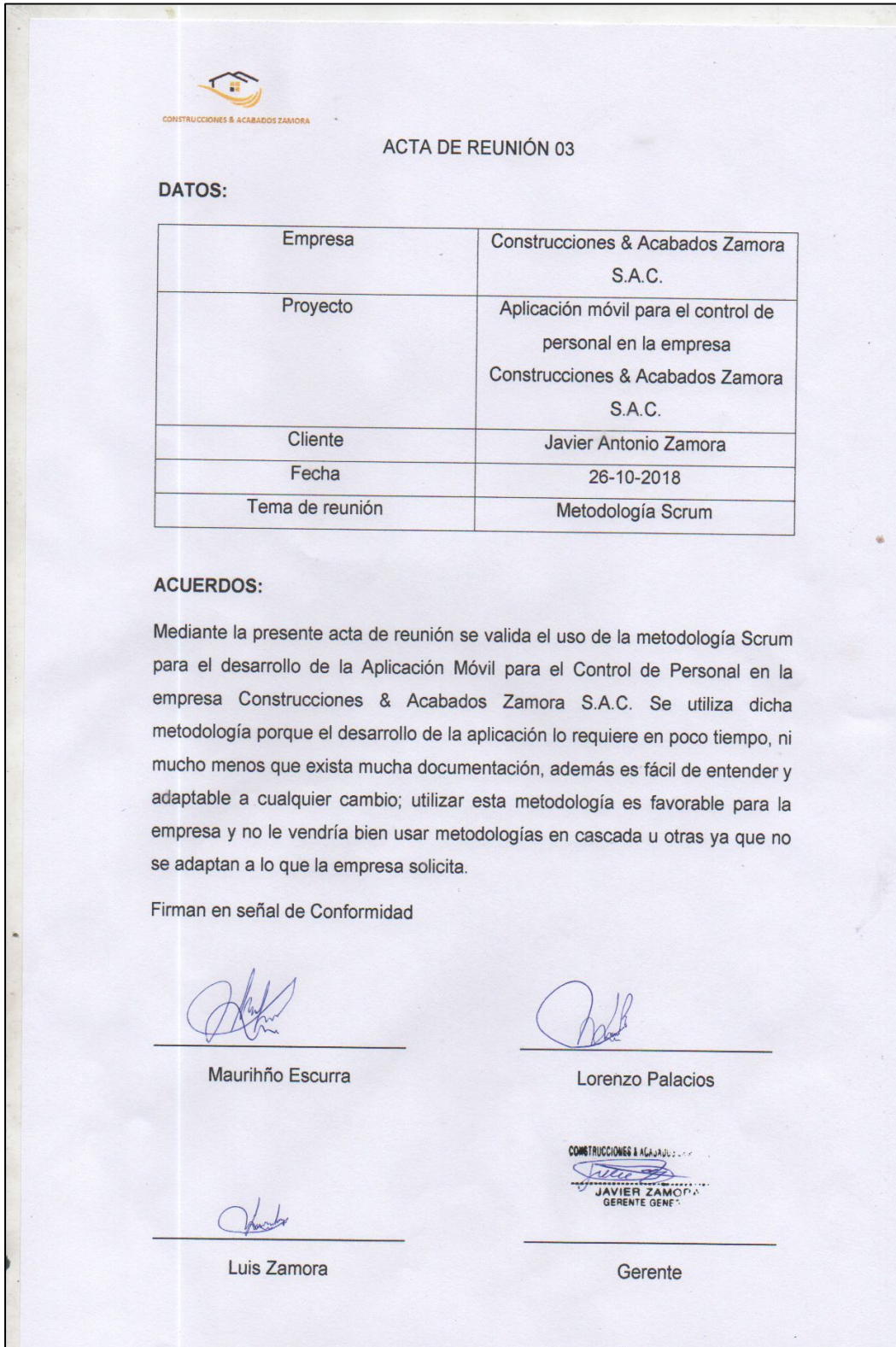


Figura 342. Acta de reunión 3

El tema de reunión de esta acta es sobre la metodología a utilizar que son Scrum.

ACTA DE REUNIÓN 04


DATOS:

Empresa	Construcciones & Acabados Zamora S.A.C.
Proyecto	Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.
Cliente	Javier Antonio Zamora
Fecha	26-10-2018
Tema de reunión	Aplicación Híbrida

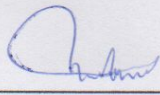
ACUERDOS:

Mediante la presente acta de reunión se valida el uso de una aplicación híbrida para el desarrollo de la Aplicación Móvil para el Control de Personal en la empresa Construcciones & Acabados Zamora S.A.C. Se utiliza dicho tipo de aplicación porque el desarrollo de la app lo requiere en diversas plataformas, además es fácil de desarrollar y portable en cualquier sistema operativo que hay en la actualidad.


Firman en señal de Conformidad



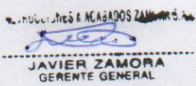
 Maurihño Escurra



 Lorenzo Palacios



 Luis Zamora



 Gerente

© Fuente: Elaboración propia

Figura 343. Acta de reunión 4

El tema de reunión de esta acta es sobre el tipo de aplicación a elegir y se eligió el híbrido, porque así lo requiere el cliente para que este en multiplataforma la app.

ACTA DE REUNIÓN 05

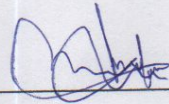
DATOS:

Empresa	Construcciones & Acabados Zamora S.A.C.
Proyecto	Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.
Cliente	Javier Antonio Zamora
Fecha	26-10-2018
Tema de reunión	Arquitectura MVC

ACUERDOS:

Mediante la presente acta de reunión se valida el uso de la arquitectura MVC para el desarrollo de la Aplicación Móvil para el Control de Personal en la empresa Construcciones & Acabados Zamora S.A.C. Se utiliza dicha arquitectura porque los servicios están en Laravel y este se basa netamente en MVC, además el marco de trabajo Ionic también.

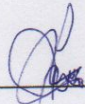
Firman en señal de Conformidad




Maurihño Escurra



Lorenzo Palacios



Luis Zamora

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

 JAVIER ZAMORA
 GERENTE GENERAL

Gerente

Figura 344. Acta de reunión 5

El tema de reunión de esta acta es sobre la arquitectura del software a elegir y se eligió el MVC, porque como la app es un híbrido es necesario sí o sí que la arquitectura sea MVC.



Figura 345. Tablero Scrum Inicial

Se puede apreciar en esta figura 345 el tablero scrum, inicialmente este vacío porque aún no hay actividades que hacer. Hay varias maneras de cómo hacer un tablero scrum, pero así también se puede trabajar, cosa que es válido. En la parte de Historias de Usuario van todas las funcionalidades que el cliente quiere que haga la aplicación, en la parte de To Do van todas las actividades o tareas que el equipo de desarrollo debe de hacer, en la parte Doing van todas las actividades que el equipo de desarrollo actualmente está trabajando y en la parte de Done van todas las actividades que se han culminado en esa historia de usuario.



Figura 346. Tablero Scrum con Actividades

Se puede apreciar en esta figura 346 el tablero scrum con actividades ya elaboradas y ejecutadas actualmente por el equipo de desarrollo. Cada persona es responsable de mover sus actividades en las columnas del tablero scrum. Este tablero se debe de actualizar diariamente y debe de haber mucha comunicación entre el equipo de desarrollo y el cliente para que la aplicación cuando pase a producción sea de calidad.

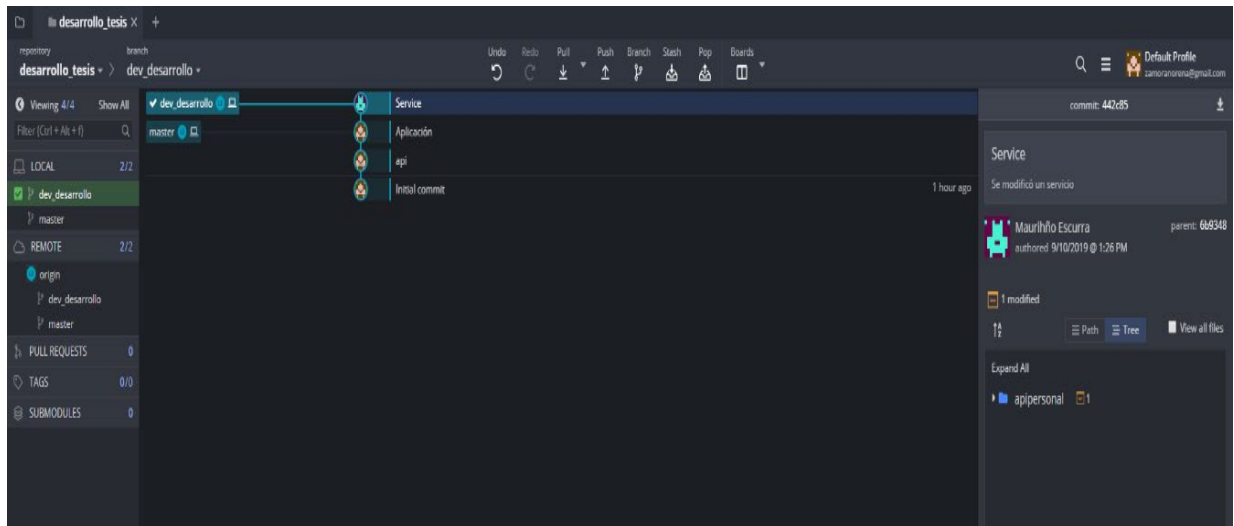


Figura 347. Herramienta GitKraken

Se puede apreciar en esta figura 347 la herramienta GitKraken en el cual es muy usada por diversos desarrolladores en la actualidad. Nosotros la usamos para tener todo el control de nuestros cambios del código fuente de la aplicación para hacer un push, pull, clonar, crear ramas, etc. Esta herramienta es muy fácil de usar ya que es muy intuitiva y te evita hacer los comandos por consola ya que te puedes equivocar.

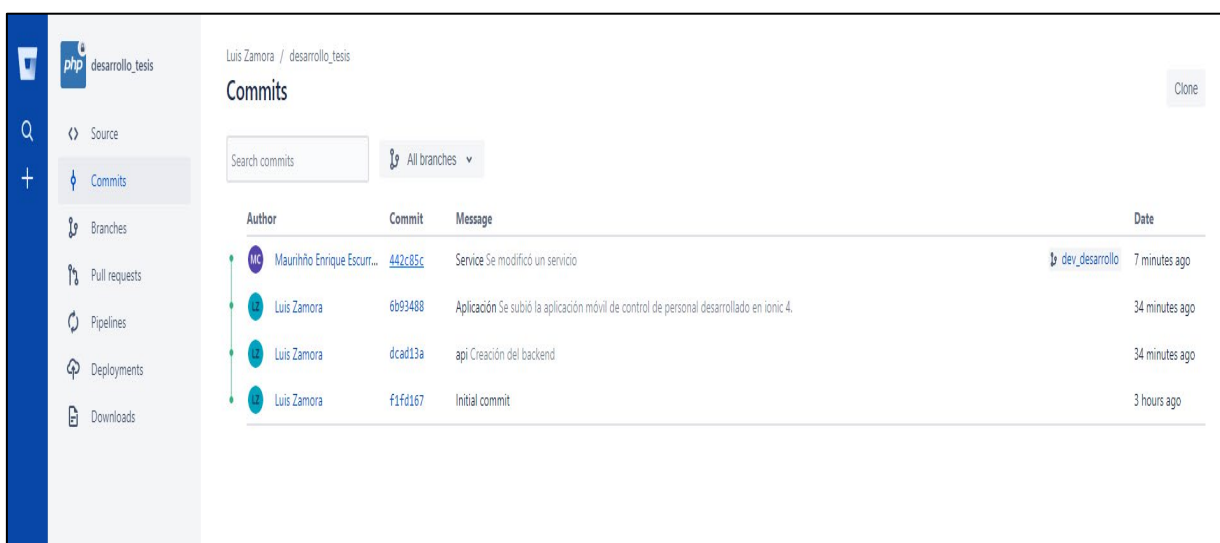


Figura 348. Commit del Repositorio Bitbucket de la aplicación

Se puede apreciar en esta figura 348 los commit del repositorio Bitbucket de la aplicación. En este repositorio está todo el código fuente de nuestra aplicación móvil. Lo único que se debe de hacer para tener el código es tener acceso al repositorio, clonarlo, crear una rama y recién ahí podrás modificar todo el código fuente. Los commit son los mensajes que hacen los desarrolladores cuando suben sus cambios del código fuente al repositorio. Por ejemplo, si dos o más desarrolladores hacen cambios en el módulo de login, uno hace una función para enviar el Request body al servidor y el otro hace la parte visual del diseño, ahí es donde los dos ponen un commit o mejor dicho un mensaje especificando que han hecho y todo esto se ve reflejado en el repositorio cuando entran a ese módulo. Actualmente la aplicación móvil está al 100% en código fuente con metodología incluida, cada uno se dividió el desarrollo en un 50%. Hay dos ramas para la app móvil, una que se llama master que es la que va estar en producción y la otra es dev_desarrollo que es la rama de desarrollo en el cual están todos los commits de cada uno de nosotros.

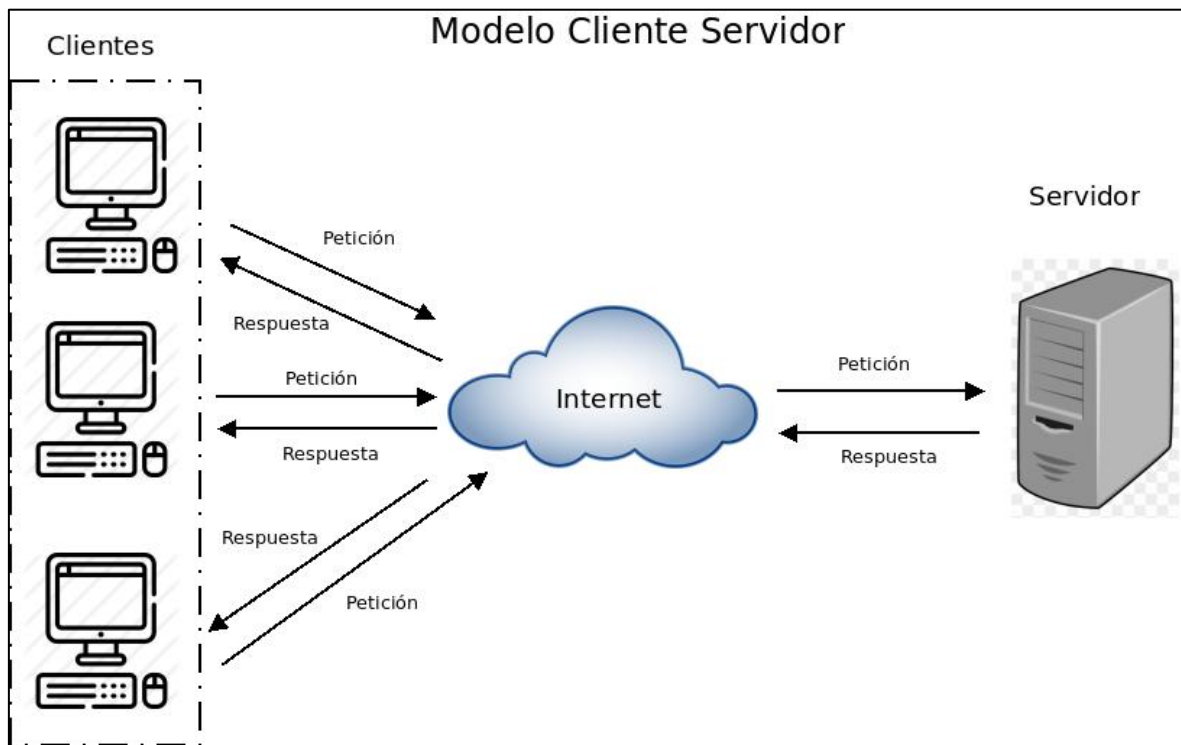


Figura 349. Arquitectura Cliente – Servidor

Se puede apreciar en esta figura 349 la arquitectura cliente servidor de la aplicación móvil de la presente investigación ya que es un modelo de diseño de software que actualmente es muy utilizado por diversos desarrolladores a nivel mundial en sus diversos proyectos que hacen. Esta arquitectura tiene dos partes fundamentales el cliente y el servidor, el primero de ellos son las estaciones de trabajos, los dispositivos, los usuarios finales que realizan peticiones al servidor mientras que el servidor se encarga de responder esas peticiones por parte de los usuarios ya que funciona como un sistema gestor de base de datos y como un depósito de datos. Para que varios usuarios interactúen con las aplicaciones desde cualquier punto o lugar donde se encuentre debe de existir la nube o mejor dicho el internet, el internet está en medio de todas las cosas para la comunicación estable entre el cliente y el servidor.

© Fuente: ic7wtrier (2016)



Figura 350. Interoperabilidad de aplicaciones híbridas

Se puede apreciar en esta figura 350 la interoperabilidad de las aplicaciones híbridas que hay en la actualidad, es mejor desarrollar una aplicación híbrida que una nativa ya que trae muchas ventajas una de ellas es la portabilidad en diferentes sistemas operativos como android, iOS y web. Desarrollas solo una vez y es adaptable para cualquier plataforma. Las aplicaciones híbridas son el futuro de desarrollo de software para ello debe de haber muchos ya desarrolladores que programen en ese tipo de tecnología para la interoperabilidad e intercambio de información de los sistemas. En el desarrollo de nuestra aplicación móvil se utiliza toda esta tecnología híbrida actualmente.



CONSTRUCCIONES & ACABADOS ZAMORA

Acta de Pruebas - 15.11.2019
“Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.”

Fecha	15/11/2019
Lugar	Construcciones & Acabados Zamora S.A.C.

Actividades Realizadas			
Fecha	15/11/2019	Módulos	Acta de Pruebas
1. Verificar todas las funcionalidades de la aplicación móvil.			
2. Probar y/o testear toda la aplicación móvil.			
3. Se cierra el acta de pruebas de la aplicación firmado y sellado por los involucrados.			

Actividades por Realizar			
Módulos	Pase a Producción		
Actividades	Responsable	Fecha Realización	
Pruebas de regresión	Julio Sosa	15/11/2019	

Observaciones
Las funcionalidades de la aplicación se ejecutaron correctamente en los dispositivos móviles.

Participantes	Firma
Lorenzo Palacios Scrum Master	
Luis Zamora Programador	
Maurihño Escurra Analista	
Julio Sosa Calidad	

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

JAVIER ZAMORA
GERENTE GENERAL

© Fuente: Elaboración propia

Figura 351. Acta de Pruebas



CONSTRUCCIONES & ACABADOS ZAMORA

Acta de Pase a Producción - 18.11.2019
“Aplicación móvil para el control de personal en la empresa Construcciones & Acabados Zamora S.A.C.”
(Mantenimiento)

Fecha	18/11/2019	
Lugar	Construcciones & Acabados Zamora S.A.C.	
Distribución		
Área	Nombres Apellido	Cargo
RRHH – Construcción Civil	Javier Antonio Zamora	Gerente General

Actividades Realizadas			
Fecha	18/11/2019	Módulos	Pase a Producción
<ol style="list-style-type: none"> 1. Reunirse todos los involucrados del pase a producción. 2. Llevar el documento de acta de pase a producción. 3. Entregar todos los scripts, código fuente, ejecutables a la oficina de Sistemas. 4. Todos los involucrados en constante comunicación. 5. Se sube la aplicación en un hosting servidor. 6. Generar los APK en los diferentes dispositivos móviles 7. Se realizan las pruebas funcionales de la aplicación. 8. Si hay errores se debería de solucionar al instante. 9. Se verifica las pruebas correctamente de la aplicación. 10. Se cierra el acta de pase a producción firmado y sellado por los involucrados. 			

Actividades por Realizar			
Módulos	Pase a Producción		
Actividades	Responsable	Fecha Realización	
Seguimiento de la aplicación	Lorenzo Palacios	18/11/2019	
Revisión del código	Luis Zamora	18/11/2019	
Revisión de documentación	Maurihño Escurra	18/11/2019	
Ejecución de pruebas	Julio Sosa	18/11/2019	

Observaciones
La aplicación se ejecutó correctamente en el servidor y en los dispositivos de usuario final.

Participantes	Firma
Lorenzo Palacios Scrum Master	
Luis Zamora Programador	
Maurihño Escurra Analista	
Julio Sosa Calidad	

CONSTRUCCIONES & ACABADOS ZAMORA S.A.C.

JAVIER ZAMORA
GERENTE GENERAL

© Fuente: Elaboración propia

Figura 352. Acta de Pase a Producción

Aquí en estas dos últimas figuras 351 – 352 se observan el acta de pruebas y el acta de pase a producción respectivamente. En el acta de pruebas se puede ver que están todas las actividades realizadas por el analista de calidad al testear la app móvil.

En el acta de pase a producción están todas las actividades realizadas por el equipo de desarrollo y la oficina de Sistemas de la organización para que la app esté ya en producción. Las dos actas están firmadas y selladas por los todos los involucrados y responsables de a app móvil.

Plan Piloto:

La app móvil actualmente ya se encuentra en producción y diferentes usuarios de la organización Construcciones & Acabados Zamora S.A.C. utilizan y prueban las funcionalidades de la app para laborar en cada punto de obra.

Después de haber culminado la app móvil con éxito, se realizó una capacitación de 2 días a los empleados para que ellos mismo se puedan adaptar a las funcionalidades de la app y entiendan de una manera óptima como es en sí el proceso de control de personal.

Actualmente los empleados están satisfechos con la app móvil implementada porque es fácil de usar, es intuitiva y es accesible para ellos, no solo los obreros sino también el administrador y recursos humanos.

La organización Construcciones & Acabados Zamora S.A.C., con su app móvil se suma a la transformación digital que hay ahora en estos tiempos en varias compañías no solo del Perú sino del mundo entero.

Para mayor información de resultados, diríjase a la parte tres de la tesis.

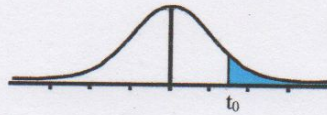


Figura 353. Construcción de casa

Aquí en esta imagen 353 está todo el personal obrero de una obra laborando. Como muestra está el uniforme de trabajo de la organización con su respectivo logo en un obrero. En la puerta de entrada de dicha obra está la app móvil para que hagan uso de sus funcionalidades.

ANEXO 10: Tabla T-Student

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

Anexo 10: T-Student

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
∞	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758