



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Migración de Datos Para el Proceso de Homologación de Ambientes de  
Calidad y Producción en la Sub Gerencia de Sistemas Aseguradoras  
Sistemas Aseguradores, Subsidios y Sociales de la Sede Central de  
EsSalud**

**TRABAJO DE SUFICIENCIA PARA OBTENER EL TÍTULO PROFESIONAL DE:  
Ingeniero de Sistemas**

**AUTOR:**

Anton Alvarado, John Paul (ORCID: 0000-0003-1386-8303)

**ASESOR:**

Mgtr. Marin Verastegui, Wilson Ricardo (ORCID: 000-0001-6175-8112)

**LÍNEA DE INVESTIGACIÓN:**

**SISTEMA DE INFORMACIÓN Y COMUNICACIONES**

LIMA – PERÚ

2021

### **Dedicatoria**

Quiero dedicar este trabajo a mi familia en especial a mi madre por que siempre me brindó su apoyo incondicional e indesmayable procurando siempre inculcárme los valores para ser un buen profesional y mejor persona.

### **Agradecimiento**

A Dios por siempre acompañarme en todo momento y concederme el cumplimiento de mis objetivos trazados.

## ÍNDICE

Resumen.....	6
Abstract.....	7
I. Introducción.....	8
1.1 Descripción del sector.....	8
1.2 Descripción general de la institución.....	11
1.3 Funciones del bachiller dentro de la institución.....	13
1.4 Actividad económica de la institución .....	14
1.5 Funciones del departamento donde labora el bachiller.....	17
1.6 Planteamiento del Problema .....	22
1.6.1 Internacional .....	22
1.6.2 Nacional.....	22
1.6.3 Local .....	23
1.7 Formulación del Problema .....	25
1.7.1 Problema General.....	25
1.7.2 Problema Específico.....	26
1.8 Objetivos.....	26
1.8.1 Objetivo General.....	26
1.8.2 Objetivos Específicos .....	26
1.9 Hipotesis .....	26
1.10 Hipotesis General .....	26
1.10.1 Hipotesis Específicas.....	26
II. Material y Métodos.....	28
2.1 Metodología de la investigación.....	28
2.2 Marco teórico .....	28
2.2.1 Migración de datos .....	28
2.2.2 Proceso de Migración de Datos.....	28
2.2.3 Arquitectura de la Migración de Datos.....	29
2.2.4 Herramientas ETL.....	30
2.2.5 Población.....	31
2.2.6 Recolección de la información.....	31
2.2.7 SCRUM .....	32
2.3 Técnicas.....	34
III. Resultados y Discusión.....	35

3.1	Limitación del Proyecto .....	35
3.2	Análisis de proceso de migración .....	36
3.3	Aspectos Técnicos .....	37
3.4	Aspectos Económicos .....	37
3.5	Aplicación de SCRUM .....	38
3.5.1	Sprint 0 .....	38
3.5.2	Sprint 1 .....	72
3.5.3	Sprint 2 .....	121
3.6	RESULTADOS .....	160
3.6.1	Prueba de Hipótesis: .....	162
IV.	Conclusiones .....	164
V.	Aportes .....	165

## Resumen

El presente trabajo de suficiencia Profesional para optar el título profesional de Ingeniero de Sistemas tuvo como objetivo el desarrollo de una herramienta ETL para la homologación de base de datos en la subgerencia de sistemas aseguradores subsidios y sociales de Essalud; esta herramienta permitió reducir los tiempos y actividades que se empleaban durante la etapa de ejecución de pruebas. Según el objetivo de la investigación el tipo del presente trabajo es importante, debido a que los tiempos son limitados para el desarrollo de proyectos informáticos y soluciones de incidencias. Se utilizaron los Entornos de Desarrollo Integrado (IDE por sus siglas en inglés) IntelliJ IDEA 2018.2.8 y Eclipse, como lenguaje de programación Java, se crearon clases para la migración de datos desde una tabla en específico y una clase principal que controlaba estas clases para ejecutar la migración de datos desde el ambiente de producción hacia al ambiente de calidad. Finalmente, en el presente trabajo, se llega a la siguiente conclusión: El implementar una herramienta ETL nos ayudó a finalizar los proyectos informáticos y solucionar las incidencias de manera oportuna reduciendo el número de actividades y el tiempo dedicado a la carga de datos durante la fase de pruebas.

## **Abstract**

The objective of this Professional Sufficiency work to opt for the professional title of Systems Engineer was the development of an ETL tool for the homologation of the database in the sub-management of subsidy and social insurance systems of Essalud; This tool allowed to reduce the time and activities that were used during the test execution stage. According to the objective of the investigation, the type of the present work is important, because the times are limited for the development of computer projects and incident solutions. The Integrated Development Environments (IDE for its acronym in English) IntelliJ IDEA 2018.2.8 and Eclipse were used, as a Java programming language, classes were created for data migration from a specific table and a main class that controlled these classes to execute the migration of data from the production environment to the quality environment. Finally, in the present work, the following conclusion is reached: Implementing an ETL tool helped us to finish computer projects and solve incidents in a timely manner, reducing the number of activities and the time dedicated to loading data during the phase of tests.

## I.Introducción

### 1.1 Descripción del sector

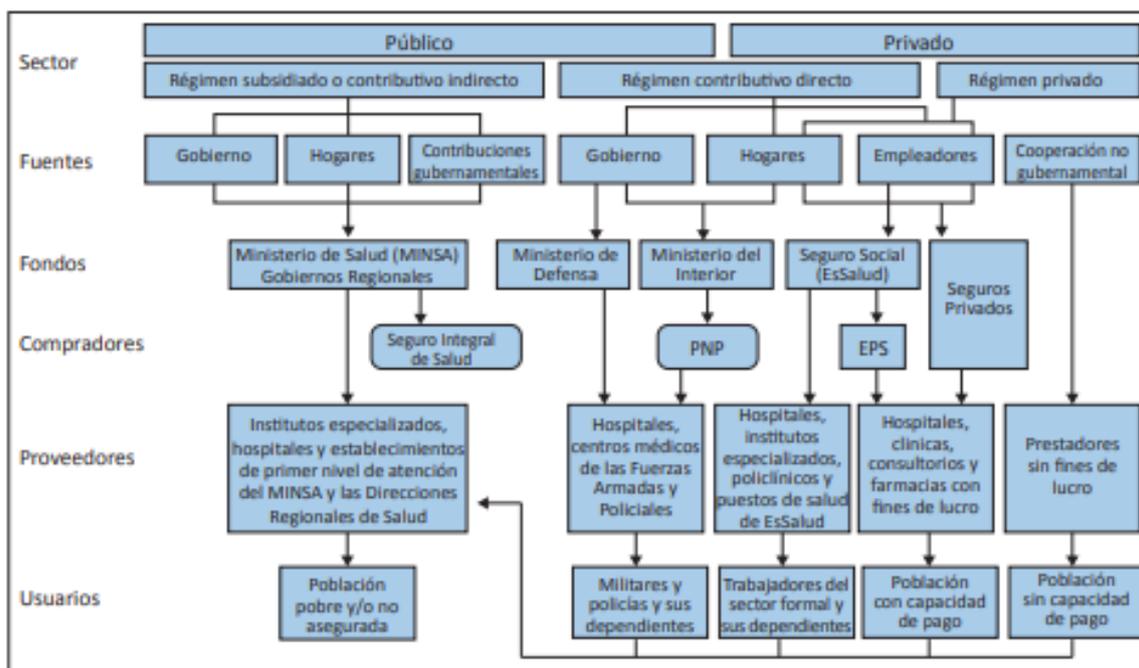
El Perú tiene un sistema de salud descentralizado, que se gestiona mediante cinco entidades: el Ministerio de Salud (MINSA), que oferta servicios de salud para el 60% de la población; EsSalud, que atiende el 30% de la población; y las Fuerzas Armadas (FFAA), la Policía Nacional (PNP), y el sector privado, que brindan servicios sanitarios al 10% restante, realidad que en los últimos 12 años no ha tenido grandes cambios. En todo sistema de Salud los actores son el paciente o asegurado, la entidad prestadora del servicio, el financiador o seguro que funge como administrador del fondo, el mismo que puede ser único o múltiples administradores, públicos, privados o mixtos.

Primeramente, el Gobierno ofrece servicios de salud a la población no asegurada a cambio del desembolso de una cuota de recuperación de montos variables, mediante el Seguro Integral de Salud (SIS) que subsidia la prestación de servicios a la población de escasos recursos económicos. La provisión de servicios, para los planes subsidiados de población abierta y población afiliada al SIS, se lleva a cabo a través de la red de establecimientos de los Gobiernos regionales y del Ministerio de Salud (MINSA), que se localizan en las regiones y en la capital del país (Alcalde-Rabanal et al., 2011). Este sistema secundario cuenta con tres niveles: nacional, regional y local. El nivel nacional lo integran el MINSA, los organismos descentralizados del MINSA (Dirección de Abastecimiento de Recursos Estratégicos en Salud y los Institutos especializados). El nivel regional esta constituido por las Direcciones Regionales de Salud (DIRESA), miembros de los gobiernos regionales y, el nivel local, por ciertas municipalidades responsables de la administración y el presupuesto de los establecimientos de salud de sus territorios (MINSA, 2010).

Los otros sistemas secundarios que ofrecen servicios de salud son cuatro: i) el Seguro Social de Salud - EsSalud adscrito al Ministerio de Trabajo y Promoción del Empleo, que maneja su propia red de hospitales y centros de salud; ii) las

Sanidades de las Fuerzas Armadas (Marina, Aviación y Ejército), adscritas al Ministerio de Defensa, que dispone de sus instalaciones propias; iii) la Sanidad de la Policía Nacional del Perú (PNP), adscrita al Ministerio del Interior, que de igual manera tiene sus propias instalaciones; y iv) las instituciones del sector privado: entidades prestadoras de salud (EPS), aseguradoras privadas, clínicas y organizaciones de la sociedad civil (OSC) (Wilson et al., 2009; Alcalde-Rabanal et al., 2011).

Figura 1. Estructura del sistema de salud del Perú



Fuente: Alcalde-Rabanal *et al.* (2011).

EsSalud brinda servicios de salud a la población que percibe un sueldo y a sus familias en sus mismas instalaciones, sin embargo, desde la promulgación de la Ley de Modernización de la Seguridad Social, en 1997, el sector privado le ha comercializado servicios personales mediante las EPS. La finalidad de estas es incorporar servicios de menor complejidad en la provisión de los servicios brindados por EsSalud a los trabajadores que se afiliaron por voluntad propia a alguna de esas instituciones. Las EPS ofrecen servicios de salud en establecimientos privados, propios o de terceros, no obstante, también tienen acuerdos con establecimientos públicos (MINSA, 2010).

La Sanidad de las Fuerzas Armadas y la Sanidad de la PNP ofrecen servicios únicamente a su personal, familiares directos y colaboradores, a través de su propia red de establecimientos. Pero a pesar de comportarse al igual que los seguros de salud, la financiación de los subsistemas procede del Tesoro público y de los copagos que realizan los derechohabientes de los beneficiarios titulares (MINSA, 2010).

Las prestaciones del sector privado se dividen en el servicio privado lucrativo y el servicio privado no lucrativo. Pertenecen al privado lucrativo las EPS, las aseguradoras particulares, las clínicas particulares especializadas y no especializadas, los centros médicos y policlínicos, los consultorios médicos y odontológicos, los laboratorios, los servicios de diagnóstico por imágenes y los establecimientos de salud de ciertas empresas mineras, petroleras y azucareras. Por otra parte, el sector privado no lucrativo es tradicionalmente conformado por un grupo variado de asociaciones civiles no lucrativas (como la Cruz Roja Peruana, los Bomberos Voluntarios y las Organizaciones de acción social de algunas iglesias, entre otras). La gran parte de estas entidades brinda servicios de primer nivel y usualmente percibe financiamiento de cooperantes externos, donantes internos, del Gobierno y de hogares (AlcaldeRabanal et al., 2011).

La mas alta autoridad sanitaria y el organismo rector del Sistema Nacional de Salud es el Ministerio de Salud -MINSA. La Ley General de Salud (Ley N° 26842) concede al MINSA la dirección y administración de la política nacional de salud, además de la supervisión de su cumplimiento de conformidad con la política general del Gobierno. Asimismo, determina que el Gobierno es garante de suministrar los servicios de salud pública a toda la población y que la salud personal es de competencia compartida entre el individuo, la sociedad y el Estado (Alcalde-Rabanal et al., 2011). Por otro lado, la Superintendencia Nacional de Aseguramiento en Salud - SUNASA (Organismo Público Técnico Especializado adscrito al Ministerio de Salud) esta representada como el ente responsable de registrar, autorizar, supervisar y regular a las instituciones administradoras de fondos de aseguramiento en salud, además de supervisar a las instituciones prestadoras de servicios de salud en el área de su competencia. También, es la institución responsable de registrar, autorizar, regular y supervisar las operaciones

de las entidades prepagas de salud y de todas las entidades públicas, privadas o mixtas que ofertan servicios bajo el sistema de pago regular o anticipado. Para desempeñar sus labores, la SUNASA posee facultades sancionadoras (Casalí y Pena, 2012).

Por otro lado, EsSalud, adjunto al Ministerio de Trabajo y Promoción del Empleo, dispone de un órgano de dirección que es el Consejo Directivo. El cual lo conforman representantes del Estado, de los trabajadores, de los empleadores y de los pensionados. El Presidente Ejecutivo, electo por el Poder Ejecutivo, es la mas alta autoridad de EsSalud. Aún cuando estos representantes son electos por dos años, con la opción de prorrogar por dos años más, se advierte una considerable inestabilidad en los puestos, lo que verdaderamente hace difícil la gestión de EsSalud.

## **1.2 Descripción general de la institución**

El Seguro Social de Salud, que además puede identificarse con el nombre de EsSalud, es un organismo público descentralizado, establecido con la finalidad de dar cobertura a los asegurados y sus derechohabientes mediante el otorgamiento de prestaciones de salud, que abarca la prevención, promoción, recuperación y rehabilitación; así como de prestaciones económicas y sociales. La Ley de Creación del Seguro Social de Salud (ESSALUD) es la N° 27056.

La Gerencia Central de Tecnologías de Información y Comunicaciones es el órgano de apoyo que es el encargado de la planificación, diseño, implementación y evaluación de las tecnologías de información y comunicaciones de la institución en el ámbito nacional, esta supeditada a la Gerencia General y tiene las siguientes funciones:

- a) Diseñar y plantear a la Gerencia General las políticas, estrategias, normas, planes y programas conducentes al desarrollo, mantenimiento e implementación de tecnología información y comunicación en los procesos de ESSALUD.
- b) Dirigir, oragnizar y examinar la formulación del Plan Estratégico de

Tecnologías de Información y Comunicaciones basado en los objetivos estratégicos institucionales, primando el cumplimiento de los planes aprobados.

- c) Plantear las disposiciones técnicas que facilite establecer los procesos, métodos y procedimientos para una apropiada administración, seguridad y administración de los recursos informáticos, medir su impacto y vigilar su cumplimiento.
- d) Dirigir, organizar y manejar los planes y proyectos en materia de modernización de tecnologías de la información y comunicaciones de la Institución, que facilite maximizar los procesos y recursos informáticos y de comunicaciones asignados.
- e) Examinar los requerimientos de tecnologías de la información para el desarrollo informático institucional, a fin de plantear a la Gerencia General las prioridades de su atención y los recursos que se necesitan para su implementación.
- f) Elaborar, preparar y plantear modelos de gestión innovadores para el desarrollo informático, y estandarizar los procedimientos relacionados a la gestión de los recursos informáticos de la Institución.
- g) Definir los lineamientos y normas de seguridad informática institucional, en busca de alcanzar la integridad, confidencialidad y suministro de la información obtenida de los sistemas corporativos implementados en la Institución.
- h) Preparar, elaborar, implementar y renovar los sistemas de información aprobados, primando los relacionados a productos y servicios asistenciales y aseguradores que ofrece la Institución.
- i) Proveer un apropiado soporte técnico informático, a fin de lograr una constante, segura y máxima operatividad de los sistemas de información corporativos y de sus plataformas tecnológicas.
- j) Diseñar, plantear y dirigir la implementación de los planes de contingencia, a fin de garantizar la continuidad de los procesos y de la información necesaria para la Institución.
- k) Plantear, elaborar y gestionar los sistemas centrales de información gerencial, basados en los sistemas de información institucionales, a fin de optimizar la toma de decisiones en la institución, en colaboración con la Gerencia Central de Planeamiento y Desarrollo órgano encargado del sistema de

información gerencial en ESSALUD.

- l) Contribuir al mejoramiento constante de los procesos informáticos en las unidades orgánicas, basados en el desarrollo e implementación de proyectos considerando las innovaciones tecnológicas que brinda el mercado.
- m) Implantar los aspectos referentes al área de las tecnologías de información que fijen los contratos en la modalidad de Asociación Público Privada suscritos por ESSALUD, así como vigilar su cumplimiento.
- n) Diseñar informes técnicos en el área de su competencia funcional, acerca de los Contratos de Promoción suscritos entre ESSALUD y los Inversionistas, a petición de la Gerencia Central de Promoción y Gestión de Contratos de Inversiones.
- o) Planear, aplicar y examinar las actividades y el uso de los recursos otorgados para el cumplimiento de sus funciones, de acuerdo a la normatividad vigente.
- p) Definir las medidas para implantar el control interno y gestión de los riesgos que correspondan en el ámbito de sus funciones, en el marco de las políticas y procedimientos establecidos.
- q) Otras labores que le asigne el Gerente General, en el área de su competencia.

### **1.3 Funciones del bachiller dentro de la institución**

Las funciones que realizo en la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales (SGSASS) de la Gerencia de Sistemas e Innovación Tecnológica (GSIT) de la Sede Central de EsSalud son las siguientes:

- Elaboración de scripts para automatización de pruebas.
- Elaboración de scripts de pruebas unitarias.
- Gestión y ejecución de pruebas funcionales.
- Gestión y ejecución de pruebas de servicios.
- Reportes de defectos al equipo de desarrollo.
- Administración y control del ambiente de pruebas.

#### **1.4 Actividad económica de la institución**

EsSalud es responsable del Régimen Contributivo de la Seguridad Social en Salud. Fue implantado en 1999 (mediante la Ley N° 27056) basado en el ex Instituto Peruano de Seguridad Social (IPSS). Cuya finalidad es de dar cobertura a los asegurados y sus derechohabientes, a través de el otorgamiento de prestaciones de prevención, promoción, recuperación, rehabilitación, prestaciones económicas y prestaciones sociales que pertenece al régimen contributivo de la Seguridad Social en Salud, además de otros seguros de riesgos humanos. Este organismo se relaciona con otras instituciones del sector público de diferentes modos. Primeramente, está supeditado orgánicamente del Ministerio de Trabajo, por cuanto es un organismo adjunto a esta institución. Por otra parte, se articula con el MINSA, dado que esta es la institución responsable de definir los lineamientos generales del sector Salud. Finalmente, EsSalud se relaciona con el Ministerio de Economía y Finanzas debido a que este autoriza el presupuesto de EsSalud mediante el FONAFE (Fondo Nacional de Financiamiento de la Actividad Empresarial del Estado).

EsSalud cuenta con varios seguros de acuerdo a la inserción laboral de los trabajadores. Todos estos seguros tienen variadas tasas de aportación, bases imponibles, reglas de acreditación, prestaciones de salud y demás beneficios. Primeramente, se encuentra el seguro regular (SR), al cual están afiliados obligatoriamente todos los trabajadores activos que laboran bajo relación de dependencia o como socios de cooperativas de trabajadores. Además son afiliados a este seguro los pensionistas que obtienen pensión de cesantía, jubilación, por incapacidad y de supervivencia (viudez y orfandad), independientemente del régimen legal al cual pertenezcan. Los trabajadores en situación de dependencia proporcionan un 9% (a través de aporte del empleador), en tanto los pensionistas aportan el 4%. El seguro brinda cobertura a los derechohabientes (cónyuge o concubina/o, hijos menores de edad e hijos mayores de edad incapacitados en forma total y permanente para el trabajo, y madre gestante de hijo extramatrimonial).

La alícuota de aportaciones al seguro de salud para los trabajadores en relación de dependencia (empleado y empleador) está por debajo de la alícuota de países como Argentina (13,2%), el Estado Plurinacional de Bolivia (10%), Colombia (12,5%), Costa Rica (14,75%), Panamá (17,25%) y Uruguay (entre 8% y 11%), pero es mayor a la alícuota vigente en Chile (7%) y Ecuador (5,7%). Por otro lado, en el caso de los pensionados, la alícuota actual es menor a la de países como Argentina (entre 3% y 6%), el Estado Plurinacional de Bolivia (5%), Chile (7%), Costa Rica (5%) y Ecuador (4,15%). Sin embargo, cabe indicar que estas comparaciones son parciales ya que no se incluye la base sobre la que se está aplicando la alícuota ni los complementos del sector público al seguro, entre otros aspectos (SSA, 2012).

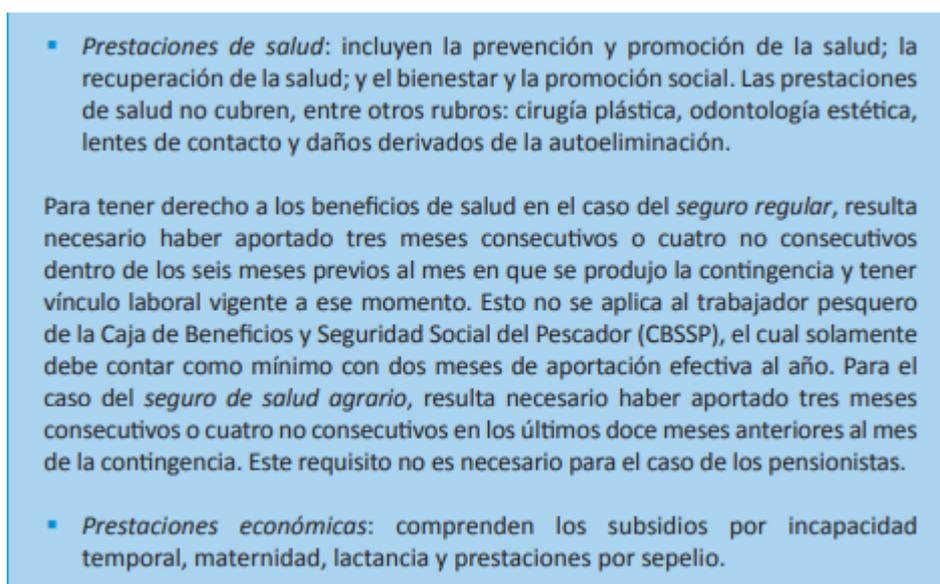
Los asegurados pueden optar por trasladar la cuarta parte de su aportación a la EPS de su elección (es decir, 2,25 p.p. de los aportes), estando esta en obligación de prestarle solo la llamada “capa simple” de atención de salud. Por otro lado, la “capa compleja” continua bajo responsabilidad de EsSalud. La elección de optar por una EPS se toma de forma colectiva en cada centro de trabajo y no de manera individual. Los trabajadores además pueden elegir el seguir con su atención de “capa simple” en EsSalud, en ese caso brindando al organismo la totalidad de su aporte (Casalí y Pena, 2012).

Por otro lado, EsSalud además cuenta con el seguro independiente. Elaborado en el marco de la Ley de Aseguramiento Universal en Salud, este seguro está disponible para cualquiera que resida en el Perú (cualquier edad), particularmente para trabajadores independientes (profesionales, técnicos, programadores, artesanos, comerciantes, transportistas, artistas), estudiantes, demás emprendedores y todas las demás personas independientes que quieran contar con un seguro en salud. También, brinda cobertura a los dependientes del titular (cónyuge o concubina/o, hijos menores de edad e hijos mayores de edad incapacitados en forma total y permanente para el trabajo). Este seguro otorga la cobertura que señala el PEAS y más de 230 prestaciones que EsSalud ha agregado. La cobertura tiene una prima de S/. 64 por una persona (titular) e incrementa en función de las personas dependientes: S/. 114 titular y un

dependiente, S/. 169 titular y dos dependientes y S/. 228 titular y 3 dependientes o más.

Además, hay un régimen de la seguridad social especial, llamado seguro agrario, que provee prestaciones de salud, económicas y sociales a los trabajadores dependientes e independientes que realicen actividades de cultivo y/o crianza, avícola, agroindustrial o acuícola, excepto la industria forestal. En el caso de los trabajadores dependientes, la aportación (a cargo del empleador) es del 4%, en tanto los trabajadores independientes aportan un 4% de la Remuneración Mínima Vital vigente. En este seguro los beneficiarios titulares tienen la posibilidad de afiliarse a sus derechohabientes.

*Figura 2. Prestaciones de EsSalud*



- *Prestaciones de salud:* incluyen la prevención y promoción de la salud; la recuperación de la salud; y el bienestar y la promoción social. Las prestaciones de salud no cubren, entre otros rubros: cirugía plástica, odontología estética, lentes de contacto y daños derivados de la autoeliminación.

Para tener derecho a los beneficios de salud en el caso del *seguro regular*, resulta necesario haber aportado tres meses consecutivos o cuatro no consecutivos dentro de los seis meses previos al mes en que se produjo la contingencia y tener vínculo laboral vigente a ese momento. Esto no se aplica al trabajador pesquero de la Caja de Beneficios y Seguridad Social del Pescador (CBSSP), el cual solamente debe contar como mínimo con dos meses de aportación efectiva al año. Para el caso del *seguro de salud agrario*, resulta necesario haber aportado tres meses consecutivos o cuatro no consecutivos en los últimos doce meses anteriores al mes de la contingencia. Este requisito no es necesario para el caso de los pensionistas.

- *Prestaciones económicas:* comprenden los subsidios por incapacidad temporal, maternidad, lactancia y prestaciones por sepelio.

Fuente: Ley N° 26790 (Ley de la modernización de de la seguridad social en salud).

El seguro tiene copagos (o deducibles) que forman costos enfrentados o compartidos por el afiliado (titular o sus dependientes) bajo la modalidad de un monto fijo o proporción a ser abonada por cada atención o procedimiento. Respecto a los trabajadores independientes hay copagos en algunas prestaciones ambulatorias, en hospitalización, en exámenes y procedimientos en EsSalud y en instituciones prestadoras de servicios de salud particulares.

Los planes ofertados por las EPS también pueden incorporar copagos. Sin embargo, la normativa fija algunos límites en este tipo de financiamiento. La Ley N° 27056 (de creación de EsSalud) define que en el plan de salud los copagos no pueden ser superiores (salvo consentimiento expreso del trabajador) al 2% del ingreso mensual del asegurado, por cada atención de carácter ambulatorio, ni el 10%, por cada hospitalización.

Como en otros seguros sociales, EsSalud tiene una reserva técnica, para afrontar escenarios imprevistos, la que debe conservarse como mínimo en un nivel equivalente al 30,0% del gasto incurrido para la producción de prestaciones durante el ejercicio anterior. En 2011, tal reserva se ubicaba en S/. 1,3 millones (0,25% del PBI).

### **1.5 Funciones del departamento donde labora el bachiller**

La Gerencia de Sistemas e Innovación Tecnológica (GSIT) es la unidad orgánica responsable de la elaboración de sistemas integrados de la función aseguradora y prestadora de ESSALUD, esta sujeta a la Gerencia Central de Tecnologías de Información y Comunicaciones y tiene las siguientes funciones:

- a) Proponer y examinar el Plan de Desarrollo de Tecnología de Información, en el área de su competencia.
- b) Proponer y formular las metas y estrategias de desarrollo e implementación de las nuevas tecnologías de información, definiendo las políticas respectivas en el ámbito nacional.
- c) Colaborar con la Gerencia de Producción, la Oficina de Seguridad Informática y el área usuaria encargada de la implementación de nuevos productos de tecnologías de información, acordando los procesos respectivos.
- d) Examinar y plantear las nuevas tecnologías de información de carácter estratégico y de gestión de las unidades orgánicas de ESSALUD, a nivel nacional.
- e) Organizar, examinar y definir conjuntamente con la Gerencia de Producción y la Oficina de Seguridad Informática los modelos, arquitectura y tecnología de base de datos y aplicaciones institucionales, plantear los procedimientos y herramientas computarizadas para la implantación de los procesos de gestión de

datos.

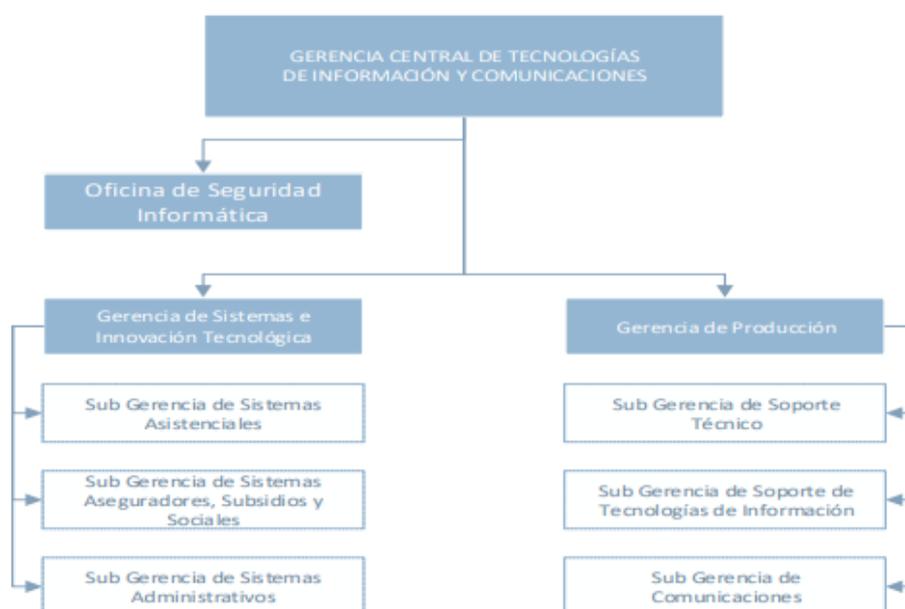
- f) Proponer, formular e implantar las metodologías para el desarrollo, implantación y mantenimiento de software de aplicación, definir las políticas respectivas a nivel institucional y examinar su cumplimiento.
- g) Aplicar, examinar y asegurar el desarrollo y mantenimiento de los sistemas de información automatizados para las unidades orgánicas de ESSALUD, definir los respectivos lineamientos a nivel nacional.
- h) Dirigir la formulación de métricas que faciliten el control de la ejecución de los contratos de desarrollo y/o mantenimiento de sistemas de información que contrate la Institución.
- i) Plantear los criterios y estándares requeridos que faciliten la implementación de los sistemas de información, a fin de un apropiado aprovechamiento de la Tecnología de Información, en todas las unidades orgánicas de ESSALUD.
- j) Supervisar, examinar y dar conformidad al trabajo desarrollado por terceros, en el área de su competencia, plantear y adoptar de manera oportuna las medidas correctivas necesarias.
- k) Conducir el desarrollo de nuevos proyectos y servicios dentro de su área, enmarcados en los planes, basados en las prioridades y planes operativos aprobados.
- l) Definir y sugerir software a utilizar en la Institución, requerido para el análisis, diseño, desarrollo, implantación y mantenimiento de sistemas y procesos informáticos, según las necesidades tecnológicas de ESSALUD y definir las políticas a nivel nacional.
- m) Organizar, en el área de su competencia la implementación de nuevos sistemas de información en las áreas usuarias a fin de conjugar las mejoras de procesos, colaborar en la capacitación que facilite su adecuada operatividad.
- n) Planear y examinar las actividades de su competencia y administrar los recursos para el cumplimiento de los mismos.
- o) Elaborar las acciones para la implementación del control interno y administración de los riesgos que correspondan en el área de sus funciones, en el marco de las políticas y procedimientos establecidos.
- p) Otras funciones que le sean asignadas por el Gerente Central de

Tecnologías de Información y Comunicaciones, en el área de su competencia.

La Gerencia de Sistemas e Innovación Tecnológica está conformada por las siguientes unidades orgánicas:

- Sub Gerencia de Sistemas Asistenciales
- Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales
- Sub Gerencia de Sistemas Administrativos

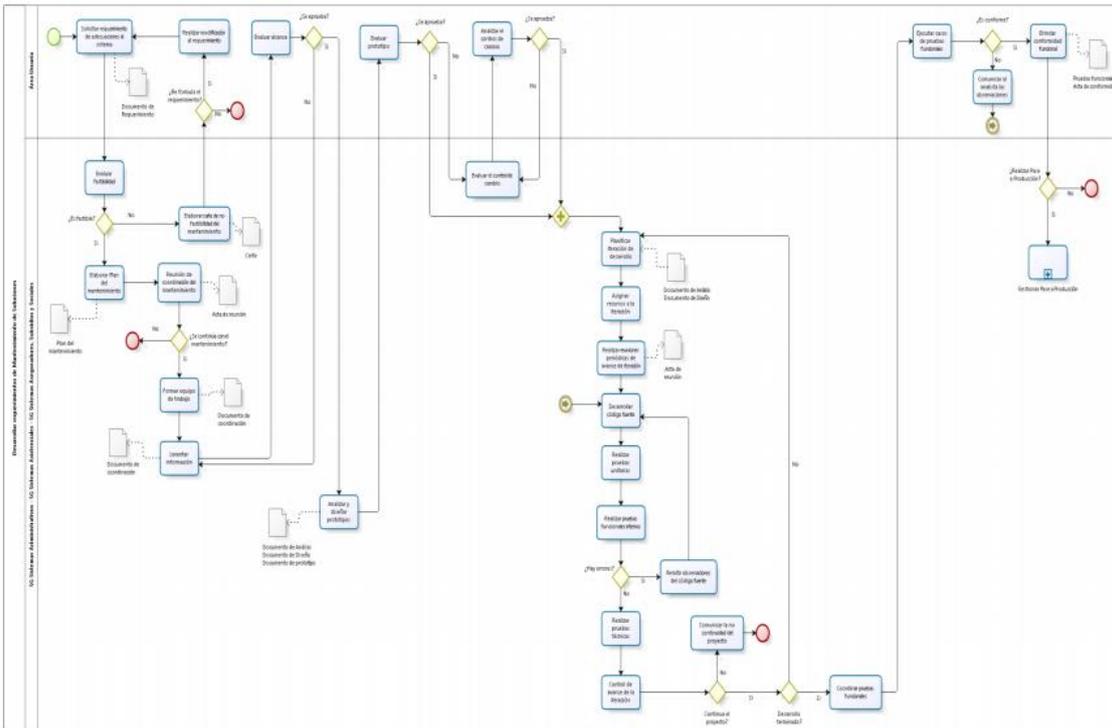
Figura 3. Organigrama Gerencia Central de Información y Comunicaciones



Fuente: Manual de Procesos y Procedimientos de GCTIC.

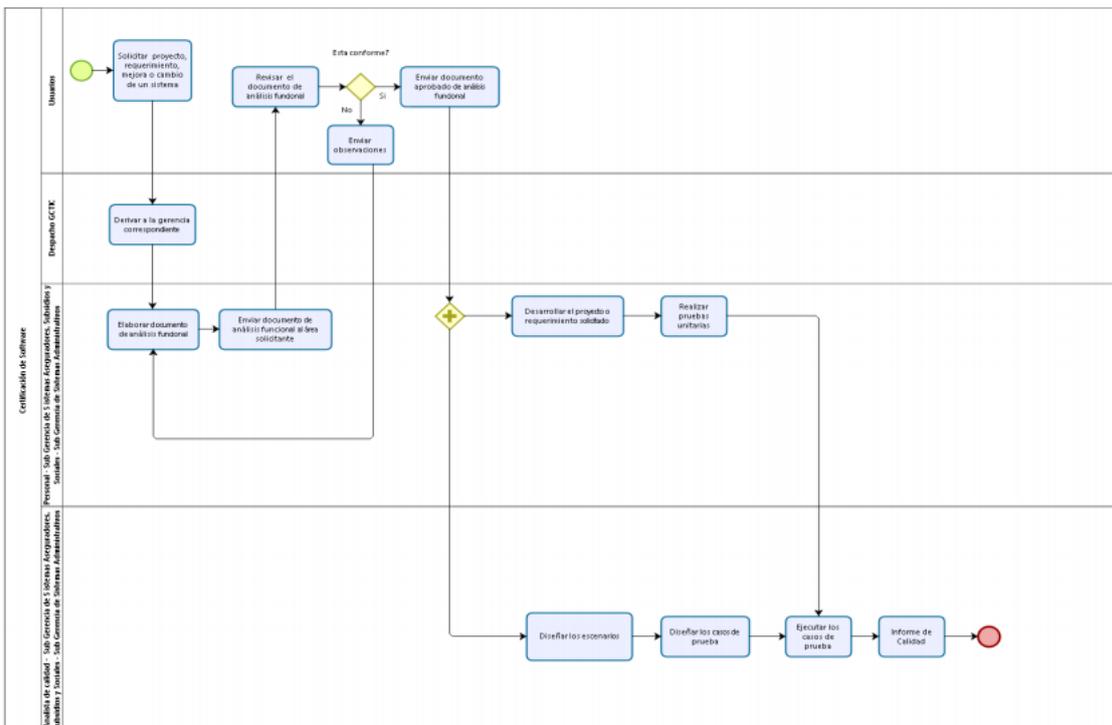
Actualmente en Essalud, la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales (SGSASS) es la unidad orgánica encargada de proveer solución tecnológica a las necesidades de sistemas de las unidades orgánicas aseguradoras y de prestaciones económicas y sociales, con base en la arquitectura de sistemas de la Institución; gestionando el soporte y mantenimiento a las mismas; además de su evaluación, implantación e integración de nuevas tecnologías, ejecutando las pruebas de control de calidad y del adecuado funcionamiento de los sistemas anterior al pase a producción, a fin de asegurar





Fuente: Manual de Procesos y Procedimientos de GCTIC.

Figura 6. Proceso de certificación de software



Fuente: Manual de Procesos y Procedimientos de GCTIC.

## **1.6 Planteamiento del Problema**

### **1.6.1 Internacional**

En la actualidad un sistema informático cuenta con un doble papel. Es un producto y simultáneamente es el vehículo para suministrar un producto. Como producto facilita el uso del hardware, mediante un ordenador personal, un teléfono móvil. Como vehículo usado para entregar el producto, trabaja como base de control, tales como, un sistema operativo, o un sistema gestor de redes. Un sistema informático suministra lo que se considera como el producto más significativo del siglo veintiuno, la información. Los sistemas de información procesan datos personales para que sean de mayor utilidad en un entorno local, gestionan información comercial para incrementar la competitividad, brindan el acceso a redes a nivel mundial, y proporcionan el medio para adquirir información en todas sus clases.

La ingeniería de sistemas aborda áreas muy variadas de la informática y de las ciencias de la computación, que se ponen en práctica en un amplio espectro de campos; por ejemplo, negocios, investigación científica, medicina, producción, logística, banca, meteorología, derecho, redes, entre otras muchas.

Aunque, es usual que en la práctica diaria profesional no se aplique prácticamente ninguno de las sugerencias más básicas de la ingeniería de sistemas. Es común que el desarrollo de software se asemeje más al descontrol del cuento de «si los programadores fueran albañiles...» que a una ideal y bien estructurada "fabrica de software" (concepto de gran vigencia a finales de los ochenta).

Efectivamente, los estudios de los procesos productivos de software efectuados a raíz de los modelos de procesos de software corroboran que el desarrollo de software se mantiene en estado caótico tanto en pequeñas empresas de países como España como en grandes proyectos en países como EE UU y Japón.

### **1.6.2 Nacional**

El director de Consulting & Management S.A.C. Ing. Oscar Camacho Carrillo quién ha liderado operaciones de TI en el Perú y otros países, señaló que el proceso de cada proyecto informático incluye la gestión, el ciclo de vida, la metodología, los estándares, la ingeniería, la arquitectura, el componente técnico

– administrativo, entre otros. Por lo tanto, si la ingeniería está mal diseñada o la gestión es deficiente esta fallará. Según el, las cuatro razones principales del fracaso de los proyectos de TI en el Estado son la pésima definición del alcance del Proyecto; la falta de aplicación de metodologías en el desarrollo de software, así como en la gestión de proyectos; el uso de tecnologías en proceso de maduración; y la poca acertada selección de los profesionales.

Con respecto al alcance, el cual consiste en determinar claramente lo que se va incluir y lo que no se incluirá en el Proyecto, es común observar en las bases o términos de referencia expresiones ambiguas, de “alto nivel” y la falta de aquello que no se incluirá, lo que después generará peticiones de especificación, controversias, adendas y arbitrajes, el cual provocará un desgaste a contratantes y contratados, subirán los costos, demorarán las entregas y terminará en el fracaso del Proyecto.

Respecto a las metodologías, frecuentemente se puede constatar la carencia de disciplinas como la ingeniería de software, evidenciándose la falta de métodos y herramientas fiables para “comprender el problema” y desarrollar el software, así como la poca planificación de las fases y etapas, lo que suele multiplicar los riesgos del Proyecto, más aún cuando se trata de emprendimientos de mediana o alta complejidad.

En adición se puede decir que no son pocos los casos en que la metodología termina siendo utilizada como una pantalla formal para satisfacer ciertas normas y requisitos, contrario a lo que verdaderamente es, el eje del proyecto.

Respecto al estado de madurez de las tecnologías, hemos observado en el Perú una gran cantidad de intentos recientes de adopción de tecnologías, sin que antes se haya analizado el impacto que esta incorporación podría tener en la estrategia, procesos, sistemas, finanzas y cultura de la organización. Casos emblemáticos son la adquisición del BPM del PRODUCE, el BPM de EsSalud y el BPM/SOA del MEF (recientemente comentado en América Sistemas), donde la dificultad mayor no fue la tecnología misma, sino el escaso grado de conocimiento que tenían de ella los contratantes y los representantes locales de los productos adquiridos.

### **1.6.3 Local**

En la SGSASS para el desarrollo de los proyectos informáticos se utiliza el

modelo de desarrollo en cascada el cual es un proceso secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada, las fases que componen el modelo son las siguientes: Requisitos del software, Diseño, Implementación, Verificación, Instalación y Mantenimiento.

Durante las fases de verificación y mantenimiento se llevan a cabo las pruebas, las cuales su ejecución y culminación tienen un tiempo establecido en los cronogramas de desarrollo de los proyectos en la fase de verificación, así como también un tiempo límite para la solución de las incidencias en la fase de mantenimiento.

La necesidad de realizar la entrega de los proyectos informáticos de una manera rápida y eficiente aprovechando al máximo los recursos disponibles es actualmente un elemento crucial para mantener y mejorar los niveles de desempeño individual y organizacional. Sobre todo, en instituciones que tienen un alto impacto en el ámbito social y público, como es el caso de ESSALUD que es el principal ente encargado de las prestaciones de salud a los asegurados y sus derechohabientes. Ante la presión de maximizar los cronogramas de los proyectos informáticos en los cuales se incluye información acerca de los tiempos estimados para culminar estos proyectos y sus diferentes actividades, las jefaturas y los gestores de proyectos muchas veces están en la búsqueda continua de herramientas que ayuden a eliminar algunas actividades o reducir los tiempos de ejecución de las mismas.

En los cronogramas de los proyectos informáticos de la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales se registraba una actividad llamada "Preparación de ambiente de prueba", Esta actividad consistía básicamente en una migración de datos entre base de datos en el cual se realizaba la copia de uno o varios registros de una tabla del ambiente de producción y registraba en una tabla del ambiente de calidad, esto con la finalidad de tener los datos actualizados de los asegurados y realizar las pruebas de tal manera que podamos observar el comportamiento del sistema desarrollado tal como se comportaría en el ambiente de producción, cabe indicar que el desarrollo de sistemas es

realizado durante las actividades previas a las pruebas en las fases de desarrollo. Esta migración de datos se realizaba de forma manual, es decir, se tenía que seleccionar, copiar y pegar registro por registro o por grupos de registros desde una tabla hacia otra, para tener los datos mínimos necesarios para las pruebas esta operación se realizaba sobre 25 tablas las cuales según los datos registrados del asegurado se tenía que migrar 1 registro en algunas tablas o más de 100 registros en otras, esta migración manual se complicaba toda vez que se intentaba migrar desde una tabla un grupo de registros y en ellos se tenía uno ya existente en la tabla del ambiente de calidad por que el motor de base de datos mostraba un mensaje de error que nos alertaba de una restricción por llave única lo que nos decía que había un registro que contenía los mismos datos en los campos que conformaban la llave única de esa tabla, en estos casos se tenía que realizar la migración registro por registro, otro problema que se presentaba con la duplicidad de llaves era que si bien un registro en el ambiente de calidad tenía los mismo datos en los campos de la llave se tenía que observar que tenga los mismos datos en todos los campos, si estos registros eran totalmente iguales se omitía la migración de dicho registro, en caso contrario se determinaba que el registro no estaba actualizado y se tenía que primero borrar el registro en la tabla del ambiente de calidad y luego migrar el registro actualizado del ambiente de producción. El tener datos desactualizados ocurre muchas veces ya que al ser un ambiente de pruebas la data es manipulada constantemente.

Todas estas operaciones antes descritas obligaban a considerar dentro del cronograma del proyecto dos días aproximadamente para la ejecución de la actividad de migración de datos durante la fase de pruebas el cual se quería reducir este tiempo de algún modo utilizando los recursos con las que contaba el área.

## **1.7 Formulación del Problema**

### **1.7.1 Problema General**

¿Cómo influye la herramienta ETL en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo?

### **1.7.2 Problema Específico**

¿De qué manera la herramienta ETL influye en el tiempo de carga de data en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo?

¿De qué manera la herramienta ETL influye en el número de actividades en los proyectos informáticos en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo?

## **1.8 Objetivos**

### **1.8.1 Objetivo General**

Determinar la influencia de la herramienta ETL en el proceso de homologación de base de datos en los ambientes de calidad y productivo.

### **1.8.2 Objetivos Específicos**

Determinar la influencia de la herramienta ETL en el tiempo de carga de data en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo

Determinar la influencia de la herramienta ETL en el número de actividades en los proyectos informáticos en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo

## **1.9 Hipotesis**

### **1.10 Hipotesis General**

La herramienta ETL influye significativamente en el proceso de homologación de base de datos en los ambientes de calidad y productivo.

### **1.10.1 Hipotesis Específicas**

La herramienta ETL influye significativamente en el tiempo de carga de data en el

proceso de homologación de las bases de datos en los ambientes de calidad y productivo

La herramienta ETL influye significativamente en el número de actividades en los proyectos informáticos en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo.

## **II. Material y Métodos**

### **2.1 Metodología de la investigación**

Para el desarrollo de la herramienta ETL se trabajó con el framework SCRUM y y la metodología de migración de datos el cual es un proceso mediante el cual se puede realizar una transferencia de datos desde unos sistemas de almacenamiento de datos a otros, de unos formatos de datos a otros o entre diferentes sistemas informáticos.

### **2.2 Marco teórico**

#### **2.2.1 Migración de datos**

El almacenamiento de datos tarde o temprano se encontrará en la necesidad imperiosa de realizar una migración. La migración es un proceso complejo donde se debe considerar los puntos a tener en cuenta durante su ejecución, como podrían ser la fuente de datos, el destino de los mismos, la estructura actual y la movilidad a otra, el análisis de la persistencia y depuración de los datos, las plataformas donde se ejecutará la migración, entre otros.

El tamaño de espacio disponible en el servidor de datos, actualizaciones de software, así como también nuevas solicitudes del usuario final, incuestionablemente, hace considerar en trasladar la información desde una estructura hacia otra que podría ser similar o diferente.

Las actuales necesidades, el número de procesos que se realizan día a día en la base de datos, junto a la información que ya existe nos da una idea de lo que se necesitará en el futuro considerando el crecimiento de los mismos, la actualización de la información y los tipos de datos que se quiere almacenar.(Antaño et al., 2014)

#### **2.2.2 Proceso de Migración de Datos**

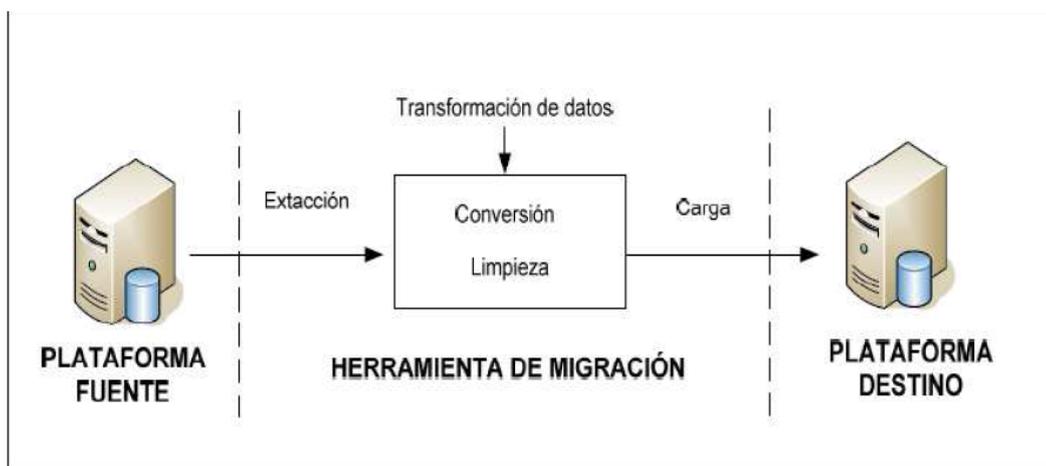
La migración es un proceso que podría ser muy complejo y, debido a la existencia de diferentes bases de datos, dificulta dar una solución que pueda satisfacer todos los casos. Por otro lado, además de la complejidad de trasladar la información entre dos motores de base de datos, también influenciará mucho en la dificultad del

problema el tipo de los datos de las tablas que serán procesadas. Tale como, las fechas, los campos numéricos con decimales o los boléanos podrían causar problemas al moverlos de un sistema a otro debido a que podrían almacenarse de diferente manera o, en lo que repecta a los números, con una precisión diferente. Para hacer la migración de datos, es necesario definir en forma temprana, los datos del sistema origen que se necesitan en el sistema destino (heterogéneo). En adición se requiere saber quienes son los propietarios de la información y los parámetros de depuración que llevarán los datos seleccionados, para garantizar la calidad ideal de ellos. (Rodriguez & Rosso, 2013)

### 2.2.3 Arquitectura de la Migración de Datos

La arquitectura básica de la migración de datos se identifica con tres componentes principales tal como se observa en la siguiente figura:

Figura 7. Arquitectura de migración de datos



Fuente: Rodriguez & Rosso, 2013

#### **Plataforma Fuente**

Es la ubicación donde se almacena la información principal que se analizará para su correspondiente migración hacia otra base de datos.

#### **Plataforma Destino**

Es la ubicación donde estarán los datos de la base de datos fuente al culminar su transformación y limpieza respectiva.

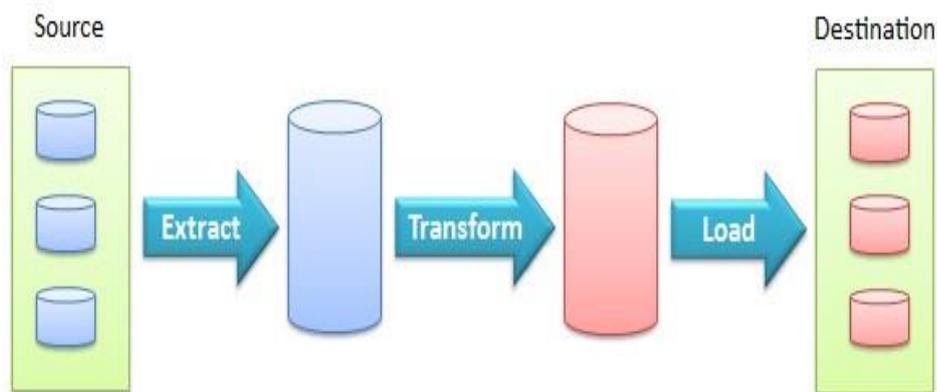
### **Herramienta de Migración**

Esta herramienta posibilita realizar la migración de datos entre plataformas, posibilitará la extracción, la transformación y la carga de los datos desde la base de datos fuente hacia la base de datos destino según los requerimientos definidos en una entidad.

#### **2.2.4 Herramientas ETL**

Según Vitt (2003), sostiene que es un componente esencial en las labores de implementación que se concentra en extraer, transformar y cargar los datos procedentes de las fuentes del sistema. Este proceso posibilita a las organizaciones trasladar datos de diferentes fuentes, reformatearlos, limpiarlos y cargarlos en otra base de datos como un datamart o un datawarehouse para su análisis, Adicionalmente de moverlo a un diferente sistema operacional que ayudará un proceso de negocio.

*Figura 8. Herramienta ETL*



Fuente: Sun Microsystem (2012).

Las fases de la herramienta ETL son las siguientes:

#### **Fase de Extracción:**

Esta fase tiene como base extraer e integrar la información de distintas fuentes (ERP, CRM, Excel) en el datawarehouse. En esta fase se estandariza un formato

para todos los datos que llenarán el datawarehouse, debido a que su origen es de distintas fuentes y de manera individual inicialmente tendrá un formato propio.

#### **Fase de Transformación:**

Esta fase utiliza un conjunto de reglas de negocio para seleccionar solamente la información que se necesita en el datawarehouse. Usando métodos de filtrado, manejo de datos y cálculos que evitarán guardar información innecesaria, prescindible o incorrecta.

#### **Fase de Carga:**

Durante la fase de carga los datos son almacenados en el sistema de destino. De acuerdo con los requerimientos de la organización, este proceso podría incluir un extenso número de actividades variadas. Algunas veces en la base de datos se reemplaza la información antigua con datos nuevos. Los datawarehouse conservan un historial de los registros para facilitar la auditoría de los mismos y contar con el rastro de la historia completa de un valor a lo largo del tiempo.

### **2.2.5 Población**

La población es el grupo de datos en el cual se utilizan procedimientos para llevar a cabo un estudio detallado de un número de personas, la cual se le denomina población (Cabezas, Andrade y Torres, 2018, p. 88).

Por esta razón, se puede comprender que la población es el conjunto donde se va a realizar la experimentación, pero la mayor importancia de la población radica en que se enfoca en el objeto de estudio siendo este el centro de atracción debido a que de ella se facilita la elaboración e identificación de instrumentos que son usados según lo requerido por el investigador, de manera tal que le permite tener resultados exactos que producen las conclusiones y resultados de las investigaciones (Cabezas, Andrade y Torres, 2018, p. 88).

Se tomará 30 procesos de homologación de migración de datos.

### **2.2.6 Recolección de la información**

Para la recolección de datos utilizaremos la observación directa para poder medir

los tiempos en cada una de las fases.

### 2.2.7 SCRUM

El concepto de Scrum comienza a inicios de los 90. Se basa en un estudio de administración de equipos de 1986 elaborado por Hirotaka Takeuchi e Ikujiro Nonaka llamado "The New New Product Development Game". Este consistía en un estudio acerca de novedosos procesos de desarrollo usados en productos que tuvieron mucho éxito en Japón y EEUU (cámaras de fotos de Canon, fotocopiadoras de Xerox, automóviles de Honda, ordenadores de HP y otros). En este estudio se evaluaba la forma de trabajo de equipos de desarrollo, sumamente productivos y multidisciplinarios, con la cooperación entre los jugadores de Rugby y su formación de Scrum.

Scrum empieza con el supuesto que el mundo es complejo y por consiguiente "no se puede predecir o planear definitivamente lo que se entregará, cuando se entregará y que calidad y costo tendrá" dice Ken Schwaber. No se fundamenta en continuar con un plan sino en el ajuste continuo según las circunstancias en el desarrollo del proyecto.

#### Artefactos

Según la guía Scrum se dispone de 3 artefactos o herramientas.

- **ProductBacklog:** Consiste en una lista de funcionalidades técnicas y de negocio prioritarias. Estos son requerimientos a muy alto nivel de lo que tiene que realizar una aplicación, en el cual se registran características, funciones, tecnología, mejoras, bugs, etc. que se aplicarán al producto. El ProductBacklog es el punto inicial.
  - **Sprint Backlog:** Lista de tareas de un sprint. El Sprint Backlog fija y establece el trabajo que deberá ser logrado por el equipo de desarrollo a lo largo de un Sprint. En un nivel el Sprint Backlog define las características mientras que en otro nivel, define las tareas necesarias para aplicar esas características.
  - **Incremento:** Comienza a partir de un sistema desarrollado en un Sprint. Este producto desarrollado es posible de entregarse al termino de cada Sprint, significa

que todo esta completamente terminado en cada Sprint; y se puede efectivamente empaquetar o desplegar en seguida posterior a la Revisión del Sprint con un poco cantidad de tareas, a pesar de que a veces es necesario realizar algunos trabajos de finalizado entre ellos pruebas o documentación.

## **Roles**

El equipo Scrum consta en un dueño del producto (ProductOwner), El equipo de Desarrollo (DevelopmentTeam), y un Scrum Master.

En Scrum los equipos tienen que ser auto-organizados y multifuncionales.

Los equipos auto-organizados deben tomar decisiones y elegir el mejor modo para realizar su trabajo en vez de ser conducidos por personas ajenas a su equipo.

Los equipos multifuncionales disponen de todas las competencias que se necesitan para realizar el trabajo. Este modelo de trabajo se elaboró para optimizar la flexibilidad, la creatividad y la productividad.

Los equipos en Scrum llevan a cabo entregas incrementales del producto, garantizando que se cuenta con una versión del producto valiosa y funcional en potencia.

- **Dueño del Producto (ProductOwner).** Es el encargado de optimizar el valor del producto y del trabajo del equipo de desarrollo. El ProductOwner es el único responsable de administrar el ProductBacklog o lista de Requerimientos funcionales. Esta gestión abarca:

- Definir y detallar los requerimientos tal que sean entendibles para el equipo.
- Establecer el valor del negocio y definir la prioridad de cada requerimiento para que el equipo de desarrollo conozca cuales son los requerimientos prioritarios.
- Garantizar que el equipo de desarrollo comprende cada requerimiento de manera clara. El ProductOwner es una persona no un comité, cada una de las decisiones y definiciones que adopte el ProductOwner basado en el producto tiene que atendidos por el equipo.

- **Equipo de Desarrollo (DevelopmentTeam).** Son los encargados responsables de entregar un incremento del producto “Hecho”, que puede ser utilizable, al termino de cada Sprint. Al equipo desarrollo se le concede el poder de

parte de la organización para que pueda encargarse y realizar su trabajo, este trabajo en conjunto logra eficiencia y efectividad en el equipo.

Los equipos de desarrollo deben contar con estas características:

- Son auto-organizados.
- Son Multifuncionales.
- No contienen sub-equipos.
- **Scrum Master.** Es el encargado de que el proceso de Scrum se comprenda y que se ponen en practica las reglas adecuadamente. Es un líder al servicio del equipo Scrum.

Servicios del ScrumMaster al ProductOwner:

- a) Capacitar a los miembros del equipo de desarrollo en ser auto-organizados y multifuncionales.
- b) Establecer y dirigir a los equipos de desarrollo a construir productos con el máximo valor.
- c) Superar los obstaculos que alteren el avance del equipo.
- d) Facilitar los eventos de Scrum.
- e) Enseñar al equipo a entender Scrum.

### **El proceso Scrum**

Scrum establece un marco de gestión de proyecto en el cual las actividades de desarrollo – recolección de requerimientos, diseño, programación – se llevan a cabo. El plazo de desarrollo es de una iteración de 30 días llamada Sprint, aunque podría realizarse con sprints de menor duración. El marco de trabajo de Scrum tiene tres componentes o cuatro según como se los mire.

## **2.3 Técnicas**

Se utilizará la Observación directa y como instrumento la ficha de observación.

### **III. Resultados y Discusión**

#### **3.1 Limitación del Proyecto**

Se trabajará solo con 25 tablas a las cuales tenemos permisos de solo lectura en ambiente de producción, en estas tablas se registran los datos correspondientes a las solicitudes y pagos de subsidios, así como también datos personales de los asegurados que necesitan ser migrados como parte del inicio de la fase de ejecución de pruebas.

### 3.2 Análisis de proceso de migración

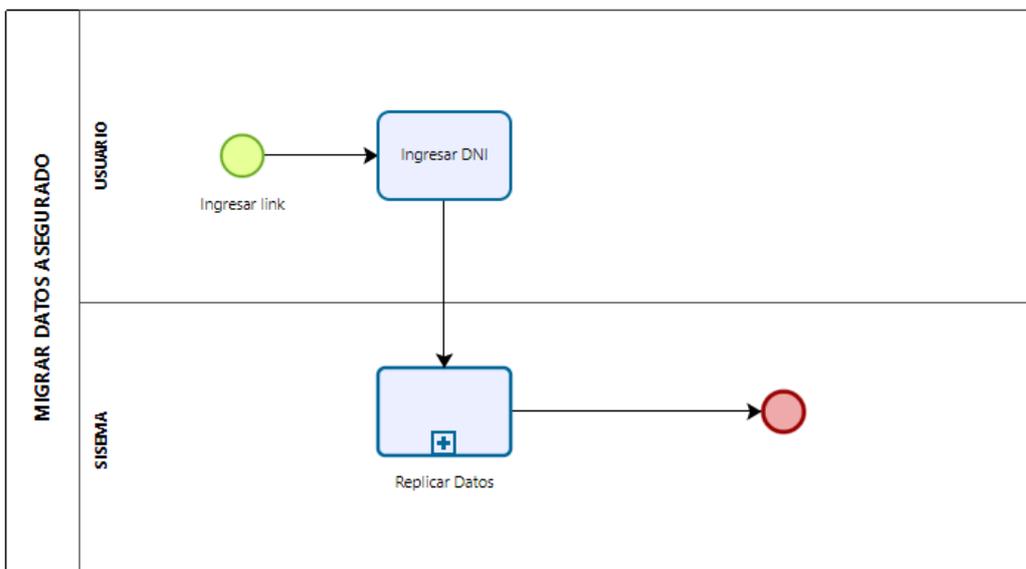
Según el análisis del proceso de migración realizado, la solución propuesta fue la migración de todas las tablas que contribuyen directamente en la obtención de los datos actualizados de los asegurados, esta migración se desarrollará en la herramienta web propuesta de la siguiente forma:

Tabla 1. Funcionalidad según usuario

<b>Usuario</b>
Migración datos de asegurado

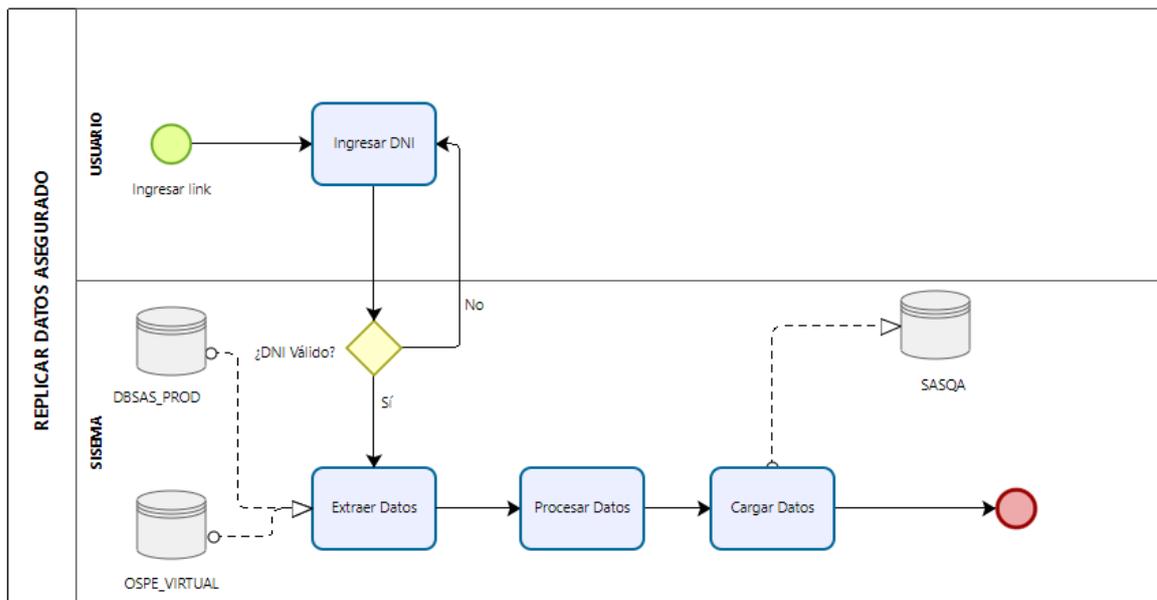
Fuente: Elaboración propia

A continuación, el diagrama BPMN propuesto del proceso de migración sería:



Y el subproceso sería:

- Replicar datos de Asegurado:



### 3.3 Aspectos Técnicos

Este proyecto se llevó a cabo aplicando la metodología SCRUM, el desarrollo de esta herramienta ETL se realizó utilizando los IDE (entorno de desarrollo integrado por sus siglas en inglés) IntelliJ IDEA y Eclipse, en lenguaje de programación JAVA y el MVC (modelo vista controlador) se programó integrando el framework SPRING y Bootstrap.

Tabla 2. Software a utilizar para el desarrollo de la herramienta

Software	Versión
Java	1.8
Spring	2.4.3
Bootstrap	v4.0.0
IntelliJ IDEA	2018.2.8
Eclipse IDE	4.12.0
Arquitectura	86_64
Sistema Operativo	Windows

Fuente: Elaboración propia

### 3.4 Aspectos Económicos

Respecto los aspectos económicos de esta herramienta se puede precisar que se ahorró en horas hombre ya que el personal involucrado en las tareas de migración de datos podrá realizar el trabajo en un tiempo menor.

Respecto a los gastos, no se generó ninguno ya que se utilizó la computadora personal del desarrollador como servidor de la herramienta ETL, posteriormente se deberá implementar la herramienta en un servidor con más capacidades cuando el número de usuarios de la herramienta web incrementa.

### 3.5 Aplicación de SCRUM

De acuerdo al análisis del proceso y funcionalidades identificados el aplicativo a desarrollar tendrá los siguientes alcances:

- **Proceso – Migrar datos asegurado:** Se realizará un módulo para que el usuario pueda migrar los datos de los asegurados mediante el DNI una vez concluido el proceso sistema le indicará mediante un mensaje que el proceso finalizó.

#### 3.5.1 Sprint 0

En el Sprint 0 se procedió con la reunión inicial para la creación del equipo de trabajo, además se instauró y se definió el Product Backlog.

##### 3.5.1.1 Conformación del equipo

Se puede observar en la siguiente tabla cómo estuvo conformado el equipo de trabajo y los roles que tuvieron en el desarrollo del sistema.

Tabla 3. Equipo de Trabajo y Roles

Rol	Persona
Product Owner	Paulo César Santamaría Teque
SCRUM Master	John Paúl Antón Alvarado
Team	John Paúl Antón Alvarado

Fuente: Elaboración propia.

### 3.5.1.2 Definición del Product Backlog

El Product Backlog incorpora todas las funcionalidades que el producto final debe poseer, es aquí donde se precisan las historias de usuario y se indica el Sprint al que serán incorporados, primando cada historia y otorgando una estimación del tiempo necesario para su implantación. Siendo el resultado del estudio de los procesos y requerimientos definidos en la etapa de análisis, se fija el Product Backlog en la siguiente tabla:

Tabla 4. Product Backlog

Identificador de la historia	Enunciado	Estado	Dimensión/ Esfuerzo	Iteración (Sprint)	Prioridad
HU_001	Como usuario necesito los datos personales actualizados del asegurado en ambiente de calidad	OK	10	1	Alta
HU_002	Como usuario necesito los datos de vínculo familiar actualizados del asegurado en ambiente de calidad	OK	5	1	Alta
HU_003	Como usuario necesito los datos de solicitudes de subsidios actualizados del asegurado en ambiente de calidad	OK	5	2	Media
HU_003	Como usuario necesito replicar todos los datos de un asegurado desde un aplicativo web	OK	10	2	Media

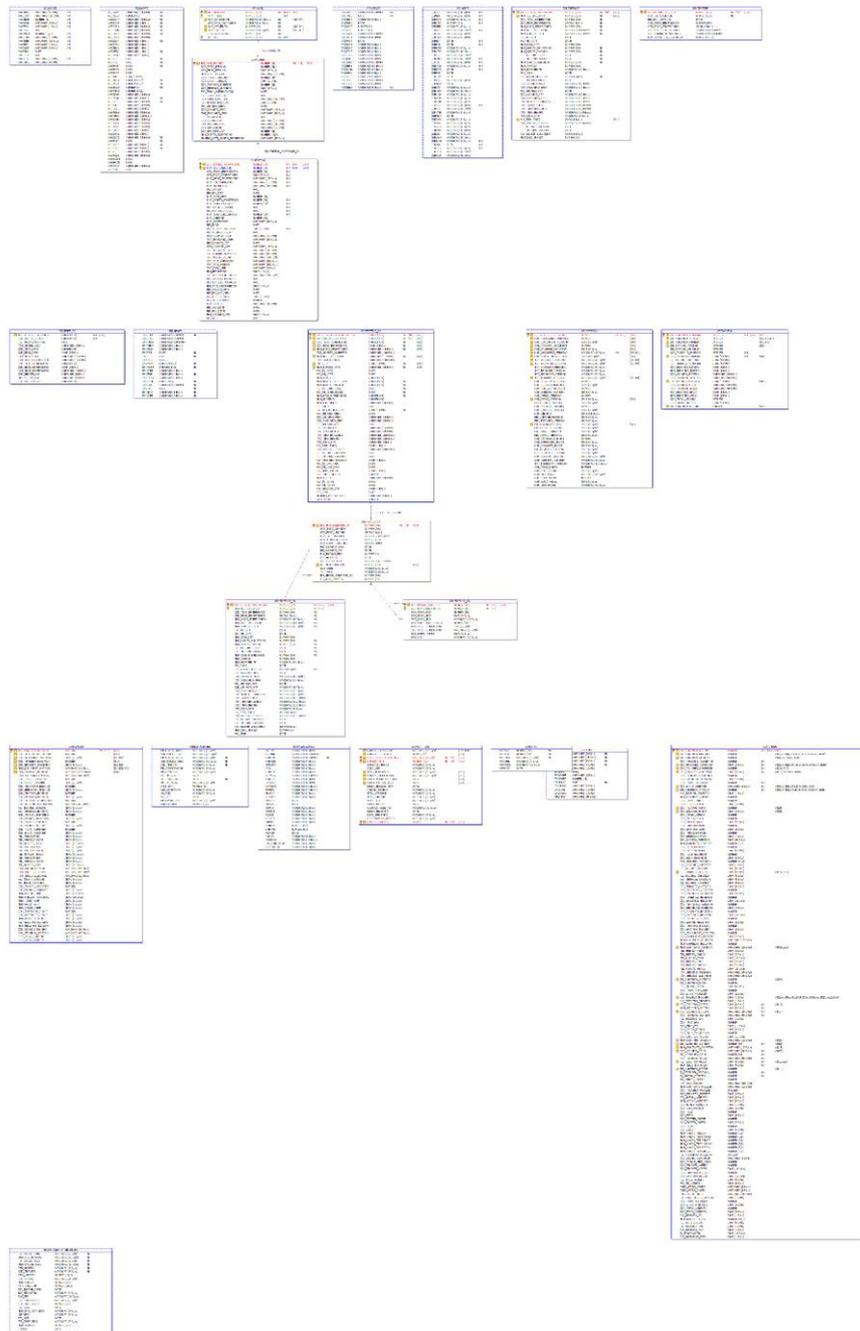
Fuente: Elaboración propia

La importancia se estima según los requerimientos del Product Owner y es calificada a partir de estados Alto, Medio y Bajo. El tiempo necesario para el desarrollo es proporcionado por los factores de tiempo sugeridos por SCRUM

para la realización de cada iteración y la experiencia en la implantación de componentes de sistemas del equipo de trabajo.

### 3.5.1.3 Modelo de Datos

Para la herramienta ETL se empleó el siguiente Diagrama de Entidad-Relación de la Base de Datos.



### 3.5.1.4 Diccionario de Datos

En el Diccionario de Datos, para ser precisos, se definen los datos que se usaron

en el sistema y se detalla la estructura de las tablas que se emplearon, está dispuesto de la siguiente manera.

Tabla 5. Modelo de Tabla del Diccionario de datos

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
[Nombre de la tabla]	[Tipo de Columna]	[Condición de nulidad]	[Descripción de la columna]	¿PK?	¿FK?

Fuente: Elaboración propia.

Tabla 6. csamperson: persona

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_NUMERICO_PERSONA	NUMBER	NOT NULL	CORRELATIVO DE TABLA PERSONA NUMERICO ENTERO DIFERENTE DE CERO	Yes	No
COD_TDOCUMENT_PERSONA	NUMBER	NULL	CODIGO TABLA TIPO DE DOCUMENTO DE IDENTIFICACION - TABLA COMPENDIO 101 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EDOCUMENT_PERSONA	CHAR(2)	NULL	CODIGO ELEMENTO TIPO DE DOCUMENTO DE IDENTIFICACION - TABLA COMPENDIO 101 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TPAISEMI_DOCUMENT	NUMBER	NULL	CODIGO TABLA PAIS EMISOR DOCUMENTO IDENTIDAD - TABLA COMPENDIO 608 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EPAISEMI_DOCUMENT	CHAR(3)	NULL	CODIGO ELEMENTO PAIS EMISOR DOCUMENTO IDENTIDAD - TABLA COMPENDIO 608 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_DOCUMENT_PERSONA	VARCHAR2(15)	NOT NULL	NUMERO DOCUMENTO IDENTIDAD PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_AUTOGENE_PERSONA	CHAR(15)	NULL	CODIGO AUTOGENERADO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TXT_APEPATER_PERSONA	VARCHAR2(40)	NULL	APELLIDO PATERNO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TXT_APEMATER_PERSONA	VARCHAR2(40)	NULL	APELLIDO MATERNO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TXT_APECASAD_PERSONA	VARCHAR2(40)	NULL	APELLIDO CASADA PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TXT_PRINOMBR_PERSONA	VARCHAR2(40)	NULL	PRIMER NOMBRE PERSONA TEXTO LIBRE DE	No	No

			CARACTERES ALFANUMERICOS		
<b>TXT_SEGNOMBR_PERSONA</b>	VARCHAR2(40)	NULL	SEGUNDO NOMBRE PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_NOMBRES_PERSONA</b>	VARCHAR2(40)	NULL	NOMBRES PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TESTCIVI_PERSONA</b>	NUMBER	NULL	CODIGO TABLA ESTADO CIVIL - TABLA COMPENDIO 106 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EESTCIVI_PERSONA</b>	CHAR(2)	NULL	CODIGO ELEMENTO ESTADO CIVIL - TABLA COMPENDIO 106 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TUBIGEO_NACIMIEN</b>	NUMBER	NULL	CODIGO TABLA UBIGEO NACIMIENTO - TABLA COMPENDIO 609 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EUBIGEO_NACIMIEN</b>	CHAR(9)	NULL	CODIGO ELEMENTO UBIGEO NACIMIENTO - TABLA COMPENDIO 609 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TSEXO_PERSONA</b>	NUMBER	NULL	CODIGO TABLA SEXO - TABLA COMPENDIO 105 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ESEXO_PERSONA</b>	CHAR(1)	NULL	CODIGO ELEMENTO SEXO - TABLA COMPENDIO 105 TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>FLG_ESTADO_INHABILI</b>	CHAR(1)	NULL	FLAG ESTADO INHABILITADO TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_TMOTIVO_INHABILI</b>	NUMBER	NULL	CODIGO TABLA TIPO MOTIVO INHABILITADO - TABLA COMPENDIO 806 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EMOTIVO_INHABILI</b>	CHAR(2)	NULL	CODIGO ELEMENTO TIPO MOTIVO INHABILITADO - TABLA COMPENDIO 806 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>NUM_CIE_RUC</b>	CHAR(11)	NULL	NUMERO RUC DEL EMPLEADOR DE TH / CIE: CODIGO DE IDENTIFICACION DE EMPLEADOR PARA TH TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FEC_INICVIGE_PERSONA</b>	CHAR(8)	NULL	FECHA INICIO VIGENCIA PERSONA. FORMATO AAAAMMDD	No	No
<b>FEC_TERMVIGE_PERSONA</b>	CHAR(8)	NULL	FECHA TERMINO VIGENCIA PERSONA	No	No
<b>FEC_NACIMI_PERSONA</b>	CHAR(8)	NULL	FECHA NACIMIENTO PERSONA. FORMATO AAAAMMDD	No	No
<b>FEC_FALLEC_PERSONA</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>FEC_INVALIDI_PERPARCI</b>	CHAR(8)	NULL	FECHA INVALIDEZ PARCIAL PERSONA. FORMATO	No	No

			AAAAMDD		
FEC_INVALIDI_PERTOTAL	CHAR(8)	NULL	FECHA INVALIDEZ TOTAL PERSONA. FORMATO AAAAMDD	No	No
COD_TOOPERACIO_SUNAT	NUMBER	NULL	CODIGO TABLA OPERACION SUNAT - TABLA COMPENDIO 316 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EOPERACIO_SUNAT	CHAR(6)	NULL	CODIGO ELEMENTO OPERACION SUNAT - TABLA COMPENDIO 316 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_PAQUETE_SUNAT	CHAR(8)	NULL	NUMERO PAQUETE SUNAT TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO AAAAMDD	No	No
HOR_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO: HH:MM:SS	No	No
COD_USUARIO_SISTEMA	VARCHAR2(20)	NULL	NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TERMINAL_SISTEMA	VARCHAR2(80)	NULL	CODIGO TERMINAL SISTEMA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TXT_NOMBCOMP_PERSONA	VARCHAR2(120)	NULL	NOMBRE COMPLETO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TDOC_MADRE	INTEGER	NULL	CODIGO TABLA DOCUMENTO DE LA MADRE - TABLA COMPENDIO 101 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EDOC_MADRE	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_DOC_MADRE	VARCHAR2(15)	NULL	NUMERO DOCUMENTO IDENTIDAD PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TDOC_PADRE	INTEGER	NULL	CODIGO TABLA DOCUMENTO DE LA PADRE - TABLA COMPENDIO 101 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EDOC_PADRE	CHAR(2)	NULL	CODIGO ELEMENTO TIPO DE DOCUMENTO DE IDENTIFICACION - TABLA COMPENDIO 101 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_DOC_PADRE	VARCHAR2(15)	NULL	NUMERO DOCUMENTO IDENTIDAD PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

Fuente: Elaboración propia.

Tabla 7. acreditaciones: acreditaciones del asegurado

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
------------------	--------------	--------------	---------------------------	-------	-------

		NULIDAD			
TIP_DOCIDE_EMPL	VARCHAR2(2)	NULL	CORRELATIVO DE TABLA PERSONA NUMERICO ENTERO DIFERENTE DE CERO	No	No
NUM_DOCIDE_EMPL	VARCHAR2(15)	NULL	CODIGO TABLA TIPO DE DOCUMENTO DE IDENTIFICACION - TABLA COMPENDIO 101 NUMERICO ENTERO DIFERENTE DE CERO	No	No
TIP_DOCIDE_ASEG	VARCHAR2(2)	NOT NULL	CODIGO ELEMENTO TIPO DE DOCUMENTO DE IDENTIFICACION - TABLA COMPENDIO 101 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_DOCIDE_ASEG	VARCHAR2(15)	NOT NULL			
COD_TRIBUTO	VARCHAR2(6)	NOT NULL	CODIGO TABLA PAIS EMISOR DOCUMENTO IDENTIDAD - TABLA COMPENDIO 608 NUMERICO ENTERO DIFERENTE DE CERO	No	No
TIP_ACREDITACION	VARCHAR2(3)	NULL	CODIGO ELEMENTO PAIS EMISOR DOCUMENTO IDENTIDAD - TABLA COMPENDIO 608 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TIP_ATENCION	VARCHAR2(1)	NULL	NUMERO DOCUMENTO IDENTIDAD PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_INVIG_ACRE	DATE	NULL	CODIGO AUTOGENERADO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_FINVIG_ACRE	DATE	NOT NULL	APELLIDO PATERNO PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
IND_FACTURAC	VARCHAR2(1)	NULL	INDICADOR DE FACTURACION TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
CANT_EMPL	NUMBER (2)	NULL	CANTIDAD DE EMPLEADORES	No	No
NUM_CONTRATO	VARCHAR2(8)	NULL	NUMERO CONTRATO	No	No
USUARIO	VARCHAR2(8)	NULL	CODIGO USUARIO	No	No
FEC_ACT	DATE	NULL	FECHA ACTUALIZACION	No	No
TIP_OPERACION	VARCHAR2(3)	NULL	TIPO DE OPERACION	No	No
NUM_CORNOV	NUMBER(10)	NULL	ORDEN DE SECUENCIA DE LA NOVEDAD PROCESADA	No	No

Fuente: Elaboración propia.

Tabla 8. nuevo\_cuenta\_individual: cuenta individual del asegurado

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
TIP_DOCIDE_EMPL	VARCHAR2 (2 Byte)	NOT NULL	Tipo de documento de identidad del empleador	No	No
NUM_DOCIDE_EMPL	VARCHAR2 (15 Byte)	NOT NULL	Número del documento de identidad del empleador	No	No
TIP_DOCIDE_ASEG	VARCHAR2 (2 Byte)	NOT NULL	Tipo de documento de identidad del asegurado	No	No
NUM_DOCIDE_ASEG	VARCHAR2 (15 Byte)	NOT NULL	Número del documento de identidad del asegurado	No	No
PER_APORTA	VARCHAR2 (6 Byte)	NOT NULL	Periodo de aportación (aaaamm)	No	No
COD_TRIBUTO	VARCHAR2 (6 Byte)	NOT NULL	Código de tributo	No	No
MTO_APORTA	NUMBER (14,2)	NULL	Monto de la aportación	No	No
COD_FORMUL	VARCHAR2 (4 Byte)	NULL	Número de formulario	No	No
NUM_ORDEN	NUMBER (10)	NULL	Número de orden del formulario	No	No
MTO_BASE_IMP	NUMBER (14,2)	NULL	Monto de la base imponible	No	No
FEC_INSCRIP_ASEG	DATE	NULL	FECHA DE INSCRIPCION (mm/dd/aaaa)	No	No
IND_SITUACION	VARCHAR2 (2 Byte)	NULL	Indicador de situación	No	No
RUC_EPS	VARCHAR2 (11 Byte)	NULL	Número de ruc de EPS	No	No
TIP_TRABAJADOR	VARCHAR2 (2 Byte)	NULL	Tipo de trabajador	No	No
CTD_DIAS_LABOR	NUMBER (10)	NULL	Días laborados	No	No
FEC_CESE	DATE	NULL	FECHA DE CESE O DE BAJA (mm/dd/aaaa)	No	No
NUM_PAQ_ACTUALIZA	VARCHAR2 (8 Byte)	NULL	Número de paquete que actualiza el registro	No	No
USUARIO	VARCHAR2 (8 Byte)	NULL	Código del usuario que genera el archivo.	No	No
FEC_ACT	DATE	NULL	Fecha y hora de actualización del registro	No	No
TIP_OPERACION	VARCHAR2 (3 Byte)	NULL	Tipo de operación	No	No
NUM_CORNOV	NUMBER (10)	NULL	Orden de secuencia de la novedad procesada	No	No
FCARGA	DATE	NULL	Fecha de carga del registro en EsSalud	No	No
IND_OPORTUNIDAD	VARCHAR2 (1 Byte)	NULL	Indicador de oportunidad	No	No

Fuente: Elaboración propia.

Tabla 9. ssahceit: citts del asegurado

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>CEITCIT</b>	VARCHAR2 (1 Byte)	NOT NULL		Yes	No
<b>CEIPERI</b>	VARCHAR2 (2 Byte)	NOT NULL		Yes	No
<b>CEICCAS</b>	VARCHAR2 (3 Byte)	NOT NULL		Yes	No
<b>CEINCER</b>	NUMBER (10)	NOT NULL		Yes	No
<b>CEICSER</b>	VARCHAR2 (3 Byte)	NULL		No	No
<b>CEICAUT</b>	VARCHAR2 (15 Byte)	NOT NULL		No	No
<b>CEIDIAG</b>	VARCHAR2 (7 Byte)	NULL		No	No
<b>CEIDIA2</b>	VARCHAR2 (7 Byte)	NULL		No	No
<b>CEIFINI</b>	DATE	NOT NULL		No	No
<b>CEIFFIN</b>	DATE	NOT NULL		No	No
<b>CEICMED</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>CEIACTM</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>CEIHCLI</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>CEINOMB</b>	VARCHAR2 (40 Byte)	NOT NULL		No	No
<b>CEITIAT</b>	VARCHAR2 (2 Byte)	NOT NULL		No	No
<b>CEICTIG</b>	VARCHAR2 (2 Byte)	NOT NULL		No	No
<b>CEIFOTD</b>	DATE	NOT NULL		No	No
<b>CEIFEAL</b>	DATE	NULL		No	No
<b>CEIFEPP</b>	DATE	NULL		No	No
<b>CEICUAC</b>	VARCHAR2 (8 Byte)	NULL		No	No
<b>CEIFACT</b>	DATE	NULL		No	No
<b>CEIHACT</b>	VARCHAR2 (8 Byte)	NULL		No	No
<b>CEICTER</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEIFLAG</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEICONT</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEIVCMP</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>CEITDOC</b>	VARCHAR2 (1 Byte)	NOT NULL		No	No
<b>CEINDOC</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEITANU</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>CEIFANU</b>	DATE	NULL		No	No
<b>CEICVER</b>	VARCHAR2 (7 Byte)	NULL		No	No
<b>CEITFLA</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEITPRO</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEIICOP</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEIICSU</b>	VARCHAR2 (1 Byte)	NULL		No	No

<b>CEICASU</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEICUAH</b>	VARCHAR2 (8 Byte)	NOT NULL		No	No
<b>CEIFACH</b>	DATE	NOT NULL		No	No
<b>CEIHACH</b>	VARCHAR2 (8 Byte)	NOT NULL		No	No
<b>CEICTEH</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEINRUC</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEICVEE</b>	VARCHAR2 (7 Byte)	NULL		No	No
<b>CEICNIT</b>	VARCHAR2 (30 Byte)	NULL		No	No
<b>CEICTCE</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEIPANU</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>CEICTRA</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>CEICCAR</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>CEIFFUR</b>	DATE	NULL		No	No
<b>CEICINP</b>	VARCHAR2 (1 Byte)	NULL		No	No

Fuente: Elaboración propia.

Tabla 10. contribuyentes: contribuyentes

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>LLTTT</b>	VARCHAR2 (7 Byte)	NULL		No	No
<b>NOMBRE</b>	VARCHAR2 (100 Byte)	NULL		No	No
<b>NUMRUC</b>	VARCHAR2 (11 Byte)	NOT NULL		Yes	No
<b>NUMREG</b>	VARCHAR2 (4 Byte)	NULL		No	No
<b>TIPOEMP</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>CIU</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>UBIGEO</b>	VARCHAR2 (6 Byte)	NULL		No	No
<b>NOMVIA</b>	VARCHAR2 (25 Byte)	NULL		No	No
<b>NUMER1</b>	VARCHAR2 (4 Byte)	NULL		No	No
<b>INTER1</b>	VARCHAR2 (4 Byte)	NULL		No	No
<b>NOMZON</b>	VARCHAR2 (25 Byte)	NULL		No	No
<b>REFER1</b>	VARCHAR2 (40 Byte)	NULL		No	No
<b>FLAG22</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>ESTADO</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>FECALT</b>	DATE	NULL		No	No
<b>FECBAJ</b>	DATE	NULL		No	No
<b>TIPVIA</b>	VARCHAR2 (2 Byte)	NULL		No	No

<b>TIPZON</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>DOBLE</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>DEPORI</b>	VARCHAR2 (4 Byte)	NULL		No	No
<b>OPERAC</b>	VARCHAR2 (3 Byte)	NULL		No	No
<b>CORNOV</b>	NUMBER (10)	NULL		No	No
<b>NUMPRO</b>	NUMBER (10)	NULL		No	No
<b>FCARGA</b>	DATE	NULL		No	No
<b>INDICA</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>ARCHIVO</b>	VARCHAR2 (40 Byte)	NULL		No	No
<b>IND_EMPHOGAR</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>EXISTENCIA</b>	VARCHAR2 (1 Byte)	NULL		No	No

Fuente: Elaboración propia.

Tabla 11. csamentidad: entidad

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>IDE_NUMERICO_ENTIDAD</b>	INTEGER	NOT NULL	COORELATIVO TABLA ENTIDAD NUMERICO ENTERO DIFERENTE DE CERO	Yes	No
<b>COD_TDOCUMENT_ENTIDAD</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EDOCUMENT_ENTIDAD</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TPAISEMI_DOCUMENT</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EPAISEMI_DOCUMENT</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>NUM_DOCUMENT_ENTIDAD</b>	VARCHAR2(15)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_RASOCIAL_ENTIDAD</b>	VARCHAR2(100)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_MARCOMER_ENTIDAD</b>	VARCHAR2(40)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TTIPO_CONTRIBU</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ETIPO_CONTRIBU</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TRELACION_ESSALUD</b>	INTEGER	NULL	COMPENDIO: 904 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ERELACION_ESSALUD</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TESTADO_ENTIDAD</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EESTADO_ENTIDAD</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TDEPENDEN_ORISUNAT</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No

<b>COD_EDEPENDEN_ORISUNAT</b>	VARCHAR2(6)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FEC_INICVIGE_ENTIDAD</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>FEC_TERMVIGE_ENTIDAD</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>COD_TINDICA_NOHABIDO</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EINDICA_NOHABIDO</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TINDICA_MULTINSC</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EINDICA_MULTINSC</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TTIPO_PERSONER</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ETIPO_PERSONER</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>TEL_PREFIJ_PAIS1</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_PREFIJ_CIUDAD1</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_NUMERO_FIJO1</b>	CHAR(10)	NULL	NUMERO ENTERO, CODIFICACION LOCAL TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_PREFIJ_PAIS2</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_PREFIJ_CIUDAD2</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_NUMERO_MOVI2</b>	CHAR(10)	NULL	NUMERO ENTERO, CODIFICACION LOCAL TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_PREFIJ_PAIS3</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_PREFIJ_CIUDAD3</b>	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TEL_NUMERO_FAX3</b>	CHAR(10)	NULL	NUMERO ENTERO, CODIFICACION LOCAL TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_DIRPOR_ENTIDAD</b>	VARCHAR2(50)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_DIRECC_ELECTRO1</b>	VARCHAR2(50)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_DIRECC_ELECTRO2</b>	VARCHAR2(50)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FEC_CONSTI_ENTIDAD</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>FEC_INICIO_ACTIVIDA</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>COD_TINDICA_COMEXTER</b>	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EINDICA_COMEXTER</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

NUM_PARTID_RRPP	CHAR(11)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_NUMERO_PATRONAL	CHAR(14)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_TOMO_RRPP	CHAR(8)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_FOLIO_RRPP	CHAR(5)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_ASIENT_RRPP	CHAR(5)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TOOPERACIO_SUNAT	INTEGER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EOPERACIO_SUNAT	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_PAQUETE_SUNAT	CHAR(8)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
HOR_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO: HH:MM:SS	No	No
COD_USUARIO_SISTEMA	VARCHAR2(20)	NULL	NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TERMINAL_SISTEMA	VARCHAR2(80)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

Fuente: Elaboración propia.

Tabla 12. scchemot: histórico de emisiones

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
EMOCTDI	VARCHAR2 (2 Byte)	NOT NULL		Yes	No
EMONDID	VARCHAR2 (15 Byte)	NOT NULL		Yes	No
EMOPACR	VARCHAR2 (6 Byte)	NOT NULL		Yes	No
EMOCTRI	VARCHAR2 (6 Byte)	NOT NULL		Yes	No
EMOIREM	VARCHAR2 (1 Byte)	NULL		No	No
EMOFACT	DATE	NULL		No	No
EMOCOPE	VARCHAR2 (3 Byte)	NULL		No	No
EMOOSSEC	NUMBER (10)	NULL		No	No
EMOFCAR	DATE	NULL		No	No
EMOTIVO	VARCHAR2 (2 Byte)	NULL		No	No
EMOIVAL	VARCHAR2 (1 Byte)	NULL		No	No
EMOIFLAG	VARCHAR2 (1 Byte)	NULL		No	No
ARCHIVO	VARCHAR2 (30 Byte)	NULL		No	No

Fuente: Elaboración propia.

Tabla 13. sscmcitt: citts procesados

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>CITCCAS</b>	VARCHAR2 (4 Byte)	NOT NULL			
<b>CITNCER</b>	NUMBER (10)	NOT NULL			
<b>CITCSER</b>	VARCHAR2 (2 Byte)	NULL			
<b>CITCAUT</b>	VARCHAR2 (15 Byte)	NOT NULL			
<b>CITCDIA</b>	VARCHAR2 (6 Byte)	NULL			
<b>CITFIIN</b>	DATE	NULL			
<b>CITFFIN</b>	DATE	NULL			
<b>CITFECE</b>	DATE	NULL			
<b>CITFALT</b>	DATE	NULL			
<b>CITCMED</b>	VARCHAR2 (5 Byte)	NOT NULL			
<b>CITCTIP</b>	VARCHAR2 (1 Byte)	NULL			
<b>CITCTAT</b>	VARCHAR2 (1 Byte)	NULL			
<b>CITCTSU</b>	VARCHAR2 (2 Byte)	NULL			
<b>CITFPPA</b>	DATE	NULL			
<b>CITNEXP</b>	VARCHAR2 (10 Byte)	NULL			
<b>CITNSEC</b>	NUMBER (5)	NULL			
<b>CITCZDE</b>	VARCHAR2 (3 Byte)	NOT NULL			
<b>CITICIT</b>	VARCHAR2 (1 Byte)	NOT NULL			
<b>CITCUAC</b>	VARCHAR2 (10 Byte)	NOT NULL			
<b>CITFACT</b>	DATE	NOT NULL			
<b>CITHACT</b>	VARCHAR2 (8 Byte)	NOT NULL			
<b>CITCTER</b>	VARCHAR2 (5 Byte)	NOT NULL			
<b>CITFLAG</b>	VARCHAR2 (1 Byte)	NULL			

Fuente: Elaboración propia.

Tabla 14. ssumaset: expedientes de subsidios

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>ASECEXP</b>	VARCHAR2 (10 Byte)	NOT NULL		Yes	No
<b>ASENSEX</b>	NUMBER (5)	NOT NULL		Yes	No
<b>ASECAUT</b>	VARCHAR2 (15 Byte)	NOT NULL		No	No
<b>ASECTAS</b>	VARCHAR2 (2 Byte)	NOT NULL		No	No
<b>ASETAPA</b>	VARCHAR2 (20 Byte)	NOT NULL		No	No

<b>ASETAMA</b>	VARCHAR2 (20 Byte)	NOT NULL		No	No
<b>ASETPNO</b>	VARCHAR2 (10 Byte)	NOT NULL		No	No
<b>ASETSNO</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>ASECLEY</b>	VARCHAR2 (1 Byte)	NOT NULL		No	No
<b>ASECTSU</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>ASECBAN</b>	VARCHAR2 (4 Byte)	NULL		No	No
<b>ASECESO</b>	VARCHAR2 (1 Byte)	NOT NULL		No	No
<b>ASECMOT</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>ASEFEST</b>	DATE	NOT NULL		No	No
<b>ASEFISI</b>	DATE	NOT NULL		No	No
<b>ASEFPRE</b>	DATE	NOT NULL		No	No
<b>ASEFPRO</b>	DATE	NULL		No	No
<b>ASEFATR</b>	DATE	NULL		No	No
<b>ASEITRE</b>	CHAR (1 Byte)	NULL		No	No
<b>ASEMRHA</b>	NUMBER (15,2)	NOT NULL		No	No
<b>ASEMSDI</b>	NUMBER (15,2)	NOT NULL		No	No
<b>ASEQDSU</b>	NUMBER (5)	NOT NULL		No	No
<b>ASEMTSU</b>	NUMBER (15,2)	NULL		No	No
<b>ASENDOC</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>ASETDOC</b>	VARCHAR2 (2 Byte)	NOT NULL		No	No
<b>ASECLEL</b>	VARCHAR2 (10 Byte)	NOT NULL		No	No
<b>ASENTEL</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>ASETDBE</b>	VARCHAR2 (2 Byte)	NULL		No	No
<b>ASENDBE</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>ASETBEN</b>	VARCHAR2 (120 Byte)	NULL		No	No
<b>ASECAMA</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>ASECANI</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>ASEREXP</b>	VARCHAR2 (13 Byte)	NULL		No	No
<b>ASECACR</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>ASECAGE</b>	VARCHAR2 (3 Byte)	NULL		No	No
<b>ASECUAC</b>	VARCHAR2 (10 Byte)	NOT NULL		No	No
<b>ASEFACT</b>	DATE	NOT NULL		No	No
<b>ASEHACT</b>	VARCHAR2 (8 Byte)	NOT NULL		No	No
<b>ASECTER</b>	VARCHAR2 (5 Byte)	NOT NULL		No	No
<b>ASEFECE</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>ASENCBA</b>	VARCHAR2 (20)	NULL		No	No

	Byte)				
<b>ASEFICON</b>	DATE	NULL		No	No
<b>ASEFFCON</b>	DATE	NULL		No	No
<b>ASETCNV</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>ASEFIINC</b>	DATE	NULL		No	No

Fuente: Elaboración propia.

Tabla 15. sccwtact: aportes contabilizados

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>TACCAUT</b>	VARCHAR2 (15 Byte)	NOT NULL		Yes	No
<b>TACFVDE</b>	DATE	NOT NULL		Yes	No
<b>TACCECA</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>TACFVHA</b>	DATE	NULL		No	No
<b>TACNSAT</b>	NUMBER (9)	NULL		No	No
<b>TACNSAF</b>	NUMBER (9)	NULL		No	No
<b>TACCATI</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>TACCVFA</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>TACFACT</b>	DATE	NULL		No	No
<b>TACCUGD</b>	VARCHAR2 (6 Byte)	NULL		No	No
<b>TACCCAS</b>	VARCHAR2 (3 Byte)	NULL		No	No
<b>TACCTAT</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>TACCTAS</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>TACIFAC</b>	VARCHAR2 (1 Byte)	NULL		No	No
<b>TACCEMP</b>	VARCHAR2 (15 Byte)	NULL		No	No
<b>TACREPS</b>	VARCHAR2 (11 Byte)	NULL		No	No
<b>TACNCON</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>TACCPLA</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>TACHACT</b>	VARCHAR2 (8 Byte)	NULL		No	No
<b>TACCUAC</b>	VARCHAR2 (10 Byte)	NULL		No	No
<b>TACCTER</b>	VARCHAR2 (5 Byte)	NULL		No	No
<b>TACCREG</b>	VARCHAR2 (1 Byte)	NOT NULL		No	No

Fuente: Elaboración propia.

Tabla 16. ssumempt: expedientes procesados

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
EMPCEXP	VARCHAR2 (10 Byte)	NOT NULL		Yes	No
EMPCEMP	VARCHAR2 (15 Byte)	NULL		No	No
EMPTRSO	VARCHAR2 (100 Byte)	NULL		No	No
EMPTEMP	VARCHAR2 (2 Byte)	NULL		No	No
EMPCBAN	VARCHAR2 (2 Byte)	NULL		No	No
EMPFISI	DATE	NOT NULL		No	No
EMPFEST	DATE	NOT NULL		No	No
EMPFPRO	DATE	NULL		No	No
EMPMAME	NUMBER (15,2)	NOT NULL		No	No
EMPMREM	NUMBER (15,2)	NOT NULL		No	No
EMPCTRE	VARCHAR2 (1 Byte)	NULL		No	No
EMPIEXP	VARCHAR2 (1 Byte)	NOT NULL		No	No
EMPCMAN	VARCHAR2 (1 Byte)	NULL		No	No
EMPQSOL	NUMBER (5)	NOT NULL		No	No
EMPCUAC	VARCHAR2 (10 Byte)	NOT NULL		No	No
EMPFACT	DATE	NOT NULL		No	No
EMPHACT	VARCHAR2 (8 Byte)	NOT NULL		No	No
EMPCTER	VARCHAR2 (5 Byte)	NOT NULL		No	No

Fuente: Elaboración propia.

Tabla 17. ssuhanu: histórico de anulaciones

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_NUM_ANU	NUMBER(38)	NOT NULL	Clave primara	Yes	No
NUM_SOLICI	NUMBER(38)	NULL	Numero de solicitud	No	Yes
NUM_EXPEDIENTE	VARCHAR2(30)	NOT NULL	Numero de expediente	No	No
IDE_DOC	NUMBER(38)	NOT NULL	Clave primaria del documento	No	No
COD_TIPO_ANULACION	VARCHAR2(2)	NOT NULL	Código tipo anulacion	No	No
NUM_SECUENCIA	NUMBER(5)	NOT NULL	Numero de secuencia	No	No
NUM_EMISION	VARCHAR2(5)	NOT NULL	Numero de emisión	No	No
FLAG_REACTIVADO	NUMBER(1)	NOT NULL	Flag reactivado	No	No

Fuente: Elaboración propia.

Tabla 18. ssuhpagt: histórico de pagos

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
PAGNEMI	VARCHAR2 (5 Byte)	NOT NULL		Yes	No
PAGCEXP	VARCHAR2 (10 Byte)	NOT NULL		No	No
PAGNSCE	NUMBER (5)	NOT NULL		Yes	No
PAGCBAN	VARCHAR2 (4 Byte)	NOT NULL		No	No
PAGIEPA	VARCHAR2 (1 Byte)	NOT NULL		No	No
PAGFPAG	DATE	NULL		No	No
PAGMREM	NUMBER (15,2)	NOT NULL		No	No
PAGCTPA	VARCHAR2 (1 Byte)	NOT NULL		No	No
PAGNCER	VARCHAR2 (10 Byte)	NOT NULL		No	No
PAGIEST	VARCHAR2 (1 Byte)	NOT NULL		No	No
PAGCAGE	VARCHAR2 (4 Byte)	NOT NULL		No	No
PAGFEMI	DATE	NULL		No	No
PAGFVEN	DATE	NULL		No	No
PAGICTA	VARCHAR2 (1 Byte)	NOT NULL		No	No
PAGBAGE	VARCHAR2 (4 Byte)	NULL		No	No

Fuente: Elaboración propia.

Tabla 19. ssutsolici: solicitudes

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
NUM_MONTO_SUBSIDIADO	NUMBER(10,2)	NULL	Clave primaria, correlativo de solicitudes	Yes	No
IDE_NUM_SOLICITU	NUMBER(38)	NOT NULL	Tabla del tipo de documento del titular, según compendio del SAS.	No	No
COD_TDOC_TITULAR	NUMBER(38)	NULL	Código de elemento de la tabla 101 del SAS.	No	No
COD_EDOC_TITULAR	CHAR(2)	NULL	Numero de documento de identidad del titular	No	No
NUM_DOC_TITULAR	VARCHAR2(15)	NULL	Tabla del tipo de subsidio según compendio del SAS.	No	No
COD_TTIPO_SUBSIDIO	NUMBER(38)	NULL	Elemento de la tabla de tipo de subsidio según compendio del SAS, tabla 1101	No	No
COD_ETIPO_SUBSIDIO	CHAR(2)	NULL	Tabla del tipo de subsidio según compendio del SAS.	No	No
COD_TESTADO_SUBSIDIO	NUMBER(38)	NULL	Elemento de la tabla	No	No

			de estado del expediente según compendio del SAS, tabla 1108.		
<b>COD_EESTADO_SUBSIDIO</b>	CHAR(2)	NULL	Fecha de solicitud o presentación.	No	No
<b>FEC_FECHA_PRESENTACION</b>	DATE	NULL	Fecha en que se procesa la solicitud	No	No
<b>FEC_FECHA_PROCESO</b>	DATE	NULL	Clave primaria, correlativo de solicitudes	No	No
<b>COD_EXPEDIENTE_NIT</b>	VARCHAR2(30)	NULL	Código de expediente	No	No
<b>FLG_ACTIVO_INACTIVO</b>	CHAR(1)	NULL	1: activo, 0: inactivo	No	No
<b>FEC_USUARIO_CREA</b>	DATE	NULL	Fecha de creación del registro	No	No
<b>COD_USUARIO_CREA</b>	VARCHAR2(15)	NULL	Código del usuario que crea la validación. El código debe ser del sistema donde se haga el mantenimiento a la tabla.	No	No
<b>TXT_IPUSUARIO_CREA</b>	VARCHAR2(20)	NULL	IP del usuario que crea	No	No
<b>FEC_USUARIO_ACT</b>	DATE	NULL	Fecha y hora de la actualización del registro	No	No
<b>COD_USUARIO_ACT</b>	VARCHAR2(15)	NULL	Código del usuario que actualiza	No	No
<b>TXT_IPUSUARIO_ACT</b>	VARCHAR2(20)	NULL	IP del usuario que actualiza	No	No
<b>COD_EMPLEADOR</b>	VARCHAR2(11)	NULL	Ruc del empleador	No	No
<b>COD_IDE_REPOLACT</b>	NUMBER(38)	NULL	Clave foránea de la tabla replac	No	No

Fuente: Elaboración propia.

Tabla 20. ssutdetlac: detalle solicitudes de lactancia

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>IDE_DETALLE_LACTANCIA</b>	NUMBER(38)	NOT NULL	Clave primaria del detalle de lactancias	No	No
<b>COD_IDE_SOLICITUD</b>	NUMBER(38)	NOT NULL	Clave foránea. Correlativo de tabla solicitud.	No	No
<b>FEC_NACIMIENTO_HIJO</b>	DATE	NULL	Fecha de nacimiento del hijo o fecha de parto de la madre.	No	No
<b>TXT_ORIGEN_SOLI</b>	VARCHAR2(2)	NULL		No	No
<b>COD_TDOC_HIJO</b>	NUMBER(38)	NULL	Tabla del tipo de documento según compendio del SAS 101.	No	No
<b>COD_EDOC_HIJO</b>	CHAR(2)	NULL	Código de elemento de la tabla 101 del SAS.	No	No
<b>TXT_NDOC_HIJO</b>	VARCHAR2(15)	NULL	Numero de documento del hijo	No	No
<b>TXT_IDENTIFICADOR_HIJO</b>	VARCHAR2(15)	NULL	Identificador: cnv, cui, rn, etc.	No	No

<b>COD_TDOC_BENEFICIARIO</b>	NUMBER(38)	NULL	Tabla del tipo de documento según compendio del SAS 101.	No	No
<b>COD_EDOC_BENEFICIARIO</b>	CHAR(2)	NULL	Código de elemento de la tabla 101 del SAS.	No	No
<b>TXT_NDOC_BENEFICIARIO</b>	VARCHAR2(15)	NULL	Numero de documento del beneficiario	No	No
<b>COD_RENAES_CAS</b>	VARCHAR2(8)	NULL	Código renaes centro asistencial de nacimiento	No	No
<b>UBIGEO_CAS_NAC</b>	VARCHAR2(6)	NULL	Ubigeo del cas de nacimiento	No	No
<b>COD_IDE_REPOLACT</b>	NUMBER(38)	NULL	Correlativo de la solicitud cnv (ssuhreplac)	No	No

Fuente: Elaboración propia.

Tabla 21. ssutdetmat: detalle solicitudes de maternidad

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
<b>IDE_DETALLE_MATERNIDAD</b>	NUMBER (38)	NOT NULL	Clave primaria, seq de detalle maternidad (sq_ssutdetmat)	Yes	No
<b>NUM_IDE_SOLICITUD</b>	NUMBER (38)	NOT NULL	Correlativo de tabla solicitud	No	No
<b>COD_TDOC_BENEFICIARIO</b>	NUMBER (38)	NOT NULL	Tabla del tipo de documento según compendio del SAS 101.	No	No
<b>COD_EDOC_BENEFICIARIO</b>	CHAR (2 Byte)	NOT NULL	Código de elemento de la tabla 101 del sas.	No	No
<b>NUM_NDOC_BENEFICIARIO</b>	NUMBER (38)	NOT NULL	Número de documento del beneficiario	No	No
<b>NUM_RUC_EMPLEADOR</b>	VARCHAR2 (11 Byte)	NOT NULL	Número de ruc de la entidad empleadora	No	No
<b>NUM_NUMERO_CITT</b>	VARCHAR2 (20 Byte)	NOT NULL	Número de citt	No	No
<b>FEC_INI_CITT</b>	DATE	NULL	Fecha de inicio del periodo de incapacidad según el citt	No	No
<b>FEC_FIN_CITT</b>	DATE	NULL	Fecha de fin del periodo de incapacidad según el citt	No	No
<b>NUM_DIAS_CITT</b>	NUMBER (38)	NULL	Número de días de incapacidad según el citt	No	No
<b>NUM_MONTO_SOLICITADO</b>	NUMBER (38)	NOT NULL	Monto a subsidiar solicitado	No	No
<b>NUM_MONTO_PAGADO</b>	NUMBER (38)	NOT NULL	Monto a subsidiar pagado	No	No
<b>FEC_INI_SUBS_SOLICI</b>	DATE	NOT NULL	Fecha de inicio del periodo a subsidiar solicitado	No	No
<b>FEC_FIN_SUBS_SOLICI</b>	DATE	NOT NULL	Fecha de fin del periodo a subsidiar solicitado	No	No
<b>NUM_DIAS_SUBSIDIADOS</b>	NUMBER (38)	NOT NULL	Número de días a subsidiar solicitados	No	No

<b>NUM_ARMADA</b>	NUMBER (38)	NULL	Número de armada (1 o 2)	No	No
<b>NUM_EXPEDIENTE</b>	VARCHAR2 (30 Byte)	NULL	Número de expediente que ordena el pago	No	No
<b>FEC_PAGO</b>	DATE	NULL	Fecha de pago según el cronograma	No	No
<b>FLG_ACTIVO_INACTIVO</b>	CHAR (1 Byte)	NOT NULL	1: activo, 0: inactivo	No	No
<b>FEC_USUARIO_CREA</b>	DATE	NULL	Fecha de creación del registro	No	No
<b>COD_USUARIO_CREA</b>	VARCHAR2 (15 Byte)	NULL	Código del usuario que crea la validación. El código deben ser del sistema donde se haga el mantenimiento a la tabla.	No	No
<b>TXT_IPUSUARIO_CREA</b>	VARCHAR2 (20 Byte)	NULL	Ip del usuario que crea	No	No
<b>FEC_USUARIO_ACT</b>	DATE	NULL	Fecha y hora de la actualización del registro	No	No
<b>COD_USUARIO_ACT</b>	VARCHAR2 (15 Byte)	NULL	Código del usuario que actualiza	No	No
<b>TXT_IPUSUARIO_ACT</b>	VARCHAR2 (20 Byte)	NULL	Ip del usuario que actualiza	No	No
<b>TXT_NOMBRE_COMPLETO</b>	VARCHAR2 (60 Byte)	NULL	Nombre completo de la asegurada	No	No
<b>TXT_RAZON_SOCIAL</b>	VARCHAR2 (100 Byte)	NULL	Razón social de la entidad empleadora	No	No
<b>TXT_TIPO_EMPLEADO</b>	VARCHAR2 (250 Byte)	NULL	Tipo de empleado según SIA	No	No
<b>TXT_TIPO_SEGURO</b>	VARCHAR2 (250 Byte)	NULL	Tipo de seguro	No	No
<b>TXT_ZONA_UPE</b>	VARCHAR2 (3 Byte)	NULL	Código de la upe	No	No
<b>FLG_RESULTADO</b>	CHAR (1 Byte)	NULL	Flag que indica el resultado del proceso: 1,orden pago; 2,backoffice sin marcas; 3,backoffice con marcas	No	No
<b>TXT_MOTIVO_RECHAZO</b>	VARCHAR2 (200 Byte)	NULL	Motivo de rechazo. Sólo para el caso de las solicitudes derivadas al backoffice	No	No
<b>FEC_INI_DIAS_PAGADOS</b>	DATE	NULL	Inicio de los días que se van a pagar	No	No
<b>FEC_FIN_DIAS_PAGADOS</b>	DATE	NULL	Fin de los días que se van a pagar (hay veces en que se descuentan días por labor remuendada)	No	No
<b>FLG_SECTOR_EDUCACION</b>	CHAR (1 Byte)	NULL	1: es del sector educacion, 0: no es del sector educación	No	No
<b>NUM_DECL_JURADA</b>	NUMBER (1)	NULL	1: presenta declaración jurada, 0: no presenta	No	No

Fuente: Elaboración propia.

Tabla 22. ssutdetinc: detalle solicitudes de incapacidades

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_DETALLE_INCAPACIDAD	NUMBER(38)	NOT NULL	Clave primaria, secuencia de detalle incapacidad (sq_ssutdetinc).	Yes	No
NUM_IDE_SOLICITUD	NUMBER(38)	NOT NULL	Correlativo de tabla solicitud	Yes	Yes
COD_TDOC_BENEFICIARIO	NUMBER(38)	NOT NULL	Tabla del tipo de documento según compendio del SAS 101.	No	No
COD_EDOC_BENEFICIARIO	CHAR(2)	NOT NULL	Código de elemento de la tabla 101 del SAS.	No	No
NUM_NDOC_BENEFICIARIO	VARCHAR2(8)	NOT NULL	Número de documento del beneficiario	No	No
NUM_RUC_EMPLEADOR	VARCHAR2(11)	NOT NULL	Número de ruc de la entidad empleadora.	No	No
NUM_NUMERO_CITT	VARCHAR2(20)	NOT NULL	Número de CITT	No	No
FEC_INI_CITT	DATE	NULL	Fecha de inicio del periodo de incapacidad según el CITT	No	No
FEC_FIN_CITT	DATE	NULL	Fecha de fin del período de incapacidad según el CITT	No	No
NUM_DIAS_CITT	NUMBER(38)	NULL	Número de días de incapacidad según el CITT	No	No
NUM_MONTO_SOLICITADO	NUMBER(38)	NOT NULL	Monto a subsidiar solicitado	No	No
NUM_MONTO_PAGADO	NUMBER(38)	NOT NULL	Monto que corresponde ser pagado.	No	No
FEC_INI_SUBS_SOLICI	DATE	NOT NULL	Fecha de inicio del periodo a subsidiar solicitado.	No	No
FEC_FIN_SUBS_SOLICI	DATE	NOT NULL	Fecha de fin del periodo a subsidiar solicitado.	No	No
NUM_DIAS_SUBSIDIADOS	NUMBER(38)	NOT NULL	Numero de días a subsidiar solicitados.	No	No
NUM_SUBSIDIO	NUMBER(38)	NULL	-	No	No
NUM_EXPEDIENTE	VARCHAR2(30)	NULL	Número de expediente: 112E801186(para pago con reembolso) 112S801174(para pago directo) 8567-2018-NIT-0004187(para expediente observados)	No	No

<b>FEC_PAGO</b>	DATE	NULL	Fecha de pago del subsidio.	No	No
<b>FLG_ACTIVADO_INACTIVO</b>	CHAR(1)	NOT NULL	Flag si está activo o inactivo el registro.	No	No
<b>FEC_USUARIO_CREA</b>	DATE	NULL	Fecha de creación del registro.	No	No
<b>COD_USUARIO_CREA</b>	VARCHAR2(15)	NULL	Código del usuario que crea la validación. El código debe ser del sistema donde se haga el mantenimiento a la tabla.	No	No
<b>TXT_IPUSUARIO_CREA</b>	VARCHAR2(20)	NULL	IP del usuario que crea	No	No
<b>FEC_USUARIO_ACT</b>	DATE	NULL	Fecha y hora de la actualización del registro.	No	No
<b>COD_USUARIO_ACT</b>	VARCHAR2(15)	NULL	Código del usuario que actualiza.	No	No
<b>TXT_IPUSUARIO_ACT</b>	VARCHAR2(20)	NULL	IP del usuario que actualiza	No	No
<b>TXT_NOMBRE_COMPLETO</b>	VARCHAR2(60)	NULL	Nombre completo de la asegurada.	No	No
<b>TXT_RAZON_SOCIAL</b>	VARCHAR2(100)	NULL	Razón social de la entidad empleadora.	No	No
<b>TXT_TIPO_EMPLEADO</b>	VARCHAR2(250)	NULL	Tipo de empleado según SIA	No	No
<b>TXT_TIPO_SEGURO</b>	VARCHAR2(250)	NULL	Tipo de seguro.	No	No
<b>TXT_ZONA_UPE</b>	VARCHAR2(3)	NULL	Código de la UPE.	No	No
<b>FLG_RESULTADO</b>	CHAR(1)	NULL	Flag que indica el resultado del proceso: 1,orden pago; 2,backoffice sin marcas; 3,backoffice con marcas	No	No
<b>TXT_MOTIVO_RECHAZO</b>	VARCHAR2(200)	NULL	Motivo de rechazo. Sólo para el caso de las solicitudes derivadas al backoffice.	No	No
<b>FEC_INI_DIAS_PAGADOS</b>	DATE	NULL	Fecha de inicio del periodo de incapacidad consecutivo	No	No
<b>FEC_FIN_DIAS_PAGADOS</b>	DATE	NULL	Fecha de fin del periodo de incapacidad consecutivo	No	No
<b>FLG_TIPO_SECTOREDUCA</b>	CHAR(1)	NULL	Flag de l tipo de sector de educación. 0: no es esducador 1: educador en el sector público 2: educador en el sector privado.	No	No
<b>FEC_INI_VAC_PRIV</b>	DATE	NULL	Fecha de inicio de las vacaciones privadas(sector privado)	No	No
<b>FEC_FIN_VAC_PRIV</b>	DATE	NULL	Fecha de fin de las vacaciones	No	No

			privadas(sector privado)		
FLG_DECL_JURADA	CHAR(1)	NULL	Flag si es que tiene declaración jurada. 0: no tiene, 1: tiene, 9: no le corresponde	No	No
FLG_SUBS_ENPARTE	NUMBER	NULL	Flag si el CITT número días recortados.	No	No
NUM_CITT_B	VARCHAR2(20)	NULL	Número de CITT que tiene recorte.	No	No
FEC_INI_CITTB	DATE	NULL	Fecha de inicio del periodo de incapacidad según el CITT con recorte.	No	No
FEC_FIN_CITTB	DATE	NULL	Fecha de fin del período de incapacidad según el CITT con recorte.	No	No
FLG_AFILIADO_CTR	CHAR(1)	NULL	Flag si es afiliado al CTR. 0: no es afiliado, 1: si es afiliado, 9: no le corresponde.	No	No
FEC_CESE	DATE	NULL	Fecha de cese	No	No

Fuente: Elaboración propia.

Tabla 23. ssutdetsep: detalle solicitudes de sepelio

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_DETALLE_SEPELIO	NUMBER(38)	NOT NULL		Yes	No
COD_IDE_SOLICITUD	NUMBER(38)	NOT NULL		No	No
FEC_FALL_TITULAR	DATE	NULL		No	No
COD_TDOC_BENEFICIARIO	NUMBER(38)	NULL		No	No
COD_EDOC_BENEFICIARIO	CHAR(2)	NULL		No	No
TXT_NDOC_BENEFICIARIO	VARCHAR2(15)	NULL		No	No
TXT_NOMBRE_BENEFICIARIO	VARCHAR2(120)	NULL		No	No
NUM_IMPORTE_DECLARACION	NUMBER(10,2)	NULL		No	No

Fuente: Elaboración propia.

Tabla 24. ssutsolici\_ov: solicitudes subsidios ospe virtual

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_NUM_SOLICITUD_OV	NUMBER(38)	NOT NULL		Yes	No
COD_TDOC_USUARIO	NUMBER(38)	NULL		No	No
COD_EDOC_USUARIO	CHAR(2)	NULL		No	No
NUM_DOC_USUARIO	VARCHAR2(15)	NULL		No	No
COD_TTIPO_SUBSIDIO	NUMBER(38)	NULL		No	No
COD_ETIPO_SUBSIDIO	CHAR(2)	NULL		No	No

FEC_USUARIO_CREA	DATE	NULL		No	No
FEC_USUARIO_ACT	DATE	NULL		No	No
FLG_ESTADO_PRO	NUMBER(1)	NULL		No	No
FEC_PROCESO	DATE	NULL		No	No
URL_DOCUMENTO	VARCHAR2(200)	NULL		No	No
IDE_NUM_SOLICITUD	NUMBER(38)	NULL		No	No
COD_OSPE	VARCHAR2(3)	NULL		No	No
TIP_PAGO	VARCHAR2(1)	NULL		No	No
IDE_GROUP_SOLICITUD_OV	NUMBER(38)	NULL		No	No

Fuente: Elaboración propia.

Tabla 25. ssutdetinc\_ov: solicitudes de incapacidad ospe virtual

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_DETALLE_INCAPACIDAD_OV	NUMBER(38)	NOT NULL		Yes	No
IDE_NUM_SOLICITUD_OV	NUMBER(38)	NOT NULL		No	No
COD_TDOC_BENEFICIARIO	NUMBER(38)	NOT NULL		No	No
COD_EDOC_BENEFICIARIO	CHAR(2)	NOT NULL		No	No
NUM_NDOC_BENEFICIARIO	VARCHAR2(8)	NOT NULL	Número de documento del beneficiario	No	No
TXT_NOMBRE_COMPLETO	VARCHAR2(100)	NULL		No	No
NUM_RUC_EMPLEADOR	VARCHAR2(11)	NOT NULL	Número de ruc de la entidad empleadora.	No	No
TXT_RAZON_SOCIAL	VARCHAR2(100)	NULL		No	No
TIP_CITT	CHAR(1)	NOT NULL		No	No
NUM_NUMERO_CITT	VARCHAR2(20)	NOT NULL		No	No
FEC_INI_CITT	DATE	NULL		No	No
FEC_FIN_CITT	DATE	NULL		No	No
NUM_DIAS_CITT	NUMBER(38)	NULL		No	No
NUM_MONTO_SOLICITADO	NUMBER(38)	NOT NULL	Monto a subsidiar solicitado	No	No
NUM_MONTO_PAGADO	NUMBER(38)	NOT NULL	Monto que corresponde ser pagado.	No	No
FEC_INI_SUBS_SOLICI	DATE	NOT NULL		No	No
FEC_FIN_SUBS_SOLICI	DATE	NOT NULL		No	No
NUM_DIAS_SUBSIDIADOS	NUMBER(38)	NOT NULL		No	No
NUM_SUBSIDIO	NUMBER(38)	NULL		No	No
NUM_EXPEDIENTE	VARCHAR2(30)	NULL		No	No
FEC_PAGO	DATE	NULL		No	No
FLG_ACTIVO_INACTIVO	CHAR(1)	NOT NULL		No	No
FEC_USUARIO_CREA	DATE	NULL		No	No
COD_USUARIO_CREA	VARCHAR2(15)	NULL		No	No
TXT_IPUSUARIO_CREA	VARCHAR2(20)	NULL		No	No
FEC_USUARIO_ACT	DATE	NULL		No	No

COD_USUARIO_ACT	VARCHAR2(15)	NULL		No	No
TXT_IPUSUARIO_ACT	VARCHAR2(20)	NULL		No	No
TXT_TIPO_EMPLEADO	VARCHAR2(250)	NULL		No	No
TXT_TIPO_SEGURO	VARCHAR2(250)	NULL		No	No
TXT_ZONA_UPE	VARCHAR2(3)	NULL		No	No
FLG_RESULTADO	CHAR(1)	NULL		No	No
TXT_MOTIVO_RECHAZO	VARCHAR2(200)	NULL		No	No
FEC_INI_DIAS_PAGADOS	DATE	NULL		No	No
FEC_FIN_DIAS_PAGADOS	DATE	NULL		No	No
FLG_TIPO_SECTOREDUCA	CHAR(1)	NULL		No	No
FEC_INI_VAC_PRIV	DATE	NULL		No	No
FEC_FIN_VAC_PRIV	DATE	NULL		No	No
FLG_DECL_JURADA	CHAR(1)	NULL		No	No
FLG_SUBS_ENPARTE	NUMBER	NULL		No	No
NUM_CITT_B	VARCHAR2(20)	NULL		No	No
FEC_INI_CITTB	DATE	NULL		No	No
FEC_FIN_CITTB	DATE	NULL		No	No
FLG_AFILIADO_CTR	CHAR(1)	NULL		No	No
FEC_CESE	DATE	NULL		No	No

Fuente: Elaboración propia.

Tabla 26. ssutdetmat\_ov: solicitudes de maternidad ospe virtual

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_DETALLE_MATERNIDAD	NUMBER (38)	NOT NULL		Yes	No
NUM_IDE_SOLICITUD	NUMBER (38)	NOT NULL		No	Yes
COD_TDOC_BENEFICIARIO	NUMBER (38)	NOT NULL		No	No
COD_EDOC_BENEFICIARIO	CHAR (2 Byte)	NOT NULL		No	No
NUM_NDOC_BENEFICIARIO	NUMBER (38)	NOT NULL		No	No
NUM_RUC_EMPLEADOR	VARCHAR2 (11 Byte)	NOT NULL		No	No
NUM_NUMERO_CITT	VARCHAR2 (20 Byte)	NOT NULL		No	No
FEC_INI_CITT	DATE	NULL		No	No
FEC_FIN_CITT	DATE	NULL		No	No
NUM_DIAS_CITT	NUMBER (38)	NULL		No	No
NUM_MONTO_SOLICITADO	NUMBER (38)	NOT NULL		No	No
NUM_MONTO_PAGADO	NUMBER (38)	NOT NULL		No	No
FEC_INI_SUBS_SOLICI	DATE	NOT NULL		No	No
FEC_FIN_SUBS_SOLICI	DATE	NOT NULL		No	No
NUM_DIAS_SUBSIDIADOS	NUMBER (38)	NOT NULL		No	No
NUM_ARMADA	NUMBER (38)	NULL		No	No
NUM_EXPEDIENTE	VARCHAR2 (30 Byte)	NULL		No	No

FEC_PAGO	DATE	NULL		No	No
FLG_ACTIVADO_INACTIVO	CHAR (1)	NOT NULL		No	No
FEC_USUARIO_CREA	DATE	NULL		No	No
COD_USUARIO_CREA	VARCHAR2 (15)	NULL		No	No
TXT_IPUSUARIO_CREA	VARCHAR2 (20)	NULL		No	No
FEC_USUARIO_ACT	DATE	NULL		No	No
COD_USUARIO_ACT	VARCHAR2 (15)	NULL		No	No
TXT_IPUSUARIO_ACT	VARCHAR2 (20)	NULL		No	No
TXT_NOMBRE_COMPLETO	VARCHAR2 (60)	NULL		No	No
TXT_RAZON_SOCIAL	VARCHAR2 (100)	NULL		No	No
TXT_TIPO_EMPLEADO	VARCHAR2 (250)	NULL		No	No
TXT_TIPO_SEGURO	VARCHAR2 (250)	NULL		No	No
TXT_ZONA_UPE	VARCHAR2 (3)	NULL		No	No
FLG_RESULTADO	CHAR (1)	NULL		No	No
TXT_MOTIVO_RECHAZO	VARCHAR2 (200)	NULL		No	No
FEC_INI_DIAS_PAGADOS	DATE	NULL		No	No
FEC_FIN_DIAS_PAGADOS	DATE	NULL		No	No
FLG_SECTOR_EDUCACION	CHAR (1 Byte)	NULL		No	No
NUM_DECL_JURADA	NUMBER (1)	NULL		No	No
FEC_CESE	DATE	NULL		No	No

Fuente: Elaboración propia.

Tabla 27. ssutdetlac\_ov: solicitudes de lactancia ospe virtual

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_DETALLE_LACT_OV	NUMBER(38)	NOT NULL		Yes	No
IDE_NUM_SOLICITUD_OV	NUMBER(38)	NOT NULL		No	No
COD_TDOC_HIJO	NUMBER(38)	NULL		No	No
COD_EDOC_HIJO	CHAR(2)	NULL		No	No
TXT_NDOC_HIJO	VARCHAR2(15)	NULL		No	No
COD_TDOC_BENEFICIARIO	NUMBER(38)	NULL		No	No
COD_EDOC_BENEFICIARIO	CHAR(2)	NULL		No	No
TXT_NDOC_BENEFICIARIO	VARCHAR2(15)	NULL		No	No
COD_ETIPO_PARTO	CHAR(1)	NULL		No	No
COD_CUI	VARCHAR2(15)	NULL		No	No

Fuente: Elaboración propia.

Tabla 28: csarvinfam: vínculo familiar

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_NUMERICO_VINCFAMI	INTEGER	NOT NULL	CORRELATIVO DE TABLA NUMERICO ENTERO DIFERENTE DE CERO	Yes	No
IDE_NUMERICO_TITULAR	INTEGER	NOT NULL	CORRELATIVO DEL ELEMENTO DE TABLA CSAMPERSON - PARA EL TITULAR NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
IDE_NUMERICO_FAMILIAR	INTEGER	NOT NULL	CORRELATIVO DEL ELEMENTO DE TABLA CSAMPERSON - PARA EL FAMILIAR NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
IDE_ENTIDAD_TITULAR	INTEGER	NULL	CORRELATIVO DEL ELEMENTO DE TABLA CSAMENTIDA - PARA LA ENTIDAD TITULAR NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
IDE_ENTIDAD_VINCULAD	INTEGER	NULL	CORRELATIVO DEL ELEMENTO DE TABLA CSAMENTIDA - PARA LA ENTIDAD VINCULADA NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
COD_TVINCULO_FAMILIAR	INTEGER	NOT NULL	CODIGO TABLA VINCULO FAMILIAR - TABLA COMPENDIO 326 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EVINCULO_FAMILIAR	CHAR(4)	NOT NULL	CODIGO ELEMENTO DE TABLA VINCULO FAMILIAR - TABLA COMPENDIO 326 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_INICVIGE_VINCFAMI	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
FEC_TERMVIGE_VINCFAMI	CHAR(8)	NULL	FECHA TERMINO VIGENCIA DE TABLA VINCULO FAMILIAR - Formato texto aaaamdd	No	No
FEC_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
HOR_REGISTRO_SISTEMA	CHAR(8)	NULL	FORMATO: HH:MM:SS	No	No
COD_USUARIO_SISTEMA	VARCHAR2(20)	NULL	NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TERMINAL_SISTEMA	VARCHAR2(80)	NULL	CODIGO TERMINAL SISTEMA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FLG_VIGENCIA_VINCULO	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
COD_USUARIO_AFILIA	VARCHAR2(20)	NULL	NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TERMINAL_AFILIA	VARCHAR2(80)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FEC_REGISTRO_AFILIA	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
HOR_REGISTRO_AFILIA	CHAR(8)	NULL	FORMATO: HH:MM:SS	No	No
COD_TBAJA_VINCULO	INTEGER	NULL	CODIGO TABLA BAJA VINCULO NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EBAJA_VINCULO	CHAR(2)	NULL	CODIGO ELEMENTO BAJA VINCULO TEXTO LIBRE DE CARACTERES ALFANUMERICO	No	No
IDE_NUMERICO_INSCRIP	NUMBER	NULL	IDENTIFICADOR NUMERO INSCRIPCION NUMERICO ENTERO DIFERENTE DE CERO	No	No

Fuente: Elaboración propia.

Tabla 29. csatinscri: Inscripción

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
IDE_NUMERICO_INSCRIP	NUMBER	NOT NULL	CORRELATIVO TABLA DE INSCRIPCIONES NUMERICO ENTERO DIFERENTE DE CERO	Yes	No
IDE_NUMERICO_PERSONA	NUMBER	NULL	CORRELATIVO DE TABLA PERSONA NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
IDE_NUMERICO_ENTIDAD	NUMBER	NULL	CORRELATIVO TABLA ENTIDAD NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
COD_TMODALID_COBERTUR	NUMBER	NOT NULL	COMPENDIO TABLA :301 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EMODALID_COBERTUR	CHAR(2)	NOT NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TCLASIFIC_COBERTUR	NUMBER	NOT NULL	CODIGO TABLA CLASIFICADOR DE COBERTURA - TABLA COMPENDIO 302 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ECLASIFIC_COBERTUR	CHAR(4)	NOT NULL	CODIGO TABLA CLASIFICADOR DE COBERTURA - TABLA COMPENDIO 302 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TCONDICIO_PERSONA	NUMBER	NULL	COMPENDIO:102 Y 325 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ECONDICIO_PERSONA	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TTIPO_OPERACIO	NUMBER	NOT NULL	COMPENDIO:316 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ETIPO_OPERACIO	VARCHAR2(6)	NOT NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
IDE_NUMERICO_TITULAR	NUMBER	NOT NULL	CORRELATIVO DE TABLA PERSONA NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
FLG_INDIVIDU_COLECTIV	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
NUM_ESTABLEC_ANEXO	NUMBER	NULL	IDENTIFICADOR DEL ESTABLECIMIENTO ANEXO AL QUE PERTENECE QUIEN REALIZA EL REGISTRO DE LA AFILIACIÓN (REGISTRADOR). ES UN IDENTIFICADOR QUE VIENE COMO LLAVE FORÁNEA DESDE LA TABLA CSARDIRENT NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_TTIPO_CARGO	NUMBER	NULL	COMPENDIO: 1009 Y 108 ALFANUMERICO DIFERENTE DE CERO	No	No
COD_ETIPO_CARGO	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TCLANAE_SUNAT	NUMBER	NULL	COMPENDIO: 310 Y 6002 NUMERO ENTERO	No	No
COD_ECLANAE_SUNAT	CHAR(5)	NULL	ACTIVIDAD ECONOMICA COMPENDIO 6002 SUNAT TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TTASA_RIESGO	NUMBER	NULL	COMPENDIO: 339 Y 508 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_TPLAN_SALUD	NUMBER	NULL	COMPENDIO :406 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EPLAN_SALUD	CHAR(5)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TMOTIVO_BAJA	NUMBER	NULL	104 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EMOTIVO_BAJA	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TFRECUEN_PAGO	NUMBER	NULL	COMPENDIO: 503 Y 1006 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EFRECUEN_PAGO	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
COD_NUMERO_INTERBAN	CHAR(20)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

NUM_PORCENTA_PARTICIP	NUMBER(8,2)	NULL	NULL NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_TCIU_PRINCIPA	NUMBER	NULL	310 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ECIU_PRINCIPA	CHAR(6)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TCIU_SECUNDAR	NUMBER	NULL	310 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ECIU_SECUNDAR	CHAR(6)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
MON_INGRESO_PROANUAL	NUMBER (14,2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
FLG_CONTINUI_COBERTUR	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
FEC_INICVIGE_INSCRIPC	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
FEC_TERMVIGE_INSCRIPC	CHAR(8)	NULL	FECHA DE TERMINO DE LA VIGENCIA	No	No
FEC_INICVIGE_CARENCIA	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
FEC_TERMVIGE_CARENCIA	CHAR(8)	NULL	FECHA FIN DEL PERIODO DE CARENCIA	No	No
FEC_INICVIGE_LATENCIA	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
FEC_TERMVIGE_LATENCIA	CHAR(8)	NULL	FECHA TERMINO DEL PERIODO DE LATENCIA	No	No
COD_TENTIDAD1_REGISTRA	NUMBER	NULL	COMPENDIO: 214 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EENTIDAD1_REGISTRA	CHAR(4)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TAGENTE1_REGISTRA	NUMBER	NULL	COMPENDIO: 215 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EAGENTE1_REGISTRA	CHAR(10)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TENTIDAD2_REGISTRA	NUMBER	NULL	COMPENDIO: 201 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EENTIDAD2_REGISTRA	CHAR(4)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TAGENTE2_REGISTRA	NUMBER	NULL	COMPENDIO: 213 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EAGENTE2_REGISTRA	CHAR(10)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TCANAL_REGISTRO	NUMBER	NULL	COMPENDIO: 315 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_ECANAL_REGISTRO	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TMOTIVO_INSCRIPC	NUMBER	NULL	COMPENDIO: 309 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EMOTIVO_INSCRIPC	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
COD_TDOCUMENT_REGISTRA	NUMBER	NULL	COMPENDIO: 101 NUMERICO ENTERO DIFERENTE DE CERO	No	No
COD_EDOCUMENT_REGISTRA	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_DOCUMENT_REGISTRA	VARCHAR2(15)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
NUM_CORRELAT_REGISTRO	NUMBER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
NUM_CONTRATO_INSCRIPC	VARCHAR2(12)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TEL_PREFIJO_PAIS1	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TEL_PREFIJO_CIUADAD1	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TEL_NUMERO_FIJO1	CHAR(10)	NULL	NUMERO ENTERO, CODIFICACION LOCAL TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TEL_PREFIJO_PAIS2	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
TEL_PREFIJO_CIUADAD2	CHAR(3)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

<b>TEL_NUMERO_MOVI2</b>	CHAR(10)	NULL	NUMERO ENTERO, CODIFICACION LOCAL TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_DIRPORTA_PERSONA</b>	VARCHAR2(50)	NULL	DIRECCION DE PORTAL WEB DE LA PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>TXT_DIRECCIO_ELECTRON</b>	VARCHAR2(50)	NULL	DIRECCION EKLECTRONICA (CORREO) DE LA PERSONA TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>IDE_NUMERICO_ANTERIOR</b>	NUMBER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>IDE_NUMERICO_INICIAL</b>	NUMBER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>FEC_TRANSF_DATOS</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>COD_TTIPO_PROCEDIM</b>	NUMBER	NULL	320 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ETIPO_PROCEDIM</b>	CHAR(2)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FLG_VIGENCIA_INSCRIPC</b>	CHAR(1)	NULL	INDICA LA VIGENCIA DEL REGISTRO DE INSCRIPCION SE CONSIDERA: 0 : NO ACTIVO 1: ACTIVO TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>FLG_REGISTRO_PROVISIO</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>FEC_REGISTRO_SISTEMA</b>	CHAR(8)	NOT NULL	FECHA EN LA QUE SE REALIZA EL REGISTRO DE LA INSCRIPCION; SEA PARA DAR DE ALTA MODIFICAR O DAR DE BAJA UNA INSCRIPCION. FORMATO: AAAAMMDD.	No	No
<b>HOR_REGISTRO_SISTEMA</b>	CHAR(8)	NOT NULL	HORA EN LA QUE SE REALIZA EL REGISTRO DE LA INSCRIPCION; SEA PARA DAR DE ALTA MODIFICAR O DAR DE BAJA UNA INSCRIPCION. FORMATO HH:MM:SS.	No	No
<b>COD_USUARIO_SISTEMA</b>	VARCHAR2(20)	NOT NULL	IDENTIFICADOR DEL USUARIO QUE REGISTRA UNA INSCRIPCION; SEA PARA DAR DE ALTA MODIFICAR O DAR DE BAJA UNA INSCRIPCION. NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TERMINAL_SISTEMA</b>	VARCHAR2(80)	NOT NULL	IDENTIFICADOR DEL TERMINAL DESDE EL CUAL EL USUARIO REGISTRA UNA INSCRIPCION; SEA PARA DAR DE ALTA MODIFICAR O DAR DE BAJA UNA INSCRIPCION. TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FLG_INDICADO_EPS</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_TRUC_EPS</b>	NUMBER	NULL	COMPENDIO: 223 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ERUC_EPS</b>	CHAR(11)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_ETASA_RIESGO</b>	CHAR(4)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TRUC_EPS_RIESGO</b>	NUMBER	NULL	COMPENDIO: 223 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ERUC_EPS_RIESGO</b>	CHAR(11)	NULL	TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>NUM_CONTPRIN_INSCRIPC</b>	VARCHAR2(12)	NULL	NÚMERO DE CONTRATO PRINCIPAL. REGISTRA EL NÚMERO DE CONTRATO DE UN SEGURO PRINCIPAL AL CUAL SE LE COMPLEMENTA CON OTRO SEGURO (SEGURO COMPLEMENTARIO) POR EJEMPLO: SEGUROS PRINCIPALES: REGULAR ESSALUD INDEPENDIENTE SEGUROS COMPLEMENTARIOS: ESSALUD VIDA SCTR. PARA EL CASO DE SEGUROS REGULARES NO SE GENERAN	No	No

			CONTRATOS; SIN EMBARGO SE MANEJARÁ UN NÚMERO SECUENCIAL DE USO INTERNO QUE HARÁ LAS VECES DE NÚMERO DE CONTRATO PARA LOS SEGUROS REGULARES. TEXTO LIBRE DE CARACTERES ALFANUMERICOS		
<b>IDE_DEPENDEN_SISTEMA</b>	NUMBER(10)	NOT NULL	IDENTIFICADOR DE LA DEPENDENCIA DEL USUARIO QUE REGISTRA UNA INSCRIPCIÓN; SEA PARA DAR DE ALTA MODIFICAR O DAR DE BAJA UNA INSCRIPCIÓN. NUMERICO ENTERO DIFERENTE DE CERO DE DIEZ DIGITOS	No	No
<b>NUM_CONTRATO_COLECTIV</b>	VARCHAR2(12)	NULL	NUMERO DEL CONTRATO COLECTIVO DE SEGUROS TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_USUARIO_AFILIA</b>	VARCHAR2(20)	NOT NULL	IDENTIFICADOR DEL USUARIO QUE REGISTRA UNA INSCRIPCIÓN CON TIPO DE OPERACIÓN "ALTA". EL VALOR DE ESTE CAMPO NO DEBE CAMBIAR A LO LARGO DEL TIEMPO AUNQUE EL REGISTRO PASE A LA TABLA HISTÓRICA DE INSCRIPCIONES. NUMERICO IDENTIFICADOR DEL USUARIO. NUMERO ENTERO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>IDE_DEPENDEN_AFILIA</b>	NUMBER(10)	NOT NULL	IDENTIFICADOR DE LA DEPENDENCIA DEL USUARIO QUE REGISTRA UNA INSCRIPCIÓN CON TIPO DE OPERACIÓN "ALTA". EL VALOR DE ESTE CAMPO NO DEBE CAMBIAR A LO LARGO DEL TIEMPO AUNQUE EL REGISTRO PASE A LA TABLA HISTÓRICA DE INSCRIPCIONES. NUMERICO ENTERO DIFERENTE DE CERO DE DIEZ DIGITOS	No	No
<b>COD_TERMINAL_AFILIA</b>	VARCHAR2(80)	NOT NULL	IDENTIFICADOR DEL TERMINAL DESDE EL CUAL EL USUARIO REGISTRA UNA INSCRIPCIÓN CON TIPO DE OPERACIÓN"ALTA". EL VALOR DE ESTE CAMPO NO DEBE CAMBIAR A LO LARGO DEL TIEMPO AUNQUE EL REGISTRO PASE A LA TABLA HISTÓRICA DE INSCRIPCIONES. TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FEC_REGISTRO_AFILIA</b>	CHAR(8)	NOT NULL	FECHA EN LA QUE SE REALIZA EL REGISTRO DE LA INSCRIPCIÓN CON TIPO DE OPERACIÓN "ALTA". EL VALOR DE ESTE CAMPO NO DEBE CAMBIAR A LO LARGO DEL TIEMPO AUNQUE EL REGISTRO PASE A LA TABLA HISTÓRICA DE INSCRIPCIONES. FORMATO: AAAAMMDD.	No	No
<b>HOR_REGISTRO_AFILIA</b>	CHAR(8)	NOT NULL	HORA EN LA QUE SE REALIZA EL REGISTRO DE LA INSCRIPCIÓN CON TIPO DE OPERACIÓN "ALTA". EL VALOR DE ESTE CAMPO NO DEBE CAMBIAR A LO LARGO DEL TIEMPO AUNQUE EL REGISTRO PASE A LA TABLA HISTÓRICA DE INSCRIPCIONES. FORMATO HH:MM:SS.	No	No
<b>IDE_NUMERICO_DIRENT</b>	NUMBER	NULL	IDENTIFICADOR DEL ESTABLECIMIENTO ANEXO DEL ASEGURADO (PARA SEGURO REGULAR O PARA GRUPO EN EL SEGURO INDEPENDIENTE-AFILIACIÓN COLECTIVA). ES UN IDENTIFICADOR QUE VIENE COMO LLAVE FORÁNEA DESDE LA TABLA CSARDIRENT NUMERICO ENTERO DIFERENTE DE CERO	No	Yes

<b>ID_DEPENDEN_PERSONA</b>	NUMBER	NOT NULL	IDENTIFICADOR DE LA DEPENDENCIA A LA QUE PERTENECE LA PERSONA (ASEGURADO). ES EL IDENTIFICADOR DE DEPENDENCIA EQUIVALENTE AL IDENTIFICADOR DE ESTABLECIMIENTO ANEXO (CSATINSCRI.NUM_ESTABLEC_ANEXO) NOTA: LOS REGISTROS DE CSARDIRENT MANTIENEN UNA EQUIVALENCIA (UNO A UNO) CON LOS REGISTROS DE TATD_DEPENDENCIAS PARA EL CASO DE ENTIDADES EMPLEADORAS DIFERENTES A ESSALUD. NUMERICO ENTERO DIFERENTE DE CERO DE DIEZ DIGITOS	No	No
<b>ID_OPCION_REGISTRA</b>	NUMBER	NOT NULL	IDENTIFICADOR DE LA OPCIÓN (MÓDULO) DEL SISTEMA DESDE LA QUE SE REALIZA EL REGISTRO CON TIPO DE OPERACIÓN: ALTA BAJA O MODIFICACIÓN. (POR EJEMPLO: CÓDIGO DE MÓDULO DEL FUR-P DEL FUR-NOMINA DE LA AFILIACIÓN WEB SIMPLIFICADA ETC). NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>IDE_DIRPER_ADSCRIP</b>	NUMBER	NULL	IDENTIFICADOR DE LA DIRECCIÓN DE ADSCRIPCIÓN (DIRECCIÓN PRINCIPAL) DEL ASEGURADO. VIENE DE LA TABLA CSARDIRPER. NUMERICO ENTERO DIFERENTE DE CERO	No	Yes
<b>TXT_OBSERVACION</b>	VARCHAR2(200)	NULL	OBSERVACION EN EL PROCESO DE INSCRIPCION O AFILIACION O MODIFICACIONES TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>NUM_CONTRATO_PRECEDE</b>	VARCHAR2(12)	NULL	NUMERO DE CONTRATO PRECEDENTE MIGRADO TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TCLASIFIC_PRECEDE</b>	INTEGER	NULL	CLASIFICADOR DE COBERTURA PRECEDENTE - TABLA COMPENDIO 302 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ECLASIFIC_PRECEDE</b>	CHAR(4)	NULL	CORRELATIVO ELEMENTO CLASIFICADOR DE COBERTURA PRECEDENTE - TABLA COMPENDIO 302 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>FEC_ULTIMA_ACREDITA</b>	CHAR(8)	NULL	FECHA DE LA ULTIMA ACREDITACION	No	No
<b>HOR_ULTIMA_ACREDITA</b>	CHAR(8)	NULL	FORMATO: HH:MM:SS	No	No
<b>COD_INDICA_ACREDITA</b>	CHAR(1)	NULL	INDICA EL TIPO DE ACREDITACION TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_OBSERVACION</b>	CHAR(2)	NULL	INDICA EL TIPO DE OBSERVACION DEL REGISTRO (01: SIN OBSERVACION 02: VALIDACION CON RENIEC DE DATOS DEL PADRE 03: PRE-REGISTRO POR SUNAT (SIN DOCUMENTO DE SUSTENTO) TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TCIO</b>	NUMBER	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ECIO</b>	CHAR(6)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_TDETER_INCPER</b>	NUMBER	NULL	TIPO DE ORIGEN DE INCAPACIDAD PERMANENTE - TABLA COMPENDIO 427 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_EDETER_INCPER</b>	CHAR(2)	NULL	CODIGO ELEMENTO DE TIPO DE ORIGEN DE INCAPACIDAD PERMANENTE SEGUN TABLA COMPENDIO 427 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No

<b>COD_TCIUO</b>	NUMBER	NULL	CUADRO OCUPACIONALES COMPENDIO 338 NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>COD_ECIUO</b>	CHAR(6)	NULL	CUADRO OCUPACIONALES COMPENDIO 338 TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>NUM_VALOR_TRIESGO</b>	NUMBER(5,2)	NULL	NUMERO VALOR TABLA RIESGO	No	No
<b>NUM_VALOR_TCONTRATO</b>	NUMBER(5,2)	NULL	NUMERO VALOR TABLA CONTRATO	No	No
<b>NUM_VALOR_TCIUPRINCIP</b>	NUMBER(5,2)	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>NUM_VALOR_TDECLACANC</b>	NUMBER(5,2)	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>NUM_VALOR_TCALCESSAL</b>	NUMBER(5,2)	NULL	NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>NUM_VALOR_TGRASINIEST</b>	NUMBER (5,2)	NULL	GRADO DE SINIESTRALIDAD	No	No
<b>FLG_PRIORIDAD_ACREDITA</b>	INTEGER	NULL	PRIORIDAD ACREDITA TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_ECONDI_PERS_PROD</b>	VARCHAR2(3)	NULL	CODIGO ELEMENTO CONDICION PERSONA PROD - TEXTO LIBRE DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TCONDI_PERS_PROD</b>	NUMBER	NULL	CODIGO TABLA CONDICION PERSONA PROD - TEXTO DE CARACTERES ALFANUMERICOS	No	No
<b>COD_TFUENTE_ACREDI</b>	NUMBER	NULL	CODIGO TABLA FUENTE ACREDITA - TEXTO DE CARACTERES ALFANUMERICOS	No	No
<b>COD_EFUENTE_ACREDI</b>	CHAR(10)	NULL	CODIGO ELEMENTO FUENTE ACREDITA - TEXTO DE CARÁCTER ALFANUMERICO	No	No
<b>COD_TMOTIVO_ACREDI</b>	NUMBER	NULL	CODIGO TABLA MOTIVO ACREDITA - TEXTO DE CARACTERES ALFANUMERICOS	No	No
<b>COD_EMOTIVO_ACREDI</b>	CHAR(12)	NULL	CODIGO ELEMENTO MOTIVO ACREDITA - TEXTO DE CARÁCTER ALFANUMERICO	No	No
<b>FEC_INICIO_ACREDI</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>FEC_FIN_ACREDI</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>PRDO_GSTCN_DESDE</b>	VARCHAR2(8)	NULL	TEXTO LIBRE DE CARACTER ALFANUMERICO	No	No
<b>PRDO_GSTCN_HASTA</b>	VARCHAR2(8)	NULL	TEXTO LIBRE DE CARACTER ALFANUMERICO	No	No
<b>CDGO_CLGIO_MDGO</b>	VARCHAR2(20)	NULL	CODIGO COLEGIO MEDICO TEXTO DE CARÁCTER ALFANUMERICO	No	No
<b>FLG_PRESTCNES_ECONMCAS</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>COD_TTIPO_ATENCION</b>	NUMBER	NULL	CODIGO TABLA TIPO ATENCION - TEXTO DE CARACTERES ALFANUMERICOS	No	No
<b>COD_ETIPO_ATENCION</b>	CHAR(2)	NULL	CODIGO ELEMENTO TIPO ATENCION - TEXTO DE CARÁCTER ALFANUMERICO	No	No
<b>COD_TTIPO_ACREDITA</b>	NUMBER	NULL	CODIGO TABLA TIPO ACREDITA - TEXTO DE CARACTERES ALFANUMERICOS	No	No
<b>COD_ETIPO_ACREDITA</b>	CHAR(3)	NULL	CODIGO ELEMENTO TIPO ACREDITA - TEXTO DE CARÁCTER ALFANUMERICO	No	No
<b>FLG_INDICADO_SP</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>NUM_MONTO_TARIFA</b>	NUMBER(14,2)	NULL	MONTO DE TARIFA NUMERICO ENTERO DIFERENTE DE CERO	No	No
<b>FEC_INICO_ESPERA</b>	CHAR(8)	NULL	FORMATO AAAAMMDD	No	No
<b>FLG_INDICADO_SMS</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No
<b>FLG_INDICADO_MAIL</b>	CHAR(1)	NULL	TEXTO DE UN CARACTER ALFANUMERICO	No	No

ID_TRANSACCION	CHAR(10)	NULL	CODIGO DE TRANSACCION	No	No
----------------	----------	------	-----------------------	----	----

Fuente: Elaboración propia.

Tabla 30. sccmvftf: vínculo trabajador familia

NOMBRE DEL CAMPO	TIPO DE DATO	CONDICION DE NULIDAD	DESCRIPCION DE LA COLUMNA	ES PK	ES FK
VTFNSAS	NUMBER(10)	NOT NULL		Yes	No
VTFNPMA	VARCHAR2(15)	NOT NULL		Yes	No
VTFNSAF	NUMBER(10)	NOT NULL		No	No
VTFIDPE	VARCHAR2(1)	NULL		No	No
VTFCVFA	VARCHAR2(1)	NULL		No	No
VTFFACT	DATE	NULL		No	No

### 3.5.2 Sprint 1

Para la elaboración del Sprint 1, este tenía como propósito desarrollar los requisitos iniciales de migración de datos del asegurado estos son, por ejemplo: migrar los datos personales y de vínculo familiar del asegurado.

Se definieron para el Sprint 1 las siguientes historias de usuario del Product Backlog:

Tabla 31. Product Backlog Sprint 1

Identificador de la historia	Enunciado	Estado	Dimensión/ Esfuerzo	Iteración (Sprint)	Prioridad
HU_001	Como usuario necesito los datos personales actualizados del asegurado en ambiente de calidad	OK	10	1	Alta
HU_002	Como usuario necesito los datos de vínculo familiar actualizados del asegurado en ambiente de calidad	OK	5	1	Alta

Fuente: Elaboración propia

#### 3.5.2.1 Análisis de Requisitos

## Casos de Uso

Luego, se desarrollaron los casos de Uso para el Sprint 1, en ellos se especifica el objetivo de cada uno, el procedimiento y las tablas que interaccionan:

Tabla 32. Caso de uso Replicar Datos Personales de Asegurado

<b>T01 Replicar datos personales asegurado</b>
Objetivo: Migrar los datos personales del asegurado desde base de datos producción a base de datos de Calidad
Procedimiento Se ingresa número identificador en scripts que migran datos personales de asegurado Dar clic en botón ejecutar
Tablas CSAMPERSON ACREDITACIONES NUEVO_CUENTA_INDIVIDUAL SSAHCEIT CONTRIBUYENTES CSAMENTIDA SCCHEMOT SSCMCITT SCCWFACT SSUMASET SSUMEMPT SSUHANU SSUHPAGT

Fuente: Elaboración propia

Tabla 33. Caso de uso Replicar Datos de Vínculo Familiar de Asegurado

<b>T02 Replicar datos vínculo familiar asegurado</b>
Objetivo: Migrar los datos de vínculo familiar del asegurado desde base de datos producción a base de datos de Calidad
Procedimiento Se ingresa número identificador en scripts que migran los datos de vínculo familiar del asegurado Dar clic en botón ejecutar

Tablas SCCMVTFT CSARVINFAM CSATINSCRI
--

Fuente: Elaboración propia

### 3.5.2.2 Construcción

En la construcción de las clases y la gestión de los objetos de las clases necesarias para la migración de datos personales y de vínculo familiar del asegurado según el Sprint 1 usamos el IDE Eclipse y la programación será realizada en JAVA, se elaboró una clase por cada tabla a migrar.

#### Clases

Las clases que se crearon para la réplica de datos personales y de vínculo familiar del asegurado se detallan a continuación:

#### Clase Cargarcsamperson

```
package go.pe.essalud.replicardatos;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarcsamperson {
    public String llenarCsamperson( String DNI){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            //numero del puerto
            String numero_puerto = "X";
            //SID
            String sid = "X";
            String sidqa = "X";
```

```

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);

System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete usrcsa.csamperson where
num_document_persona ="+""+DNI+""");
System.out.println("Borrado de datos correcto en QA para el DNI"+ DNI);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("select * from usrcsa.csamperson where
num_document_persona ="+""+DNI+""");
System.out.println("Selección de datos de producción correcta para el DNI"+ DNI);
while ( rs_ruc.next() )
{
    StringBuilder sbf = new StringBuilder("");
    for (int i=0;i<43;i++) {
        if (rs_ruc.getString(i+1) == null ||i == 0 ||i == 1||i == 3||i == 13||i == 15||i == 17||i == 20||i
== 29||i == 37||i == 40){
            sbf.append( rs_ruc.getString(i+1)+",");
        }

        else
        {
            sbf.append( ""+rs_ruc.getString(i+1)+""+",");
        }
    }
}
}

```

```

        String insert = "insert into usrcsa.csamperson (IDE_NUMERICO_PERSONA,
COD_TDOCUMENT_PERSONA, COD_EDOCUMENT_PERSONA,
COD_TPAISEMI_DOCUMENT, COD_EPAISEMI_DOCUMENT, NUM_DOCUMENT_PERSONA,
COD_AUTOGENE_PERSONA, TXT_APEPATER_PERSONA, TXT_APEMATER_PERSONA,
TXT_APECASAD_PERSONA, TXT_PRINOMBR_PERSONA, TXT_SEGNOMBR_PERSONA,
TXT_NOMBRES_PERSONA, COD_TESTCIVI_PERSONA, COD_EESTCIVI_PERSONA,
COD_TUBIGEO_NACIMIEN, COD_EUBIGEO_NACIMIEN, COD_TSEXO_PERSONA,
COD_ESEXO_PERSONA, FLG_ESTADO_INHABILI, COD_TMOTIVO_INHABILI,
COD_EMOTIVO_INHABILI, NUM_CIE_RUC, FEC_INICVIGE_PERSONA,
FEC_TERMVIGE_PERSONA, FEC_NACIMI_PERSONA, FEC_FALLEC_PERSONA,
FEC_INVALIDI_PERPARCI, FEC_INVALIDI_PERTOTAL, COD_TOPERACIO_SUNAT,
COD_EOPERACIO_SUNAT, NUM_PAQUETE_SUNAT, FEC_REGISTRO_SISTEMA,
HOR_REGISTRO_SISTEMA, COD_USUARIO_SISTEMA, COD_TERMINAL_SISTEMA,
TXT_NOMBCOMP_PERSONA, COD_TDOC_MADRE, COD_EDOC_MADRE,
NUM_DOC_MADRE, COD_TDOC_PADRE, COD_EDOC_PADRE, NUM_DOC_PADRE)
values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en csamperson QA...");
    }
    System.out.println("Registro de datos de correcta en csamperson QA para el DNI"+ DNI);
    conexion.close();
    conexion2.close();
}
catch( Exception e ){
    e.printStackTrace();
}
return "ok";
}
}

```

### **Clase CargarAcreditacion**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

public class CargarAcreditacion {
public String llenarAcreditacion( String DNI){
    try
    {
        DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
        String nombre_servidor = "X.X.X.X";
        String nombre_servidor_qa = "X.X.X.X";
        String numero_puerto = "X"
        String sid = "X";
        String sidqa = "X";

        String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
        String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

        //Nombre usuario y password
        String usuario = "X";
        String password = "X";
        String usuario_qa = "X";
        String password_qa = "X";

        //Obtiene la conexion producción
        Connection conexion = DriverManager.getConnection( url, usuario, password );
        //Para realizar una consulta
        Statement sentencia_prod = conexion.createStatement();
        System.out.println("conexion exitosa");
        ResultSet resultado = sentencia_prod.executeQuery("select * from
SIA.acreditaciones@LNK_SIA_SAS.ESSALUD c where NUM_DOCIDE_ASEG="+DNI);
        System.out.println("Selección de datos de producción correcta para el DNI"+ DNI);
        //Obtiene la conexion a QA
        Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa );
        System.out.println("conexion QA exitosa");
        Statement sentencia_qa = conexion2.createStatement();

        sentencia_qa.executeQuery("delete from usrcsa.acreditaciones where
NUM_DOCIDE_ASEG="+ DNI);
        System.out.println("Borrado en QA exitoso");
        while ( resultado.next() )
        {
            StringBuilder sbf = new StringBuilder("");

```

```

for (int i=0;i<16;i++) {
    if (resultado.getString(i+1) == null || i == 10 || i == 15){
        sbf.append( resultado.getString(i+1)+",");
    }

    else if(i == 7 || i == 8|| i == 13 ) {

        sbf.append("TO_DATE("'+resultado.getString(i+1)+'","YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( ""'+resultado.getString(i+1)+"''+",");
    }
}

String insert = "insert into usrcsa.acreditaciones (TIP_DOCIDE_EMPL,
NUM_DOCIDE_EMPL, TIP_DOCIDE_ASEG, NUM_DOCIDE_ASEG, COD_TRIBUTO,
TIP_ACREDITACION, TIP_ATENCION, FEC_INVIG_ACRE, FEC_FINVIG_ACRE,
IND_FACTURAC, CANT_EMPL, NUM_CONTRATO, USUARIO, FEC_ACT, TIP_OPERACION,
NUM_CORNOV) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement sentencia_qa_insert = conexion2.prepareStatement(insert);
sentencia_qa_insert.executeUpdate();
System.out.println("Registrando datos en acreditaciones QA...");
}
System.out.println("Registro de datos correcto en acreditaciones QA para el DNI"+ DNI);
conexion.close();
conexion2.close();
}catch( Exception e ){
    e.printStackTrace();
}
return "ok";
}
}

```

## Clase Cargarnuevocuentaindividual

```

package gob.pe.essalud;
import java.sql.Connection;

```

```

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarnuevocuentaindividual {
    public String llenarnuevocuentaindividual( String DNI){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete from USRCSA.NUEVO_CUENTA_INDIVIDUAL c
WHERE NUM_DOCIDE_ASEG =" + DNI);
            System.out.println("Borrado de datos correcto en QA para el DNI"+ DNI);
            Statement sentencia_prod = conexion.createStatement();
            ResultSet rs_ruc = sentencia_prod.executeQuery("select * from (select c.*

```

```

from SIA.NUEVO_CUENTA_INDIVIDUAL@LNK_SIA_SAS.ESSALUD c WHERE
NUM_DOCIDE_ASEG =" +DNI+"order by per_aporta desc)where rownum <=50");
    System.out.println("Selección de datos de producción correcta para el DNI"+ DNI);
    while ( rs_ruc.next() )
    {
        StringBuilder sbf = new StringBuilder("");
        for (int i=0;i<23;i++) {
            if (rs_ruc.getString(i+1) == null || i == 6 || i == 8 || i == 9 || i == 14 || i == 20) {
                sbf.append( rs_ruc.getString(i+1)+",");
            }

            else if(i == 10 || i == 15 ||i == 18 || i == 21) {

                sbf.append("TO_DATE(" +rs_ruc.getString(i+1)+",'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
            }

            /*else if (i == 15){

                sbf.append("TO_DATE(" +rs_ruc.getString(i+1)+",'dd-mm-yy
HH24:MI:SS'),").toString().substring(0,29);
            }*/

            else
            {
                sbf.append( ""+rs_ruc.getString(i+1)+""+",");
            }

        }

        System.out.println(sbf.toString().substring(0, sbf.toString().length()-1));
        String insert = "insert into USRCSA.NUEVO_CUENTA_INDIVIDUAL
(TIP_DOCIDE_EMPL, NUM_DOCIDE_EMPL, TIP_DOCIDE_ASEG, NUM_DOCIDE_ASEG,
PER_APORTA, COD_TRIBUTO, MTO_APORTA, COD_FORMUL, NUM_ORDEN,
MTO_BASE_IMP, FEC_INSCRIP_ASEG, IND_SITUACION, RUC_EPS, TIP_TRABAJADOR,
CTD_DIAS_LABOR, FEC_CESE, NUM_PAQ_ACTUALIZA, USUARIO, FEC_ACT,
TIP_OPERACION, NUM_CORNOV, FCARGA, IND_OPORTUNIDAD) values(" +
sbf.toString().substring(0, sbf.toString().length()-1)+)";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en NUEVO_CUENTA_INDIVIDUAL QA...");
    }

```

```

        System.out.println("Registro de datos de correcto en NUEVO_CUENTA_INDIVIDUAL QA
para el DNI"+ DNI);
        conexion.close();
        conexion2.close();
    }

    catch( Exception e ){
    e.printStackTrace();
    }

    return "ok";
    }
}

```

### **Clase CargarSsahceit**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class CargarSsahceit {
    public String LlenarSsahceit( String DNI){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";

```

```

String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);

System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete usrcsa.ssahceit c where CEINDOC =" + DNI);
System.out.println("Borrado de datos correcto en QA para el DNI" + DNI);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.ssahceit@LNK_SIA_SAS.ESSALUD c where CEINDOC =" + DNI);
System.out.println("Selección de datos de producción correcta para el DNI" + DNI);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<41;i++) {
if (rs_ruc.getString(i+1) == null || i == 3 ){
sbf.append( rs_ruc.getString(i+1)+",");
}

else if(i == 8 || i == 9 || i == 16 || i == 17 || i == 18 || i == 20 || i == 29 || i == 37) {

sbf.append("TO_DATE('" + rs_ruc.getString(i+1) + "', 'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

/*else if (i == 15){

sbf.append("TO_DATE('" + rs_ruc.getString(i+1) + "', 'dd-mm-yy
HH24:MI:SS'),").toString().substring(0,29);
}*/

else
{

```

```

        sbf.append( ""+rs_ruc.getString(i+1)+"",");
    }
}

String insert = "insert into usrcsa.sshceit (CEITCIT, CEIPERI, CEICCAS, CEINCER,
CEICSER, CEICAUT, CEIDIAG, CEIDIA2, CEIFINI, CEIFFIN, CEICMED, CEIACTM, CEIHCLI,
CEINOMB, CEITIAT, CEICTIG, CEIFOTD, CEIFEAL, CEIFEPP, CEICUAC, CEIFACT, CEIHACT,
CEICTER, CEIFLAG, CEICONT, CEIVCMP, CEITDOC, CEINDOC, CEITANU, CEIFANU,
CEICVER, CEITFLA, CEITPRO, CEIICOP, CEIICSU, CEICASU, CEICUAH, CEIFACH, CEIHACH,
CEICTEH, CEINRUC) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en sshceit QA...");
}
System.out.println("Registro de datos de correcta en sshceit QA para el DNI"+ DNI);
conexion.close();
conexion2.close();
}
catch( Exception e ){
e.printStackTrace();
}
return "ok";
}
}

```

## Clase Cargarcontribuyentes

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarcontribuyentes {

    public String llenarcontribuyentes( String RUC){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );

```

```

String nombre_servidor = "X.X.X.X";
String nombre_servidor_qa = "X.X.X.X";
String numero_puerto = "X";
String sid = "X";
String sidqa = "X";

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete from usrcsa.CONTRIBUYENTES c where numruc
="+ RUC);
System.out.println("Borrado de datos correcto en QA para el RUC"+ RUC);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("select * from
SIA.CONTRIBUYENTES@LNK_SIA_SAS.ESSALUD c where numruc =" +RUC);

System.out.println("Selección de datos de producción correcta para el RUC"+ RUC);
while ( rs_ruc.next() )
{
StringBuffer sbf = new StringBuffer("");
for (int i=0;i<28;i++) {
if (rs_ruc.getString(i+1) == null) {
sbf.append( rs_ruc.getString(i+1)+",");
}
}
}

```

```

else if(i == 14 || i == 23 || i == 15) {

    sbf.append("TO_DATE('"+rs_ruc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( ""+rs_ruc.getString(i+1)+"",");
    }
    }

String insert = "insert into usrcsa.CONTRIBUYENTES (LLTTT,NOMBRE, NUMRUC,
NUMREG, TIPOEMP, CIU, UBIGEO, NOMVIA, NUMER1, INTER1, NOMZON, REFER1,
FLAG22, ESTADO, FECALT, FECBAJ, TIPVIA, TIPZON, DOBLE, DEPORI, OPERAC, CORNOV,
NUMPRO, FCARGA, INDICA, ARCHIVO, IND_EMPHOGAR, EXISTENCIA) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en CONTRIBUYENTES QA...");
}
System.out.println("Registro de datos correcto en CONTRIBUYENTES QA para el RUC"+
RUC);
conexion.close();
conexion2.close();
}

catch( Exception e ){
e.printStackTrace();
}

return "ok";
}
}
}

```

## Clase Cargarsamentida

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.Statement;

public class Cargarcsamentida {
    public String llenarcsamentida( String RUC){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete from usrcsa.CSAMENTIDA C where
num_document_entidad =" + RUC);
            System.out.println("Borrado de datos correcto en QA para el RUC"+ RUC);
            Statement sentencia_prod = conexion.createStatement();
            ResultSet rs_ruc = sentencia_prod.executeQuery("select * from
usrcsa.CSAMENTIDA where num_document_entidad =" +RUC);
            System.out.println("Selección de datos de producción correcta para el RUC"+ RUC);
            while ( rs_ruc.next() )

```

```

{
StringBulder sbf = new StringBulder("");
for (int i=0;i<52;i++) {
    if (rs_ruc.getString(i+1) == null) {
        sbf.append( rs_ruc.getString(i+1)+",");
    }

    else
    {
        sbf.append( ""+rs_ruc.getString(i+1)+""+",");
    }
}
}

```

```

String insert = "insert into usrcsa.CSAMENTIDA (IDE_NUMERICO_ENTIDAD,
COD_TDOCUMENT_ENTIDAD, COD_EDOCUMENT_ENTIDAD, COD_TPAISEMI_DOCUMENT,
COD_EPAISEMI_DOCUMENT, NUM_DOCUMENT_ENTIDAD, TXT_RASOCIAL_ENTIDAD,
TXT_MARCOMER_ENTIDAD, COD_TTIPO_CONTRIBU, COD_ETIPO_CONTRIBU,
COD_TRELACION_ESSALUD, COD_ERELACION_ESSALUD, COD_TESTADO_ENTIDAD,
COD_EESTADO_ENTIDAD, COD_TDEPENDEN_ORISUNAT, COD_EDEPENDEN_ORISUNAT,
FEC_INICVIGE_ENTIDAD, FEC_TERMVIGE_ENTIDAD, COD_TINDICA_NOHABIDO,
COD_EINDICA_NOHABIDO, COD_TINDICA_MULTINSC, COD_EINDICA_MULTINSC,
COD_TTIPO_PERSONER, COD_ETIPO_PERSONER, TEL_PREFIJ_PAIS1,
TEL_PREFIJ_CIUADAD1, TEL_NUMERO_FIJ01, TEL_PREFIJ_PAIS2, TEL_PREFIJ_CIUADAD2,
TEL_NUMERO_MOVI2, TEL_PREFIJ_PAIS3, TEL_PREFIJ_CIUADAD3, TEL_NUMERO_FAX3,
TXT_DIRPOR_ENTIDAD, TXT_DIRECC_ELECTRO1, TXT_DIRECC_ELECTRO2,
FEC_CONSTI_ENTIDAD, FEC_INICIO_ACTIVIDA, COD_TINDICA_COMEXTER,
COD_EINDICA_COMEXTER, NUM_PARTID_RRPP, NUM_NUMERO_PATRONAL,
NUM_TOMO_RRPP, NUM_FOLIO_RRPP, NUM_ASIENT_RRPP, COD_TOPERACIO_SUNAT,
COD_EOPERACIO_SUNAT, NUM_PAQUETE_SUNAT, FEC_REGISTRO_SISTEMA,
HOR_REGISTRO_SISTEMA, COD_USUARIO_SISTEMA, COD_TERMINAL_SISTEMA) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+")";

```

```

    PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
    qa_insert_contribuyentes.executeUpdate();
    System.out.println("Registrando datos en CSAMENTIDA QA...");
}

```

```

System.out.println("Registro de datos corrceto en CSAMENTIDA QA para el RUC"+ RUC);
conexion.close();
conexion2.close();
}

```

```

        catch( Exception e ){
e.printStackTrace();
        }
        return "ok";
    }
}

```

## Clase Cargarscchemot

```
package gob.pe.essalud;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
```

```
public class Cargarscchemot {
    public String LlenarScchemot( String RUC){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

```

```

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion BD Prod exitosa");
Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion BD QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete usrcsa.scchemot c where EMOCTDI = '06' and
EMONDID = "+ RUC);
System.out.println("Borrado de datos correcto en QA para el RUC"+ RUC);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.scchemot@LNK_SIA_SAS.ESSALUD c where EMOCTDI = '06' and EMONDID = "+RUC);

System.out.println("Selección de datos de producción correcta para el RUC"+ RUC);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<13;i++) {
if (rs_ruc.getString(i+1) == null ||i == 7){
sbf.append( rs_ruc.getString(i+1)+"",");
}

else if(i == 5 || i == 8) {

sbf.append("TO_DATE(""+rs_ruc.getString(i+1)"+",'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

else
{
sbf.append( ""+rs_ruc.getString(i+1)+"""+",");
}

}

String insert = "insert into usrcsa.scchemot (EMOCTDI, EMONDID, EMOPACR,
EMOCTRI, EMOIREM, EMOFACT, EMOCOPE, EMOOSEC, EMOFCAR, EMOTIVO, EMOIVAL,
EMOIFLAG, ARCHIVO) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+"");
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();

```

```

        System.out.println("Registrando en scchemot QA...");
    }
    System.out.println("Registro de datos correcto en scchemot QA para el RUC"+ RUC);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
    e.printStackTrace();
    }
        return "ok";
    }
}

```

### **Clase Cargarsscmcitt**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarsscmcitt {
    public String LlenarSscmcitt( String AUT){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";

```

```

String usuario_qa = "X";
String password_qa = "X";
//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete USRCSA.SSCMCITT c WHERE CITCAUT =" +
AUT);
System.out.println("Borrado de datos correcto en QA para el Autogenerado" + AUT);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.SSCMCITT@LNK_SIA_SAS.ESSALUD c WHERE CITCAUT =" +AUT);

System.out.println("Selección de datos de producción correcta para el Autogenerado" +
AUT);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<23;i++) {
if (rs_ruc.getString(i+1) == null ||i == 1 ||i == 15){
sbf.append( rs_ruc.getString(i+1)+",");
}

else if(i == 5 ||i == 6 || i == 7|| i == 8|| i == 13||i == 19) {

sbf.append("TO_DATE(" +rs_ruc.getString(i+1)+", 'YYYY-MM-DD
HH24:MI:SS'),").substring(0,29);
}

else
{
sbf.append( "" +rs_ruc.getString(i+1)+""+",");
}

}
}
}

```

```

        String insert = "insert into USRCSA.SSCMCITT (CITCCAS, CITNCER, CITCSER,
CITCAUT, CITCDIA, CITFIIN, CITFFIN, CITFECE, CITFALT, CITCMED, CITCTIP, CITCTAT,
CITCTSU, CITFPPA, CITNEXP, CITNSEC, CITCZDE, CITICIT, CITCUAC, CITFACT, CITHACT,
CITCTER, CITFLAG) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en SSCMCITT QA...");
    }
    System.out.println("Registro de datos de correcta en SSCMCITT QA para el
Autogenerado"+ AUT);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
    e.printStackTrace();
}
        return "ok";
    }
}

```

### **Clase Cargarsccwtact**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarsccwtact {
    public String LlenarSccwtact( String AUT){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

```

```

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);

System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete usrcsa.sccwtact c where taccaut =" + AUT);
System.out.println("Borrado de datos correcto en QA para el Autogenerado"+ AUT);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.sccwtact@LNK_SIA_SAS.ESSALUD c where taccaut =" +AUT);
System.out.println("Selección de datos de producción correcta para el Autogenerado"+
AUT);
while ( rs_ruc.next() )
{
StringBuffer sbf = new StringBuffer("");
for (int i=0;i<22;i++) {
if (rs_ruc.getString(i+1) == null ||i == 4 ||i == 5){
sbf.append( rs_ruc.getString(i+1)+",");
}

else if(i == 1 ||i == 3 || i == 8) {

sbf.append("TO_DATE('" +rs_ruc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").substring(0,29);
}
}
}

```

```

        else
        {
            sbf.append( ""+rs_ruc.getString(i+1)+"",");
        }
    }

    String insert = "insert into usrcsa.sccwtact (TACCAUT, TACFVDE, TACCECA,
TACFVHA, TACNSAT, TACNSAF, TACCATI, TACCVFA, TACFACT, TACCUGD, TACCCAS,
TACCTAT, TACCTAS, TACIFAC, TACCEMP, TACREPS, TACNCON, TACCPLA, TACHACT,
TACCUAC, TACCTER, TACCREG) values("+ sbf.toString().substring(0, sbf.toString().length()-
1)+)";
    PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
    qa_insert_contribuyentes.executeUpdate();
    System.out.println("Registrando datos en sccwtact QA...");
}
System.out.println("Registro de datos correcto en sccwtact QA para el Autogenerado"+
AUT);
conexion.close();
conexion2.close();
}

    catch( Exception e ){
    e.printStackTrace();
    }

    return "ok";
}
}

```

### **Clase Cargarsumaset**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

public class Cargarssumaset {
    public String LlenarSsumaset( String AUT){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete USRCSA.SSUMASET c WHERE ASECAUT =" +
AUT);
            System.out.println("Borrado de datos correcto en QA para el Autogenerado"+ AUT);
                Statement sentencia_prod = conexion.createStatement();
                ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.SSUMASET@LNK_SIA_SAS.ESSALUD c WHERE ASECAUT =" +AUT);

            System.out.println("Selección de datos de producción correcta para el Autogenerado"+
AUT);
            while ( rs_ruc.next() )

```

```

{
StringBulder sbf = new StringBulder("");
for (int i=0;i<45;i++) {
    if (rs_ruc.getString(i+1) == null ||i == 1 ||i == 19||i == 20||i == 21||i == 22){
        sbf.append( rs_ruc.getString(i+1)+",");
    }

    else if(i == 13 ||i == 14 || i == 15|| i == 16|| i == 17||i == 36||i == 41 ||i == 42 ||i ==
44) {

        sbf.append("TO_DATE("'+rs_ruc.getString(i+1)+'",'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( ""'+rs_ruc.getString(i+1)+'""+',");
    }
}

String insert = "insert into USRCSA.SSUMASET (ASECEXP, ASENSEX, ASECAUT,
ASECTAS, ASETAPA, ASETAMA, ASETPNO, ASETSNO, ASECLEY, ASECTSU, ASECBAN,
ASECESO, ASECMOT, ASEFEST, ASEFISI, ASEFPRE, ASEFPRO, ASEFATR, ASEITRE,
ASEMRHA, ASEMSDI, ASEQDSU, ASEMTSU, ASENDOC, ASETDOC, ASECLEL, ASENTEL,
ASETDBE, ASENDBE, ASETBEN, ASECAMA, ASECANI, ASEREXP, ASECACR, ASECAGE,
ASECUAC, ASEFACT, ASEHACT, ASECTER, ASEFECE, ASECNBA, ASEFICON, ASEFFCON,
ASETCNV, ASEFIINC) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+")";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SSUMASET QA...");
}
System.out.println("Registro de datos correcto en SSUMASET QA para el Autogenerado"+
AUT);
conexion.close();
conexion2.close();
}

catch( Exception e ){
e.printStackTrace();
}

return "ok";

```

```
}  
}
```

## Clase Cargarssumempt

```
package gob.pe.essalud;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.Statement;  
  
public class Cargarssumempt {  
    public String LlenarSsumempt( String EXP){  
        try  
        {  
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );  
            String nombre_servidor = "X.X.X.X";  
            String nombre_servidor_qa = "X.X.X.X";  
            String numero_puerto = "X";  
            String sid = "X";  
            String sidqa = "X";  
  
            String url = "jdbc:oracle:thin:@" + nombre_servidor + "." + numero_puerto + "." + sid;  
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + "." + numero_puerto + "." +  
sidqa;  
  
            //Nombre usuario y password  
            String usuario = "X";  
            String password = "X";  
            String usuario_qa = "X";  
            String password_qa = "X";  
  
            //Obtiene la conexion producci3n  
            Connection conexion = DriverManager.getConnection( url, usuario, password );  
            System.out.println("conexion exitosa");  
            //Para realiza una consulta  
  
            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa  
);  
            System.out.println("conexion QA exitosa");
```

```

Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete USRCSA.SSUMEMPT c WHERE EMPCEXP =" +
EXP);
System.out.println("Borrado de datos correcto en QA para el RUC" + EXP.substring(0,11));
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.SSUMEMPT@LNK_SIA_SAS.ESSALUD c WHERE EMPCEXP =" + EXP);

System.out.println("Selección de datos de producción correcta para el EXPEDIENTE" +
EXP);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<18;i++) {
if (rs_ruc.getString(i+1) == null ||i == 8 ||i == 9||i == 13){
sbf.append( rs_ruc.getString(i+1)+",");
}

else if(i == 5 ||i == 6 || i == 7|| i == 15) {

sbf.append("TO_DATE(" +rs_ruc.getString(i+1)+", 'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

else
{
sbf.append( ""+rs_ruc.getString(i+1)+""+",");
}

}

String insert = "insert into USRCSA.SSUMEMPT (EMPCEXP, EMPCEMP, EMPTRSO,
EMPTEMP, EMPCBAN, EMPFISI, EMPFEST, EMPFPRO, EMPMAME, EMPMREM, EMPCTRE,
EMPIEXP, EMPCMAN, EMPQSOL, EMPCUAC, EMPFACT, EMPHACT, EMPCTER) values(" +
sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SSUMEMPT QA...");
}
System.out.println("Registro de datos correcto en SSUMEMPT QA para el EXPEDIENTE" +

```

```

EXP);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
    e.printStackTrace();
        }
        return "ok";
    }
}

```

## Clase Cargarssuhanu

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssuhanu {
    public String llenarSsuhanu( String EXP){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

```

```

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete USRCSA.SSUHANU c where
NUM_EXPEDIENTE="+ EXP);
System.out.println("Borrado de datos correcto en QA para el Expediente"+ EXP);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
USRCSA.SSUHANU c where NUM_EXPEDIENTE =" +EXP);
System.out.println("Selección de datos de producción correcta para el Expediente"+ EXP);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<8;i++) {
if (rs_ruc.getString(i+1) == null ||i == 0 ||i == 1||i == 6 ||i == 7){
sbf.append( rs_ruc.getString(i+1)+",");
}

/*else if(i == 5 || i == 11|| i == 12) {

sbf.append("TO_DATE("'+rs_ruc.getString(i+1)+'",'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

else if (i == 15){

sbf.append("TO_DATE("'+rs_ruc.getString(i+1)+'','dd-mm-yy
HH24:MI:SS'),").toString().substring(0,29);
}*/

else
{
sbf.append( ""'+rs_ruc.getString(i+1)+"''+",");
}
}
}

```

```

        }
    }

    String insert = "insert into USRCSA.SSUHANU (IDE_NUM_ANU, NUM_SOLICI,
NUM_EXPEDIENTE, COD_TIPO_ANULACION, NUM_SECUENCIA, NUM_EMISION,
FLAG_REACTIVADO, IDE_DOC) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
    PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
    qa_insert_contribuyentes.executeUpdate();
    System.out.println("Registrando datos en SSUHANU QA...");
}
System.out.println("Registro de datos de correcta en QA para el Expediente"+ EXP);
conexion.close();
conexion2.close();
}
catch( Exception e ){
e.printStackTrace();
}
return "ok";
}
}

```

### **Clase Cargarssuhpagt**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssuhpagt {
    public String llenarSsuhpagt( String EXP){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

```

```

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";
//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete USRCSA.SSUHPAGT c where PAGCEXP="+
EXP);
System.out.println("Borrado de datos correcto en QA para el Expediente"+ EXP);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.SSUHPAGT@LNK_SIA_SAS.ESSALUD c where PAGCEXP="+EXP);
System.out.println("Selección de datos de producción correcta para el Expediente"+ EXP);
while ( rs_ruc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<15;i++) {
if (rs_ruc.getString(i+1) == null ||i == 2 ||i == 6){
sbf.append( rs_ruc.getString(i+1)+",");
}

else if(i == 5 || i == 11|| i == 12) {

sbf.append("TO_DATE('" +rs_ruc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}
}
}

```

```

        else
        {
            sbf.append( ""+rs_ruc.getString(i+1)+"",");
        }
    }

    String insert = "insert into USRCSA.SSUHPAGT (PAGNEMI, PAGCEXP, PAGNSCE,
PAGCBAN, PAGIEPA, PAGFPAG, PAGMREM, PAGCTPA, PAGNCER, PAGIEST, PAGCAGE,
PAGFEMI, PAGFVEN, PAGICTA, PAGBAGE) values("+ sbf.toString().substring(0,
sbf.toString().length()-1)+)";
    PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
    qa_insert_contribuyentes.executeUpdate();
    System.out.println("Registrando datos en SSUHPAGT QA...");
}
System.out.println("Registro de datos correcto en SSUHPAGT QA para el Expediente"+
EXP);
conexion.close();
conexion2.close();
}

    catch( Exception e ){
        e.printStackTrace();
    }
    return "ok";
}
}

```

### **Clase CargarSccmvftf**

```

package gob.pe.essalud;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class CargarSccmvftf {
    public String llenarSccmvftf( String SAS){
        try

```

```

{
    DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
String nombre_servidor = "X.X.X.X";
String nombre_servidor_qa = "X.X.X.X";
String numero_puerto = "X";
String sid = "X";
String sidqa = "X";

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete USRCSA.SCCMVTFT where VTFNSAS = "+
SAS);
System.out.println("Borrado de datos correcto en QA para el Autogenerado"+ SAS);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_ruc = sentencia_prod.executeQuery("Select * from
SIA.SCCMVTFT@LNK_SIA_SAS.ESSALUD where VTFNSAS =" +SAS);
System.out.println("Selección de datos de producción correcta para el SAS"+ SAS);
while ( rs_ruc.next() )
{
StringBulder sbf = new StringBulder("");
for (int i=0;i<6;i++) {
if (rs_ruc.getString(i+1) == null ||i == 0 ||i == 1){
sbf.append( rs_ruc.getString(i+1)+",");
}
}
}
}

```

```

    }

    else if(i == 5 ) {

        sbf.append("TO_DATE('"+rs_ruc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( ""+rs_ruc.getString(i+1)+" "+"");
    }
}

String insert = "insert into USRCSA.SCCMVTFT (VTFNSAS, VTFNSAF, VTFCVFA,
VTFIDPE, VTFNPMA, VTFFACT) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+"");
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SCCMVTFT QA...");
}
System.out.println("Registro de datos de correcta en SCCMVTFT QA para el SAS"+ SAS);
conexion.close();
conexion2.close();
}

catch( Exception e ){
e.printStackTrace();
}

return "ok";

}
}
}

```

### **Clase Cargarcsarvinfam**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

public class Cargarcsarvinfam {
    public String llenarCsarvinfam( String IDT){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";
            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete usrcsa.csarvinfam where ide_numerico_titular =" +
IDT);
            System.out.println("Borrado de datos correcto en QA para el DNI"+ IDT);
                Statement sentencia_prod = conexion.createStatement();
                ResultSet rs_ruc = sentencia_prod.executeQuery("select * from
usrcsa.csarvinfam where ide_numerico_titular =" +IDT);
            System.out.println("Selección de datos de producción correcta para el DNI"+ IDT);
            while ( rs_ruc.next() )
            {
                StringBuilder sbf = new StringBuilder("");
                for (int i=0;i<21;i++) {

```

```

        if (rs_ruc.getString(i+1) == null ||i == 0 ||i == 1||i == 2 ||i == 3||i == 4 ||i == 5||i == 18
||i == 20){
                sbf.append( rs_ruc.getString(i+1)+",");
        }

        else
        {
                sbf.append( ""+rs_ruc.getString(i+1)+"",");
        }

        }

        String insert = "insert into usrcsa.csarvinfam (IDE_NUMERICO_VINCFAMI,
IDE_NUMERICO_TITULAR, IDE_NUMERICO_FAMILIAR, IDE_ENTIDAD_TITULAR,
IDE_ENTIDAD_VINCULAD, COD_TVINCULO_FAMILIAR, COD_EVINCULO_FAMILIAR,
FEC_INICVIGE_VINCFAMI, FEC_TERMVIGE_VINCFAMI, FEC_REGISTRO_SISTEMA,
HOR_REGISTRO_SISTEMA, COD_USUARIO_SISTEMA, COD_TERMINAL_SISTEMA,
FLG_VIGENCIA_VINCULO, COD_USUARIO_AFILIA, COD_TERMINAL_AFILIA,
FEC_REGISTRO_AFILIA, HOR_REGISTRO_AFILIA, COD_TBAJA_VINCULO,
COD_EBAJA_VINCULO, IDE_NUMERICO_INSCRIP) values("+ sbf.toString().substring(0,
sbf.toString().length()-1)+")";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en csarvinfam QA...");
    }
    System.out.println("Registro de datos de correcta en QA para el ID Titular:"+ IDT);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
            e.printStackTrace();
        }
        return "ok";
    }
}

```

### **Clase Cargarcsatinscri**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;

```

```

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarcsatinscri {
    public String llenarCsatinscri( String IDT){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete usrcsa.csatinscri where ide_numerico_titular =" +
IDT);
            System.out.println("Borrado de datos correcto en QA para el DNI"+ IDT);
            Statement sentencia_prod = conexion.createStatement();
            ResultSet rs_ruc = sentencia_prod.executeQuery("select * from
usrcsa.csatinscri where ide_numerico_titular =" +IDT);

```

```

System.out.println("Selección de datos de producción correcta para el DNI"+ IDT);
while ( rs_ruc.next() )
{
    StringBuilder sbf = new StringBuilder("");
    for (int i=0;i<136;i++) {
        if (rs_ruc.getString(i+1) == null ||i == 0 ||i == 1||i == 2 ||i == 3||i == 5||i == 7||i == 9||i
== 11||i == 13||i == 14||i == 16||i == 18||i == 19||i == 21||i == 23||i == 26||i == 27||i == 29||i == 31||i
== 39||i == 41||i == 43||i == 45||i == 47||i == 49||i == 51||i == 54||i == 64||i == 65||i == 67||i == 76||i
== 79||i == 82||i == 85||i == 89||i == 90||i == 91||i == 92||i == 95||i == 101||i == 103||i == 105||i ==
107||i == 108||i == 109||i == 110||i == 111||i == 112||i == 113||i == 115||i == 116||i == 118||i == 126||i
== 128||i == 131){

            sbf.append( rs_ruc.getString(i+1)+",");

        }

        else
        {
            sbf.append( ""+rs_ruc.getString(i+1)+""+",");

        }

    }
}

```

```

String insert = "insert into usrcsa.csatinscri (IDE_NUMERICO_INSCRIP,
IDE_NUMERICO_PERSONA, IDE_NUMERICO_ENTIDAD, COD_TMODALID_COBERTUR,
COD_EMODALID_COBERTUR, COD_TCLASIFIC_COBERTUR, COD_ECLASIFIC_COBERTUR,
COD_TCONDICIO_PERSONA, COD_ECONDICIO_PERSONA, COD_TTIPO_OPERACIO,
COD_ETIPO_OPERACIO, IDE_NUMERICO_TITULAR, FLG_INDIVIDU_COLECTIV,
NUM_ESTABLEC_ANEXO, COD_TTIPO_CARGO, COD_ETIPO_CARGO,
COD_TCLANAE_SUNAT, COD_ECLANAE_SUNAT, COD_TTASA_RIESGO,
COD_TPLAN_SALUD, COD_EPLAN_SALUD, COD_TMOTIVO_BAJA, COD_EMOTIVO_BAJA,
COD_TFRECUEN_PAGO, COD_EFRECUEN_PAGO, COD_NUMERO_INTERBAN,
NUM_PORCENTA_PARTICIP, COD_TCIUU_PRINCIPA, COD_ECIUU_PRINCIPA,
COD_TCIUU_SECUNDAR, COD_ECIUU_SECUNDAR, MON_INGRESO_PROANUAL,
FLG_CONTINUI_COBERTUR, FEC_INICVIGE_INSCRIPC, FEC_TERMVIGE_INSCRIPC,
FEC_INICVIGE_CARENCIA, FEC_TERMVIGE_CARENCIA, FEC_INICVIGE_LATENCIA,
FEC_TERMVIGE_LATENCIA, COD_TENTIDAD1_REGISTRA, COD_EENTIDAD1_REGISTRA,
COD_TAGENTE1_REGISTRA, COD_EAGENTE1_REGISTRA, COD_TENTIDAD2_REGISTRA,
COD_EENTIDAD2_REGISTRA, COD_TAGENTE2_REGISTRA, COD_EAGENTE2_REGISTRA,
COD_TCANAL_REGISTRO, COD_ECANAL_REGISTRO, COD_TMOTIVO_INSCRIPC,
COD_EMOTIVO_INSCRIPC, COD_TDOCUMENT_REGISTRA, COD_EDOCUMENT_REGISTRA,

```

```

NUM_DOCUMENT_REGISTRA, NUM_CORRELAT_REGISTRO, NUM_CONTRATO_INSCRIPC,
TEL_PREFIJO_PAIS1, TEL_PREFIJO_CIUADAD1, TEL_NUMERO_FIJO1, TEL_PREFIJO_PAIS2,
TEL_PREFIJO_CIUADAD2, TEL_NUMERO_MOVI2, TXT_DIRPORTA_PERSONA,
TXT_DIRECCIO_ELECTRON, IDE_NUMERICO_ANTERIOR, IDE_NUMERICO_INICIAL,
FEC_TRANSF_DATOS, COD_TTIPO_PROCEDIM, COD_ETIPO_PROCEDIM,
FLG_VIGENCIA_INSCRIPC, FLG_REGISTRO_PROVISIO, FEC_REGISTRO_SISTEMA,
HOR_REGISTRO_SISTEMA, COD_USUARIO_SISTEMA, COD_TERMINAL_SISTEMA,
FLG_INDICADO_EPS, COD_TRUC_EPS, COD_ERUC_EPS, COD_ETASA_RIESGO,
COD_TRUC_EPS_RIESGO, COD_ERUC_EPS_RIESGO, NUM_CONTPRIN_INSCRIPC,
IDE_DEPENDEN_SISTEMA, NUM_CONTRATO_COLECTIV, COD_USUARIO_AFILIA,
IDE_DEPENDEN_AFILIA, COD_TERMINAL_AFILIA, FEC_REGISTRO_AFILIA,
HOR_REGISTRO_AFILIA, IDE_NUMERICO_DIRENT, ID_DEPENDEN_PERSONA,
ID_OPCION_REGISTRA, IDE_DIRPER_ADSCRIP, TXT_OBSERVACION,
NUM_CONTRATO_PRECEDE, COD_TCLASIFIC_PRECEDE, COD_ECLASIFIC_PRECEDE,
FEC_ULTIMA_ACREDITA, HOR_ULTIMA_ACREDITA, COD_INDICA_ACREDITA,
COD_OBSERVACION, COD_TCIIO, COD_ECIIO, COD_TDETER_INCPER,
COD_EDETER_INCPER, COD_TCIUO, COD_ECIUO, NUM_VALOR_TRIESGO,
NUM_VALOR_TCONTRATO, NUM_VALOR_TCIUPRINCIP, NUM_VALOR_TDECLACANC,
NUM_VALOR_TCALCESSAL, NUM_VALOR_TGRASINIEST, FLG_PRIORIDAD_ACREDITA,
COD_ECONDI_PERS_PROD, COD_TCONDI_PERS_PROD, COD_TFUENTE_ACREDI,
COD_EFUENTE_ACREDI, COD_TMOTIVO_ACREDI, COD_EMOTIVO_ACREDI,
FEC_INICIO_ACREDI, FEC_FIN_ACREDI, PRDO_GSTCN_DESDE, PRDO_GSTCN_HASTA,
CDGO_CLGIO_MDICO, FLG_PRESTCNES_ECONMCAS, COD_TTIPO_ATENCION,
COD_ETIPO_ATENCION, COD_TTIPO_ACREDITA, COD_ETIPO_ACREDITA,
FLG_INDICADO_SP, NUM_MONTO_TARIFA, FEC_INICO_ESPERA, FLG_INDICADO_SMS,
FLG_INDICADO_MAIL, ID_TRANSACCION) values("+ sbf.toString().substring(0,
sbf.toString().length()-1)+")";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en csatinscri QA...");
    }
    System.out.println("Registro de datos de correcta en QA para el ID Titular"+ IDT);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
            e.printStackTrace();
        }
        return "ok";
    }
}

```

}

### 3.5.2.3 Sprint Backlog

La lista de tareas del Sprint 1 se detalla en la tabla siguiente:

Tabla 34. Sprint01 Product Backlog

Identificador de la historia	Enunciado	Tareas	Dueño/ Voluntario	Estado
HU_001	Como usuario necesito los datos personales actualizados del asegurado en ambiente de calidad	Creación de la clase Cargarcsamperson	John Antón	Terminado
		Creación de la clase CargarAcreditacion	John Antón	Terminado
		Creación de la clase Cargarnuevocuentaindividual	John Antón	Terminado
		Creación de la clase CargarSsahceit	John Antón	Terminado
		Creación de la clase Cargarcontribuyentes	John Antón	Terminado
		Creación de la clase Cargarcsamentida	John Antón	Terminado
		Creación de la clase Cargarcschemot	John Antón	Terminado
		Creación de la clase Cargarsscmcitt	John Antón	Terminado
		Creación de la clase Cargarscctact	John Antón	Terminado
		Creación de la clase Cargarssumaset	John Antón	Terminado
		Creación de la clase Cargarssuhanu	John Antón	Terminado
		Creación de la clase Cargarssuhpagt	John Antón	Terminado
		HU_002	Como usuario necesito los datos de vínculo familiar	Creación de la clase CargarSccmvttf
Creación de la clase Cargarcsarvinfam	John Antón			Terminado
Creación de la clase	John			Terminado

	actualizados del asegurado en ambiente de calidad	Cargarcsatinscri	Antón	
--	---	------------------	-------	--

Fuente: Elaboración propia

### 3.5.2.4 Pruebas

Con el fin de poder poner a prueba nuestras clases desarrolladas a lo largo del Sprint, gracias al desarrollo completo de las historias de usuario establecidas, por cada historia de usuario hay un conjunto de variados criterios de aceptación, definiendo un límite de 4 de acuerdo a la metodología de Scrum, los cuales precisan de manera adecuada las funcionalidades de cada historia de usuario.

Para este Sprint, se mostrará el resultado final satisfactorio.

Seguidamente, se detallan los casos de prueba ejecutados durante el Sprint:

Tabla 35. Casos de prueba (HU\_001)

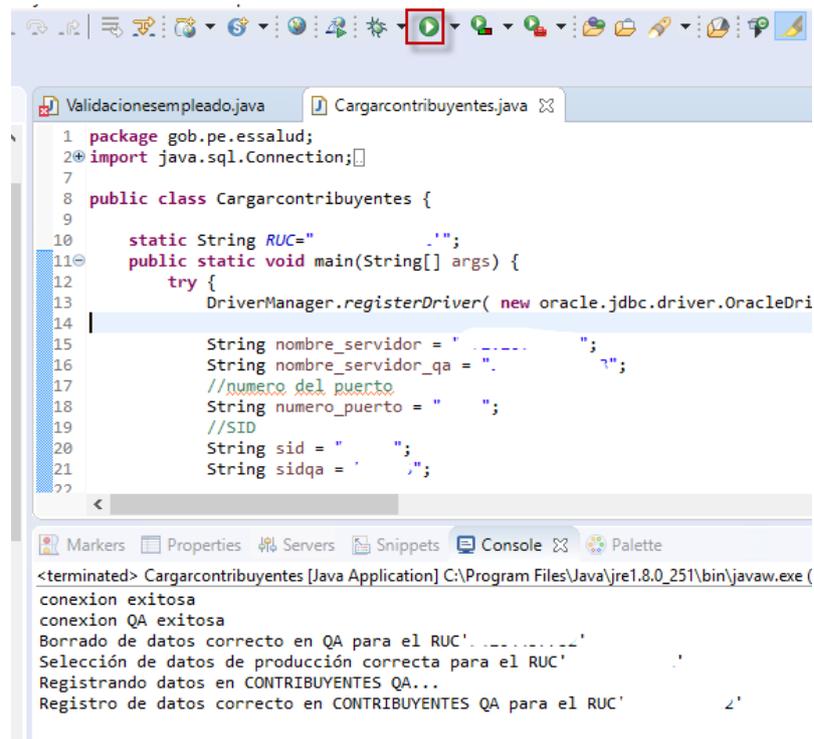
HU_001	Como usuario necesito los datos personales actualizados del asegurado en ambiente de calidad		
	Rol	Características /Funcionalidad	Razón/ Resultado
	Usuario	Necesito replicar los datos personales del asegurado	Con la finalidad de obtener los datos personales del asegurado en ambiente de calidad
Criterio de Aceptación	Contexto	Evento	Resultado/ Comportamiento esperado
Replicar datos personales del asegurado	En caso de que ingrese un número identificador en las clases que replican los datos personales del asegurado	cuando hago clic en el botón Ejecutar del IDE Eclipse	Se deberá migrar los datos personales del asegurado.

Fuente: Elaboración propia





Figura 11. Prueba de la clase Cargarcontribuyentes (HU\_001)

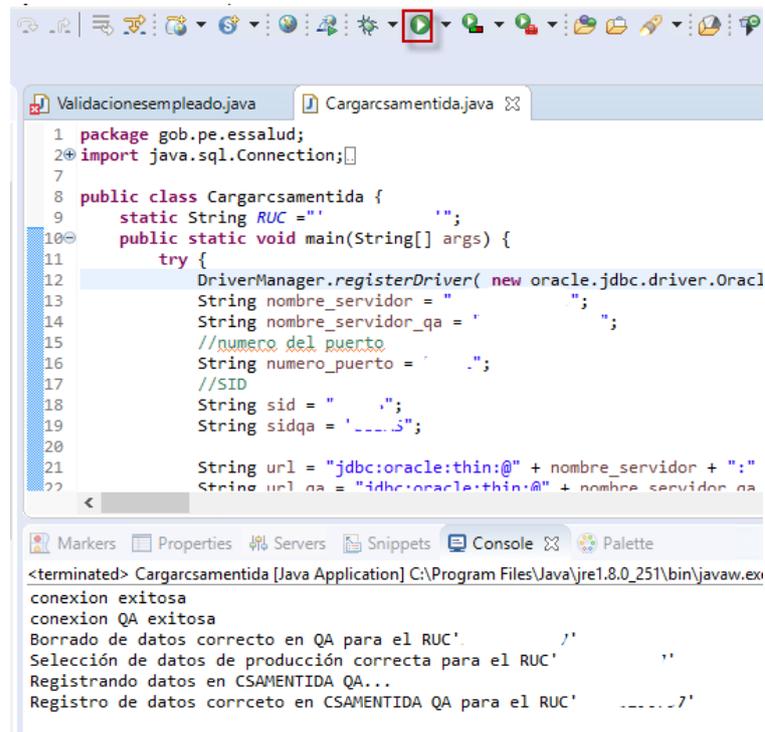


```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarcontribuyentes {
9
10     static String RUC="";
11     public static void main(String[] args) {
12         try {
13             DriverManager.registerDriver( new oracle.jdbc.driver.OracleDri
14
15             String nombre_servidor = " ";
16             String nombre_servidor_qa = " ";
17             //numero del puerto
18             String numero_puerto = " ";
19             //SID
20             String sid = " ";
21             String sidqa = " ";
22
23         }
24     }
25 }
```

<terminated> Cargarcontribuyentes [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el RUC' .....'  
Selección de datos de producción correcta para el RUC' .....'  
Registrando datos en CONTRIBUYENTES QA...  
Registro de datos correcto en CONTRIBUYENTES QA para el RUC' .....2'

Fuente: Elaboración propia

Figura 12. Prueba de la clase Cargarcsamentida (HU\_001)



```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarcsamentida {
9     static String RUC ="";
10     public static void main(String[] args) {
11         try {
12             DriverManager.registerDriver( new oracle.jdbc.driver.OracleDri
13             String nombre_servidor = " ";
14             String nombre_servidor_qa = " ";
15             //numero del puerto
16             String numero_puerto = " ";
17             //SID
18             String sid = " ";
19             String sidqa = ".....S";
20
21             String url = "jdbc:oracle:thin:@" + nombre_servidor + ":"
22             String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa
23         }
24     }
25 }
```

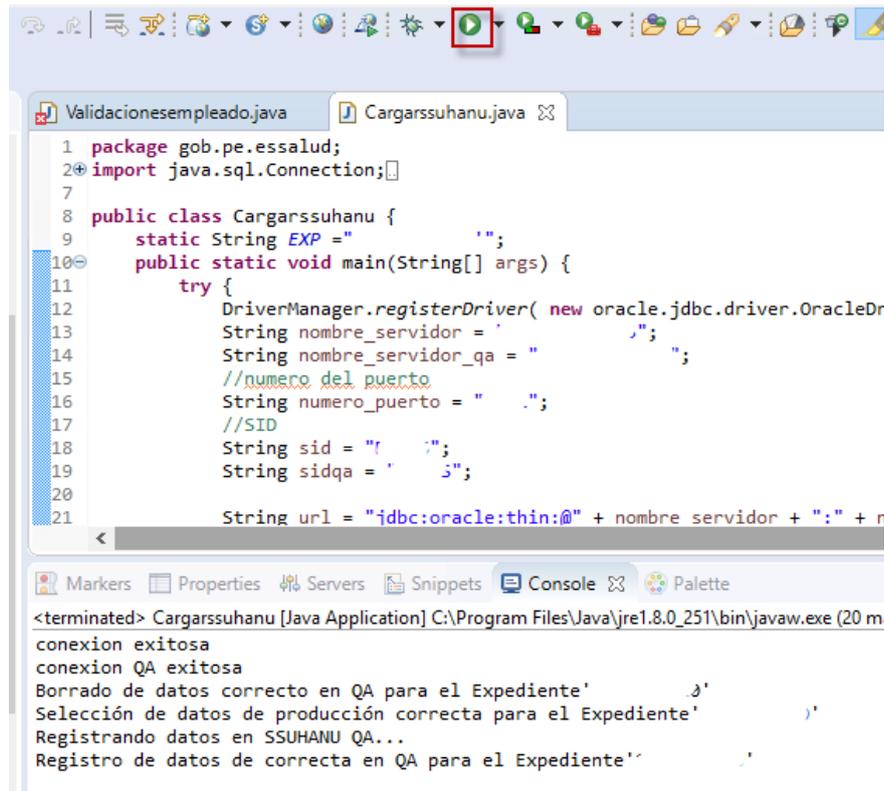
<terminated> Cargarcsamentida [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el RUC' .....'  
Selección de datos de producción correcta para el RUC' .....'  
Registrando datos en CSAMENTIDA QA...  
Registro de datos corrceto en CSAMENTIDA QA para el RUC' .....7'

Fuente: Elaboración propia





Figura 17. Prueba de la clase Cargarssuhanu (HU\_001)

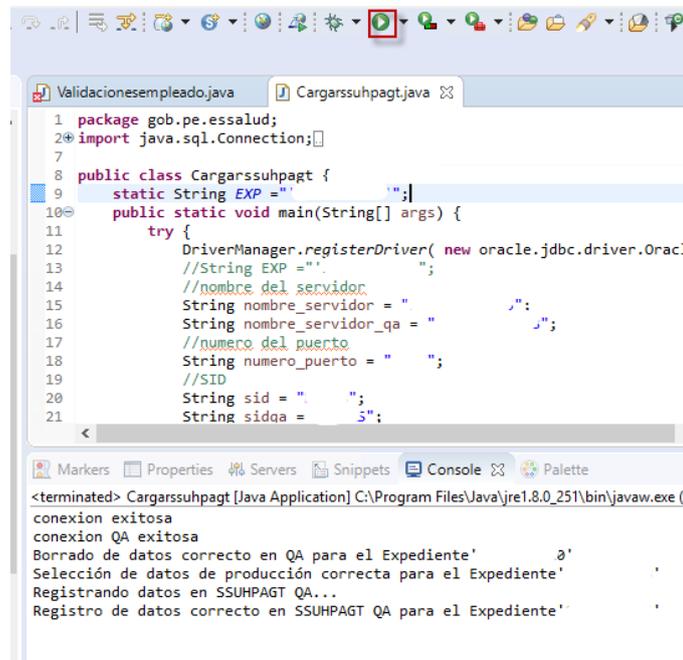


```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarssuhanu {
9     static String EXP = " ";
10 public static void main(String[] args) {
11     try {
12         DriverManager.registerDriver( new oracle.jdbc.driver.OracleDr
13         String nombre_servidor = " ";
14         String nombre_servidor_qa = " ";
15         //numero del puerto
16         String numero_puerto = " ";
17         //SID
18         String sid = " ";
19         String sidqa = " ";
20
21         String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + n
```

<terminated> Cargarssuhanu [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 m  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el Expediente'                    '  
Selección de datos de producción correcta para el Expediente'                    ')'  
Registrando datos en SSUHANU QA...  
Registro de datos de correcta en QA para el Expediente''                    '

Fuente: Elaboración propia

Figura 18. Prueba de la clase Cargarssuhpagt (HU\_001)



```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarssuhpagt {
9     static String EXP = " ";
10 public static void main(String[] args) {
11     try {
12         DriverManager.registerDriver( new oracle.jdbc.driver.Orac
13         //String EXP = " ";
14         //nombre del servidor
15         String nombre_servidor = " ";
16         String nombre_servidor_qa = " ";
17         //numero del puerto
18         String numero_puerto = " ";
19         //SID
20         String sid = " ";
21         String sidqa = " ";
```

<terminated> Cargarssuhpagt [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el Expediente'                    '  
Selección de datos de producción correcta para el Expediente'                    '  
Registrando datos en SSUHPAGT QA...  
Registro de datos correcto en SSUHPAGT QA para el Expediente''                    '

Fuente: Elaboración propia

Tabla 36. Casos de prueba (HU\_002)

HU_002	Como usuario necesito los datos de vínculo familiar actualizados del asegurado en ambiente de calidad		
	Rol	Características /Funcionalidad	Razón/ Resultado
	Usuario	Necesito replicar los datos personales y de vínculo familiar del asegurado.	Con la finalidad de obtener los datos de vínculo familiar del asegurado en ambiente de calidad
Criterio de Aceptación	Contexto	Evento	Resultado/ Comportamiento esperado
Replicar datos de vínculo familiar del asegurado	En caso de que ingrese un número de DNI en las clases que replican los datos de vínculo familiar del asegurado	cuando hago clic en el botón Ejecutar del IDE Eclipse	Se deberá migrar los datos de vínculo familiar del asegurado.

Fuente: Elaboración propia

Figura 19. Prueba de la clase CargarSccmvftf (HU\_001)

```

1 package gob.pe.essalud;
2
3 import java.sql.Connection;
4
5
6
7
8
9 public class CargarSccmvftf {
10     static String SAS = " ";
11     public static void main(String[] args) {
12         try {
13             DriverManager.registerDriver( new oracle.jdbc.driver.Ora
14             String nombre_servidor = " ";
15             String nombre_servidor_qa = " ";
16             //numero del puerto
17             String numero_puerto = " ";
18             //SID
19             String sid = " ";
20             String sidqa = " ";
21         }
22     }
23 }

```

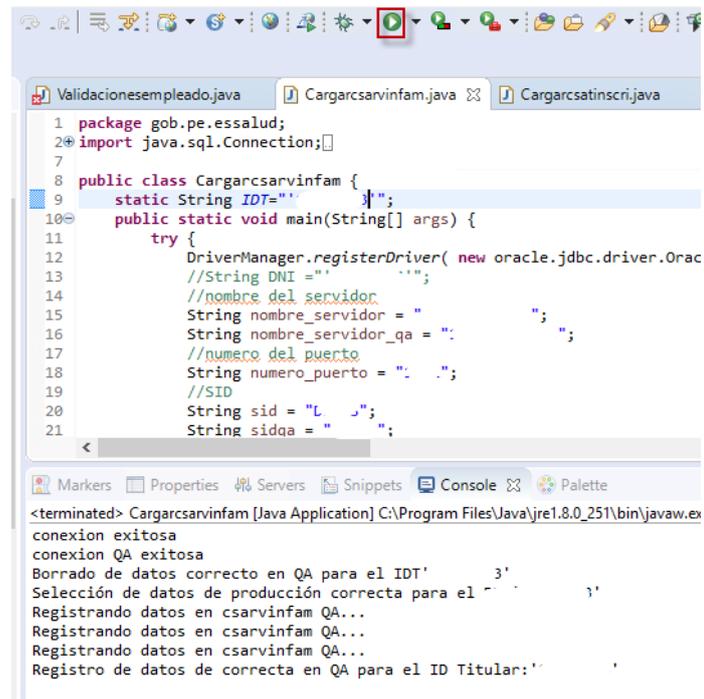
```

<terminated> CargarSccmvftf [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe
conexion exitosa
conexion QA exitosa
Borrado de datos correcto en QA para el Autogenerado'
Selección de datos de producción correcta para el SAS'
Registrando datos en SCCMVFTF QA...
Registrando datos en SCCMVFTF QA...
Registro de datos de correcta en QA para el SAS'

```

Fuente: Elaboración propia

Figura 20. Prueba de la clase Cargarcsarvinfam (HU\_001)

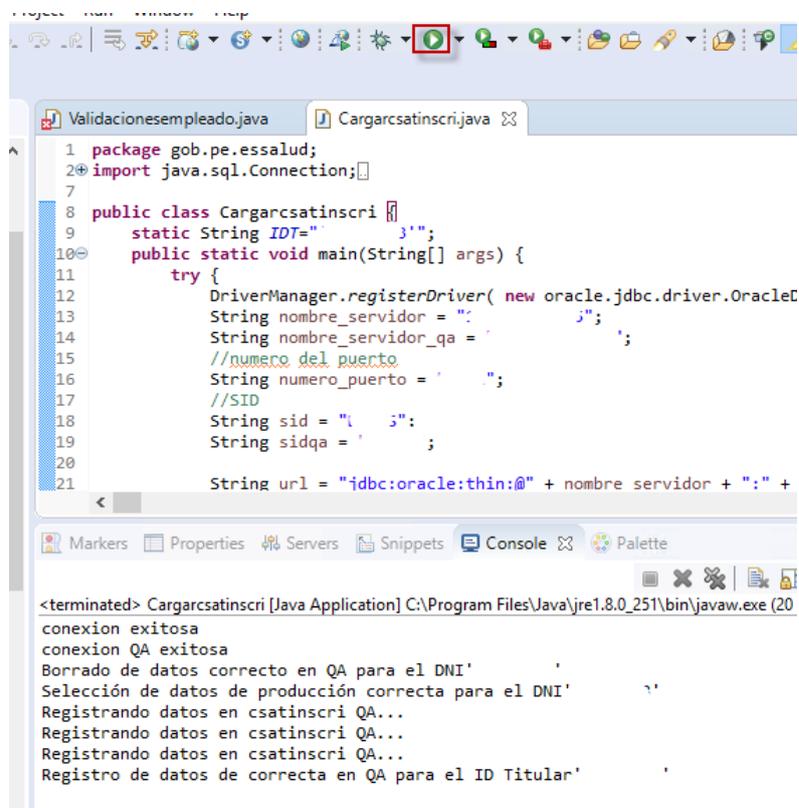


```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarcsarvinfam {
9     static String IDT="";
10 public static void main(String[] args) {
11     try {
12         DriverManager.registerDriver( new oracle.jdbc.driver.Oracle
13             //String DNI = " ";
14             //nombre del servidor
15             String nombre_servidor = " ";
16             String nombre_servidor_qa = " ";
17             //numero del puerto
18             String numero_puerto = " ";
19             //SID
20             String sid = " ";
21             String sidqa = " ";
```

<terminated> Cargarcsarvinfam [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el IDT' 3'  
Selección de datos de producción correcta para el ' ' }'  
Registrando datos en csarvinfam QA...  
Registrando datos en csarvinfam QA...  
Registrando datos en csarvinfam QA...  
Registro de datos de correcta en QA para el ID Titular:' ' }

Fuente: Elaboración propia

Figura 21. Prueba de la clase Cargarcsatinscri (HU\_001)



```
1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarcsatinscri {
9     static String IDT=" ";
10 public static void main(String[] args) {
11     try {
12         DriverManager.registerDriver( new oracle.jdbc.driver.Oracle
13             String nombre_servidor = " ";
14             String nombre_servidor_qa = " ";
15             //numero del puerto
16             String numero_puerto = " ";
17             //SID
18             String sid = " ";
19             String sidqa = " ";
20
21             String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" +
```

<terminated> Cargarcsatinscri [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para el DNI' ' '  
Selección de datos de producción correcta para el DNI' ' '  
Registrando datos en csatinscri QA...  
Registrando datos en csatinscri QA...  
Registrando datos en csatinscri QA...  
Registro de datos de correcta en QA para el ID Titular' ' ' }

Fuente: Elaboración propia

### 3.5.3 Sprint 2

Para la elaboración del Sprint 2, se tenía como propósito desarrollar los requisitos finales estos son, por ejemplo: migrar los datos de las solicitudes de subsidios del asegurado y la construcción de un aplicativo web que migre los datos del asegurado.

Se definieron para el Sprint 2 las siguientes historias de usuario del Product Backlog:

Tabla 37. Product Backlog Sprint 2

Identificador de la historia	Enunciado	Estado	Dimensión/ Esfuerzo	Iteración (Sprint)	Prioridad
HU_003	Como usuario necesito los datos de solicitudes de subsidios actualizados del asegurado en ambiente de calidad	OK	5	2	Media
HU_004	Como usuario necesito los datos de vínculo familiar actualizados del asegurado en ambiente de calidad	OK	10	2	Media

Fuente: Elaboración propia

#### 3.5.3.1 Análisis de Requisitos

##### Casos de Uso

Seguidamente, se desarrolló los casos de Uso para este Sprint, en ellos se especifica el propósito de cada uno, el procedimiento y las tablas que se relacionan:

Tabla 38. Caso de uso Replicar Solicitudes de Subsidio de Asegurado

T03 Replicar datos de solicitudes de subsidios de asegurado
Objetivo: Migrar los datos solicitudes de subsidios del asegurado desde base de datos producción a

base de datos de Calidad
<p>Procedimiento</p> <p>Se ingresa número identificador en scripts que migran datos personales de asegurado</p> <p>Dar clic en botón ejecutar</p>
<p>Tablas</p> <p>SSUTSOLICI</p> <p>SSUTDETLAC</p> <p>SSUTDETMAT</p> <p>SSUTDETINC</p> <p>SSUTDETSEP</p> <p>SSUTSOLICI_OV</p> <p>SSUTDETINC_OV</p> <p>SSUTDETMAT_OV</p> <p>SSUTDETLAC_OV</p>

Fuente: Elaboración propia

Tabla 39. Caso de uso replicar datos desde aplicativo web

T04 Replicar todos datos del asegurado
<p>Objetivo:</p> <p>Migrar datos del asegurado desde base de datos producción a base de datos de Calidad mediante aplicativo web</p>
<p>Procedimiento</p> <p>Se ingresa número de DNI en aplicativo web</p> <p>Dar clic en botón ejecutar</p>
<p>Tablas</p> <p>CSAMPERSON</p> <p>ACREDITACIONES</p> <p>NUEVO_CUENTA_INDIVIDUAL</p> <p>SSAHCEIT</p> <p>CONTRIBUYENTES</p> <p>CSAMENTIDA</p> <p>SCCHEMOT</p> <p>SSCMCITT</p> <p>SCCWTACT</p> <p>SSUMASET</p> <p>SSUMEMPT</p> <p>SSUHANU</p> <p>SSUHPAGT</p>

SCCMVTFT
CSARVINFAM
CSATINSCRI
SSUTSOLICI
SSUTDETLAC
SSUTDETMAT
SSUTDETINC
SSUTDETSEP
SSUTSOLICI_OV
SSUTDETINC_OV
SSUTDETMAT_OV
SSUTDETLAC_OV

Fuente: Elaboración propia

### 3.5.3.2 Construcción

Para la creación de las clases y el manejo de los objetos de dichas clases necesarias para la migración de datos personales y de vínculo familiar del asegurado según el Sprint 2 usamos el IDE Eclipse y la programación será realizada en JAVA se elaboró una clase por cada tabla a migrar, para la construcción de la web se reutilizará las clases antes creada en el IDE IntelliJ IDEA, se utilizó el framework spring boot y se seguirá utilizando el lenguaje de programación JAVA.

#### Clases

Las clases que se crearon para la réplica de datos de solicitudes de subsidios del asegurado y el aplicativo web se detallan a continuación:

#### Cargar Cargarssutsolici

```
package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutsolici {
    public String Llenarssutsolici( String SOL){
```

```

        try
    {
        DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
        String nombre_servidor = "X.X.X.X";
        String nombre_servidor_qa = "X.X.X.X";
        String numero_puerto = "X";
        String sid = "X";
        String sidqa = "X";

        String url = "jdbc:oracle:thin:@" + nombre_servidor + "." + numero_puerto + "." + sid;
        String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + "." + numero_puerto + "." +
sidqa;

        //Nombre usuario y password
        String usuario = "X";
        String password = "X";
        String usuario_qa = "X";
        String password_qa = "X";

        //Obtiene la conexion producción
        Connection conexion = DriverManager.getConnection( url, usuario, password );
        System.out.println("conexion exitosa");
        //Para realiza una consulta

        Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
        System.out.println("conexion QA exitosa");
        Statement sentencia_ruc_qa = conexion2.createStatement();
        sentencia_ruc_qa.executeQuery("delete SSUTSOLICI WHERE IDE_NUM_SOLICITU =" +
SOL);
        System.out.println("Borrado de datos correcto en tabla solicitud QA para solicitud"+ SOL);
        Statement sentencia_prod = conexion.createStatement();
        ResultSet rs_sol = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTSOLICI WHERE IDE_NUM_SOLICITU =" +SOL);
        System.out.println("Selección de datos de producción correcta en tabla solicitud para
Solicitud"+ SOL);
        while ( rs_sol.next() )
        {
            StringBuilder sbf = new StringBuilder("");
            for (int i=0;i<21;i++) {

```

```

        if (rs_sol.getString(i+1) == null || i == 0 || i == 1 || i == 4 || i == 6 || i == 19 || i == 20
    ){
        sbf.append( rs_sol.getString(i+1)+",");
    }

    else if(i == 8 || i == 9|| i == 12 || i == 15) {

        sbf.append("TO_DATE('" +rs_sol.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( "" +rs_sol.getString(i+1)+" ""+",");
    }
}

String insert = "insert into SSUTSOLICI (IDE_NUM_SOLICITU, COD_TDOC_TITULAR,
COD_EDOC_TITULAR, NUM_DOC_TITULAR, COD_TTIPO_SUBSIDIO,
COD_ETIPO_SUBSIDIO, COD_TESTADO_SUBSIDIO, COD_EESTADO_SUBSIDIO,
FEC_FECHA_PRESENTACION, FEC_FECHA_PROCESO, COD_EXPEDIENTE_NIT,
FLG_ACTIVADO_INACTIVO, FEC_USUARIO_CREA, COD_USUARIO_CREA,
TXT_IPUSUARIO_CREA, FEC_USUARIO_ACT, COD_USUARIO_ACT, TXT_IPUSUARIO_ACT,
COD_EMPLEADOR, COD_IDE_REPOLACT, NUM_MONTO_SUBSIDIADO) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SSUTSOLICI QA...");
}
System.out.println("Registro de datos de correcta en tabla SOLICITUD QA para Solicitud"+
SOL);
conexion.close();
conexion2.close();
}

catch( Exception e ){
e.printStackTrace();
}

return "ok";
}
}

```

Cargar Cargarssutdetlac

```
package gob.pe.essalud;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.Statement;
```

```
public class Cargarssutdetlac {
```

```
    public String Llenarssutdetlac( String SOL){
```

```
        try
```

```
        {
```

```
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
```

```
            String nombre_servidor = "X.X.X.X";
```

```
            String nombre_servidor_qa = "X.X.X.X";
```

```
            String numero_puerto = "X";
```

```
            String sid = "X";
```

```
            String sidqa = "X";
```

```
            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
```

```
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
```

```
sidqa;
```

```
            //Nombre usuario y password
```

```
            String usuario = "X";
```

```
            String password = "X";
```

```
            String usuario_qa = "X";
```

```
            String password_qa = "X";
```

```
            //Obtiene la conexion producción
```

```
            Connection conexion = DriverManager.getConnection( url, usuario, password );
```

```
            System.out.println("conexion exitosa");
```

```
            //Para realiza una consulta
```

```
            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
```

```
);
```

```
            System.out.println("conexion QA exitosa");
```

```
            Statement sentencia_ruc_qa = conexion2.createStatement();
```

```
            sentencia_ruc_qa.executeQuery("delete SSUTDETLAC WHERE COD_IDE_SOLICITUD
```

```

="+ SOL);
    System.out.println("Borrado de datos correcto en QA para la solicitud"+ SOL);
        Statement sentencia_prod = conexion.createStatement();
        ResultSet rs_lac = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTDETLAC WHERE COD_IDE_SOLICITUD =" +SOL);

    System.out.println("Selección de datos de producción correcta para la solicitud"+ SOL);
    while ( rs_lac.next() )
    {
        StringBuilder sbf = new StringBuilder("");
        for (int i=0;i<14;i++) {
            if (rs_lac.getString(i+1) == null || i == 0 || i == 1 || i == 4 || i == 8 || i == 13){
                sbf.append( rs_lac.getString(i+1)+",");
            }

            else if(i == 2) {

                sbf.append("TO_DATE('" +rs_lac.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
            }

            /*else if (i == 15){

                sbf.append("TO_DATE('" +rs_ruc.getString(i+1)+"','dd-mm-yy
HH24:MI:SS'),").toString().substring(0,29);
            }*/

            else
            {
                sbf.append( "" +rs_lac.getString(i+1)+""+",");
            }

        }

        String insert = "insert into SSUTDETLAC (IDE_DETALLE_LACTANCIA,
COD_IDE_SOLICITUD, FEC_NACIMIENTO_HIJO, TXT_ORIGEN_SOLI, COD_TDOC_HIJO,
COD_EDOC_HIJO, TXT_NDOC_HIJO, TXT_IDENTIFICADOR_HIJO,
COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO, TXT_NDOC_BENEFICIARIO,
COD_RENAES_CAS, UBIGEO_CAS_NAC, COD_IDE_REPOLACT) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+")";

        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);

```

```

        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en SSUTDETLAC QA...");
    }
    System.out.println("Registro de datos de correcta en tabla LACTANCIA QA para la
solicitud"+ SOL);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
        e.printStackTrace();
    }
    return "ok";
}
}

```

## **Cargar Cargarssutdetmat**

```

package gob.pe.essalud;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetmat {
    public String Llenarssutdetmat( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

```

```

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";
//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);

System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete SSUTDETMAT WHERE NUM_IDE_SOLICITUD
="+ SOL);
System.out.println("Borrado de datos correcto en QA para la solicitud"+ SOL);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_mat = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTDETMAT WHERE NUM_IDE_SOLICITUD="+SOL);

System.out.println("Selección de datos de producción correcta para la solicitud"+ SOL);
while ( rs_mat.next() )
{
StringBuffer sbf = new StringBuffer("");
for (int i=0;i<36;i++) {
if (rs_mat.getString(i+1) == null || i == 0 || i == 1 || i == 2 || i == 4 || i == 9 || i == 10
|| i == 11 || i == 14 || i == 15 || i == 35){
sbf.append( rs_mat.getString(i+1)+"");
}

else if(i == 7 || i == 8|| i == 12 || i == 13 || i == 17 ||i == 19 ||i == 22 ||i == 32 ||i ==
33) {

sbf.append("TO_DATE("'+rs_mat.getString(i+1)+'",'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

else

```

```

        {
            sbf.append( ""+rs_mat.getString(i+1)+"",");
        }
    }

    String insert = "insert into SSUTDETMAT(IDE_DETALLE_MATERNIDAD,
NUM_IDE_SOLICITUD, COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO,
NUM_NDOC_BENEFICIARIO, NUM_RUC_EMPLEADOR, NUM_NUMERO_CITT, FEC_INI_CITT,
FEC_FIN_CITT, NUM_DIAS_CITT, NUM_MONTO_SOLICITADO, NUM_MONTO_PAGADO,
FEC_INI_SUBS_SOLICI, FEC_FIN_SUBS_SOLICI, NUM_DIAS_SUBSIDIADOS, NUM_ARMADA,
NUM_EXPEDIENTE, FEC_PAGO, FLG_ACTIVADO_INACTIVO, FEC_USUARIO_CREA,
COD_USUARIO_CREA, TXT_IPUSUARIO_CREA, FEC_USUARIO_ACT, COD_USUARIO_ACT,
TXT_IPUSUARIO_ACT, TXT_NOMBRE_COMPLETO, TXT_RAZON_SOCIAL,
TXT_TIPO_EMPLEADO, TXT_TIPO_SEGURO, TXT_ZONA_UPE, FLG_RESULTADO,
TXT_MOTIVO_RECHAZO, FEC_INI_DIAS_PAGADOS, FEC_FIN_DIAS_PAGADOS,
FLG_SECTOR_EDUCACION, NUM_DECL_JURADA) values("+ sbf.toString().substring(0,
sbf.toString().length()-1)+)";
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en SSUTDETMAT QA...");
    }
    System.out.println("Registro de datos de correcta en tabla MATERNIDAD QA para la
solicitud"+ SOL);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
            e.printStackTrace();
        }
        return "ok";
    }
}

```

### **Cargar Cargarssutdetinc**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

```

```

import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetinc {
    public String Llenarssutdetinc( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";
            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete SSUTDETINC WHERE NUM_IDE_SOLICITUD
="+ SOL);
            System.out.println("Borrado de datos correcto en QA para la solicitud"+ SOL);
            Statement sentencia_prod = conexion.createStatement();
            ResultSet rs_inc = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTDETINC WHERE NUM_IDE_SOLICITUD =" +SOL);

            System.out.println("Selección de datos de producción correcta para la solicitud"+ SOL);

```

```

while ( rs_inc.next() )
{
    StringBuilder sbf = new StringBuilder("");
    for (int i=0;i<44;i++) {
        if (rs_inc.getString(i+1) == null || i == 0 || i == 1 || i == 2 || i == 9 || i == 10 || i == 11
|| i == 14 || i == 15 || i == 38){
            sbf.append( rs_inc.getString(i+1)+",");
        }

        else if(i == 7 || i == 8 || i == 12 || i == 13 || i == 17 || i == 19 || i == 22 || i == 32 || i ==
33 || i == 35 || i == 36 || i == 40 || i == 41 || i == 43) {

            sbf.append("TO_DATE('" +rs_inc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
        }

        else
        {
            sbf.append( ""+rs_inc.getString(i+1)+"",");
        }
    }

    String insert = "insert into SSUTDETINC(IDE_DETALLE_INCAPACIDAD,
NUM_IDE_SOLICITUD, COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO,
NUM_NDOC_BENEFICIARIO, NUM_RUC_EMPLEADOR, NUM_NUMERO_CITT, FEC_INI_CITT,
FEC_FIN_CITT, NUM_DIAS_CITT, NUM_MONTO_SOLICITADO, NUM_MONTO_PAGADO,
FEC_INI_SUBS_SOLICI, FEC_FIN_SUBS_SOLICI, NUM_DIAS_SUBSIDIADOS,
NUM_SUBSIDIO, NUM_EXPEDIENTE, FEC_PAGO, FLG_ACTIVO_INACTIVO,
FEC_USUARIO_CREA, COD_USUARIO_CREA, TXT_IPUSUARIO_CREA,
FEC_USUARIO_ACT, COD_USUARIO_ACT, TXT_IPUSUARIO_ACT,
TXT_NOMBRE_COMPLETO, TXT_RAZON_SOCIAL, TXT_TIPO_EMPLEADO,
TXT_TIPO_SEGURO, TXT_ZONA_UPE, FLG_RESULTADO, TXT_MOTIVO_RECHAZO,
FEC_INI_DIAS_PAGADOS, FEC_FIN_DIAS_PAGADOS, FLG_TIPO_SECTOREDUCA,
FEC_INI_VAC_PRIV, FEC_FIN_VAC_PRIV, FLG_DECL_JURADA, FLG_SUBS_ENPARTE,
NUM_CITT_B, FEC_INI_CITTB, FEC_FIN_CITTB, FLG_AFILIADO_CTR, FEC_CESE) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+)";

    PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
    qa_insert_contribuyentes.executeUpdate();
    System.out.println("Registrando datos en SSUTDETINC QA...");
}

```

```

        System.out.println("Registro de datos de correcta en QA en tabla INCAPACIDAD para la
solicitud"+ SOL);
        conexion.close();
        conexion2.close();
    }
        catch( Exception e ){
        e.printStackTrace();
    }
        return "ok";
    }
}

```

### **Cargar Cargarssutdetsep**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetsep {
    public String Llenarssutdetsep( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";

```

```

String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);

System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete SSUTDETSEP WHERE COD_IDE_SOLICITUD
="+ SOL);
System.out.println("Borrado de datos correcto en QA para la solicitud"+ SOL);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_inc = sentencia_prod.executeQuery("Select * from
SSUTDETSEP WHERE COD_IDE_SOLICITUD =" +SOL);
System.out.println("Selección de datos de producción correcta para la solicitud"+ SOL);
while ( rs_inc.next() )
{
StringBuilder sbf = new StringBuilder("");
for (int i=0;i<8;i++) {
if (rs_inc.getString(i+1) == null || i == 0 || i == 1 || i == 3 || i == 7){
sbf.append( rs_inc.getString(i+1)+",");
}

else if(i == 2) {

sbf.append("TO_DATE('" +rs_inc.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}

else
{
sbf.append( ""+rs_inc.getString(i+1)+""+",");
}

}

String insert = "insert into SSUTDETSEP(IDE_DETALLE_SEPELIO,
COD_IDE_SOLICITUD, FEC_FALL_TITULAR, COD_TDOC_BENEFICIARIO,

```

```

COD_EDOC_BENEFICIARIO, TXT_NDOC_BENEFICIARIO, TXT_NOMBRE_BENEFICIARIO,
NUM_IMPORTE_DECLARACION) values("+ sbf.toString().substring(0, sbf.toString().length()-
1)+");
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en SSUTDETSEP QA...");
    }
    System.out.println("Registro de datos de correcta en QA en tabla INCIDENCIA para la
solicitud"+ SOL);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
        e.printStackTrace();
    }
    return "ok";
}
}

```

## **Cargar Cargarssutsolici\_ov**

```
package gob.pe.essalud;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

public class Cargarssutsolici_ov {
    public String Llenarssutsolici_ov( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";

```

```

String nombre_servidor_qa = "X.X.X.X";
String numero_puerto = "X";
String sid = "X";
String sidqa = "X";

String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete SSUTSOLICI_OV WHERE
IDE_NUM_SOLICITUD_OV =" + SOL);
System.out.println("Borrado de datos correcto en tabla solicitud QA para solicitud" + SOL);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_sol = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTSOLICI_OV WHERE IDE_NUM_SOLICITUD_OV =" + SOL);

System.out.println("Selección de datos de producción correcta en tabla solicitud para
Solicitud" + SOL);
while ( rs_sol.next() )
{
StringBuffer sbf = new StringBuffer("");
for (int i=0;i<15;i++) {
if (rs_sol.getString(i+1) == null || i == 0 || i == 1 || i == 4 || i == 8 || i == 11 || i == 14
){
sbf.append( rs_sol.getString(i+1)+",");

```

```

    }

    else if(i == 6 || i == 7 || i == 9 ) {

        sbf.append("TO_DATE('"+rs_sol.getString(i+1)+"','YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
    }

    else
    {
        sbf.append( ""+rs_sol.getString(i+1)+"",");
    }
}

String insert = "insert into SSUTSOLICI_OV (IDE_NUM_SOLICITUD_OV,
COD_TDOC_USUARIO, COD_EDOC_USUARIO, NUM_DOC_USUARIO,
COD_TTIPO_SUBSIDIO, COD_ETIPO_SUBSIDIO, FEC_USUARIO_CREA,
FEC_USUARIO_ACT, FLG_ESTADO_PRO, FEC_PROCESO, URL_DOCUMENTO,
IDE_NUM_SOLICITUD, COD_OSPE, TIP_PAGO, IDE_GROUP_SOLICITUD_OV) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SSUTSOLICI_OV QA...");
}
System.out.println("Registro de datos correcto en tabla SOLICITUD QA para Solicitud"+
SOL);
conexion.close();
conexion2.close();
}
catch( Exception e ){
e.printStackTrace();
}

return "ok";
}
}

```

### **Cargar Cargarssutdetinc\_ov**

```
package gob.pe.essalud;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetinc_ov {
    public String Llenarssutdetinc_ov( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete SSUTDETINC_OV WHERE
IDE_NUM_SOLICITUD_OV =" + SOL);
            System.out.println("Borrado de datos correcto en QA para la solicitud"+ SOL);
            Statement sentencia_prod = conexion.createStatement();

```

```
ResultSet rs_inc = sentencia_prod.executeQuery("Select * from  
OSPE_VIRTUAL.SSUTDETINC_OV WHERE IDE_NUM_SOLICITUD_OV =" + SOL);
```

```
System.out.println("Selección de datos de producción correcta para la solicitud" + SOL);  
while ( rs_inc.next() )  
{  
    StringBuilder sbf = new StringBuilder("");  
    for (int i=0;i<45;i++) {  
        if (rs_inc.getString(i+1) == null || i == 0 || i == 1 || i == 2 || i == 12 || i == 13 || i ==  
14 || i == 17 || i == 18 || i == 39){  
            sbf.append( rs_inc.getString(i+1)+",");  
        }  
  
        else if(i == 10 || i == 11 || i == 15 || i == 16 || i == 20 || i == 22 || i == 25 || i == 33 || i  
== 34 || i == 36 || i == 37 || i == 41 || i == 42 || i == 44) {  
  
            sbf.append("TO_DATE(" + rs_inc.getString(i+1) + ", 'YYYY-MM-DD  
HH24:MI:SS')").toString().substring(0,29);  
        }  
  
        else  
        {  
            sbf.append( "" + rs_inc.getString(i+1) + "" + ",");  
        }  
    }  
}
```

```
String insert = "insert into SSUTDETINC_OV (IDE_DETALLE_INCAPACIDAD_OV,  
IDE_NUM_SOLICITUD_OV, COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO,  
NUM_NDOC_BENEFICIARIO, TXT_NOMBRE_COMPLETO, NUM_RUC_EMPLEADOR,  
TXT_RAZON_SOCIAL, TIP_CITT, NUM_NUMERO_CITT, FEC_INI_CITT, FEC_FIN_CITT,  
NUM_DIAS_CITT, NUM_MONTO_SOLICITADO, NUM_MONTO_PAGADO,  
FEC_INI_SUBS_SOLICI, FEC_FIN_SUBS_SOLICI, NUM_DIAS_SUBSIDIADOS,  
NUM_SUBSIDIO, NUM_EXPEDIENTE, FEC_PAGO, FLG_ACTIVO_INACTIVO,  
FEC_USUARIO_CREA, COD_USUARIO_CREA, TXT_IPUSUARIO_CREA,  
FEC_USUARIO_ACT, COD_USUARIO_ACT, TXT_IPUSUARIO_ACT, TXT_TIPO_EMPLEADO,  
TXT_TIPO_SEGURO, TXT_ZONA_UPE, FLG_RESULTADO, TXT_MOTIVO_RECHAZO,  
FEC_INI_DIAS_PAGADOS, FEC_FIN_DIAS_PAGADOS, FLG_TIPO_SECTOREDUCIA,  
FEC_INI_VAC_PRIV, FEC_FIN_VAC_PRIV, FLG_DECL_JURADA, FLG_SUBS_ENPARTE,  
NUM_CITT_B, FEC_INI_CITTB, FEC_FIN_CITTB, FLG_AFILIADO_CTR, FEC_CESE) values(" +
```

```

sbf.toString().substring(0, sbf.toString().length()-1)+"");
        PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
        qa_insert_contribuyentes.executeUpdate();
        System.out.println("Registrando datos en SSUTDETINC_OV QA...");
    }
    System.out.println("Registro de datos correcto en QA en tabla INCIDENCIA para la
solicitud"+ SOL);
    conexion.close();
    conexion2.close();
}
        catch( Exception e ){
        e.printStackTrace();
    }
    return "ok";
}
}

```

### **Cargar Cargarssutdetmat\_ov**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetmat_ov {
    public String Llenarssutdetmat_ov( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

```

```

//Nombre usuario y password
String usuario = "X";
String password = "X";
String usuario_qa = "X";
String password_qa = "X";

//Obtiene la conexion producción
Connection conexion = DriverManager.getConnection( url, usuario, password );
System.out.println("conexion exitosa");
//Para realiza una consulta

Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
System.out.println("conexion QA exitosa");
Statement sentencia_ruc_qa = conexion2.createStatement();
sentencia_ruc_qa.executeQuery("delete SSUTDETMAT_OV WHERE
NUM_IDE_SOLICITUD =" + SOL);
System.out.println("Borrado de datos correcto en QA para la solicitud" + SOL);
Statement sentencia_prod = conexion.createStatement();
ResultSet rs_mat = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTDETMAT_OV WHERE NUM_IDE_SOLICITUD =" + SOL);

System.out.println("Selección de datos de producción correcta para la solicitud" + SOL);
while ( rs_mat.next() )
{
StringBuffer sbf = new StringBuffer("");
for (int i=0;i<37;i++) {
if (rs_mat.getString(i+1) == null || i == 0 || i == 1 || i == 2 || i == 9 || i == 10 || i ==
11 || i == 14 || i == 15 || i == 35){
sbf.append( rs_mat.getString(i+1)+" ");
}

else if(i == 7 || i == 8 || i == 12 || i == 13 || i == 17 || i == 19 || i == 22 || i == 32 || i ==
33 || i == 36) {

sbf.append("TO_DATE('" + rs_mat.getString(i+1) + "', 'YYYY-MM-DD
HH24:MI:SS'),").toString().substring(0,29);
}
}
}

```

```

else
{
    sbf.append( ""+rs_mat.getString(i+1)+"",");
}
}

String insert = "insert into SSUTDETMAT_OV (IDE_DETALLE_MATERNIDAD,
NUM_IDE_SOLICITUD, COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO,
NUM_NDOC_BENEFICIARIO, NUM_RUC_EMPLEADOR, NUM_NUMERO_CITT, FEC_INI_CITT,
FEC_FIN_CITT, NUM_DIAS_CITT, NUM_MONTO_SOLICITADO, NUM_MONTO_PAGADO,
FEC_INI_SUBS_SOLICI, FEC_FIN_SUBS_SOLICI, NUM_DIAS_SUBSIDIADOS, NUM_ARMADA,
NUM_EXPEDIENTE, FEC_PAGO, FLG_ACTIVADO_INACTIVO, FEC_USUARIO_CREA,
COD_USUARIO_CREA, TXT_IPUSUARIO_CREA, FEC_USUARIO_ACT, COD_USUARIO_ACT,
TXT_IPUSUARIO_ACT, TXT_NOMBRE_COMPLETO, TXT_RAZON_SOCIAL,
TXT_TIPO_EMPLEADO, TXT_TIPO_SEGURO, TXT_ZONA_UPE, FLG_RESULTADO,
TXT_MOTIVO_RECHAZO, FEC_INI_DIAS_PAGADOS, FEC_FIN_DIAS_PAGADOS,
FLG_SECTOR_EDUCACION, NUM_DECL_JURADA, FEC_CESE) values("+
sbf.toString().substring(0, sbf.toString().length()-1)+)";

PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();

System.out.println("Registrando datos en SSUTDETMAT_OV QA...");
}

System.out.println("Registro de datos correcto en tabla MATERNIDAD QA para la
solicitud"+ SOL);

conexion.close();
conexion2.close();
}

catch( Exception e ){
e.printStackTrace();
}

return "ok";
}
}

```

### **Cargar Cargarssutdetlac\_ov**

```

package gob.pe.essalud;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

```

```

import java.sql.ResultSet;
import java.sql.Statement;

public class Cargarssutdetlac_ov {
    public String Llenarssutdetlac_ov( String SOL){
        try
        {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            String nombre_servidor = "X.X.X.X";
            String nombre_servidor_qa = "X.X.X.X";
            String numero_puerto = "X";
            String sid = "X";
            String sidqa = "X";

            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sidqa;

            //Nombre usuario y password
            String usuario = "X";
            String password = "X";
            String usuario_qa = "X";
            String password_qa = "X";

            //Obtiene la conexion producción
            Connection conexion = DriverManager.getConnection( url, usuario, password );
            System.out.println("conexion exitosa");
            //Para realiza una consulta

            Connection conexion2 = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
            System.out.println("conexion QA exitosa");
            Statement sentencia_ruc_qa = conexion2.createStatement();
            sentencia_ruc_qa.executeQuery("delete SSUTDETLAC_OV WHERE
IDE_NUM_SOLICITUD_OV =" + SOL);
            System.out.println("Borrado de datos correcto en QA para la solicitud" + SOL);
            Statement sentencia_prod = conexion.createStatement();
            ResultSet rs_lac = sentencia_prod.executeQuery("Select * from
OSPE_VIRTUAL.SSUTDETLAC_OV WHERE IDE_NUM_SOLICITUD_OV =" + SOL);

```

```

System.out.println("Selección de datos de producción correcta para la solicitud"+ SOL);
while ( rs_lac.next() )
{
StringBulder sbf = new StringBulder("");
for (int i=0;i<10;i++) {
if (rs_lac.getString(i+1) == null || i == 0 || i == 1 || i == 2 || i == 5){
sbf.append( rs_lac.getString(i+1)+",");
}

else
{
sbf.append( ""+rs_lac.getString(i+1)+" "+",");
}

}

String insert = "insert into SSUTDETLAC_OV (IDE_DETALLE_LACT_OV,
IDE_NUM_SOLICITUD_OV, COD_TDOC_HIJO, COD_EDOC_HIJO, TXT_NDOC_HIJO,
COD_TDOC_BENEFICIARIO, COD_EDOC_BENEFICIARIO, TXT_NDOC_BENEFICIARIO,
COD_ETIPO_PARTO, COD_CUI) values("+ sbf.toString().substring(0, sbf.toString().length()-1)+)";
PreparedStatement qa_insert_contribuyentes = conexion2.prepareStatement(insert);
qa_insert_contribuyentes.executeUpdate();
System.out.println("Registrando datos en SSUTDETLAC_OV QA...");
}
System.out.println("Registro de datos correcto en tabla LACTANCIA QA para la solicitud"+
SOL);
conexion.close();
conexion2.close();
}
catch( Exception e ){
e.printStackTrace();
}
return "ok";
}
}

```

## Clase Asegurado

```
package go.pe.essalud.replicardatos;
```

```

public class Asegurado {
    public String getDni() {
        return dni;
    }

    public Asegurado(String dni) {
        this.dni = dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

    @Override
    public String toString() {
        return "Asegurado{" +
            "dni=" + dni + "\" +
            "};
    }

    public Asegurado() {
    }

    public String dni;
}

```

## **Clase ReplicardatosApplication**

```

package go.pe.essalud.replicardatos;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
@SpringBootApplication
public class ReplicardatosApplication extends SpringBootServletInitializer {
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

```

```

        return application.sources(ReplicardatosApplication.class);
    }

    public static void main(String[] args) {
        SpringApplication.run(ReplicardatosApplication.class, args);
    }
}

```

## **Clase ServletInitializer**

```

package go.pe.essalud.replicardatos;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(ReplicardatosApplication.class);
    }

}

```

## **Clase ReplicardatosController**

```

package go.pe.essalud.replicardatos;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import javax.swing.text.html.Option;

@Controller
public class ReplicardatosController {

```

```

@Autowired
Replicardata servicio;

@RequestMapping("/")
public String index(Model model) {
    model.addAttribute("aseguradodato", new Asegurado());
    return "inicio";
}

@GetMapping(value = "/enviar/{docu}")
public ResponseEntity<String> enviardatos(@PathVariable("docu") String docu) {
    System.out.println(docu);
    servicio.Replicar(docu);
    return new ResponseEntity<String>("OK", HttpStatus.OK);
}
}

```

## Clase Replicardata

```

package go.pe.essalud.replicardatos;
import org.springframework.stereotype.Service;
import java.sql.*;
@Service
public class Replicardata {
    public String Replicar(String DNI) {
        try {
            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver() );
            //Base de datos QA
            String nombre_servidor_qa = "172.20.0.138";
            String numero_puerto = "1521";
            String sid = "DBSAS";
            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" +
sid;
            String usuario_qa = "USRCSA";
            String password_qa = "usrcsa22";
            //Base de datos producción SAS
            String nombre_servidor_prod = "172.20.0.185";
            String urlprod = "jdbc:oracle:thin:@" + nombre_servidor_prod + ":" + numero_puerto + ":" +
sid;

```

```

String usuarioprod = "USR_GSIT_KRUIZ";
String passwordprod = "ItTflm#cW7tm";
//Base de datos producción OSPE
String urlprodospe = "jdbc:oracle:thin:@" + nombre_servidor_prod + ":" + numero_puerto +
":" + sid;
String usuarioprodospe = "USR_GSIT_PSANTAMARIA";
String passwordprodospe = "Essalud19";

Cargarcsamperson persona = new Cargarcsamperson();
persona.llenarCsamperson(DNI);

CargarAcreditacion generar = new CargarAcreditacion();
generar.llenarAcreditacion(DNI);

Cargarnuevocuentaindividual cargarne = new Cargarnuevocuentaindividual();
cargarne.llenarnuevocuentaindividual(DNI);
//Cargarsccmdgat cargar = new Cargarsccmdgat();
//cargar.llenarSccmdgat(DNI);
CargarSsahceit cargarssah = new CargarSsahceit();
cargarssah.llenarSsahceit(DNI);
//Conexiones
Connection conexionqa = DriverManager.getConnection( url_qa, usuario_qa, password_qa
);
Connection conexionprod = DriverManager.getConnection( urlprod, usuarioprod,
passwordprod );
Connection conexionprodospe = DriverManager.getConnection( urlprodospe,
usuarioprodospe, passwordprodospe );
//Obtener ID's de tablas
Statement sentencia_qa = conexionqa.createStatement();
ResultSet rs_ruc = sentencia_qa.executeQuery("select distinct num_docide_empl from
usrca.acreditaciones where NUM_DOCIDE_ASEG ="+"""+DNI+""");
Statement sentencia_qa_2 = conexionqa.createStatement();
ResultSet rs_aut = sentencia_qa_2.executeQuery("select DGACAUT from
SIA.sccmdgat@LNK_SIA_SAS.ESSALUD c where DGANDID ="+"""+DNI+""");
Statement sentencia_qa_3 = conexionqa.createStatement();
ResultSet rs_aut2 = sentencia_qa_3.executeQuery("select DGACAUT from
SIA.sccmdgat@LNK_SIA_SAS.ESSALUD c where DGANDID ="+"""+DNI+""");

while ( rs_ruc.next() )

```

```

{
    String RUC = ""+rs_ruc.getString(1)+"";

    Cargarcontribuyentes llenarcont = new Cargarcontribuyentes();
    llenarcont.llenarcontribuyentes(RUC);

    Cargarcsamentida llenarent = new Cargarcsamentida();
    llenarent.llenarcsamentida(RUC);

    Cargarscchemot llenar = new Cargarscchemot();
    llenar.LlenarScchemot(RUC);
}

while ( rs_aut.next() )
{
    String AUT = ""+rs_aut.getString(1)+"";

    Cargarsscmcitt cargar2 = new Cargarsscmcitt();
    cargar2.LlenarSscmcitt(AUT);

    Cargarssumaset cargar3 = new Cargarssumaset();
    cargar3.LlenarSsumaset(AUT);

    Cargarscwtact cargar4 = new Cargarscwtact();
    cargar4.LlenarScwtact(AUT);
}

while ( rs_aut2.next() )
{
    String AUT = ""+rs_aut2.getString(1)+"";
    Statement sentencia_qa_4 = conexionqa.createStatement();
    ResultSet rs_exp = sentencia_qa_4.executeQuery("SELECT DISTINCT ASECEXP
FROM USRCSA.SSUMASET WHERE ASECAUT = "+AUT);
    while ( rs_exp.next() )
    {
        String EXP = ""+rs_exp.getString(1)+"";
        Cargarssumempt cargar5 = new Cargarssumempt();
        cargar5.LlenarSsumempt(EXP);
        Cargarssuhanu cargar6 = new Cargarssuhanu();
        cargar6.llenarSsuhanu(EXP);
    }
}

```

```

        Cargarssuhpagt cargar7 = new Cargarssuhpagt();
        cargar7.llenarSsuhpagt(EXP);

    }
}
Statement sentencia_prod = conexionprod.createStatement();
ResultSet rs_sol = sentencia_prod.executeQuery("SELECT * FROM
USRCSA.SSUTSOLICI WHERE NUM_DOC_TITULAR ="+""+DNI+"");
while ( rs_sol.next() )
{
    String SOL = ""+rs_sol.getString(1)+"";
    String COD = rs_sol.getString(6);
    Statement sentencia_qa_5 = conexionqa.createStatement();
    sentencia_qa_5.executeQuery("delete USRCSA.SSUTDETSEP WHERE
COD_IDE_SOLICITUD =" + SOL);
    sentencia_qa_5.executeQuery("delete USRCSA.SSUTDETLAC WHERE
COD_IDE_SOLICITUD =" + SOL);
    sentencia_qa_5.executeQuery("delete USRCSA.SSUTDETMAT WHERE
NUM_IDE_SOLICITUD =" + SOL);
    sentencia_qa_5.executeQuery("delete USRCSA.SSUTDETINC WHERE
NUM_IDE_SOLICITUD =" + SOL);

    Cargarssutsolici cargarsol = new Cargarssutsolici();
    cargarsol.llenarssutsolici(SOL);
    System.out.println(COD);
    switch(COD)
    {
        case "LT": case "LC":
            Cargarssutdetlac cargarlac = new Cargarssutdetlac();
            cargarlac.llenarssutdetlac(SOL);
            break;
        case "MA": case "MD":
            Cargarssutdetmat cargarmat = new Cargarssutdetmat();
            cargarmat.llenarssutdetmat(SOL);
            break;
        case "EN": case "AC": case "AT": case "EP":
            Cargarssutdetinc cargarin = new Cargarssutdetinc();
            cargarin.llenarssutdetinc(SOL);
            break;
        case "SE":

```

```

        Cargarssutdetsep cargarsep = new Cargarssutdetsep();
        cargarsep.Llenarssutdetsep(SOL);
        break;
    }
}

Statement sentencia_prodospe = conexionprodospe.createStatement();
ResultSet rs_inc = sentencia_prodospe.executeQuery("SELECT DISTINCT
IDE_NUM_SOLICITUD_OV FROM OSPE_VIRTUAL.SSUTDETINC_OV WHERE
NUM_NDOC_BENEFICIARIO ="+""+DNI+"");

while ( rs_inc.next() )
{
    String SOL = ""+rs_inc.getString(1)+"";
    Statement sentencia_qa_6 = conexionqa.createStatement();
    sentencia_qa_6.executeQuery("delete USRCSA.ssutdetinc_ov WHERE
ide_num_solicitud_ov =" + SOL);
    Cargarssutsolici_ov cargar = new Cargarssutsolici_ov();
    cargar.Llenarssutsolici_ov(SOL);
    Cargarssutdetinc_ov cargar2 = new Cargarssutdetinc_ov();
    cargar2.Llenarssutdetinc_ov(SOL);
}

ResultSet rs_mat = sentencia_prodospe.executeQuery("SELECT NUM_IDE_SOLICITUD
FROM OSPE_VIRTUAL.SSUTDETMAT_OV WHERE NUM_NDOC_BENEFICIARIO
="+""+DNI+"");

while ( rs_mat.next() )
{
    String SOL = ""+rs_mat.getString(1)+"";
    Statement sentencia_qa_7 = conexionqa.createStatement();
    sentencia_qa_7.executeQuery("delete USRCSA.ssutdetmat_ov WHERE
num_ide_solicitud =" + SOL);

    Cargarssutsolici_ov cargar = new Cargarssutsolici_ov();
    cargar.Llenarssutsolici_ov(SOL);

    Cargarssutdetmat_ov cargar1 = new Cargarssutdetmat_ov();
    cargar1.Llenarssutdetmat_ov(SOL);
}

```

```

        ResultSet rs_lac = sentencia_prodospe.executeQuery("SELECT
IDE_NUM_SOLICITUD_OV FROM OSPE_VIRTUAL.SSUTDETLAC_OV WHERE
TXT_NDOC_BENEFICIARIO ="+""+DNI+""");

        while ( rs_lac.next() )
        {
            String SOL = ""+rs_lac.getString(1)+"";
            Statement sentencia_qa_8 = conexionqa.createStatement();
            sentencia_qa_8.executeQuery("delete USRCSA.ssutdetlac_ov WHERE
ide_num_solicitud_ov ="+ SOL);

            Cargarssutsolici_ov cargar = new Cargarssutsolici_ov();
            cargar.Llenarssutsolici_ov(SOL);

            Cargarssutdetlac_ov llenar = new Cargarssutdetlac_ov();
            llenar.Llenarssutdetlac_ov(SOL);
        }

        Statement sentencia_prod_2 = conexionprod.createStatement();
        ResultSet rs_idt = sentencia_prod_2.executeQuery("select ide_numerico_persona from
usrca.csamperson where num_document_persona ="+""+DNI+""");
        while ( rs_idt.next() )
        {
            String IDT = ""+rs_idt.getString(1)+"";
            Cargarcsarvinfam generar2 = new Cargarcsarvinfam();
            generar2.llenarCsarvinfam(IDT);

            Cargarcsatinscri generar3 = new Cargarcsatinscri();
            generar3.llenarCsatinscri(IDT);
        }

        Statement sentencia_prod_3 = conexionprod.createStatement();
        ResultSet rs_sas = sentencia_prod_3.executeQuery("SELECT DGANSAS FROM
SIA.sccmdgat@LNK_SIA_SAS.ESSALUD WHERE DGANDID ="+""+DNI+""");
        while ( rs_sas.next() )
        {
            String SAS = ""+rs_sas.getString(1)+"";
            CargarSccmvfft cargar1 = new CargarSccmvfft();
            cargar1.llenarSccmvfft(SAS);
        }

```

```

    }

    conexionqa.close();
    conexionprod.close();
    conexionprodospe.close();
    System.out.println("Registro de datos correcto en QA para el DNI "+ DNI);
}

catch (SQLException e) {
    e.printStackTrace();
}

return "ok";

}
}

```

### 3.5.3.3 Sprint Backlog

La lista de tareas del Sprint 2 se definen en la tabla siguiente:

Tabla 40. Sprint02 Product Backlog

Identificador de la historia	Enunciado	Tareas	Dueño/ Voluntario	Estado
HU_003	Como usuario necesito los datos personales actualizados de las solicitudes de subsidios en ambiente de calidad	Creación de la clase Cargarssutsolici	John Antón	Terminado
		Creación de la clase Cargarssutdetlac	John Antón	Terminado
		Creación de la clase Cargarssutdetmat	John Antón	Terminado
		Creación de la clase Cargarssutdetinc	John Antón	Terminado
		Creación de la clase Cargarssutdetsep	John Antón	Terminado
		Creación de la clase Cargarssutsolici_ov	John Antón	Terminado
		Creación de la clase Cargarssutdetinc_ov	John Antón	Terminado
		Creación de la clase	John	Terminado

		Cargarssutdetmat_ov	Antón	
		Creación de la clase	John	Terminado
		Cargarssutdetlac_ov	Antón	
HU_004	Como usuario necesito replicar los datos del asegurado desde un aplicativo web	Creación de la clase	John	Terminado
		Asegurado	Antón	
		Creación de la clase	John	Terminado
		Replicardata	Antón	
		Creación del front del aplicativo	John Antón	Terminado

Fuente: Elaboración propia

### 3.5.3.4 Pruebas

Con la finalidad de probar nuestras clases creadas y la herramienta ETL detallada en las historias de usuario establecidas, por cada historia de usuario hay una variedad de criterios de aceptación, definiendo un límite de 4 de acuerdo a la metodología de Scrum, los mismos que están detallados de mejor forma en las funcionalidades de cada historia de usuario. Para el Sprint 2, se muestra el resultado final satisfactorio.

Seguidamente, se muestra el detalle de los casos de prueba ejecutados durante el Sprint 2:

Tabla 41. Casos de prueba (HU\_003)

HU_003	<b>Como usuario necesito los datos personales actualizados de las solicitudes de subsidios en ambiente de calidad</b>		
	<b>Rol</b>	<b>Características /Funcionalidad</b>	<b>Razón/ Resultado</b>
	Usuario	Necesito replicar los datos de solicitudes de subsidios del asegurado	Con la finalidad de obtener los datos de solicitudes de subsidios del asegurado en ambiente de calidad
<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado/ Comportamiento esperado</b>
Replicar datos de solicitudes de	En caso de que ingrese un número	cuando hago clic en el botón Ejecutar del	Se deberá migrar los datos de las

subsidio del asegurado	de DNI en las clases que replican las solicitudes de subsidio del asegurado	IDE Eclipse	solicitudes de subsidio del asegurado.
------------------------	---	-------------	--

Fuente: Elaboración propia

Figura 22. Prueba de la clase Cargarssutsolici (HU\_003)

```

1 package gob.pe.essalud;
2 import java.sql.Connection;
7
8 public class Cargarssutsolici {
9     static String SOL="";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver()
13            String nombre_servidor = " ";
14            String nombre_servidor_qa = " ";
15            //numero del puerto
16            String numero_puerto = "1521";
17            //SID
18            String sid = "DBSAS";
19            String sidqa = "DBSAS";
20
21            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_p
22            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + nu
23
24            //Nombre usuario y password
25            String usuario = " ";

```

<terminated> Cargarssutsolici [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 mar. 2021 11:00:00 AM)
conexion exitosa
conexion QA exitosa
Borrado de datos correcto en tabla solicitud QA para solicitud' '
Selección de datos de producción correcta en tabla solicitud para Solicitud' '
Registrando datos en SSUTSOLICI QA...
Registro de datos de correcta en tabla SOLICITUD QA para Solicitud' '

Fuente: Elaboración propia

Figura 23. Prueba de la clase Cargarssutdetlac (HU\_003)

```

1 package gob.pe.essalud;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8
9 public class Cargarssutdetlac {
10     static String SOL=" ";
11     public static void main(String[] args) {
12         try {
13             DriverManager.registerDriver( new oracle.jdbc.driver.OracleD
14                 //String SOL = "'401'";
15                 //nombre del servidor
16                 String nombre_servidor = " ";
17                 String nombre_servidor_qa = " ";
18                 //numero del puerto
19                 String numero_puerto = "1521";
20                 //SID
21                 String sid = "DBSAS";

```

<terminated> Cargarssutdetlac [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (2  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para la solicitud' '  
Selección de datos de producción correcta para la solicitud' '  
Registrando datos en SSUTDETLAC QA...  
Registro de datos de correcta en tabla LACTANCIA QA para la solicitud' '

Fuente: Elaboración propia

Figura 24. Prueba de la clase Cargarssutdetmat (HU\_003)

```

1 package gob.pe.essalud;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class Cargarssutdetmat {
9     static String SOL=" ";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.OracleD
13                String nombre_servidor = " ";
14                String nombre_servidor_qa = " ";
15                //numero del puerto
16                String numero_puerto = "1";
17                //SID
18                String sid = ".S";
19                String sidqa = ".S";
20
21                String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" +

```

<terminated> Cargarssutdetmat [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (2  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para la solicitud' '  
Selección de datos de producción correcta para la solicitud' '  
Registro de datos de correcta en tabla MATERNIDAD QA para la solicitud' '

Fuente: Elaboración propia

Figura 25. Prueba de la clase Cargarssutdetinc (HU\_003)

```

1 package gob.pe.essalud;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class Cargarssutdetinc {
9     static String SOL = "1";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDri
13            String nombre_servidor = " ";
14            String nombre_servidor_qa = " ";
15            //numero del puerto
16            String numero_puerto = " ";
17            //SID
18            String sid = " ";
19            String sidqa = " ";
20
21            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + nu

```

<terminated> Cargarssutdetinc [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 mar. 2018)
conexion exitosa
conexion QA exitosa
Borrado de datos correcto en QA para la solicitud' 1'
Selección de datos de producción correcta para la solicitud' 1'
Registro de datos de correcta en QA en tabla INCIDENCIA para la solicitud' 1'

Fuente: Elaboración propia

Figura 26. Prueba de la clase Cargarssutdetsep (HU\_003)

```

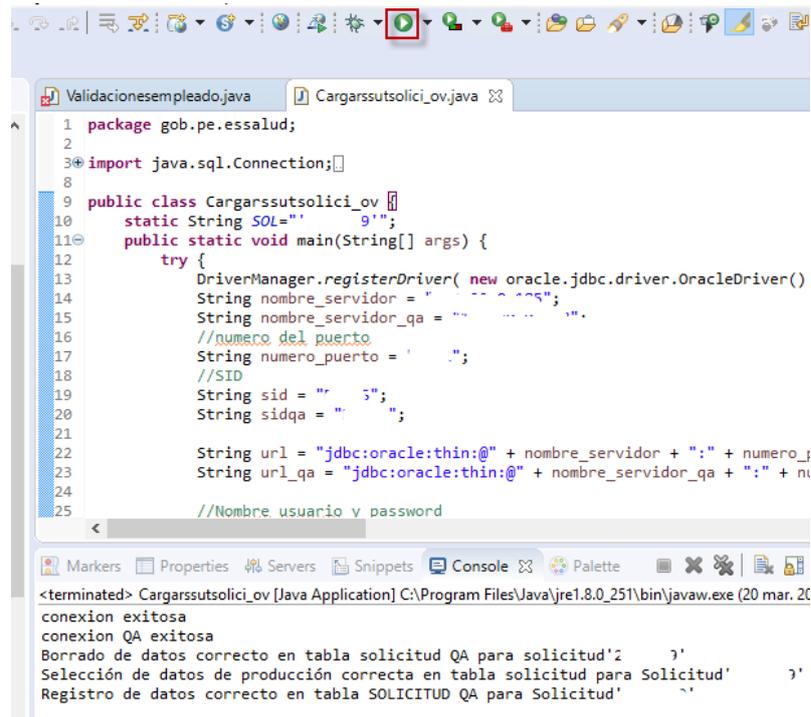
1 package gob.pe.essalud;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class Cargarssutdetsep {
9     static String SOL="1";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver(
13            String nombre_servidor = " ";
14            String nombre_servidor_qa = " ";
15            //numero del puerto
16            String numero_puerto = " ";
17            //SID
18            String sid = " ";
19            String sidqa = ' ';
20
21            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero

```

<terminated> Cargarssutdetsep [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 mar. 2018)
conexion exitosa
conexion QA exitosa
Borrado de datos correcto en QA para la solicitud' 1'
Selección de datos de producción correcta para la solicitud' 1'
Registro de datos de correcta en QA en tabla INCIDENCIA para la solicitud' 1'

Fuente: Elaboración propia

Figura 27. Prueba de la clase Cargarssutsolici\_ov (HU\_003)

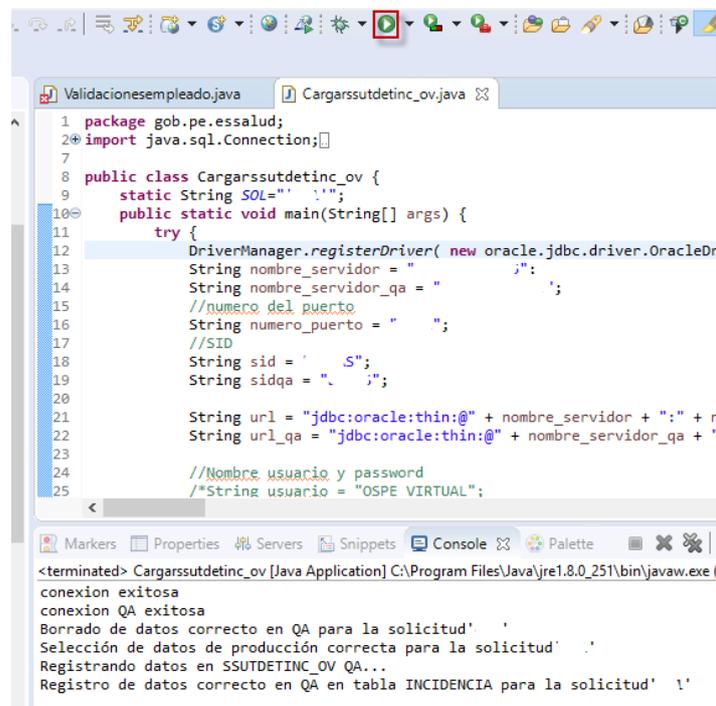


```
1 package gob.pe.essalud;
2
3 import java.sql.Connection;
4
5
6
7
8
9 public class Cargarssutsolici_ov {
10     static String SOL="";
11     public static void main(String[] args) {
12         try {
13             DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver()
14             String nombre_servidor = " ";
15             String nombre_servidor_qa = " ";
16             //numero del puerto
17             String numero_puerto = " ";
18             //SID
19             String sid = " ";
20             String sidqa = " ";
21
22             String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
23             String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" + sidqa;
24
25             //Nombre usuario y password
26             String usuario = "OSPE VIRTUAL";
27         } catch (Exception e) {
28             e.printStackTrace();
29         }
30     }
31 }
```

<terminated> Cargarssutsolici\_ov [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 mar. 2016 10:05:00 AM)  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en tabla solicitud QA para solicitud'2'  
Selección de datos de producción correcta en tabla solicitud para Solicitud'  
Registro de datos correcto en tabla SOLICITUD QA para Solicitud'

Fuente: Elaboración propia

Figura 28. Prueba de la clase Cargarssutdetinc\_ov (HU\_003)

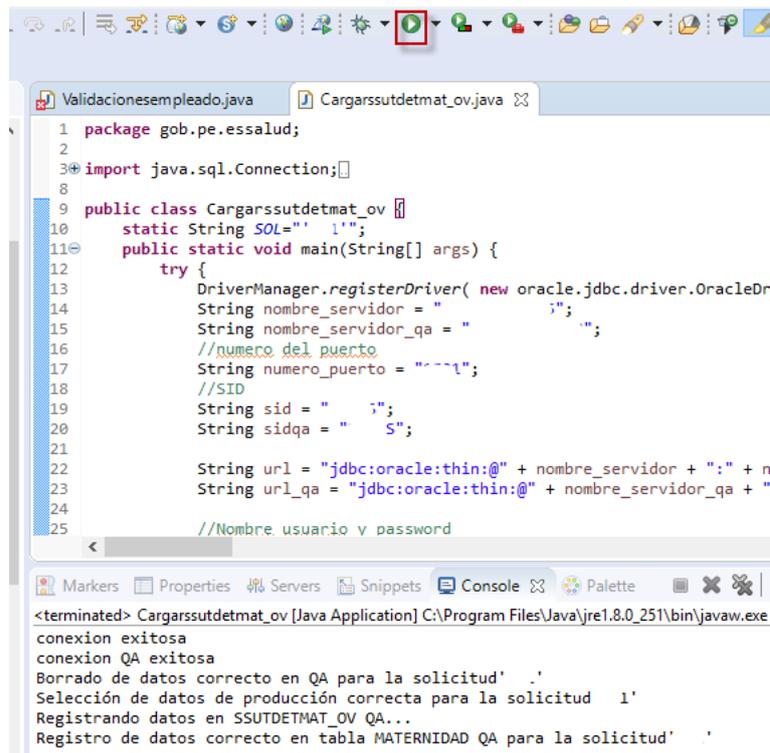


```
1 package gob.pe.essalud;
2 import java.sql.Connection;
3
4
5
6
7
8 public class Cargarssutdetinc_ov {
9     static String SOL="";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver()
13            String nombre_servidor = " ";
14            String nombre_servidor_qa = " ";
15            //numero del puerto
16            String numero_puerto = " ";
17            //SID
18            String sid = " ";
19            String sidqa = " ";
20
21            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + numero_puerto + ":" + sid;
22            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + ":" + numero_puerto + ":" + sidqa;
23
24            //Nombre usuario y password
25            String usuario = "OSPE VIRTUAL";
26        } catch (Exception e) {
27            e.printStackTrace();
28        }
29    }
30 }
```

<terminated> Cargarssutdetinc\_ov [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe (20 mar. 2016 10:05:00 AM)  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para la solicitud'  
Selección de datos de producción correcta para la solicitud'  
Registrando datos en SSUTDETIINC\_OV QA...  
Registro de datos correcto en QA en tabla INCIDENCIA para la solicitud' 1'

Fuente: Elaboración propia

Figura 29. Prueba de la clase Cargarssutdetmat\_ov (HU\_003)



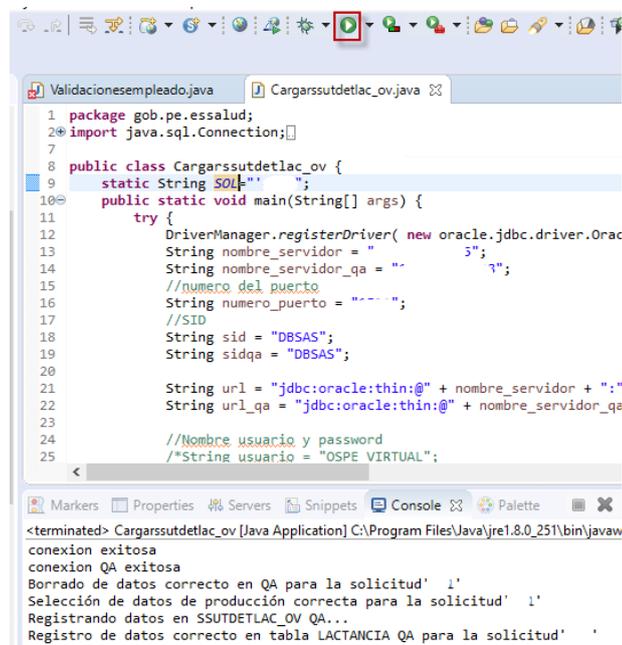
```
1 package gob.pe.essalud;
2
3 import java.sql.Connection;
4
5
6
7
8
9 public class Cargarssutdetmat_ov {
10     static String SOL="1";
11     public static void main(String[] args) {
12         try {
13             DriverManager.registerDriver( new oracle.jdbc.driver.OracleDr
14             String nombre_servidor = " ";
15             String nombre_servidor_qa = " ";
16             //numero del puerto
17             String numero_puerto = "1";
18             //SID
19             String sid = " ";
20             String sidqa = " S";
21
22             String url = "jdbc:oracle:thin:@" + nombre_servidor + ":" + n
23             String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa + "
24
25             //Nombre usuario y password

```

<terminated> Cargarssutdetmat\_ov [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw.exe  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para la solicitud' 1'  
Selección de datos de producción correcta para la solicitud 1'  
Registrando datos en SSUTDETMAT\_OV QA...  
Registro de datos correcto en tabla MATERNIDAD QA para la solicitud' 1'

Fuente: Elaboración propia

Figura 30. Prueba de la clase Cargarssutdetlac\_ov (HU\_003)



```
1 package gob.pe.essalud;
2 import java.sql.Connection;
3
4
5
6
7
8 public class Cargarssutdetlac_ov {
9     static String SOL="1";
10    public static void main(String[] args) {
11        try {
12            DriverManager.registerDriver( new oracle.jdbc.driver.Orac
13            String nombre_servidor = "5";
14            String nombre_servidor_qa = "1";
15            //numero del puerto
16            String numero_puerto = "1";
17            //SID
18            String sid = "DBSAS";
19            String sidqa = "DBSAS";
20
21            String url = "jdbc:oracle:thin:@" + nombre_servidor + ":"
22            String url_qa = "jdbc:oracle:thin:@" + nombre_servidor_qa
23
24            //Nombre usuario y password
25            /*String usuario = "OSPE VIRTUAL";

```

<terminated> Cargarssutdetlac\_ov [Java Application] C:\Program Files\Java\jre1.8.0\_251\bin\javaw  
conexion exitosa  
conexion QA exitosa  
Borrado de datos correcto en QA para la solicitud' 1'  
Selección de datos de producción correcta para la solicitud' 1'  
Registrando datos en SSUTDETLAC\_OV QA...  
Registro de datos correcto en tabla LACTANCIA QA para la solicitud' 1'

Fuente: Elaboración propia

Tabla 42. Casos de prueba (HU\_004)

HU_004	<b>Como usuario necesito replicar los datos del asegurado desde un aplicativo web</b>		
	<b>Rol</b>	<b>Características /Funcionalidad</b>	<b>Razón/ Resultado</b>
	Usuario	Necesito replicar los datos del asegurado desde un aplicativo web	Con la finalidad de replicar los datos del asegurado en ambiente de calidad
<b>Criterio de Aceptación</b>	<b>Contexto</b>	<b>Evento</b>	<b>Resultado/ Comportamiento esperado</b>
Replicar datos del asegurado desde un aplicativo web.	En caso de que ingrese un número de DNI	cuando hago clic en el botón "Replicar Datos"	Se deberá migrar los datos del asegurado según DNI ingresado en aplicativo web

Fuente: Elaboración propia

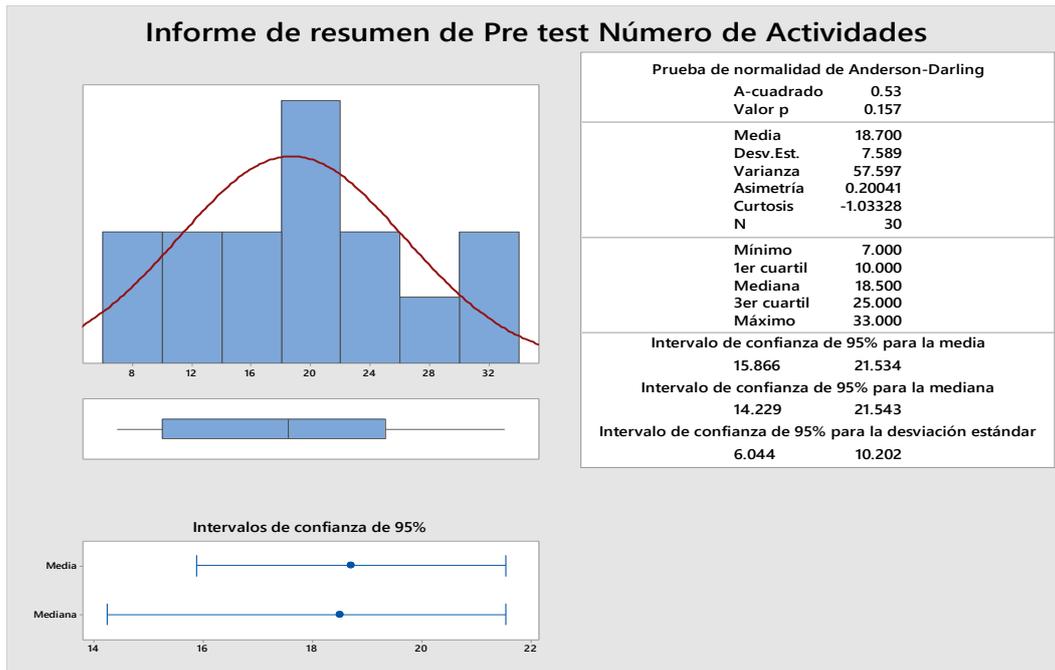
Figura 31. Prueba de réplica de datos desde aplicativo web (HU\_004)



Fuente: Elaboración propia

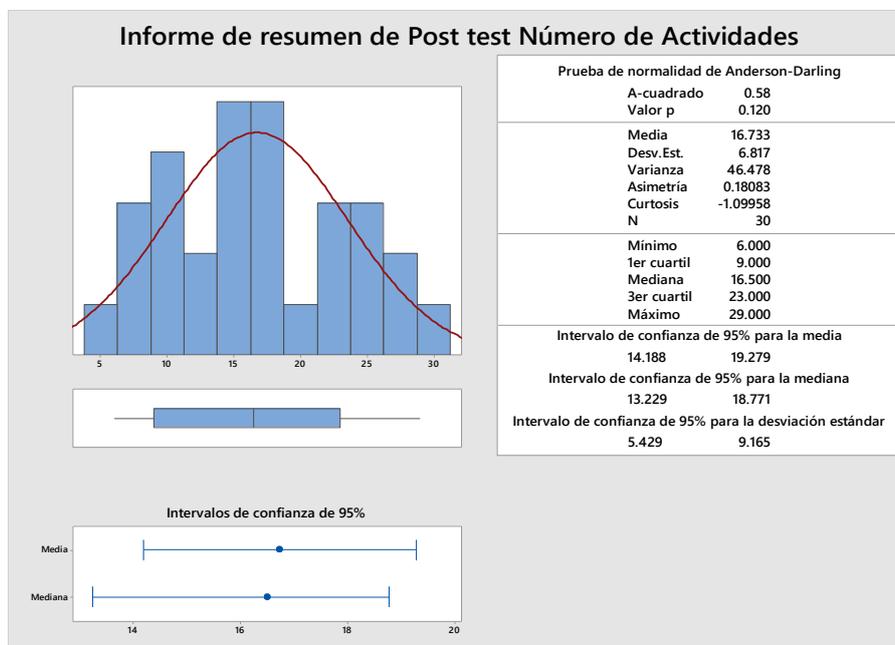
### 3.6 RESULTADOS

Figura 32. Informe estadístico del pre test Numero de actividades



Fuente: Elaboración propia.

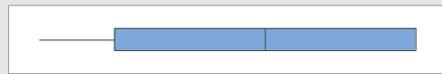
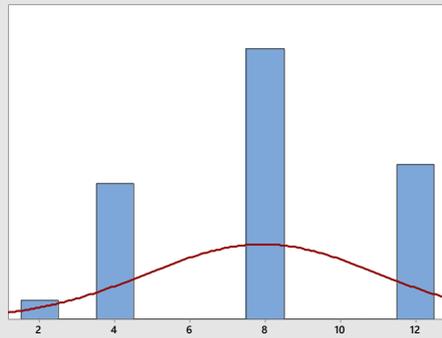
Figura 33. Informe estadístico del pos test Numero de actividades



Fuente: Elaboración propia.

Figura 34. Informe estadístico del pre test Tiempo de Carga de Datos

### Informe de resumen de Pre test Tiem de Carga de Datos

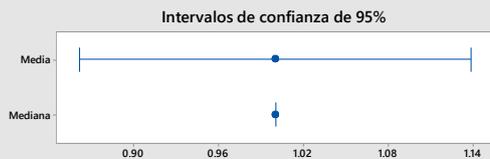
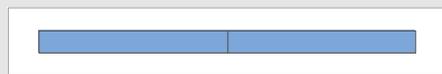
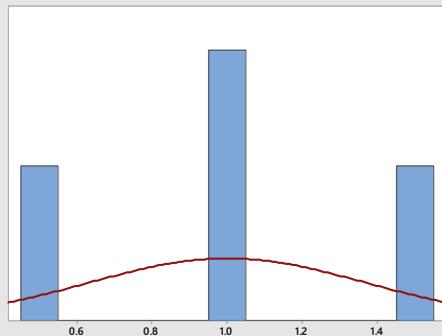


Prueba de normalidad de Anderson-Darling	
A-cuadrado	2.08
Valor p	<0.005
Media	7.9333
Desv.Est.	3.0843
Varianza	9.5126
Asimetría	-0.121918
Curtosis	-0.945245
N	30
Mínimo	2.0000
1er cuartil	4.0000
Mediana	8.0000
3er cuartil	12.0000
Máximo	12.0000
Intervalo de confianza de 95% para la media	
	6.7817 9.0850
Intervalo de confianza de 95% para la mediana	
	8.0000 8.0000
Intervalo de confianza de 95% para la desviación estándar	
	2.4563 4.1462

Fuente: Elaboración propia.

Figura35: Informe estadístico del pos test Tiempo de Carga de Datos

### Informe de resumen de Post test Tiem de Carga de Dato



Prueba de normalidad de Anderson-Darling	
A-cuadrado	2.31
Valor p	<0.005
Media	1.0000
Desv.Est.	0.3714
Varianza	0.1379
Asimetría	0.00000
Curtosis	-1.10764
N	30
Mínimo	0.5000
1er cuartil	0.5000
Mediana	1.0000
3er cuartil	1.5000
Máximo	1.5000
Intervalo de confianza de 95% para la media	
	0.8613 1.1387
Intervalo de confianza de 95% para la mediana	
	1.0000 1.0000
Intervalo de confianza de 95% para la desviación estándar	
	0.2958 0.4993

Fuente: Elaboración propia.

### 3.6.1 Prueba de Hipótesis:

## Prueba T e IC de dos muestras: Pre test Número de Actividades; Post test Número de Actividades

T de dos muestras para Pre test Número de Actividades vs. Post test Número de Actividades

	N	Media	Desv.Est.	Error estándar de la media
Pre test Número de Activ	30	18.70	7.59	1.4
Post test Número de Acti	30	16.73	6.82	1.2

Diferencia =  $\mu$  (Pre test Número de Actividades) -  $\mu$  (Post test Número de Actividades)

Estimación de la diferencia: 1.97

IC de 95% para la diferencia: (-1.76; 5.70)

Prueba T de diferencia = 0 (vs.  $\neq$ ): Valor T = 1.06 Valor p = 0.295 GL = 57

Como se puede apreciar en los resultados estadísticos ambas muestras tienen una distribución normal en ese sentido se aplicó la prueba de T, observándose que  $p > 0.005$  se acepta la hipótesis nula, que considera que no existe una diferencia significativa entre las dos pruebas.

## Prueba de Mann-Whitney e IC: Pre test Tiempo de carga de datos; Post test Tiempo de carga de datos

	N	Mediana
Pre test Tiem de Carga de Datos	30	8.000
Post test Tiem de Carga de Dato	30	1.000

La estimación del punto para  $\eta_1 - \eta_2$  es 7.000

95.2 El porcentaje IC para  $\eta_1 - \eta_2$  es (6.500;7.500)

W = 1365.0

Prueba de  $\eta_1 = \eta_2$  vs.  $\eta_1 \neq \eta_2$  es significativa en 0.0000

La prueba es significativa en 0.0000 (ajustado por empates)

Como se puede apreciar en los resultados estadísticos ambas muestras no tienen una distribución normal en ese sentido se aplicó la prueba de Mann-Whitney, observándose que  $p < 0.005$  en ese sentido se rechaza la hipótesis nula y se acepta la hipótesis alterna, que considera que existe una diferencia significativa entre las dos pruebas.

#### **IV. Conclusiones**

Se Determinó que existe una influencia significativa de la herramienta ETL en el tiempo de carga de data en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo, porque se bajó de 7.93 a 1 minuto.

Se Determinó que existe una influencia no tan significativa de la herramienta ETL en el número de actividades en los proyectos informáticos en el proceso de homologación de las bases de datos en los ambientes de calidad y productivo, porque se bajó de 18 a 16 actividades.

## RECOMENDACIONES

Se determinó que los aportes mas significativos del presente trabajo son los siguientes:

- Se logró elaborar scripts para reducir los tiempos de migración de datos personales y de vinculo familiar de un asegurado en una primera fase de la construcción de la herramienta ETL, esto satisfizo de manera parcial la necesidad del usuario de contar con datos replicados de los asegurados en el ambiente de calidad.
- Se logró elaborar scripts para reducir los tiempos de migración de datos de las solicitudes de subsidios de los asegurados en la etapa final de la construcción de la herramienta ETL, esto satisfizo la necesidad de contar con datos en la totalidad de tablas solicitadas por el usuario.
- Se logró elaborar y desplegar una herramienta ETL que nos ayude a reducir los tiempos de migración de datos de los asegurados a través de la web desde cualquier computadora conectada a la red corporativa.
- Se logró elaborar una herramienta ETL escalable al cual se le pueden adicionar mas scripts de migración de datos de otras tablas no consideradas en el presente trabajo.
- Se logró reducir el número de actividades en la planificación de los proyectos en la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales (SGSASS).

## VI. Referencias

- Antaño, A. C. M., Castro, J. M. M., & Valencia, R. E. C. (2014). Migración de Bases de Datos SQL a NoSQL. *Revista Tlamati, Especial*, 3, 144-148.
- Cetrángolo, O., Bertranou, F., Casanova, L., Casalí, P., International Labour Office, & EsSalud (Peru). (2013). *El sistema de salud del Perú: Situación actual y estrategias para orientar la extensión de la cobertura contributiva*. OIT: EsSalud.
- Rodríguez, M., & Rosso, M. (2013). *ProMiDa Proceso de Migración de Datos*.
- Barco, Carlos Andrés; Guzmán, Paola Andrea; Vivas Guzmán, Álvaro José. (2015). Criterios para la implementación del estándar SCRUM como marco de trabajo para el desarrollo de proyectos. Trabajo de Grado. Universidad San Buenaventura Cali
- Kniberg, Henrik; Skarin, Mattias. (2010). Kanban y Scrum – obteniendo lo mejor de ambos– Editorial: C4Media – EE.UU - Año 2010 .ISBN:978-0-557-13832-6.
- Schwaber, K., Beedle M., Martin R. (2001) "Agile Software Development with SCRUM". Editorial: Pearson Prentice Hall. Stoughton, MA – Estados Unidos - Año 2002 .ISBN: 0130676349, 9780130676344.
- León Silvestre, Javier J. (2014). App Web SCRUM – de <http://hdl.handle.net/2099.1/23350>
- En qué consiste el modelo en cascada*. (s. f.). OpenClassrooms. Recuperado 13 de mayo de 2021, de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>
- ESSALUD. (2020). *ROF\_Institucional\_Sistematizado\_24082020.pdf*. [http://www.essalud.gob.pe/transparencia/pdf/ROF\\_Institucional\\_Sistematizado\\_24082020.pdf](http://www.essalud.gob.pe/transparencia/pdf/ROF_Institucional_Sistematizado_24082020.pdf)
- Estadística Institucional | EsSalud*. (s. f.). Recuperado 13 de mayo de 2021, de <http://www.essalud.gob.pe/estadistica-institucional/>
- Estrategias de migración de datos a entornos de producción*. (2016, noviembre 29).

BeLike Software. <https://www.belikesoftware.com/estrategias-migracion-de-datos-entornos-produccion/>

Fernandez, J., & GIRALDO, J. C. (2011). Metodología para la construcción de un migrador universal de bases de datos. *Dyna*, 78(165), 266-275.

García, B. S. M., & Aya, J. T. C. (s. f.). *SOFTWARE MIGRADOR DE SCRIPTS .SQL ORACLE PARA BASES DE DATOS SQLSERVER EN LA EMPRESA EXPINN TECHNOLOGY*. 83.

Migración de datos. (2021). En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=Migraci%C3%B3n\\_de\\_datos&oldid=132306258](https://es.wikipedia.org/w/index.php?title=Migraci%C3%B3n_de_datos&oldid=132306258)

*Migración de datos a la nube: Herramientas, técnicas y tipos—Ekon*. (s. f.). Recuperado 13 de mayo de 2021, de <https://www.ekon.es/migracion-de-datos-a-la-nube-herramientas-tecnicas-y-tipos/>

*Migración de datos: Definición, desafíos y mejores prácticas para afrontarla*. (s. f.).

Recuperado 13 de mayo de 2021, de <https://www.powerdata.es/migracion-de-datos#:~:text=Llamamos%20migraci%C3%B3n%20de%20datos%20al,la%20aplicaci%C3%B3n%20que%20los%20gestiona.>

*¿Por qué fracasan los proyectos de TI en el estado Peruano? | América Sistemas*. (s. f.).

Recuperado 13 de mayo de 2021, de <https://www.americasistemas.com.pe/por-que-fracasan-los-proyectos-de-ti-en-el-estado-peruano/>

## Anexos

FICHA DE REGISTRO PARA EL LEVANTAMIENTO DE INFORMACIÓN PRE TEST/POST TEST		 Seguridad Social para todos
Investigador:	Antón Alvarado John Paúl	
Título de la Investigación:	Migración de Datos Para el Proceso de Homologación de Ambientes de Calidad y Producción en la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales de la Sede Central de EsSalud	
Organización:	Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales	
Dirección:	Jirón Domingo Cueto 119 - Lince	
Levantamiento de Información:	Periodo Pre test	01/07/2019 AL 31/12/2019
	Periodo Post test	01/01/2020 AL 30/06/2020
Variable en Estudio:	Proceso de Homologación de Ambientes de Calidad y Producción	
Proceso	Pre test Pre test Tiempo de Carga de Datos	Post test Tiempo de Carga de Datos
1	12	1.5
2	2	0.5
3	8	1
4	8	1
5	8	1
6	12	1.5
7	4	0.5
8	4	0.5
9	12	1.5
10	12	1.5
11	8	1
12	8	1
13	8	1
14	8	1
15	4	0.5
16	4	0.5
17	12	1.5
18	12	1.5
19	8	1
20	12	1.5
21	8	1
22	8	1
23	12	1.5
24	8	1
25	8	1
26	4	0.5
27	4	0.5
28	8	1
29	8	1
30	4	0.5
<b>Total</b>	238	30
<b>AUTORIZADO POR:</b>		
Apellidos y Nombres	Firma y Sello	
Kathy Ruiz Carrasco		

*Anexo 1. Ficha de registro de Pre test y Post test de Carga de datos*

FICHA DE REGISTRO PARA EL LEVANTAMIENTO DE INFORMACIÓN PRE TEST/POST TEST		 Seguridad Social para todos
Investigador:	Antón Alvarado John Paúl	
Título de la Investigación:	Migración de Datos Para el Proceso de Homologación de Ambientes de Calidad y Producción en la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales de la Sede Central de EsSalud	
Organización:	Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales	
Dirección:	Jirón Domingo Cueto 119 - Lince	
Levantamiento de información:	Periodo Pre test	01/07/2019 AL 31/12/2019
	Periodo Post test	01/01/2020 AL 30/06/2020
Variable en Estudio:	Proceso de Homologación de Ambientes de Calidad y Producción	
Proceso	Pre test Número de Actividades	Post test Número de Actividades
1	30	26
2	9	8
3	18	16
4	20	17
5	22	19
6	14	12
7	9	8
8	10	9
9	33	29
10	30	26
11	15	14
12	19	17
13	25	23
14	18	16
15	10	9
16	9	8
17	29	27
18	25	23
19	17	15
20	27	25
21	14	13
22	18	16
23	30	27
24	20	18
25	19	17
26	10	9
27	7	6
28	20	18
29	24	22
30	10	9
<b>Total</b>	<b>561</b>	<b>502</b>
<b>AUTORIZADO POR:</b>		
Apellidos y Nombres	Firma y Sello	
Kathy Ruiz Carrasco		

Anexo 2. Ficha de registro de Pre test y Post test de Número de actividades



## Declaratoria de Autenticidad del Asesor

Yo, Marín Verástegui, Wilson Ricardo, docente de la Facultad / Escuela de posgrado Ingeniería de Sistemas y Escuela Profesional / Programa académico Taller de Suficiencia Profesional de la Universidad César Vallejo Sede Lima (Filial Callao), asesor (a) del Trabajo de Investigación / Tesis titulada:

“Migración de Datos Para el Proceso de Homologación de Ambientes de Calidad y Producción en la Sub Gerencia de Sistemas Aseguradores, Subsidios y Sociales de la Sede Central de EsSalud”

del (los) autor (autores) Antón Alvarado, John Paúl, constato que la investigación tiene un índice de similitud de 24% verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender el trabajo de investigación / tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

Callao, 20 de mayo de 2021

Marín Verástegui, Wilson Ricardo	
DNI 45801046	Firma 
ORCID 000-0001-6175-8112	