



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Software

**Desarrollo e implementación de un sistema de
continuidad de negocio para las operaciones más
usadas en una entidad bancaria**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Software

AUTOR

Bryan Willians ROSAS QUISPE

ASESOR

Mg. Jorge Raúl DIAZ MUÑANTE

Lima, Perú

2022



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Rosas, B. (2022). *Desarrollo e implementación de un sistema de continuidad de negocio para las operaciones más usadas en una entidad bancaria*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Software]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Bryan Willians Rosas Quispe
Tipo de documento de identidad	DNI
Número de documento de identidad	72655073
URL de ORCID	https://orcid.org/0000-0002-3992-7546
Datos de asesor	
Nombres y apellidos	Jorge Raúl Díaz Muñante
Tipo de documento de identidad	DNI
Número de documento de identidad	07216161
URL de ORCID	https://orcid.org/0000-0001-5176-923X
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Mario Agustin Huapaya Chumpitaz
Tipo de documento	DNI
Número de documento de identidad	15386891
Miembro del jurado 1	
Nombres y apellidos	Víctor Hugo Bustamante Olivera
Tipo de documento	DNI
Número de documento de identidad	25655590
Miembro del jurado 2 (Asesor)	
Nombres y apellidos	Jorge Raúl Díaz Muñante
Tipo de documento	DNI
Número de documento de identidad	07216161
Datos de investigación	
Línea de investigación	No aplica

Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2022
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Software

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 19:00 horas del día 18 de agosto del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Huapaya Chumpitaz Mario Agustín (Presidente), Lic. Bustamante Olivera Víctor Hugo (Miembro) y el Mg. Díaz Muñante Jorge Raúl (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/osg-sdpp-cfc>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTINUIDAD DE NEGOCIO PARA LAS OPERACIONES MÁS USADAS EN UNA ENTIDAD BANCARIA”**, por el Bachiller Rosas Quispe Bryan Willians; para obtener el Título Profesional de Ingeniero de Software.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **17 (DIECISIETE)**.

A continuación el Presidente de Jurados el Mg. Huapaya Chumpitaz Mario Agustín, declara al Bachiller **Ingeniero de Software**.

Siendo las 19:40 horas, se levantó la sesión.

Presidente

Mg. Huapaya Chumpitaz Mario Agustín

Miembro

Lic. Bustamante Olivera Víctor Hugo

Miembro Asesor

Mg. Díaz Muñante Jorge Raúl



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Profesional de Ingeniería Software

INFORME DE EVALUACIÓN DE ORIGINALIDAD
Nº 026-EPISW-FISI-2022

1. Autoridad Académica que emite el Informe de Originalidad:	Directora de la Escuela Profesional de Ingeniería de Software
2. Apellidos y Nombres de la autoridad académica:	Dra. Nora Bertha La Serna Palomino
3. Operador del programa informático de similitudes:	Dra. Nora Bertha La Serna Palomino
3. Documento evaluado:	Tesis para Pregrado Título: “Desarrollo e implementación de un sistema de continuidad de negocio para las operaciones más usadas en una entidad bancaria”
5. Autores del documento:	Rosas Quispe, Bryan Willians
6. Fecha de recepción de documento	Recepción: 20/09/2022
7. Fecha de aplicación del programa detector de similitudes:	Revisión: 20/09/2022
8. Software utilizado:	Turnitin
9. Configuración del programa detector de similitudes:	Excluye textos entrecomillados: Sí Excluye biografías: Sí Excluye cadenas menores a 40 palabras: Sí Otro criterio (especificar): No
10. Porcentaje de similitudes según programa detector de similitudes	Diez por ciento (10%)
11. Fuentes originales de las similitudes encontradas	Se adjuntan en 01 (una) foja al presente informe
12. Observaciones:	Ninguna
13. Calificación de originalidad i. Documento cumple criterios de originalidad, sin observaciones ii. Documento cumple criterios de originalidad, con observaciones iii. Documento no cumple criterios de originalidad.	Documento cumple criterio de originalidad, sin observación
14. Fecha del Informe:	14/11/2022

Dra. Nora Bertha La Serna Palomino
Directora (e) de la EPISW

DEDICATORIA:

Dedico este trabajo y todos mis logros a mis padres Martin Rosas Guizado y Susana Quispe Polo, y a mi hermano Anthony Rosas Quispe, por su amor y su cariño que siempre me están dando, sin ellos no sería quien soy ahora.

AGRADECIMIENTOS:

A mi familia y seres queridos por siempre estar hay hasta en los momentos difíciles

A mí alma mater, la Universidad Nacional Mayor de San Marcos donde me formé y adquirí conocimientos que me han servido para ser el profesional que soy ahora.

A mi asesor por sus sugerencias y recomendaciones durante la elaboración del presente trabajo

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

Desarrollo e implementación de un sistema de continuidad de negocio para las operaciones más usadas en una entidad bancaria

Autor: Rosas Quispe, Bryan Willians
Asesor: Diaz Muñante, Jorge Raúl
Título: Trabajo de Suficiencia Profesional para optar para el Título Profesional de Ingeniero de Software
Fecha: Agosto 2022

RESUMEN

El presente trabajo describe el Desarrollo e implementación de un Sistema de Continuidad de Negocio para las operaciones más usadas en una entidad bancaria, para la realización del proyecto se hizo uso de la metodología RUP ya que se tenía claro el alcance, tiempo y presupuesto para este proyecto, también se hizo empleo de herramientas y buenas prácticas de desarrollo. El objetivo principal del proyecto es brindar una continuidad de negocio frente a una pérdida de comunicación con el mainframe ubicado fuera del país dándoles un servicio continuo a los clientes. Para la solución se hizo uso de schedulers de java para la carga periódica de información del mainframe principal a la base de datos alterna aquí en Perú, también se hizo uso del patrón MVC el cual permitió una mejor organización y claridad del código. Como consecuencia se logró tener un sistema que sirva de emergencia frente una caída del mainframe principal del banco, manteniendo la operatividad dentro la entidad bancaria y cumpliendo así con las normativas impuestas por la Superintendencia de Banca, Seguros y AFP (SBS).

Palabra claves: Sistema de Continuidad, Operaciones Bancarias, RUP, MVC, Scheduler

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF SYSTEMS AND INFORMATICS ENGINEERING

PROFESSIONAL SCHOOL OF SOFTWARE ENGINEERING

Development and implementation of a business continuity system for the most frequently used operations in a banking institution

Autor: Rosas Quispe, Bryan Willians

Advisor: Diaz Muñante, Jorge Raúl

Title: Professional Sufficiency Work to opt for the Professional Title of Software Engineer

Date: August 2022

ABSTRACT

This work describes the development and implementation of a Business Continuity System for the most used operations in a banking entity, for the realization of the project the RUP methodology was used since the scope, time, and budget for this project were clear, and the tools and good development practices were also used. The main objective of the project is to provide business continuity in the event of a loss of communication with the mainframe located outside the country, providing continuous service to customers. For the solution we made use of java schedulers for the periodic loading of information from the mainframe to the alternate database here in Peru, we also made use of the MVC pattern which allowed a better organization and clarity of the code. As a consequence, it was possible to have a system that could be used as an emergency backup in case of a crash of the main mainframe of the bank, maintaining the operability within the bank and complying with the regulations imposed by the Superintendence of Banking, Insurance and AFP (SBS).

Keywords: Continuity System, Banking Operations, RUP, MVC, Scheduler

ÍNDICE GENERAL

DEDICATORIA:	iii
AGRADECIMIENTOS:	iv
RESUMEN	v
ABSTRACT	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	x
CAPÍTULO I: TRAYECTORIA PROFESIONAL.....	3
1.1 Experiencia laboral	3
1.2 Formación académica.....	4
1.3 Cursos	5
1.4 Otros conocimientos	5
CAPÍTULO II: CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	7
2.1 Empresa – actividad que realiza	7
2.2 Visión	7
2.3 Misión	7
2.4 Organización de la empresa	7
2.5 Área, cargo y funciones desempeñadas	8
2.6 Experiencia realizada en la empresa	9
CAPÍTULO III: ACTIVIDADES DESARROLLADAS.....	10
3.1 Situación problemática.....	10
3.1.1 Definición del problema	10
3.2 Solución	13
3.2.1 Objetivo	15
3.2.2 Alcance.....	15
3.2.3 Etapas y Metodológica.....	17
3.2.4 Fundamentos Utilizados	18
3.2.5 Implementación de las áreas, procesos, sistemas y buenas prácticas..	29
3.3 Evaluación	45
3.3.1 Evaluación Económica.....	45
3.3.2 Interpretación del VAN y del TIR.....	46
CAPÍTULO IV: REFLEXIÓN CRÍTICA DE LA EXPERIENCIA.....	48

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....	49
5.1 Conclusiones	49
5.2 Recomendaciones	51
5.3 Fuentes de información.....	52
5.4 Glosario	54
ANEXOS	56
Anexo 1: Casos de uso del sistema	56
Anexo 2: Caso de uso login	57
Anexo 3: Documento técnico	59
Anexo 4: Modelo de procesos.....	60

ÍNDICE DE FIGURAS

Figura 1. Organigrama de la empresa	8
Figura 2. Normas de Regulación dadas por la SBS	10
Figura 3. Normativa de Infracciones Graves	12
Figura 4. Diagrama de Solución.....	14
Figura 5. Representación de patrón MVC	20
Figura 6. Proceso de compilación	22
Figura 7. Integración Continúa utilizando Jenkins	25
Figura 8. Estructura del proyecto PRONABEC	26
Figura 9. Diagrama de casos de usos para registro y cierre de reclamos	27
Figura 10. Posibles riesgos que podría sufrir el negocio financiero.....	28
Figura 11. Meet de la coordinación del Cronograma del aplicativo	29
Figura 12. Acta de reunión de revisión de los documentos	30
Figura 13. Arquitectura Final del Aplicativo	31
Figura 14. Prueba de concepto de la aplicación	32
Figura 15. Estructura del proyecto anterior	33
Figura 16. Tecnologías encontradas antes del desarrollo	34
Figura 17. Tecnologías actuales en desarrollo.....	34
Figura 18. Iteraciones en la etapa de desarrollo	35
Figura 19. Estructura del código del proyecto	36
Figura 20. Repositorio Bitbucket del proyecto	37
Figura 21. Flujo del Scheduler	38
Figura 22. Interfaz de carga de información del mainframe al aplicativo	39
Figura 23. Solución de Tickets.....	40
Figura 24. Consola de Jenkins.....	41
Figura 25. Análisis del inicio de aplicativo en SonarQube	42
Figura 26. Análisis final del aplicativo en SonarQube	42
Figura 27. T907 – Evidencia de pruebas	44
Figura 28. P110 - Hoja de pase a producción	45

ÍNDICE DE TABLAS

Tabla 1.	Experiencia Laboral	3
Tabla 2.	Formación Académica Profesional.....	4
Tabla 3.	Cursos	5
Tabla 4.	Otros conocimientos	5
Tabla 5.	Monto de pérdidas	13
Tabla 6.	Oficinas de Lima por Gerencia Territorial – Banca Minorista.....	17
Tabla 7.	Etapas del proyecto	17
Tabla 8.	Observaciones de las pruebas funcionales	43
Tabla 9.	Costos estimados del personal del proyecto	46
Tabla 10.	Costos Estimados Operativos del proyecto.....	46

INTRODUCCIÓN

En el presente trabajo se detalla el desarrollo e implementación de un sistema de continuidad de negocio para las operaciones más usadas dentro de una entidad bancaria con el objetivo de brindar una continuidad de los servicios frente a percances con el mainframe principal del banco.

En la actualidad muchas de las empresas son conscientes de lo vulnerable que pueden ser sus aplicativos ante siniestros, desastres naturales u otros riesgos que traigan como resultado la parálisis de sus negocios. Para ellos las empresas van optando por planes que les permitan continuar aún a pesar del percance sucedido, es ahí donde se lleva el contexto del presente proyecto a detallar.

Por ese motivo es que el banco quiere darle solución a la pérdida de comunicación con la infraestructura fuera del país que producía que clientes del banco no pudieran realizar sus operaciones en las oficinas de la entidad bancaria para lo cual se planteó construir un aplicativo web que fuera totalmente independiente del mainframe y que tuviera las operaciones más comunes que se realizan en el banco de esa manera cuando haya alguna falla de comunicación, poder mantener la operatividad dentro del banco sin afectar a los clientes. Unos de los retos más sobresalientes del proyecto fue que el sistema debía contener la misma información que el aplicativo host para ello se hizo uso de procedimientos en base de datos y scheduler que permitan la carga de información al presente aplicativo y a la vez permitan generar archivos de salida que serían expuestos al mainframe de Monterrey para su homologación de datos.

El presente trabajo se realizó con la siguiente estructura:

En el CAPÍTULO I se describe la información de la trayectoria profesional del presente autor, sus aprendizajes y sus experiencias dadas a través de su camino como programador, también se detallan sus conocimientos en los diferentes campos y cursos.

En el CAPÍTULO II se resume la función que brinda la empresa, en la que se realizó dicho trabajo, además de mencionar la visión, la misión y la organización de la empresa.

En el CAPÍTULO III se describe la situación problemática que se dio en la empresa, así como también se detalla la solución descrita (etapas, metodologías y evaluación económica).

En el CAPÍTULO IV se describe la reflexión crítica que narra la experiencia ganada de la solución propuesta en el trabajo de suficiencia profesional.

En el CAPÍTULO V se detalla la conclusión de dicho trabajo y las recomendaciones obtenidas tras la implantación de la solución.

CAPÍTULO I: TRAYECTORIA PROFESIONAL

El autor de dicho trabajo cuenta con un grado de Bachiller en Ingeniería de Software de la Universidad Nacional Mayor de San Marcos, con experiencia de 5 años en desarrollo e implementación de proyectos de TI. También posee conocimientos en metodologías ágiles como Scrum y RUP.

1.1 Experiencia laboral

A continuación, en la tabla 1 se detalla la experiencia laboral del autor:

Tabla 1.

Experiencia Laboral

Periodo	Desempeño
Julio 2021 – Presente	Empresa: Tata Consulting Cargo: Full Stack Engineer Tareas Realizadas: ▪ Mantenimiento de unos reportes de Pre-Evaluación con las tecnologías de IText y Java, para mantener comprensión del reporte en el área. ▪ Creación de un Sistema de Continuidad de Servicio con las siguientes tecnologías Spring MVC, JDBC, Hibernate, Mail, Quartz, y Java para realizar la continuidad de los servicios del banco. ▪ Mantener la calidad del código usando SonarQube para la validación estática del código para reducción de los bugs.
	Empresa: EsSalud Cargo: Full Stack Engineer Tareas Realizadas: ▪ Desarrollé un módulo procesos automáticos con las tecnologías de Maven, Lombok, Spring JDBC Template, Electrón y Bash para el área de prestaciones económicas en EsSalud, un área que gestiona el pago de los seguros de EsSalud, para acelerar el proceso de generación de expedientes. ▪ Desarrollé, gestioné, documenté y mantuve diferentes

Agosto 2016 – Julio 2021	módulos de prestaciones económicas de información crítica utilizando diferentes tecnologías Spring MVC, Struts, Angular, Spring Security, MyBatis, Oracle., para la mejora en sus procesos manuales para la gestión de expedientes y solicitudes en sus diferentes sistemas dentro EsSalud. ▪ Desarrollé procesos masivos en base de datos a través de Jobs en Oracle para procesos pesados para el área de prestaciones económicas en EsSalud, mejorando el rendimiento de la plataforma. ▪ Desarrollé sistemas de reportes a través Jasper Report y POI para mostrar el detalle de los resultados de las solicitudes transaccionadas a través de la plataforma mejorando el proceso de auditoría.
Junio 2015 – Julio 2016	Empresa: Kodevian Studio Cargos: Web System Developer Tareas Realizadas: ▪ Mantuve un sistema de compras con las tecnologías Node.js, Angular.js, Spring Framework y MongoDB para la empresa Rosaclaun, una empresa dedicada a la entrega de flores con dedicatorias especiales para mejorar su rendimiento. ▪ Desarrollé y mantuve una aplicación interna para el control de la entrega de los pedidos para la empresa Rosaclaun con las tecnologías Node.js, Ionic y MongoDB mejorando el proceso de entrega.

Nota: Formación profesional del autor.

Fuente. Elaboración propia

1.2 Formación académica

A continuación, en la tabla 2 se detalla la formación académica profesional:

Tabla 2.

Formación Académica Profesional

Tipo	Institución	Nivel alcanzado	Profesión/ Especialidad	Años de Estudio
Formación	Universidad Mayor	Bachiller	Ingeniería de	2011-2017

Universitaria	de San Marcos		Software	
----------------------	---------------	--	----------	--

Nota: Formación del autor.

Fuente. Elaboración propia

1.3 Cursos

A continuación, en la tabla 3 se detalla los cursos realizados:

Tabla 3.

Cursos

Curso	Institución	Año
Java Developer	CEPS-UNI	2016
Android Básico	CIBERTEC	2017
Android Avanzado	CIBERTEC	2018
Java Básico	CIBERTEC	2018
Java Web Developer	CIBERTEC	2018
Java Architect Developer	CIBERTEC	2019

Nota: Detalle de los cursos.

Fuente. Elaboración Propia

1.4 Otros conocimientos

A continuación, en la tabla 4 se detalla los otros conocimientos que posee el autor del presente trabajo:

Tabla 4.

Otros conocimientos

Tema	Conocimientos	Detalle de conocimientos
Lenguaje de Programación		Java, Javascript, TypeScript
Sistemas Operativos		Microsoft Windows, Mac, Linux
Base de Datos		MongoDB, Redis, MySQL, Oracle, Postgres

Tecnologías		Java: Spring Framework, Spring boot, Struts, MyBatis, PrimeFaces, JPA, Hibernate, Quartz, Spring Batch, Spring MVC, Log4j, Eureka., JasperReport.
		Mobil: Android, Ionic, Flutter
		Frontend: React, Angular, JQuery
Servidores		JBoss, Apache Tomcat
Herramientas de Desarrollo	de	Intelij IDEA, Netbeans, Spring Tool Suite, Visual Studio, Atom
Herramientas de Control de Versiones	de	Git (Gitlab, Bitbucket, GitHub)
Herramientas CI - CD		Jenkys Travis
Herramientas de Modelado	de	UML, StarUML
Metodologías		Scrum, Kanban

Nota: Otros conocimientos que tiene el autor.

Fuente. Elaboración Propia

CAPÍTULO II: CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1 Empresa – actividad que realiza

El banco es un grupo financiero con más de cien años en el mercado brindando servicios de Salud Financiera, Cheque de Gerencia, Giros y Transferencias y CCI Cuenta Interbancaria. Cuenta con sedes en diferentes partes del mundo de entre los que se destaca EE. UU., México, Turquía y gran parte de América del Sur.

Se trata de una empresa firmemente comprometida con el futuro y en la constante adaptación a las necesidades del mercado. La empresa se centra en crear un futuro mejor para todos, esforzándose por construir relaciones a largo plazo con sus clientes para cambiar la forma en que las personas entienden los negocios. Gracias a este trabajo, el banco se ha convertido en un referente mundial y uno de los bancos más reconocidos y premiados del mundo.

2.2 Visión

El banco tiene como visión otorgar un futuro mejor a las personas en las sociedades y países donde actúa, aspirando como empresa tener una identidad y posicionamiento de la mejor marca siendo esta una referencia de comportamiento empresarial responsable.

2.3 Misión

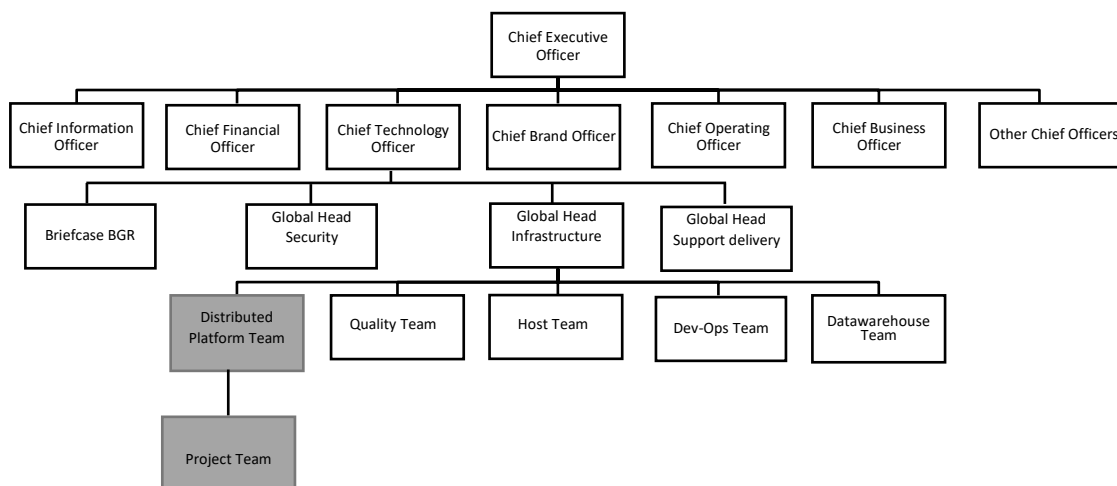
El banco tiene como misión aportar soluciones a sus clientes dándoles un crecimiento rentable y sostenible otorgando un progreso en la sociedad en la que está presente.

2.4 Organización de la empresa

De acuerdo con el organigrama general del Banco, el autor del presente trabajo señala el área donde se realizó su experiencia profesional. Como se muestra en la figura 1.

Figura 1.

Organigrama de la empresa



Nota: El gráfico representa el organigrama del banco.

Fuente. Elaboración propia

2.5 Área, cargo y funciones desempeñadas

El autor de este trabajo trata de la infraestructura en el grupo de Plataformas Distribuidas.

El banco define su posición según dos conceptos: rol y nivel. Los roles determinan la especialización de un empleado y el nivel determina el nivel de conocimiento y experiencia de ese empleado.

El autor actual del trabajo es un desarrollador de Java desde julio de 2021 hasta el presente.

El banco para el rol de Java Developer se define en los siguientes niveles:

- Trainee
- Junior
- Analyst Software Engineer
- Associate Software Engineer
- Manage Software Engineer

El presente autor del trabajo se le otorgó el nivel Analyst Software Engineer en el 2021 hasta la actualidad posee dicho nivel.

Según banco, las funciones de un Java Developer con el nivel de Analyst Software Engineer son las siguientes:

- Diseñar y desarrollar soluciones técnicas en base a especificaciones técnicas.
- Análisis previo al inicio del desarrollo
- Elaboración de la documentación previa al desarrollo del proyecto, así como realización de las modificaciones posteriores que sean necesarias.
- Desarrollo y mantenimiento de software de aplicación propia.
- Resolución de incidencias en el software propio
- Investigación y estudio de nuevas tecnologías.

2.6 Experiencia realizada en la empresa

El autor del trabajo, presenta el rol de Java Developer cumpliendo las siguiente funciones en la empresa:

- Diseñar e implementar soluciones tecnológicas, según los estándares de calidad y buenas prácticas.
- Realizar mantenimiento a software distribuidos.
- Elaborar documentación de los aplicativos previos al desarrollo.
- Realizar cursos de capacitación mensual.
- Investigar sobre nuevas tecnologías.

CAPÍTULO III: ACTIVIDADES DESARROLLADAS

3.1 Situación problemática

Los aplicativos cliente servidor son los sistemas más recurrentes dentro de un banco ya que son los menos vulnerables ante ataques cibernéticos. Pero eso no quiere decir que no sean propensos a fallos en el mainframe central, tales como falta de conexión a la base de datos, problemas en la red, virus o troyanos, problemas en hardware, entre otros.

Para ello, en el 2011 se llevó a cabo un proyecto con el fin de solucionar estos inconvenientes que afectarían la operabilidad dentro de las oficinas de la entidad bancaria. Este proyecto fue dirigido por una consultora de software reconocida a nivel mundial. La solución que se propuso en ese entonces fue la realización de un proyecto web que tendría conexión con un nuevo servidor en Perú. El proyecto como tal se entregaría a finales de ese año, pero por temas de presupuesto y gestión no se llegó a culminar el proyecto. La solución que se propuso en ese entonces se llegó a discontinuar con el tiempo, lo cual hizo que no pudiera ser mantenido por los servidores de aplicaciones recientes en el mercado.

En el 2021 tras un incidente que ocurrió en el banco, sus aplicaciones dejaron de funcionar durante varias horas; ocasionando que los clientes no pudieran usar los servicios web, ni móvil. Ese fue el incentivo que motivó nuevamente a retomar el proyecto de Sistema de continuidad de negocio en el Perú.

3.1.1 Definición del problema

En el banco donde se llevó a cabo la experiencia profesional, contaba con un sistema basado en una arquitectura cliente servidor; el cual se conectaba a un mainframe central fuera del país para el procesamiento de las operaciones y si en caso se perdiera la comunicación con la central, los servicios brindados por el banco caerían, quedando inoperativa la atención en las oficinas y generando el incumplimiento de las normas proporcionadas por la Superintendencia de Banca, Seguros y AFP (SBS) que obligan a las empresas a tener una continuidad de negocio frente algún percance como se aprecia en la figura 2.

Figura 2.

Normas de Regulación dadas por la SBS

SBS 2021: Renovación ASA	Regulación Específica
 Resolución SBS N° 2115-2009 Reglamento para el Requerimiento de Patrimonio Efectivo por Riesgo Operacional Renovación de Autorización para uso del Método Estándar Alternativo (ASA - Alternative Standard Approach): Artículo 8°.- "j. La empresa deberá implementar un sistema de gestión de la continuidad del negocio conforme a la normativa vigente, que tenga como objetivo asegurar un nivel aceptable de operatividad de sus procesos críticos, ante eventos que puedan afectar la continuidad de sus operaciones."	 Resolución S.B.S. N° 2116 -2009 - Reglamento para la Gestión del Riesgo Operacional Artículo 14°.- Subcontratación Con el fin de gestionar los riesgos operacionales asociados a la subcontratación, las empresas deberán establecer políticas y procedimientos apropiados para evaluar, administrar y monitorear los procesos subcontratados.  CIRCULAR SBS G- 139 -2009 - Gestión de la Continuidad del Negocio. Artículo 3º "... implementar respuestas efectivas para que la operatividad del negocio de la empresa continúe de una manera razonable, ..., ante la ocurrencia de eventos que pueden crear una interrupción o inestabilidad en las operaciones de la empresa." Artículo 8º "... asegurarse que sus principales proveedores de servicios cuenten con planes de continuidad." Artículo 10º.- "Las empresas analizarán el impacto que tienen los cambios significativos sobre la continuidad del negocio."

Nota: Normas del SBS sobre la continuidad de negocio.

Fuente. (Entidad Bancaria)

El incumplimiento de estas normas trae como consecuencia el pago de una multa que ronda el rango de 100 UIT a 80 UIT siendo considerada esta como una infracción grave ante SBS como se aprecia en la figura 3.

Figura 3.

Normativa de Infracciones Graves



16)	No informar o informar en forma incompleta o con errores la información relativa a la apertura, traslado, cierre de sus oficinas y locales compartidos, de acuerdo a la normativa vigente.
17)	No constituir y mantener actualizados los archivos de información de cada uno de sus agentes.
18)	Incumplir las disposiciones sobre funcionamiento establecidas en el Reglamento.

II. INFRACCIONES GRAVES

1)	Incumplir las disposiciones normativas relativas a la emisión, administración y/o seguridad aplicables a las tarjetas de crédito y/o débito y/o a su información.
2)	Utilizar modelos de contratos con cláusulas generales de contratación que no hayan sido aprobadas previamente conforme a la normativa vigente.
3)	Incluir en los contratos utilizados, sin aprobación previa de la Superintendencia, cláusulas generales de contratación aprobadas para otras empresas.
4)	Utilizar en los contratos cláusulas identificadas como abusivas por la Superintendencia.
5)	Incumplir las disposiciones normativas vigentes relacionadas con el otorgamiento de financiamientos a directores o trabajadores, o a los cónyuges o parientes de estos. <u>Sanción específica:</u> Multa equivalente al cien por ciento (100%) del monto total del exceso o de la valoración de las condiciones más ventajosas.
6)	Presentar discrepancias en la clasificación de la cartera de créditos corporativos, a grandes empresas y a medianas empresas en más del 20% y hasta el 30% de una muestra significativa de por lo menos 40 deudores revisada por la Superintendencia. El porcentaje de discrepancias se calculará en función al ratio "número de deudores discrepantes/total de deudores revisados".
7)	Incumplir las disposiciones normativas relativas a la evaluación y clasificación del deudor y la exigencia de provisiones establecidas en las normas de la materia que determine una desviación del cálculo de provisiones requeridas respecto de las constituidas como mínimo del 10%.
8)	Presentar discrepancias en la clasificación de la cartera de créditos de Pequeña Empresa, Microempresa, de Consumo y/o Hipotecarios en más del 10% y hasta el 20% del saldo de la cartera de créditos revisada por la Superintendencia, siempre y cuando alguna de las líneas de negocio revisadas, de manera individual o a nivel agregado, representen más del 20% del saldo total de la cartera de la entidad.
9)	No cumplir con el horario mínimo de atención al público conforme a lo establecido en el artículo 139 de la Ley General.
10)	No divulgar las memorias anuales o incumplir con la obligación de publicar los estados financieros en el Diario Oficial y en uno de extensa circulación nacional, cuando menos cuatro veces al año, en las oportunidades y con el detalle que establece la Superintendencia.

Nota: Listado de infracciones leves y graves del aplicativo.

Fuente. (SBS, 2021)

Otras de las consecuencias que traería la inoperatividad de las operaciones, es que no se podrían llevar a cabo las operaciones que generan ganancias al banco (como es el pago de las tarjetas de crédito) y otras que generan pérdidas y desprestigio (como el bloqueo de tarjetas) si no se realizan a tiempo.

En la tabla 5 se muestra la estimación de la pérdida generada de un 0,1 % del total de clientes que tiene el banco que han realizado operaciones de un mínimo de 100 soles.

Tabla 5.

Monto de pérdidas

Total de Clientes	Cantidad de Transacción (soles) usuarios realizando operaciones (%) (por hr)	Total de Pérdida (soles)
80,700,000	0.1%	100
		S./ 8,070,000

Nota: Montos de pérdidas realizadas en una hora.

Fuente. Elaboración Propia

Lo señalado anteriormente fue lo que motivó la reactivación del proyecto de continuidad de negocio en el Perú nuevamente.

3.2 Solución

Para la solución se planteó la realización de un aplicativo web que se conectó a una base de datos local, en un servidor en Perú; de esa manera, se seguirá operando ante una pérdida de comunicación del mainframe fuera del país.

Entre las operaciones básicas que se propuso en el nuevo sistema son las siguientes:

- Apertura y Cierre de Caja
- Retiro en efectivo
- Depósitos en efectivo
- Consulta de Saldos
- Pago de cheques propio banco
- Depósito a cheque al propio banco
- Pago de tarjeta

- Bloqueo de tarjeta
- Acceso / Reportes

Con el fin de seguir los estándares planteados se examinó la documentación anterior que se tenía del proyecto, así como el código fuente de ese entonces. Tras la examinación del proyecto se planteó incorporar un nuevo proyecto; actualizando los componentes como Java 11, Spring MVC 5, Hibernate 5.0 y Quartz 2.5. Este último componente para la automatización de la carga de información del mainframe central fuera del país a Perú. El proyecto también contará con un aplicativo batch en cobol que tendrá como función principal la carga de información desde el aplicativo web al mainframe fuera del país.

La mayor complicación de la solución fue la impresión de los tickets que se generaban en cada operación, debido a que la anterior solución estaba realizada con applets, el cual ya no es admisible para los navegadores actuales; por ello, se realizó una coordinación con la gente del banco de México para poder idear una solución en conjunto.

En la siguiente figura 4, se aprecia un diagrama de la solución, si en caso se perdiera la conexión con el servidor primario. Se usaría un servidor alternativo de Monterrey que en conjunto con el sistema de continuidad de negocio y su base de datos propia en Perú permitirían continuar con las operaciones bancarias dentro de la entidad bancaria.

Figura 4.

Diagrama de Solución



Nota: Diagrama de la solución ante una pérdida de comunicación con el mainframe central.

Fuente. (Entidad Bancaria)

La solución con más detalle se describe en las siguientes secciones de este capítulo.

A continuación, se detallan los objetivos del presente trabajo:

3.2.1 Objetivo

Objetivo general

Desarrollar e implementar un sistema de continuidad de negocio para las operaciones bancarias más usadas dentro de una entidad bancaria, resolviendo inoperatividad en las oficinas de Lima ante una posible pérdida de comunicación con la infraestructura fuera del país.

Objetivos específicos

OE1: Cumplir con las normativas dadas por el SBS.

OE2: Analizar el código que se tenía del anterior proyecto.

OE3: Tener un aplicativo moderno que cumpla con los estándares de desarrollo del banco.

OE4: La aplicación de buenas prácticas usando la metodología RUP

OE5: Elaborar documentos para el pase a producción y actualizar los documentos de diseño de la aplicación tales como: Documento de Componentes, Documento de Pruebas, Documento de Análisis, Documento de Diseño y Manuales de usuario.

3.2.2 Alcance

A continuación, se detalla los alcances del presente trabajo:

Alcance funcional

El presente trabajo cubrió el desarrollo e implementación de un sistema de continuidad de negocio para las operaciones bancarias más usadas dentro una entidad bancaria, en el cual se establecieron los siguientes alcances:

Acceso al Sistema

- Se deberá desarrollar e implementar un portal de acceso al sistema, con usuario y contraseña la cual estará conectado con el repositorio de datos local.
- En caso el usuario no se encuentre en el repositorio de datos local se deberá acceder un repositorio externo de usuarios del banco.
- Se debe obtener la información del código de la oficina y el nombre del usuario que ingresó en el sistema.

- Tras acceder al sistema se debe mostrar una ventana donde se ingresa el terminal del cual se está haciendo uso.
- El asesor del banco no podrá ingresar al sistema si no tiene una previa autorización del subgerente o gerente de su oficina.

Módulo de Seguridad

- Se desarrollará e implementará la creación de los perfiles, permisos de autorización, opciones del sistema.
- La información de este módulo solo será accesible por el usuario administrador.

Módulo de Administración

- Se desarrollará e implementará la configuración de archivos, envío de archivos, configuración de parámetros, inicio y término de aplicación.
- La información del módulo de administración sólo será accesible por el usuario subgerente o gerente y administrador.

Módulo de Operaciones

- Se desarrollará e implementará la Apertura de Caja, Bloqueo de Tarjeta, Cierre de Caja, Cierre de oficina, Confirmación de traspaso, Consulta saldo, Extorno, Operación y Traspaso.
- La información de este módulo a excepción del Cierre de oficina solo será accesible por el asesor del banco.
- La información del cierre de oficina será accesible por el usuario subgerente o gerente.

Módulo de Reportes

- Se desarrollará e implementará el reporte de Estado de la oficina, Faltantes y Sobrantes, Log de Transacciones, Operaciones, Rechazos de Procesos, Situación de Caja, Tarjetas Bloqueadas, Transacciones Significativas.
- La información de este módulo podrá ser accesible por el asesor y por el subgerente.

Alcance organizacional

El usuario final de la solución serán los empleados de las sucursales de los distintos bancos ubicados en la organización, el número de oficinas se describe a continuación.

Las oficinas de banca minorista en Lima se muestran en la Tabla 6.

Tabla 6.

Sucursales de Lima de la Banca Minorista

Oficinas	Total
Oficina Principal	1
Sucursal Lima Centro	26
Sucursal Lima Residencial	20
Sucursal Surco – La Molina	25
Sucursal Lince	23
Sucursal Callao – San Miguel	25
Sucursal Miraflores	24
Total	144

Nota: Oficinas en lima

Fuente. (Entidad Bancaria)

3.2.3 Etapas y Metodológica

Para el desarrollo de la experiencia profesional se utilizó la metodología RUP que significa Proceso Unificado Racional, la cual consiste en varias iteraciones del producto con el fin de asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios finales, dentro de unos tiempos y presupuesto predecibles.

A continuación, en la tabla 7 se detallan los entregables y herramientas por cada iteración del proyecto:

Tabla 7.

Etapas del proyecto

Etapas	iteración	Entregables	Herramientas
Inicio	iteración 1	Plan de Proyecto	Jira
Planificación	iteración 1	Documento de Análisis	StarUML

	iteración 2	Documento de Diseño	
		Documento de Pruebas	
	iteración 3	Análisis del código anterior	
Construcción	iteración 1	Construcción de operaciones y seguridad	Git Spring MVC Quartz
	iteración 2	Construcción de reportes y administración	SonarQube
Transición	iteración 1	Manual de usuario Plan de instalación	Jira

Nota: Describe los entregables y Herramientas usadas durante el Desarrollo.

Fuente. Elaboración Propia

3.2.4 Fundamentos Utilizados

3.2.4.1 RUP

El Proceso Unificado Racional (a menudo llamado RUP) es un proceso de desarrollo de software, es una la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos establecidos, sino un conjunto de metodologías que pueden adaptarse al contexto y necesidades de cada organización. Conocido con este nombre, es un software desarrollado por Rational y ahora propiedad de IBM que consiste en una información interconectada de varios artefactos y descripciones de varias operaciones.

Según:

Silvestre Mercedes (2018) señala que el proceso unificado suministra un enfoque de asignar tareas y responsabilidades dentro de una organización.

Ciclo de vida

Las características del ciclo de vida de RUP son las siguientes:

Basado en casos de uso: los casos de uso se capturan a medida que el negocio se modela y se representan a través de requisitos. Estos requisitos representan las necesidades que tiene el usuario sobre el sistema.

Dirección arquitectónica: La arquitectura es la visión general del sistema el cual el equipo de desarrollo y los usuarios acordaran, describir los elementos más importantes del modelo que se construirá. Conózcalo, desarrolle y produzca económicamente. Los RUP se desarrollan en varias iteraciones, comenzando con las CU específicas de la arquitectura. Los modelos arquitectónicos están representados por vistas que contienen diagramas UML.

Iterativo e incremental: los iterativos incluyen actividades de todos los procesos comerciales, pero básicamente abarcan algunos procesos más que otros.

Fases

En el RUP se tienen 4 fases fundamentales:

La **fase de inicio**, la visión, los objetivos y el alcance técnico del proyecto se definen durante esta fase. Esta fase también incluye el modelado de negocios y el análisis de los requisitos del proyecto. Al final de esta fase, se determina una lista de casos de uso y riesgos asociados con el proyecto.

La **fase de elaboración** el objetivo de esta fase es completar el análisis de casos de uso y determinar la arquitectura del sistema, para obtener aplicaciones ejecutables que cumplan con los casos de uso que ponen en peligro el sistema. Aunque partes del sistema se desarrollan en detalle, las decisiones arquitectónicas se basan en la comprensión de todo el sistema y los requisitos (funcionales y no funcionales) se definen por un alcance definido.

La **fase de construcción** consiste en un ciclo iterativo en el que los casos de uso se integran uno tras otro, dependiendo de los factores de riesgo del proyecto. Permitiendo obtener una versión temprana del sistema para cubrir casos de uso clave. Los cambios en los requisitos no se incorporarán hasta el comienzo de la siguiente iteración.

La **fase de transición** comienza con la versión "beta" del sistema y finaliza con el sistema de producción.

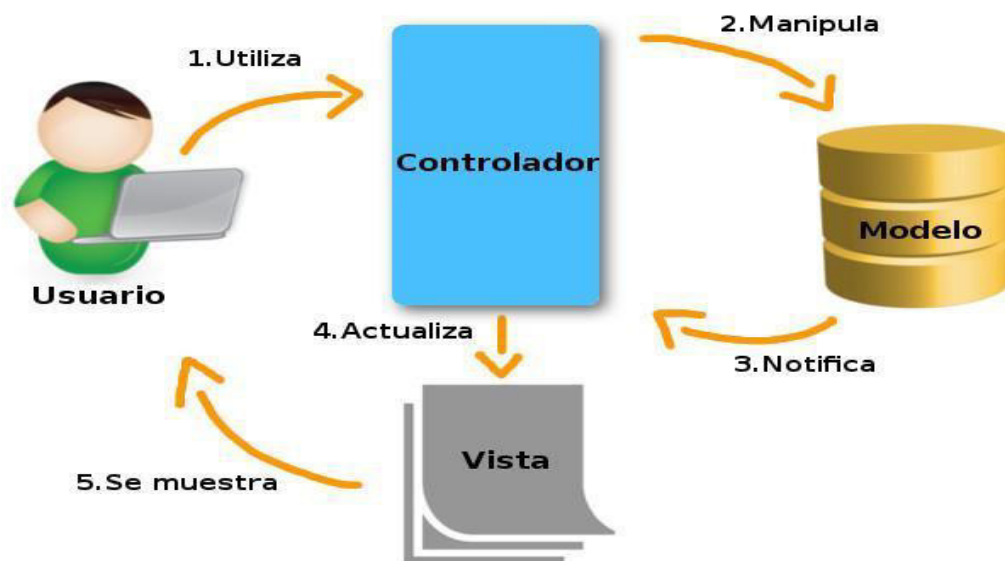
3.2.4.2 MVC

MVC es un patrón, plantilla o guía de arquitectura que muestra cómo se organizan y estructuran los componentes de un sistema de software, con se llevan a cabo las responsabilidades y las relaciones que existe en cada componente.

Luna Quinto (2021) señala que el patrón MVC aísla aplicación en partes individuales para poder desarrollar de forma aislada sin preocuparse de dañar otras partes del aplicativo, mejorando la testeabilidad y mantenimiento del este.

La arquitectura MVC, independientemente de la tecnología o el entorno en el que se desarrolla el sistema, sugiere separar los componentes de la aplicación en tres grupos principales (o clases: modelos, vistas y controles) y describe cómo se vinculan entre sí para mantener una arquitectura limpia y organizada. con un acoplamiento mínimo en cada una de las diferentes clases como se puede apreciar en la figura 5.

Figura 5.
Representación de patrón MVC



Nota: Representa el diagrama del patrón MVC.

Fuente. (Platzi, 2021)

Modelo

A nivel de modelo siempre podemos encontrar representaciones de datos de dominio, entidades que almacenan información sobre el sistema que estamos desarrollando. Si nuestra aplicación es parte de un sistema distribuido, es decir, utiliza servicios proporcionados por otros sistemas, el modelo que contiene las clases de objetos de transferencia de datos (DTO) les permitirá intercambiar información con ellos.

De igual manera, encontramos la lógica de negocios de la aplicación que nos otorga el poder de administrar entidades de dominio. Por lo tanto, el encargado de garantizar que el sistema se mantenga en un flujo constante y completo en todo momento. En pocas palabras, el modelo contiene básicamente objetos que comprometen a la lógica comercial y al dominio de nuestro sistema.

Vista

Los componentes de visualización son los encargados de crear la apariencia de nuestra aplicación, los distintos resultados que componen las pantallas, páginas o sistemas que pueden ser utilizados por los usuarios o clientes. De hecho, a menudo se hace referencia a una vista como la vista del modelo de estado en un momento dado y en el contexto de la actividad.

Cuando las vistas son parte de la aplicación, deben estar compuesto de elementos interactivos que brindan al usuario el poder enviar información e iniciar movimiento en el sistema, tales como botones o cualquier otro elemento que pueda adaptarse fácilmente a la tecnología. Para las aplicaciones web, las vistas suelen contener componentes que generan marcas para que la página se distribuya a los usuarios.

Por tanto, en esta interfaz encontramos los componentes encargados de crear una interfaz que tiene la apariencia, normalmente, pero no exclusivamente, de la interfaz de usuario de nuestra aplicación.

Controlador

La tarea principal del componente controlador es mediar la relación con el usuario y el sistema. Deberían poder capturar su acción en la vista, como presionar un botón o seleccionar un elemento de menú, interpretarlo y ejecutarlo. Por ejemplo, el sistema regresa el estado actual del sistema o invoca operaciones definidas por el modelo para leer o renovar la información. Además, realizan tareas de conversión de datos a los elementos del modelo y la vista para que lleguen a un entendimiento entre sí. Por lo tanto, traducen la información

despachada desde la interfaz, como los valores de los campos de formulario recibidos a través de HTTP, en objetos que son comprensibles para el Modelo, como una implementación de entidad de clase o dominio. Nuevamente, el administrador toma información del modelo y la adapta al formato o estructura de datos en la que puede operar la vista. Por todas estas razones, podemos pensar en el controlador como un coordinador general del sistema que regula la navegación y el flujo de información con el usuario, y actúa como intermediario entre la capa de vista y el modelo.

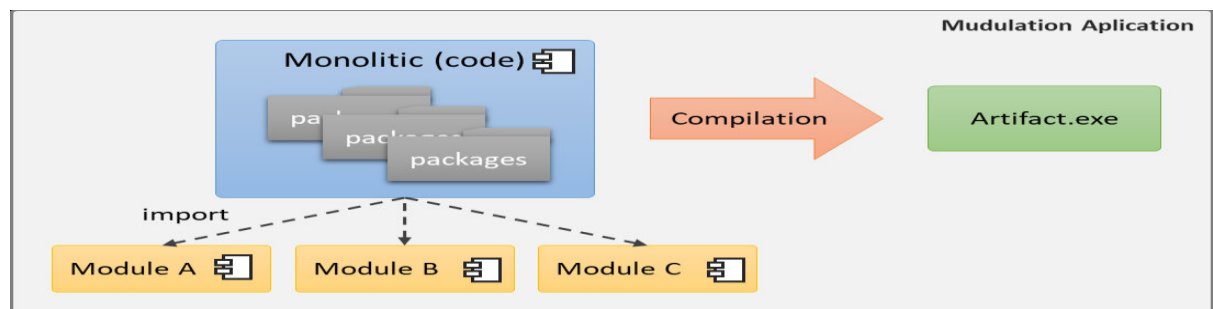
3.2.4.3 Arquitectura Monolítica

El estilo arquitectónico monolítico implica tener una aplicación independiente que lleve absolutamente toda la funcionalidad necesaria para llevar a cabo la tarea para la cual fue diseñada, sin usar dependencias externas que adicionen funcionalidad (Oscar Blancarte, 2020).

Un monolito puede consistir en una sola pieza de software compuesta por muchos módulos o bibliotecas empaquetadas en una sola pieza de software como se puede distinguir en la figura 6.

Figura 6.

Proceso de compilación



Nota: Descripción de aplicativo monolítico,

Fuente. (Introducción de Arquitectura de Software, 2020)

Características

Las características solo se pueden convertir en ventajas o desventajas cuando analizamos el problema que pretendemos resolver, mientras que estas son solo características:

1. Estas son aplicaciones independientes (no necesitan nada para ejecutarse).
2. Hacen todo el trabajo de principio a fin para completar la tarea.

3. Suelen ser aplicaciones grandes, aunque no obligatorias.
4. Por regla general, se trata de almacenes de datos privados, es decir, cada instalación gestiona su propia base de datos.
5. Todo el sistema se ejecuta en una plataforma.

3.2.4.4 Scheluders

Scheduler está diseñado para programar un subproceso o una tarea para que se ejecute durante un período de tiempo específico o periódicamente durante intervalos fijos. Hay varias formas de programar tareas de Java.

- Biblioteca Java TimerTask
- ScheduledExecutorService
- Quartz
- TaskScheduler

Quartz es un marco de programación de código abierto que proporciona funciones avanzadas de programación de tareas en Java. Éstos incluyen:

- Cualquier tarea escrita en Java se puede programar para ejecutarse en un marco.
- Podemos programar tareas donde especificamos solo la fecha y hora de vencimiento y desde allí especificamos el intervalo de ejecución (por ejemplo, ejecutar la tarea el 1 de enero a las 00:00 y hacerlo cada 3 días a partir de ahora) para programar una tarea usando una expresión Cron (por ejemplo, ejecute la tarea el segundo lunes del mes de enero a septiembre de 2008, a partir de las 8:00 am cada 15 minutos. hasta las 10:00 horas).
- Habilidad para trabajar en un ambiente de clusters.

3.2.4.5 SonarQube

SonarQube es una plataforma de calidad de código gratuito y de código abierto. Que brinda en tiempo real la calidad de tu código actual, así como lo que salió mal y lo que probablemente salga mal a futuro otorgando puntuaciones del porcentaje de código que hay que corregir.

Es una guía perfecta para los equipos de desarrollo durante la examinación de código en la fase testing y auditoría dentro del ciclo de desarrollo de la aplicación.

Características

- Permite los lenguajes de programación más populares.
- Se adecua con toda la cadena de herramientas de DevOps.

- Permite realizar exámenes automáticos de código estático para la detección de problemas que podrían afectar el código.
- Genera métricas y gráficos que ofrecen información objetiva de la calidad de los proyectos.

3.2.4.6 Jenkins

Jenkins es un servidor open source para la integración continua. Es usado para probar proyectos de software de forma continua, lo que permite a desarrolladores integrar sus cambios en el proyecto y entregar sus versiones mucho más rápido.

¿Qué puede hacer Jenkins?

Con Jenkins podemos automatizar cientos de tareas que nos ayudarán a reducir el tiempo de entrega de nuestros productos. Con esta herramienta podemos:

- Notificar al equipo de registro de errores correspondiente.
- Generar documentación del proyecto.
- Implementar cambios de código aprobados.
- Automatice el desarrollo y las pruebas de software.

¿Cómo funciona Jenkins?

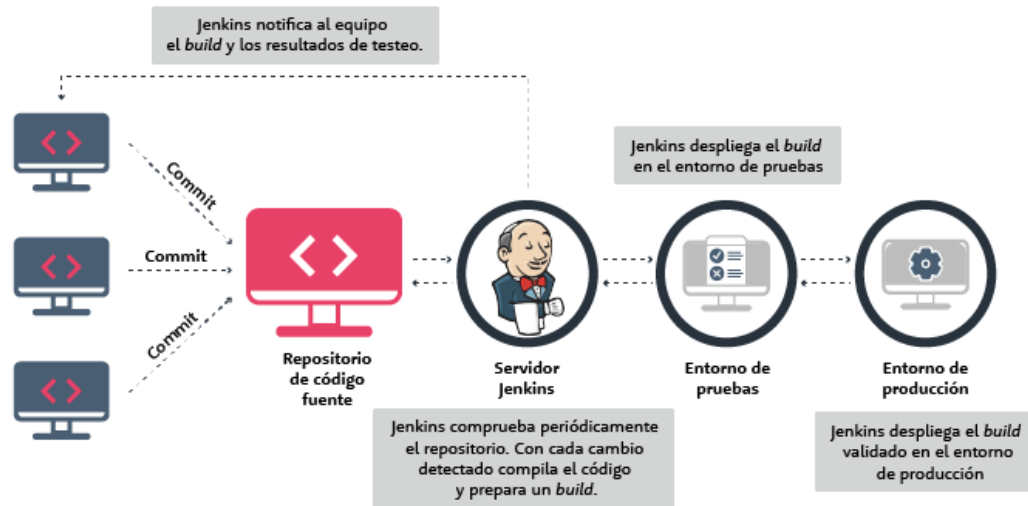
A continuación, se describe cómo utilizar esta herramienta en el proceso de integración.

- Los desarrolladores envían el código fuente.
- El servidor Jenkins comprueba periódicamente si hay cambios en el repositorio.
- Después de una confirmación, Jenkins captura los cambios en el código fuente. Ejecuta el código y se prepara para construir. Esto alertará al equipo si la construcción falla bajo cualquier circunstancia. Si tiene éxito, los cambios se envían al servidor.
- Después de la inspección. Jenkins genera informes y comunica la compilación al equipo.
- Jenkins continúa revisando frecuentemente y repetido nuevamente el proceso

El proceso descrito anteriormente se muestra en la figura 7.

Figura 7.

Integración Continúa utilizando Jenkins



Nota: Describe el flujo de proceso del aplicativo Jenkins.

Fuente. (Sentry, 2021)

3.2.4.7 Bitbucket

Es una herramienta de colaboración y alojamiento de proyecto tanto públicos como privados, está basado en git y entre sus principales características tiene:

- La creación y gestión de repositorios
- Conectarse usando Autenticación de Dos Factores
- Hacer peticiones de extracción
- Realizar revisiones de código
- Usar la edición en línea y el soporte de Markdown
- Realizar el seguimiento de los problemas
- Integración directa con Jira Bambú y Jenkins

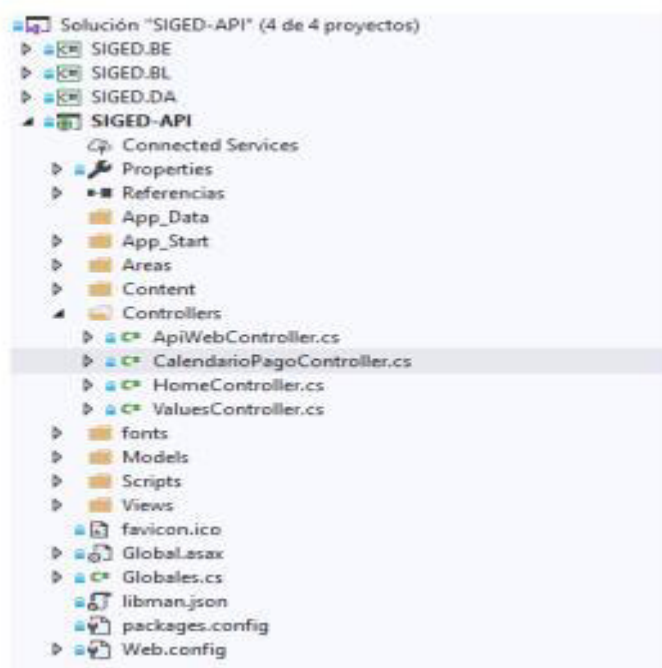
3.2.4.8 Trabajos relacionados a la presente investigación

“Automatización del proceso aprobar calendario de pago en la unidad de subvenciones y financiamiento PRONABEC” (Diego Luna, 2021)

El autor del presente trabajo detalla la implementación de sistema para aprobar los pagos en la unidad de subvenciones PRONABEC usando el patrón MVC detallando las ventajas que tiene utilizarlo en proyecto de software.

A continuación, en la figura 8 se observa la estructura de carpeta que uso para la elaboración del sistema dividiéndolas en: models, controllers y views. Siendo models donde se encuentra la lógica de acceso a datos, controller quien contiene la lógica de la aplicación y views el cual contiene la representación visual del aplicativo.

Figura 8. Estructura del proyecto PRONABEC



Nota: Estructura de carpetas del proyecto PRONABEC

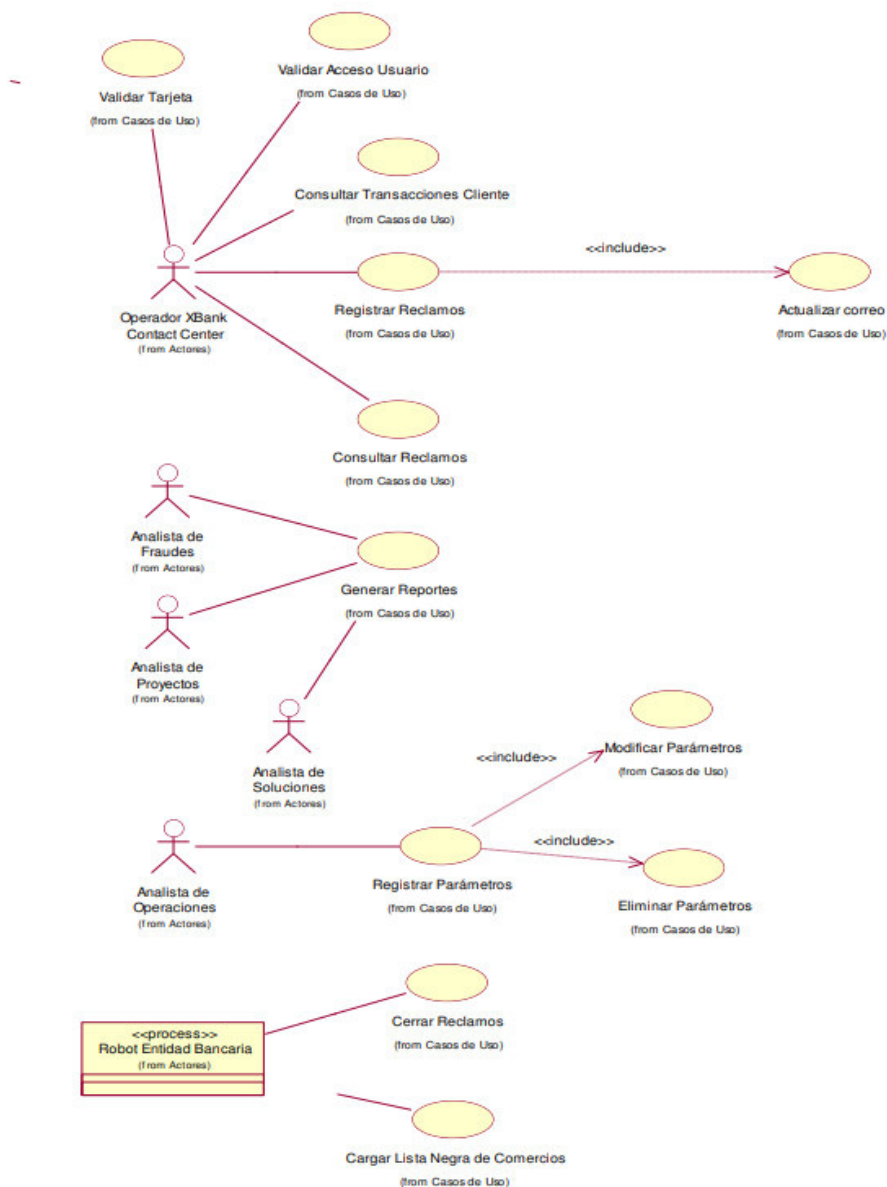
Fuente. (Diego Luna, 2021)

“Implementación de un proceso automático para el registro y cierre de reclamos en una entidad bancaria utilizando metodología RUP y la herramienta Genexus” (Pablo Quispe, 2022)

El trabajo de suficiencia profesional tiene como objetivo la creación de un aplicativo usando la herramienta Genexus y la metodología RUP. El autor define los pasos a seguir en la metodología como son la fase de inicio, elaboración, construcción y transición. También presenta otros trabajos usando la misma metodología de desarrollo y el uso de un modelado de sistemas como es UML.

En la siguiente figura 9 se muestra el esquema de casos de uso de sistema del aplicativo del presente trabajo, así como también se definieron los roles que tendrá cada uno de estos.

Figura 9. *Esquema de casos de usos para registro y cierre de reclamos*



Nota: Diagrama de caso de uso para un sistema de registro y cierre de reclamos para una entidad bancaria

Fuente. (Pablo Quispe, 2022)

“Prácticas para la gestión de continuidad de negocio en cooperativas con actividad financiera: una guía para la implementación” (Juan Arcila, 2021)

El presente trabajo detalla acerca de los riesgos que se pueden dar en el ambiente financiero. Planteando como estrategia el uso de un servidor de alerno donde enviar la información ante una posible pérdida del sistema.

En la siguiente figura 10 se muestran los posibles riesgos que podría sufrir el negocio financiero.

Figura 10. Posibles riesgos que podría sufrir el negocio financiero

Habilitador	Nombre Riesgo
Infraestructura	Indisponibilidad de infraestructura física
	Riesgo de imagen por negación de prestación de servicios
Ahorradores	Ausencia de ahorradores, o corrida de estos
	Limitación en las fuentes de fondeo
	Riesgo de liquidez
	Daño a la reputación
Servicios de TIC	Indisponibilidad de equipos de computo
	Obsolescencia de equipos de computo
	Riesgos emergentes
	Sobrecostos en la implementación de TICS
	Riesgo de imagen por negación de prestación de servicios
	Riesgos en seguridad de la información
	Bajos estándares en los ANS (Acuerdo de Nivel de Servicio)
	Indisponibilidad de servidores por caída ya que afectan servicio del ERP,
	Uso inadecuado de la tecnología
	Indisponibilidad de servicios, canales, portales, web service
Personal	Recarga de procesos concentrados solo en una persona
	No contar con personal que tenga la capacidad para afrontar los procesos clave
	Ausencia de personal clave por pandemias
	La dependencia de los roles
	Alto índice de rotación
Procesos	Indisponibilidad de los procesos
	No tener los procesos identificados y documentados
Proveedores	Alta dependencia del proveedor.
	Incumplimiento del servicio
	Interrupción del servicio
	Bajo poder de negociación

Nota: Posibles riesgos que podría sufrir el negocio financiero

Fuente. (Juan Arcila, 2021)

3.2.5 Implementación de las áreas, procesos, sistemas y buenas prácticas

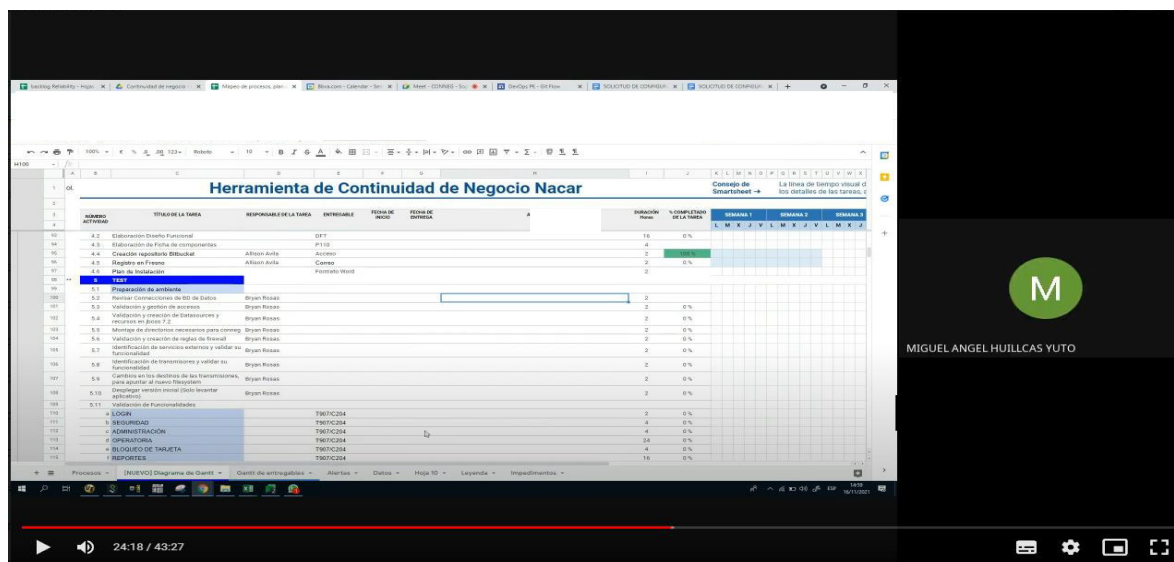
Primera Etapa: inicio

En esta etapa hubo reuniones entre el Project manager, área de operaciones y el área de gerencia del banco en donde se definieron el alcance, los objetivos, los entregables, el calendario y presupuesto, colocándolo todo en un documento de proyecto. El cual fue supervisado por las áreas competentes para su aprobación.

En la figura 11 se muestra la coordinación que hubo con las áreas acerca del cronograma de actividades del proyecto usando el meet.

Figura 11.

Meet de la coordinación del Cronograma del aplicativo



Nota: Reunión en sobre la planificación del cronograma,

Fuente. (Entidad Bancaria)

Segunda Etapa: planificación

El objetivo de esta etapa fue validar documentos antiguos del anterior proyecto con el banco con el fin de ver que no exista algún cambio en los requerimientos o diseño del sistema para esto se definieron dos iteraciones en la primera se verificó que los documentos de análisis cumplieran con todo lo propuesto por el banco, para lo cual se hicieron reuniones en donde se acordaron la conformidad de los documentos. Para la segunda iteración se corroboró la

En la figura 12 se muestra el acta de reunión tras la validación del documento de análisis del aplicativo. Uno de los primeros conceptos a validar fueron los casos de uso del sistema que se aprecia en el Anexo 1, en donde se validó si aun corresponden dichos casos con el actual proyecto. Luego, se validó la descripción de los casos de uso, primero se comenzó con el Login, esta descripción se encuentra en el Anexo 2. Seguidamente, se validó los diagramas de procesos, empezando primero por el ingreso al sistema con un perfil asesor como se aprecia en el Anexo 4. Y al último, fue validado el documento técnico que se aprecia en el Anexo 3; si las tablas aun corresponden y eran apropiadas para la carga de información del sistema del cual se encontró que algunas tablas no estaban bien cuadradas con la información de los archivos.

Figura 12.
Acta de reunión de revisión de los documentos

[illegible]

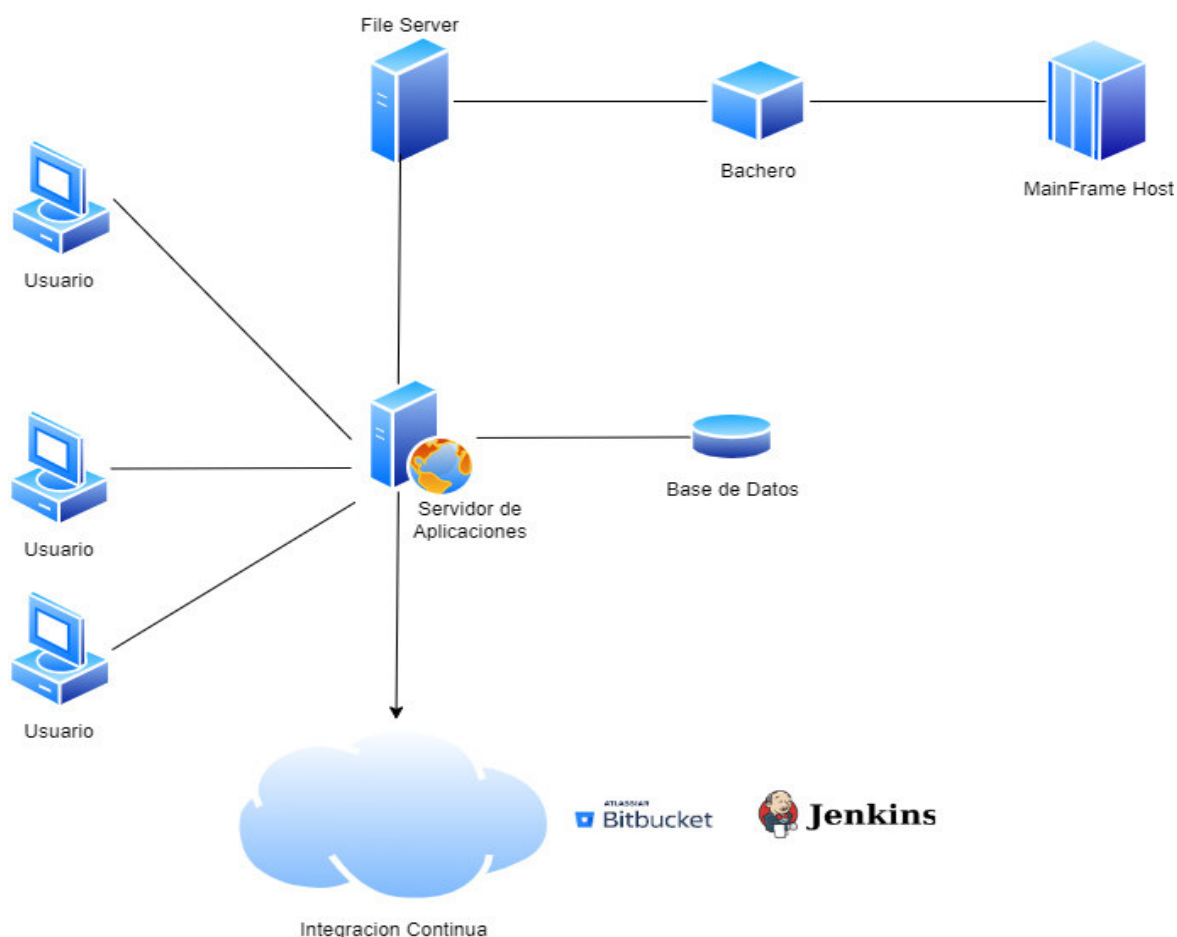
Nota: Acta de reunión donde se acordó la validación de los documentos del proyecto anterior.

Fuente. (Entidad Bancaria)

En la figura 13 se muestra la arquitectura final del aplicativo, que se diferencia con la anterior arquitectura es que en esta tendrá un despliegue continuo usando Jenkins y Bitbucket este último como herramienta de control de versiones; también, se observa el uso de un File Server que tiene dos funciones. La primera como alojador de archivos de carga de información del mainframe al aplicativo; siendo estos los siguientes: clientes, tarjetas, talonarios, parámetros, cheques, cuentas y oficinas. Y en la segunda como alojador de archivos de salida del aplicativo al mainframe; tales como: bloqueo, pagos, transacciones y cuentas. Siendo estos archivos generados por un bachero host hecho en cobol.

Figura 13.

Arquitectura Final del Aplicativo



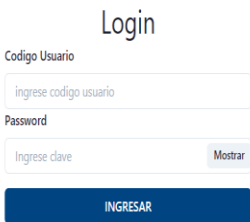
Nota: Descripción del aplicativo.

Fuente. (Entidad Bancaria)

También se hizo énfasis en la arquitectura monolítica del sistema, que se tenía de la anterior solución, para esto se planteó hacer un prueba de concepto; tal como se muestra en la figura 14, que consistía en llevar una funcionalidad en este caso el login al ser un desarrollo sencillo y construirla desde cero usando una nueva arquitectura basada en servicio como rest utilizando las tecnologías más comerciales como react y spring boot como resultado de esta prueba de concepto se encontró que la curva de aprendizaje y la complejidad de la nueva solución era alta y que por temas de tiempo y presupuesto por parte de banco no eran apropiadas aplicarlas en el proyecto por el momento, si no cuando el proyecto aumentará su complejidad.

Figura 14.

Prueba de concepto de la aplicación



The image shows a login interface with the title "Login". Below the title, there are two input fields: "Codigo Usuario" with a placeholder "Ingrese codigo usuario" and "Password" with a placeholder "Ingrese clave". A "Mostrar" button is located next to the password field. At the bottom, there is a blue button labeled "INGRESAR".

Nota: Interfaz de la prueba de concepto que se realizó usando JWT, Spring Security y una base de datos Oracle que aumentaron la complejidad de la solución.

Fuente. (Entidad Bancaria)

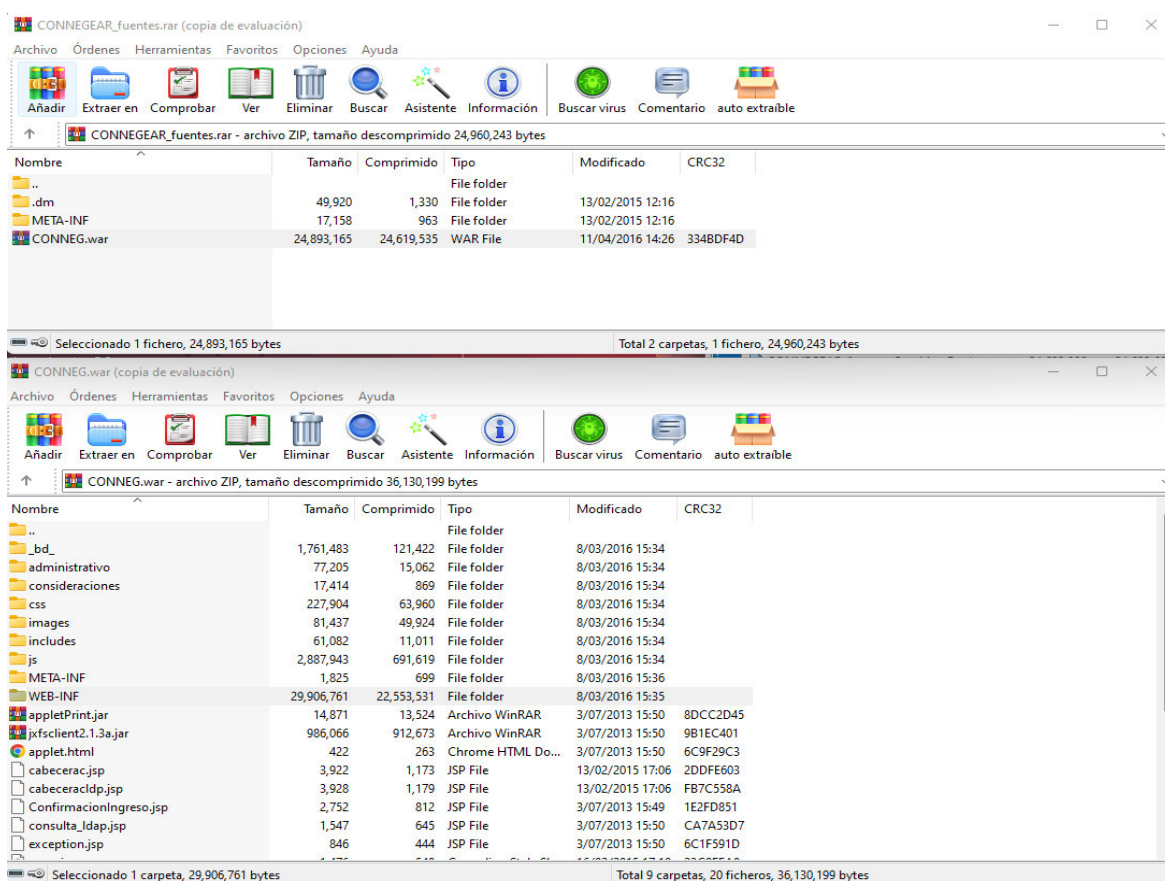
Tras la validación de los documentos de diseño, prueba de concepto y a la arquitectura se prosiguió a la examinación del código anterior del aplicativo, uno de los problemas que hubo

en esta etapa fue que no se contaba con un repositorio del código por parte del banco, a lo que se tuvo que hacer uso de ingeniería inversa al ear que se encontraba desplegado en el servidor WAS.

En la siguiente figura 15 se muestra el ear que se entregó, así como el código fuente que se tenía del anterior proyecto todo dentro un war.

Figura 15.

Estructura del proyecto anterior



Nota: Código fuente que paso y al cual se realizó la ingeniería inversa.

Fuente. (Entidad Bancaria)

Tras realizar la ingeniería inversa se encontró que proyecto trabajaba con una arquitectura monolítica con las siguientes tecnologías: Spring 3 como framework de vista, Hibernate 2.3 como framework de acceso a datos, Java 6 como kit de programación y usaba el driver Oracle 9.

El detalle de las tecnologías se muestra en la siguiente figura 16.

Figura 16.

Tecnologías encontradas antes del desarrollo



Nota: Detalle de las tecnologías usadas dentro del aplicativo anterior.

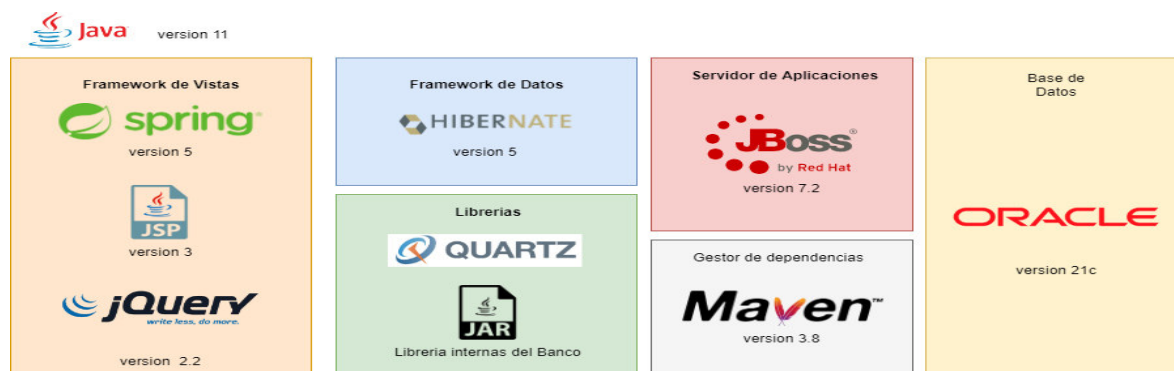
Fuente. Elaboración propia.

En esta etapa también se hizo el mapeo de los programas hosts que se comunicaban el aplicativo del 2011 para poder usarlos en la integración con el nuevo sistema a desarrollar.

También se definieron las tecnologías a usar: Java 11, Spring 5, Hibernate 5, Maven y Quartz 2.5 con servidor de aplicaciones Jboss 7.2. Como se puede apreciar en la figura 17.

Figura 17.

Tecnologías actuales en desarrollo



Nota: Detalle de las tecnologías planteadas para el nuevo aplicativo.

Fuente. Elaboración propia

Se escogió estas tecnologías porque ya no hay soporte para la versión de WAS que trabaja anteriormente el aplicativo y por qué estas tecnologías son compatibles con el servidor de aplicaciones licenciado JBoss 7.2. También se hizo uso de gestor de dependencias para mantener un mayor orden en las dependencias o librerías del proyecto.

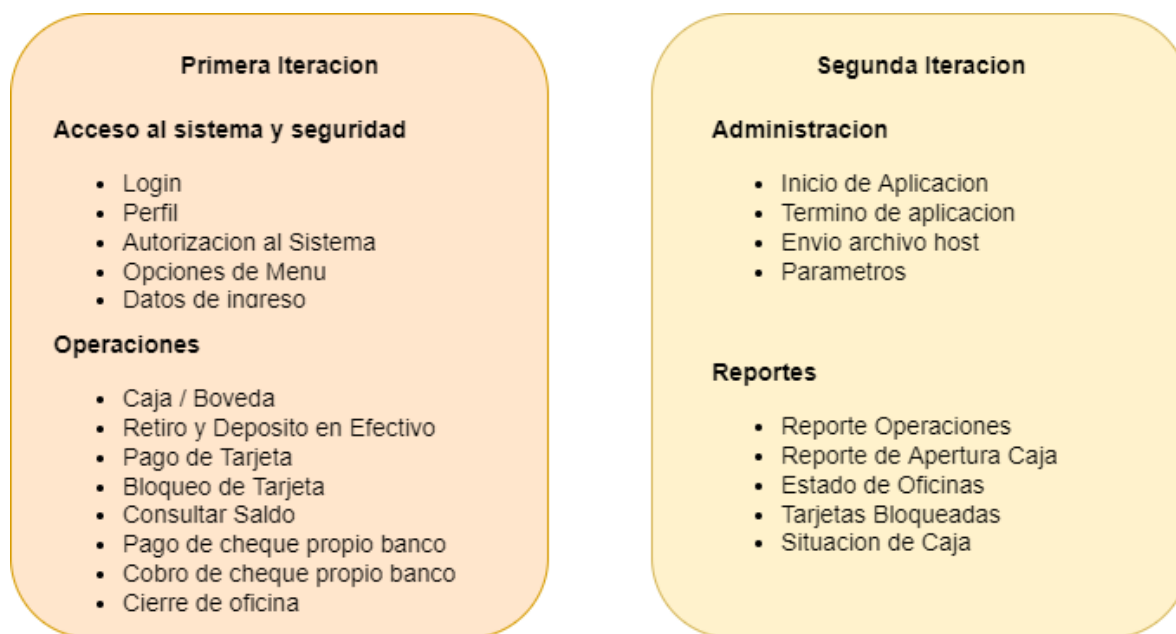
Tercera Etapa: Desarrollo

Esta etapa contó con dos desarrolladores Java, un desarrollador Cobol y un Project mánager. El desarrollo se dividió por etapas; primero se empezó con acceso al sistema, luego se desarrolló el módulo de seguridad en conjunto con el módulo de operaciones, luego continuó con módulo de administración y el módulo de reportes.

Como se muestra en la siguiente figura 18 se detallan las actividades desarrolladas en cada uno de los 4 módulos de entre el que tienen mayor peso e impacto es el módulo de operaciones siendo este el core del banco.

Figura 18.

Iteraciones en la etapa de desarrollo



Nota: Descripción de las actividades realizadas durante el Desarrollo.

Fuente. Elaboración propia

Las siguientes carpetas se usaron siguiendo el patrón MVC para un mayor organización y legibilidad del código. Estas se dividen de la siguiente manera:

- La carpeta **bean** es donde se encuentran los modelos del aplicativo.
- La carpeta **controller** es donde se encuentra la conexión con las vistas jsp.
- La carpeta **entity** es donde se encuentran las tablas de base de datos.
- La carpeta **facade** es donde se encuentra la lógica de servicio.
- La carpeta **local** es donde se encuentran las interfaces de la carpeta sesión.
- La carpeta **session** es donde se encuentra el acceso a la base de datos.

Como se muestra en la figura 19.

Figura 19.

Estructura del código del proyecto



Nota: Carpeta que se desarrolló dentro del proyecto.

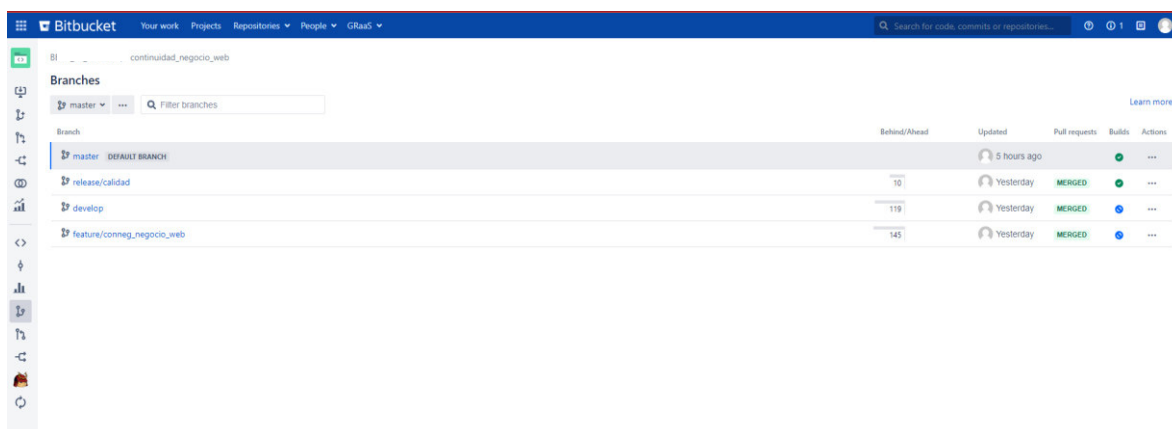
Fuente. (Entidad Bancaria)

Durante el desarrollo se hizo uso de una herramienta para alojar proyectos en la nube llamado Bitbucket, que se basa en el sistema de control de versiones llamado Git. El proyecto fue dividido en diferentes ramas como se muestra en la figura 20 en las que destacan:

- Master: Rama de producción del sistema, para desplegar aquí necesitas de documento de pases y un artifact generado en esta rama.
- Release/calidad: Rama de calidad del sistema, ambiente en donde los tester desarrollan las pruebas funcionales al sistema.
- Develop: Rama de desarrollo del sistema, es donde se desarrolla la mayor parte de las pruebas por parte del desarrollador.

Figura 20.

Repositorio Bitbucket del proyecto



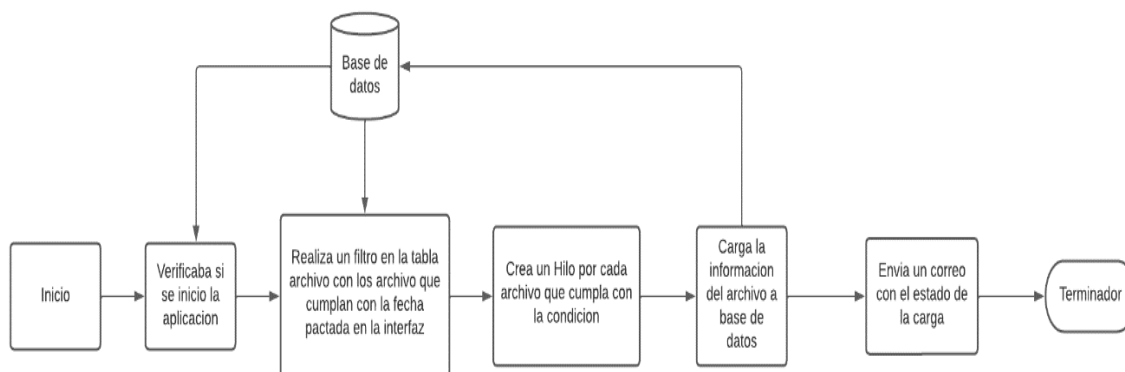
Nota: Ramas creadas durante el proyecto.

Fuente. (Entidad Bancaria)

También se hizo uso de scheduler que permitían la carga periódica de información de cada archivo que generaba el bachero. Este scheduler se dejaba correr cada 30 segundos y estaba atento a una rutina que se fijó a nivel de código. La rutina se explica en la siguiente figura 21.

Figura 21.

Flujo del Scheduler



Nota: Flujo del scheduler.

Fuente. Elaboración propia.

Para la realización de carga periódica se hizo uso Quartz una librería que por su versatilidad y facilidad de uso se escogió para este proyecto. Las cargas periódicas se acordaron de que fueran diarias en horario de inactividad entre 7pm a 4 am. Entre los principales archivos de carga están los datos del cliente, tarjetas, talonarios, cuentas, cheques y oficinas. También cabe destacar que, cuando se realicen las cargas se le enviará un correo detallando la culminación de carga si fue exitosa o hubo algún error en el proceso. Como se aprecia en la figura 22 donde se observa la interfaz de carga información del mainframe al aplicativo en la cual se especifica el nombre del archivo a extraer, tabla, secuencia y correo en caso dado se produjera un error.

Figura 22.

Interfaz de carga de información del mainframe al aplicativo

Red de Oficinas (Contingencia)
 REGISTRO: P00001, MIGUEL AROQUE VASQUEZ, OFICINA: 0750, TERMINAL: 9999, PERFIL: ADMINISTRADOR 2, FECHA: 25/06/2022, HORA: 16:57 Cerrar Sesión

Seguridad Administracion Operatoria Bloqueo de Tarjetas Reportes Ayuda

Configuración de Carga de Archivos

Nombre: Extensión:
 Ruta:

Buscar Nuevo

Opciones	Nombre del Archivo	Extensión	Ruta	Tabla	Secuence
	PEBD.PEFD.FIX.CONTNEG.DATOCCLI.D1	TXT	D:\mntfcompartido\conneg\load\	ENTRADACLIENTE	SQ_ENTRADACLIENTE

Datos del Archivo

Datos del Archivo

Código Archivo: 5 Estado: ACTIVO
 Nombre: PEBT.BGFD.FIX.CONTNEG.TARJMPA Extensión: TXT
 Tabla: ENTRADATARJETA Secuence: SQ_ENTRADATARJETA
 Ruta: D:\mntfcompartido\conneg\load\ Email: ROMANILOPEZ@GMAIL.COM

Datos de la Programación de la Carga

Mes: TODOS Tipo: MENSUAL Día: TODOS Hora: 06 Minuto: 00

Nombre Campo de Tabla	Posición	Tipo de Dato	Longitud	Formato	Activo
BINTARJETA	1	C	6		☒
NUMEROTARJETA	2	C	13		☒
TIPOTARJETA	3	C	2		☒
DESCRIPCION TARJETA	4	C	20		☒
ENTRADCONTRATO	5	C	4		☒

Grabar Retornar

Nota: Interfaz de la aplicación sobre la carga de información del aplicativo.

Fuente. (Entidad Bancaria)

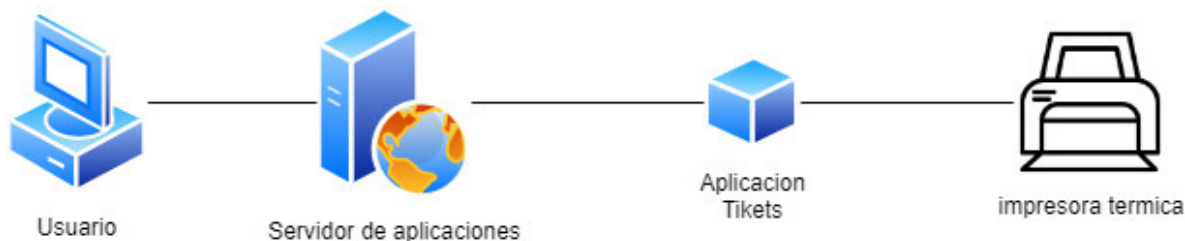
Unas de las problemáticas que se desarrolló en esta etapa fue el tema de impresión de tickets, ya que el anterior proyecto hizo uso de applets, que por temas de seguridad ya no es compatible con navegadores actuales así que para el desarrollo en esta etapa se tuvo que hacer una coordinación con el equipo de México del banco para que nos apoyen con uso de la librería de la impresora térmica.

La solución que se propuso fue crear un servicio intermedio que usando los drivers de la impresora térmica se conectara a esta y realizará la impresión de un ticket para cada operación que requiera el aplicativo web. Este servicio local se instalaría en cada oficina del banco. Para realizar dicha instalación en los ambientes del banco se creó un archivo con extensión .sh el cual se ejecutaría en la máquina Linux de la oficina y permitiría la creación de un servicio local en dicho sistema operativo.

En la siguiente figura 23 se puede apreciar la arquitectura del servicio local intermedio que se creó para la impresión de tickets desde el aplicativo web.

Figura 23.

Solución de Tickets



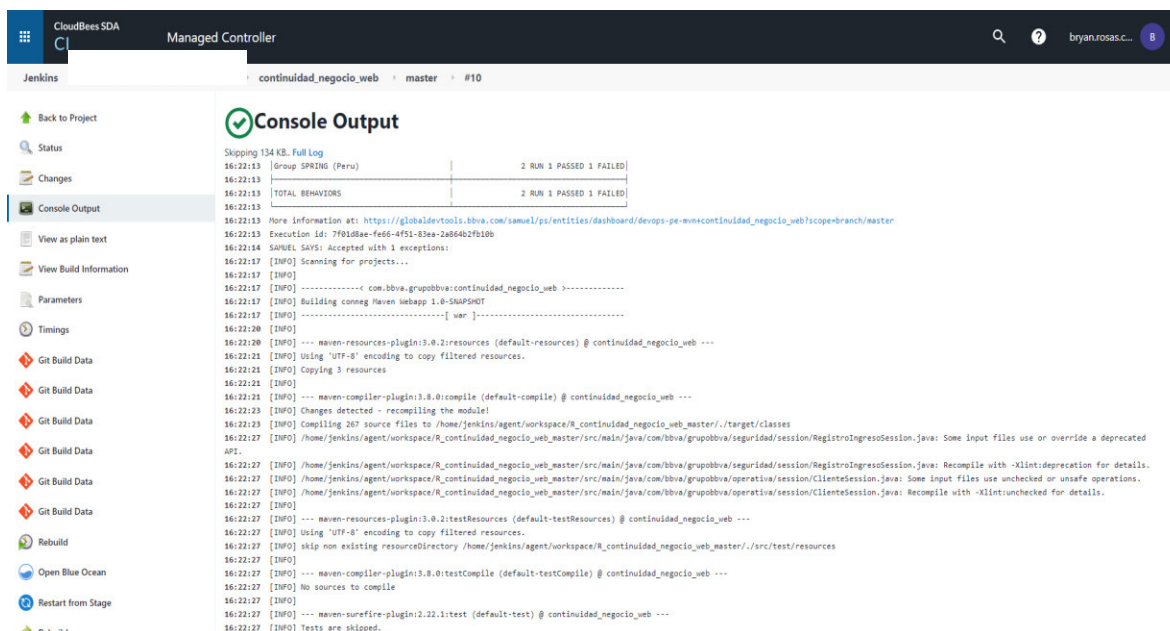
Nota: Diagrama de arquitectura del aplicativo de impresión de tickets.

Fuente. Elaboración propia

Además, se actualizaron los servicios de comunicación con otros sistemas tales como LAPD y Reniec. El primero para extraer la información de los usuarios del banco tales como username, password, código de oficina y nombre del usuario. Y el segundo para acceder a la información de los clientes tales como nombres y apellidos, fecha de fallecimiento, fecha de nacimiento, sexo y dirección, con fines de confirmar la identidad del cliente.

Como punto final para obtener un despliegue continuo con Jenkins se hizo uso de un archivo jenkins file. El cual contenía la configuración necesaria para la integración continua. En la figura 24 se puede apreciar un despliegue exitoso continuo usando Jenkins en proyecto de continuidad de negocio.

Figura 24.
Consola de Jenkins



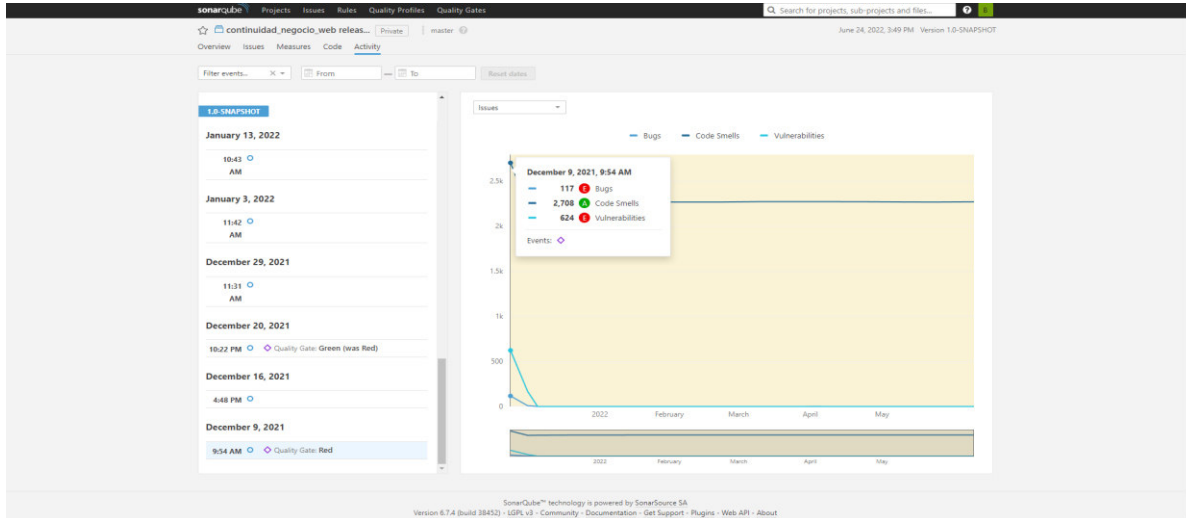
Nota: Consola de jenkins donde se puede ver el build de realizado.
Fuente. (Entidad Bancaria)

Cuarta Etapa: Transición

En esta etapa se realizaron las pruebas por parte del área de pruebas del banco y así como la carga información aplicativo para poder tener datos de pruebas. También se hizo uso de la herramienta SonarQube para análisis de código estático en el aplicativo y detección de malas prácticas en el código. Al inicio del proyecto se contaba con 117 Bug y 624 vulnerabilidades que impedían que el código del aplicativo este saludable para lo cual se realizó un arduo trabajo de 11 días en las cuales solo se veía la mejora del aplicativo a nivel de buenas prácticas, que dio como resultado final 0 Bug y 0 vulnerabilidades que se pueden ver en las figuras 25 y 26.

Figura 25.

Análisis del inicio de aplicativo en SonarQube

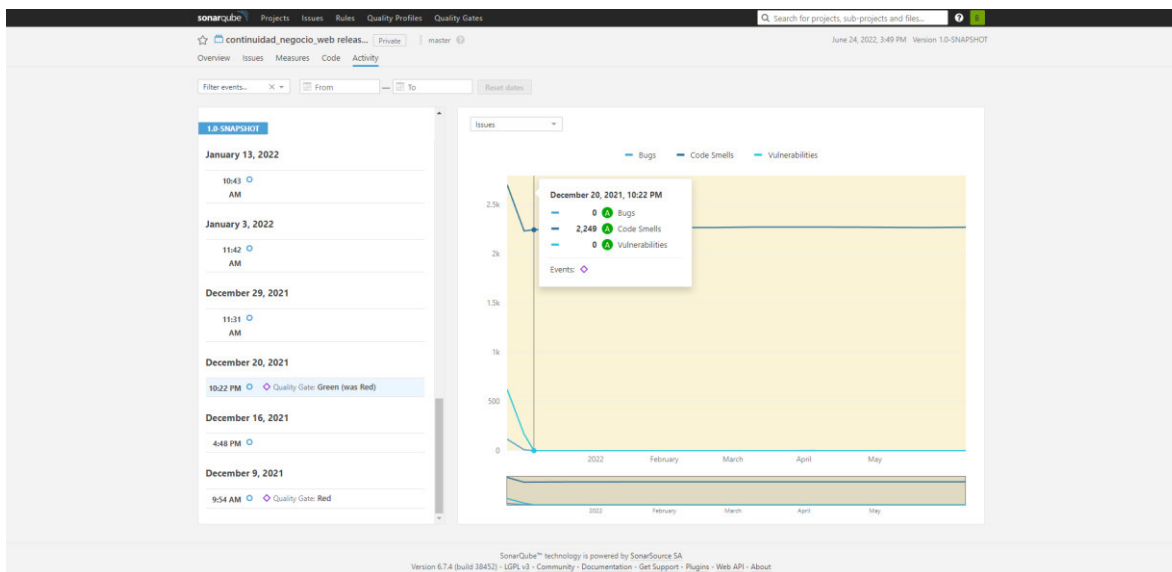


Nota: Bug y vulnerabilidades en el 9 de Diciembre.

Fuente. (Entidad Bancaria)

Figura 26.

Análisis final del aplicativo en SonarQube



Nota: Bug y vulnerabilidades en el 20 de Diciembre.

Fuente. (Entidad Bancaria)

Luego realizar buenas prácticas en el aplicativo usando SonarQube y el despliegue en los ambientes de calidad del banco. Se prosiguió a las pruebas funcionales por parte del área de tester del banco donde se observó lo siguiente (tabla 8).

Tabla 8.

Observaciones de las pruebas funcionales

Observaciones	Tipo observación
Correcciones en las posiciones de los archivos de salida	Funcional
Cambio de logo e interfaz del banco	Funcional
Errores de puntuación formato de conversión de numeral en apertura y cierre de caja	Funcional
Errores con la validación Reniec	Funcional
Errores de ortografía	Funcional

Nota: Errores encontrados tras las pruebas funcionales en el aplicativo.

Fuente. Elaboración Propia

De las pruebas funcionales se creó los documentos de pruebas llamado T907 – Evidencia de pruebas como se aprecia en la figura 27, así como también DFT – Pruebas Funcionales del aplicativo

Figura 27.

T907 – Evidencia de pruebas

T907 - Evidencia de Pruebas

Ambito de las Pruebas	Nro. Ticket Jira ó Service Now	Título del Ticket
SSDD	<u>PACDN-2</u>	Actualizar la Herramienta Web de Continuidad <u>CONNEX</u> (Nácar de Contingencia)
Fecha creación	Fecha actualización	Creado por
24.01.2021	25.01.2021	

Objetivo del documento
El objetivo de este documento es registrar las evidencias por cada Caso de Prueba descrito en el C204.

Bitácora de cambios al documento			
Versión	Fecha	Autor Modificación	Descripción del cambio
1.0	24/01/2022		Primera versión

Ejecución de los Casos de Prueba del documento C204 y registro de Evidencias
--

Identificación del Caso de Prueba	
Código de Caso:	C0001
Nombre del Caso:	Login Subgerente

Pruebas detalladas del Caso de Prueba	
Prueba efectuada:	Login Subgerente
Descripción:	Login Subgerente

Nota: Documento de evidencia de pruebas.

Fuente. (Entidad Bancaria)

También en esta etapa se realizaron los despliegues a ambientes productivos llenado de información los documentos de pase a producción tales como P 110 – Hoja de Pase a producción este documento comprende todos los componentes que se van a desplegar en ambientes productivos como se aprecia en la figura 28, así como también se creó el documento de plan de instalación en donde se detallan los pasos a seguir para una correcta instalación de los packages, war y queries.

Figura 28.

P110 - Hoja de pase a producción

P110 - Hoja de Pase SSDD -Versión 2021 .XLSX Guardado en Drive

Archivo Editar Ver Insertar Formato Datos Herramientas Ayuda Última modificación el 2 de mayo

100% \$ % .0_ .00 123 Arial 9 B I A

HOJA de PASE A PRODUCCIÓN
HOST - SSDD -

FECHA dd/mm/aaaa

HOST SS DD X

Fecha Pase UPD

N° RSD Alto Impacto Requiere Pruebas ? x

N° Ticket Jira ó ServiceNow PACDN-2 Aplicativo / Equipo Conneg (Continuidad de Negocio)

N° Tarea Calidad Lider Solutions

N° Tarea Gest Cambios Lider Reliability

Service Now

Jira Service Now

Evolutivo Incidencias

Obsolescencia x Problemas

Optimización Petición

Proyecto Especial

Proyecto Jira

Servicios Críticos

Vulnerabilidad

Descripción Detallada Funcional del cambio Migración del Aplicativo a la version 11 jdk y compatibilidad de despliegue jboss EAP 7.2.

LENGUAJE / PAQUETE	COMPONENTE	TIPO DE COMPONENTE	REFERENCIA / MOTIVO (Agregar líneas si lo requiere)	Tipo	Fecha	Hora	Tamaño
java	CONNEX.war	war	Componente de CONNEG Web	1			
sql	CONNEX.sql	sql	Script de actualización de la tabla ENTRADATARJETA_EXT	1			
sql	PQ_REPORTE.pck	pck	Paquete de reportes	1			
sql	UPDATE_ARCHIVO.sql	sql	Script de actualización de la tabla ARCHIVO de	1			

*Nota: Hoja de pase donde se describen los componentes que se van a pasar a producción.
Fuente. (Entidad Bancaria)*

3.3 Evaluación

3.3.1 Evaluación Económica

El proyecto tuvo una duración de 7 meses en las cuales estuvo involucrado dos desarrolladores, un líder de proyecto, el área de test y un analista de operaciones para el despliegue automático del aplicativo. En las siguientes tablas se puede ver el gasto en un mes de desarrollo del proyecto, así como también gasto total de las licencias y equipo que se usó en el proyecto (tabla 9 y 10)

Tabla 9.

Costos estimados del personal del proyecto

Costo	Descripción	Monto/mes
Consultora	Desarrolladores (2)	S./ 11,000
Banco	Líder de Proyecto (1)	S./ 8,000
Consultora	Testers (2)	S./ 11,000
Consultora	Líder Tester (1)	S./ 8,000
Banco	Analista de Operaciones (1)	S./ 6,000
	Total	S./ 44,000

Nota: Descripción de los costos estimados personal en el proyecto.

Fuente. Elaboración propia.

Tabla 10.

Costos Estimados Operativos del proyecto

Costo	Descripción	Monto
Licencia	Servidor de aplicaciones (JBoss)	S./ 1,321.14
Equipo	Laptop (7)	S./ 21,000
	Total	S./ 22,321.14

Nota: Descripción de los costos operativos en el proyecto.

Fuente. Elaboración propia

En total el monto gastado suma un alrededor S./ 330,321.14 durante los 7 meses de desarrollo que se llevó del proyecto.

3.3.2 Interpretación del VAN y del TIR

El proyecto demandó una inversión S./ 330,321.14 y para este flujo se consideró un generaría un flujo 8,500,000 por cada año durante 4 años considerando una tasa de descuento de 10% anual. Se realizó el siguiente cálculo:

$$van = i_o + \sum_{t=1}^n \frac{F_t}{(1+k)^t}$$

Hacemos el reemplazo de las variables:

$$VAN = -330,321.14 + \frac{8169678.86}{(1+0.10)} + \frac{8,169,678.86}{(1+0.10)^2} + \frac{8169678.86}{(1+0.10)^3} + \frac{8,169,678.86}{(1+0.10)^4} = 25,566,461.59$$

Calcula de TIR se realiza siguiendo la siguiente fórmula:

$$0 = i_o + \sum_{t=1}^n \frac{F_t}{(1+k)^t}$$

$$0 = -330,321.14 + \frac{8169678.86}{(1+k)} + \frac{8,169,678.86}{(1+k)^2} + \frac{8169678.86}{(1+k)^3} + \frac{8,169,678.86}{(1+k)^4}$$

Para una tasa de 2460 %

$$VAN = 1778.853434$$

Para una tasa de = 2480 %

$$VAN = -899.3488115$$

Haciendo interpolación:

$$\frac{2460-2480}{1778.85-(-899.34)} = \frac{2460-TIR}{1778.85-0} = TIR = 2473.28 \%$$

Este porcentaje refleja que el proyecto se beneficia de manera positiva ya que es superior al que se asumió en un principio 10%.

CAPÍTULO IV: REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El presente autor del trabajo de suficiencia profesional se desarrolló dentro del proyecto como Analista de Software, el cual le permitió analizar los requerimientos y las diferentes tecnologías comprendidas durante el proyecto.

Dentro del proyecto hubo mucha comunicación por parte del área usuario lo cual permitió que el proyecto no se extendiera más tiempo y permitiendo afinar más el software propuesto en este documento.

Durante el proyecto se analizó las tecnologías a usar para lo cual se hizo una prueba de concepto con las tecnologías más populares de mercado, pero como resultado de esto se concluyó que iba tomar mucho tiempo adaptarse a esa tecnología para la cual se optó por seguir el mismo cambio que tenía la solución propuesta al inicio.

A pesar de algunas complicaciones que hubo en la construcción de la solución de impresión de tickets se pudo afrontar muy bien con la ayuda del equipo de México. Permitiendo que esta nueva solución sea compatible con otras soluciones que requieran de la impresión de tickets.

El uso de SonarQube le permitió al autor mejorar su calidad de código, resolver bug y vulnerabilidades que hubieran afectado al proyecto en un futuro.

El uso de RUP permitió al autor poder mejorar la calidad y tener una mayor aceptación de los entregables ya que estos eran revisados en cada iteración del proyecto.

Las tecnologías que se usaron también permitieron al autor conocer nuevos conceptos y nuevas herramientas para poder afrontar nuevos proyectos.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se desarrolló e implementó el sistema de continuidad de negocio para las operaciones más usadas dentro de una entidad bancaria, dando solución al problema de inoperatividad en las oficinas de Lima ante una posible pérdida de comunicación con la infraestructura fuera del país, usando un servidor alternativo y servidor de base de datos alojado en Perú para brindar soporte dentro del país, como se puede apreciar en el punto 3.2.5 del presente documento en donde se detalla el proceso llevado a cabo durante el proyecto.
- La solución planteada cumplió con las normativas dictadas por la SBS que hablan sobre la continuidad de negocio frente a la pérdida de comunicación; esto se detalla en el subcapítulo 3.1 del presente documento, donde se especifica el uso de un servidor alternativo ante una caída del servidor primario que, en conjunto con el servidor propio alojado en Perú, ayudarían a poder afrontar el tema de la pérdida de comunicación entre servidores.
- Se analizó la documentación y el código que se tenía del anterior proyecto, esto se detalló en la segunda etapa de planificación del acápite 3.2.5 del presente documento; donde se habla como se examinó la documentación y el método de ingeniería inversa que se aplicó, con el fin de extraer la información base del proyecto y así obtener el código base para poder desarrollar el aplicativo del presente trabajo.
- Se actualizaron las tecnologías que se tenía del aplicativo, como se explica en la segunda etapa de planificación del acápite 3.2.5 del presente documento donde se detalla las tecnologías usadas en el aplicativo como son Oracle 21c, Spring Framework 5, entre otras también se actualizaron las librerías que tengan soporte con los servidores actuales como JBoss 7.2.
- Para el uso de buenas prácticas se hizo uso de la metodología RUP separando el desarrollo en iteraciones las cuales eran luego revisadas por las partes interesadas del banco esto se detalla en la tercera etapa de construcción del acápite 3.2.5 del presente documento.

- Se crearon nuevos documentos para las pruebas funcionales y para despliegue del aplicativo en ambiente productivos, con el fin de brindar documentación que ayude a tener un mayor entendimiento del aplicativo; de esa manera, brindar un apoyo en caso de alguna falla en el sistema, como se aprecia en la cuarta etapa de transición del acápite 3.2.5 del presente documento.

5.2 Recomendaciones

En base a la experiencia adquirida en este proyecto, se hacen las siguientes recomendaciones:

- A nivel de desarrollo, se pudo aplicar el uso de nuevas tecnologías de desarrollo como react y spring boot para obtener un mayor soporte en el aplicativo; debido a que son mucho más comerciales que las tecnologías planteadas en el actual desarrollo.
- A nivel de mantenimiento, si el aplicativo se vuelve más complejo con el tiempo por causa de nuevos requerimientos, es mejor irse moviendo de un modelo de MVC a un modelo basado en servicio rest. Dado que este último trae consigo un mayor soporte y escalabilidad de los servicios.
- A nivel de entregables del proyecto, se pudo hacer uso de un plan de SQA que en conjunto con la metodología RUP, hubieran mejorado aún más la calidad de los entregables y la comunicación entre las partes interesadas. Dando como resultado un mejor producto del software.
- A nivel de despliegue, se pudo aplicar inteligencia artificial para levantar los servicios de continuidad de negocio, sólo cuando sean necesarios; disminuyendo costos altos a futuro.

5.3 Fuentes de información

- Platzi (2020, 19 de Diciembre), ¿Qué es el patrón MVC?, Recuperado de https://platzi.com/tutoriales/1248-pro-arquitectura/5466-que-es-el-patron-mvc/?gclid=CjwKCAjw46CVBhB1EiwAgy6M4uBLTr1U94d0lZGtpQUY4YHraA4BFE65nBGxUhNmpq0PgUXQPK5SWxoCEvoQAvD_BwE&gclsrc=aw.ds
- Sentrio (2021, 15 de Diciembre), Qué es SonarQube, Recuperado <https://sentrio.io/blog/que-es-sonarqube/>
- Ecured (2020, 14 de Febrero), Proceso unificado de desarrollo, Recuperado de https://www.ecured.cu/Proceso_unificado_de_desarrollo
- CoderMentor (2019, 14 de Agosto), Java Scheduler, Recuperado de <https://www.codementor.io/@joydipkumar/java-scheduler-xudhcr72w>
- Silvestre Mercedes, Jorge Luis (2018), Automatización del proceso de inscripción y gestión de alumnos en el Centro Preuniversitario de la Universidad Nacional Mayor de San Marcos usando la metodología RUP, adecuándolo al uso de software libre (tesis de pregrado) Universidad Nacional Mayor de San Marcos, Perú
- Luna Quinto, Diego Armando Jesús (2021), Automatización del proceso aprobar calendario de pago en la unidad de subvenciones y financiamiento de PRONABEC (tesis de pregrado) Universidad Nacional Mayor de San Marcos, Perú
- Quispe Vilcatoma, Pablo César (2022), Implementación de un proceso automático para el registro y cierre de reclamos en una entidad bancaria utilizando metodología RUP y la herramienta Genexus (tesis de pregrado) Universidad Nacional Mayor de San Marcos, Perú
- Arcila Montoya, Juan Gabriel (2021), Prácticas para la gestión de continuidad de negocio en cooperativas con actividad financiera: una guía para la implementación (tesis de pregrado) Universidad EAFIT, Colombia
- DZone (2018, 28 de Noviembre), Working With the Java Scheduler, Recuperado de <https://dzone.com/articles/schedulers-in-java-and-spring>
- Oscar Blancarte (1 de enero del 2020), Introducción a la arquitectura de software un enfoque práctico, 1, p.134

- Sentrio (2021, 16 de Setiembre), Introducción a Jenkins: ¿qué es, para qué sirve y cómo funciona?, Recuperado de <https://sentrio.io/blog/que-es-jenkins/>
- SBS (2020, 20 de Diciembre), La gestión de continuidad de negocios en el sistema financiero en tiempos de pandemia, Recuperado de <https://www.sbs.gob.pe/boletin/detalleboletin/idbulletin/1135>
- SBS (2017, 25 de Enero), Resolución SBS N° 139-2009, Recuperado de https://www.sbs.gob.pe/app/pp/int_cn/paginas/busqueda/Download.aspx?url=p8ehJETNq5hYZsadCTQIDA1Pw7aAvNcREl6DVQhESNiouiM3fzeprEqSY0+Ubw+IMYa9V8Zay/mgvD1uvX66PIwO9B2GmahyzPxCppervruL9tT8Mqm8Ysxdsmg64A+BntTqwhBTvb5KCdYVDJ0ubRkcZuADwBvyEkye/IUwI/A2z2c/UAFf/Rc0ipWU8DUA6fJVVj8aKVpizngpXMEig==
- SBS (2009, 2 de Abril), Resolución SBS N° 2115-2009, Recuperado de https://intranet2.sbs.gob.pe/intranet/INT_CN/DV_INT_CN/840/v1.0/Adjuntos/2115-2009.r.pdf
- SBS (2009, 2 de Abril), Resolución SBS N° 2116-2009, Recuperado de https://intranet2.sbs.gob.pe/dv_int_cn/842/v2.0/Adjuntos/2116-2009.r.pdf
- SBS (2018, 16 de Julio), Resolución SBS N° 2755-2018, Recuperado de https://www.sbs.gob.pe/Portals/5/jer/REG_PER_CAM_MON/Reglamento%20de%20Sanciones.pdf

5.4 Glosario

- **Spring**

Es un framework basado en la tecnología java que ofrece todo orquestación desde la seguridad, creación de servicio rest, microservicios, schedulers, entre otros.

- **Git**

Es un sistema de control de versiones distribuido. Está pensando en el mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos en su código fuente.

- **REST**

Es una interfaz para interconectar diferentes sistemas basado en el protocolo HTTP, está basado en la actualidad en el formato JSON ya que es ligero y legible en comparación con el XML.

- **LAPD**

Librería del banco que contiene la información de todos los usuarios de la entidad bancaria.

- **JWT**

Es un estándar abierto basado en JSON para la creación de token que sirve para enviar información entre aplicaciones y de esa manera garantizar que sean seguros y válidos.

- **Bug**

Se refiere un error o defecto de software que trae como consecuencia que el programa no funcione correctamente

- **Vulnerabilidad**

Se refiere a una situación de amenaza o posibilidad de sufrir algún daño en el aplicativo.

- **Servidor de aplicaciones**

Es un programa de servidor en un equipo que proporciona la lógica de negocio y acceso a datos para un aplicativo.

- **SQA**

Es un conjunto de actividades sistemáticas que aseguran que el proceso del software y producto cumplan con los estándares planteados al inicio del proyecto garantizando unos óptimos niveles de calidad.

- **Prueba de concepto**

Se define como un método de investigación del mercado dando ideas para producto antes de realizar el desarrollo. De esa manera se puede medir su aceptación y tomar decisiones antes de realizar el desarrollo.

- **Mainframe**

Computadoras de alto rendimiento con grandes cantidades de memoria y procesadores.

- **File server**

Servidor dedicado al alojamiento de archivos de todo tipo de extensión.

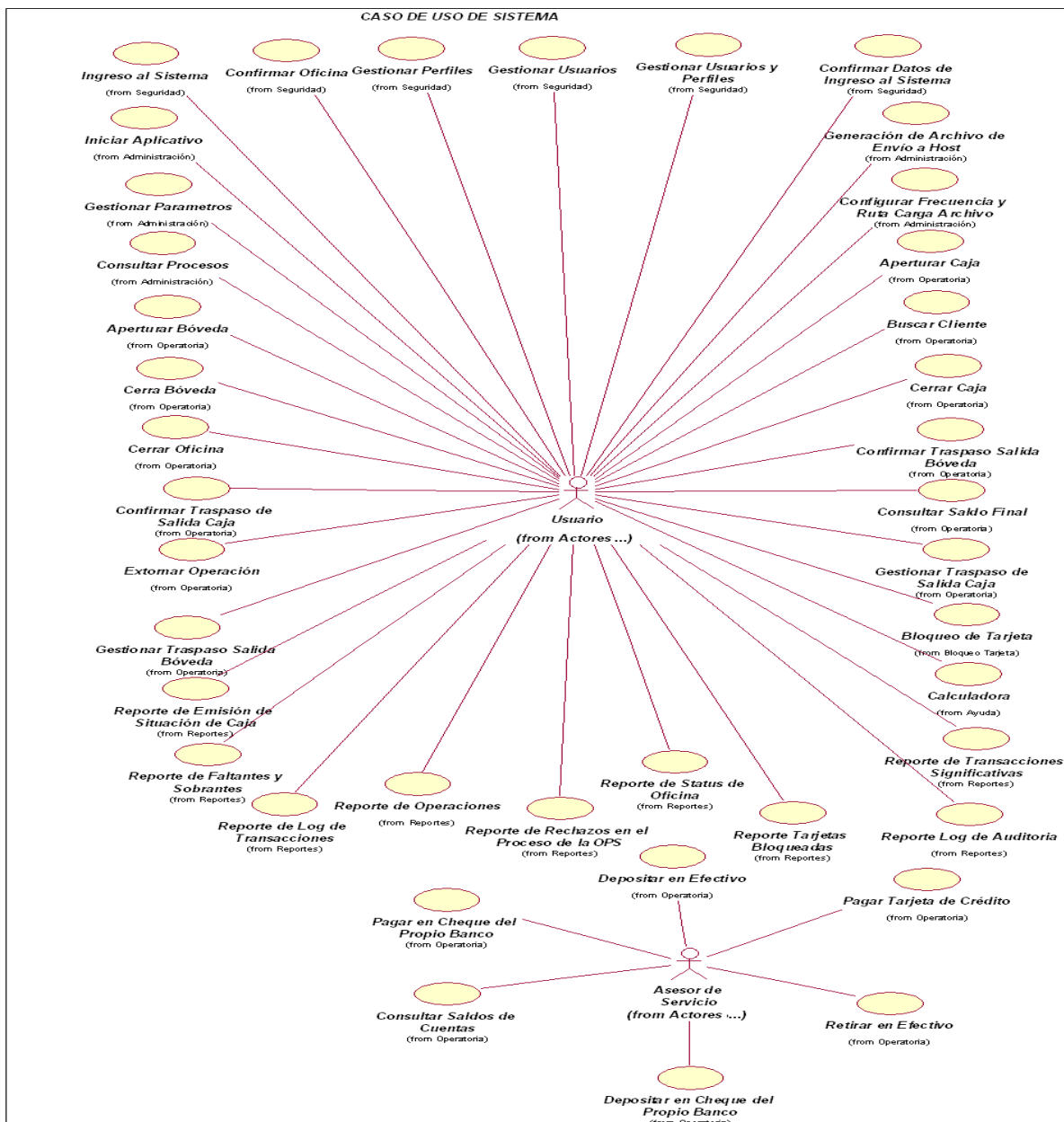
- **Bachero**

Aplicativo que procesa grandes cantidades de registros ya para inserción o actualización.

ANEXOS

Anexo 1: Casos de uso del sistema

Diagrama en donde se visualiza los casos de uso del sistema sacado del documento de análisis del presente trabajo, el cual se colocó en el punto 3.2.5 en la segunda etapa de planificación del presente trabajo.



Anexo 2: Caso de uso login

Este anexo fue mostrado en el punto 3.2.5 en la segunda etapa de planificación del presente trabajo. El cual describe el caso de uso del Login en donde se detallan las condiciones que debe cumplir el aplicativo para darse por realizado.

NOMBRE DEL CASO DE USO

CUS01 – INGRESO AL SISTEMA

OBJETIVOS

Realizar el ingreso al sistema de continuidad de negocio de banca minorista.

INVENTARIO DE ACTORES

Usuario

CARACTERÍSTICAS

PRECONDICIONES

1. El usuario, debe de estar registrado en la base de datos del aplicativo de continuidad de negocio de banca minorista.

POSCONDICIONES

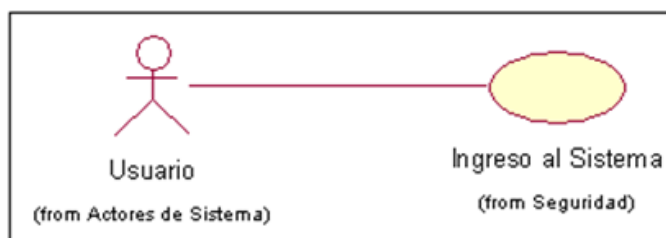
Válidas

1. Se validará los usuarios registrados en la LDAP

Inválidas

1. El usuario, no podrá tener acceso al sistema, sin el perfil asignado.

Diagrama de Caso de Uso - CUS01: INGRESO AL SISTEMA



ESCENARIO PRIMARIO

Paso	Acción
1	Desde el sistema de continuidad de negocio de la banca minorista, el usuario, indica "Ingreso al sistema".
2	El usuario, tendrá la opción <u>loguearse</u> al sistema una vez ingresado a la ventana "Ingreso al sistema".
3	El usuario, ingresará la siguiente información para realizar el ingreso al sistema. • Usuario • Clave de acceso Luego elegirá la opción INGRESAR.
4	El sistema mostrará la ventana donde se ingresará el número de terminal y la confirmación de la oficina donde va a realizar las operaciones.
5	El usuario pulsa el botón "Ingresar", el sistema mostrará la pantalla principal del sistema.
6	El usuario indica "Cerrar Sesión", el sistema cierra la ventana pantalla principal del sistema.
7	Fin de caso de uso.

VALIDACION:

Paso	Acción
01	En el punto 3, en caso de que el usuario no ingrese ningún campo, el sistema mostrara un mensaje donde indique que es obligatorio ingresar el usuario y <u>el password</u> .
02	En el punto 3, en caso de que el usuario ingrese un usuario o <u>un password</u> que no sean válidos, el sistema mostrará un mensaje donde indique la invalidez de los campos.
03	En el punto 3, <u>en caso que</u> usuario se olvide su contraseña. El usuario selecciona <u>link</u> ¿Olvido contraseña?, El sistema mostrara mensaje indicándole que debe contactarse con mesa de ayuda.
03	En el punto 4, en caso de que el usuario no ingrese ningún campo, el sistema mostrará un mensaje donde indique que es obligatorio ingresar el número de terminal y la oficina de operación.

REGLAS DE NEGOCIO (Opcional):**[RN01] Validación de usuario:**

Todo usuario debe de estar registrado en el LDAP.

[RN02] Realizar operación:

Cuando se realice cualquier operación se debe validar que el aplicativo se haya autorizado al inicio de las operaciones.

Anexo 3: Documento técnico

Este anexo fue mostrado en el punto 3.2.5 en la segunda etapa de planificación del presente trabajo Documento donde se detalla las tablas de base de datos que tiene el presente aplicativo de continuidad de negocio.

Modelo Físico c
V 2.0 (12/08/

Nombre de la Tabla		Actuación	Descripción de la Tabla				
ARCHIVO		Nueva	Almacena la información de los archivos de importación o exportación de datos				
Campo	Descripción del Campo	Actuación	Tipo	Validación / Default	Nombre de Índice	Tipo de Índice	Column
CODIGOARCHIVO	Código del archivo	Nuevo	NUMBER(6)	NOT NULL	PK_ARCHIVO	PK	CODIGOARCHIVO
TIPOARCHIVO	Tipo de archivo	Nuevo	NUMBER(1)	NOT NULL			
DESCRIPCION	Descripción del archivo	Nuevo	VARCHAR2(500)	NULL			
NOMBREARCHIVO	Nombre del archivo	Nuevo	VARCHAR2(50)	NOT NULL			
EXTENSION	Extensión del archivo	Nuevo	VARCHAR2(50)	NOT NULL			
RUTAENTRADA	Nombre del directorio de origen para la ruta de entrada del archivo	Nuevo	VARCHAR2(200)	NULL			
RUTASALIDA	Nombre del directorio de origen para la ruta de salida del archivo	Nuevo	VARCHAR2(200)	NULL			
TABLA	Nombre de la tabla a cargar	Nuevo	VARCHAR2(50)	NOT NULL			
SEQUENCE	Nombre de la secuencia a utilizar en el proceso de carga	Nuevo	VARCHAR2(50)	NULL			
CORREOELECTRONICO	Correo electrónico de los usuarios que se informará el resultado de la carga del archivo	Nuevo	VARCHAR2(500)	NULL			
PROGRAMACION_TIPO	Tipo de programación de la carga	Nuevo	VARCHAR2(2)	NULL			
PROGRAMACION_MES	Mes de programación de la carga	Nuevo	VARCHAR2(2)	NULL			
PROGRAMACION_DIA	Día de programación de la carga	Nuevo	VARCHAR2(2)	NULL			
PROGRAMACION_HORA	Hora de programación de la carga	Nuevo	VARCHAR2(2)	NULL			
PROGRAMACION_MINUTO	Minuto de programación de la carga	Nuevo	VARCHAR2(2)	NULL			
TIPO_PROCESO_ID	Código del tipo de proceso (importación o exportación)	Nuevo	NUMBER(38)	NULL			
FLAGACTIVO	Flag para indicar si es activo	Nuevo	NUMBER(1)	NULL			
USUARIOREGISTRO	Usuario de registro	Nuevo	VARCHAR2(30)	NOT NULL			
FECHAREGISTRO	Fecha de registro	Nuevo	DATE	NOT NULL			
USUARIOMODIFICACION	Usuario de modificación	Nuevo	VARCHAR2(30)	NOT NULL			
FECHAMODIFICACION	Fecha de modificación	Nuevo	DATE	NOT NULL			
PROC_LLENADO	Proceso que ejecutará la carga del llenado de datos	Nuevo	VARCHAR2(50)	NULL			
CONTADOR	Contador	Nuevo	NUMBER(38)	NULL			
Nombre de la Tabla		Actuación	Descripción de la Tabla				
ARCHIVO_EJECUCION		Nueva	Almacena la información de la ejecución de los archivos mediante el quartz				

Anexo 4: Modelo de procesos

Este anexo fue colado en el punto 3.2.5 en la segunda etapa de planificación del presente trabajo. Diagrama de procesos del ingreso al sistema de un perfil asesor en donde se detallan los pasos a seguir por el presente aplicativo a la hora de logearse en el sistema.

Ingreso al Sistema

