



FACULTAD DE INGENIERÍA DE SISTEMAS Y SOFTWARE

**“DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN SEGURA
DE COLAS EN LA ATENCIÓN AL CLIENTE DE LAS ENTIDADES
BANCARIAS”**

AUTORES:

PRADO AGUILAR, JAHIR EDEN

CALIZAYA HUAMÁN, EMILIO JORGE

TACURE SANCHEZ, TOMAS ANGEL RUBEN

CURSO:

Herramientas de Desarrollo

DOCENTE:

Amache Sanchez, Milton Freddy

Lima - Perú, 2025

AGRADECIMIENTO

Agradecemos profundamente a Dios por ser nuestro guía y fortaleza durante toda la carrera, y por permitirnos alcanzar esta meta. También reconocemos el invaluable apoyo de nuestras familias, cuyo amor, paciencia y sacrificio hicieron posible este logro, que es tanto suyo como nuestro. De igual modo, extendemos nuestro agradecimiento a todos nuestros profesores, cuya experiencia, sabiduría y dedicación enriquecieron nuestra formación académica y profesional, contribuyendo significativamente en nuestro crecimiento personal.

DEDICATORIA

A nuestra familia, por su amor y apoyo inquebrantable; a nuestros profesores, por guiarme con sabiduría; y a Dios, por ser nuestra fortaleza en este camino. Este logro es también de ustedes.

RESUMEN

El presente proyecto propone el desarrollo de un sistema de control de colas con acceso por QR y verificación biométrica cuando se realiza alguna operación dentro del sistema, este proyecto está enfocado en mejorar la eficiencia y seguridad en agentes bancarios del sistema financiero peruano. La problemática principal radica en las largas filas y los tiempos excesivos de espera que afectan especialmente a poblaciones vulnerables como adultos mayores y pensionistas, debido a procesos manuales de validación y gestión de operaciones. La solución planteada integra tres componentes clave: acceso mediante QR como alternativa a las tarjetas físicas, verificación biométrica para garantizar la seguridad en las transacciones, y un módulo de inteligencia artificial capaz de monitorear en tiempo real la cantidad de usuarios en la fila y el promedio de atención por persona. Con ello, se busca reducir la congestión en los agentes bancarios, optimizar los recursos disponibles y garantizar una experiencia de usuario más ágil, inclusiva y segura.

ABSTRACT

This project proposes the development of a queue management system with QR-based access and biometric identification, aimed at improving efficiency and security in banking agents within the Peruvian financial system. The main problem lies in long queues and excessive waiting times that particularly affect vulnerable populations such as the elderly and pensioners, due to manual processes of identity validation and transaction management. The proposed solution integrates three key components: QR-based access as an alternative to physical cards, biometric verification to ensure secure transactions, and an artificial intelligence module capable of monitoring in real time the number of users in line and the average service time per person. This approach seeks to reduce congestion in banking agents, optimize available resources, and provide a faster, more inclusive, and secure user experience.

ÍNDICE

REALIDAD PROBLEMÁTICA.....	4
JUSTIFICACIÓN.....	4
OBJETIVO GENERAL.....	5
OBJETIVOS ESPECÍFICOS.....	5
CRONOGRAMA DE ACTIVIDADES.....	6
MARCO TEÓRICO.....	8
Git.....	8
Github.....	9
MVC.....	9
JPA.....	9
Spring Boot.....	10
MySQL.....	10
Bootstrap.....	10
THYMELEAF.....	11
REQUERIMIENTOS FUNCIONALES.....	11
REQUERIMIENTOS NO FUNCIONALES.....	11
DIAGRAMA DE ACTIVIDADES.....	12
DIAGRAMA DE CASOS DE USO.....	13
ARQUITECTURA DEL SISTEMA.....	16
BASE DE DATOS.....	16
DISEÑO UX/UI.....	17
REFERENCIAS BIBLIOGRÁFICAS.....	20

ÍNDICE DE FIGURAS

Figura N°01: Tabla de tareas.....	6
Figura N°02: Hoja de recursos.....	6
Figura N°03: Cronograma grafico.....	7
Figura N°04: Diagrama de actividades.....	12
Figura N°05: El sistema debe permitir al administrador registrar un cliente y generar el código QR con todos sus datos personales y sus montos.....	13
Figura N°06: El sistema debe validar la identidad del cliente mediante el QR y la contraseña.....	14
Figura N°07: El sistema debe permitir realizar operaciones financieras básicas(depósitos, retiros y transferencias).....	14
Figura N°08: El sistema debe permitir al cliente consultar el historial de sus operaciones y transacciones realizadas asociadas a su código QR.....	15
Figura N°09: El sistema en el sistema se deben visualizar los tiempos de espera por cliente y la cantidad de clientes atendidos. El sistema usar algoritmos para monitorear el tiempo real de atención por cliente y la cantidad de clientes atendidos. El sistema debe generar alertas automáticas si el tiempo promedio de atención excede ciertos umbrales.....	15
Figura N°10: Arquitectura del sistema.....	16
Figura N° 11: Diagrama lógico de Base de Datos.....	16

REALIDAD PROBLEMÁTICA

En el sistema financiero peruano, particularmente entidades como Agente Digital Perú, los adultos mayores, pensionistas y beneficiarios se enfrentan a una problemática en común: las largas colas y tiempos de espera excesivos para realizar una operación en un agente bancario. Estas situaciones se suelen agravar porque la validación de identidad y la gestión de las operaciones suelen hacerse de manera manual y lenta, lo que genera congestión en los agentes y afecta en mayor medida a las personas más vulnerables. El tiempo que se pierde en la identificación y en el registro de cada transacción, sumado a la falta de información en tiempo real sobre la cantidad de usuarios en fila o el promedio de atención por persona, provoca que las agencias no puedan gestionar eficientemente sus recursos ni reducir los tiempos de espera. Por estas razones surge la necesidad de contar con un sistema que pueda agilizar la identificación del proceso y a la vez analice en tiempo real la cantidad de usuarios y el tiempo de atención que se tiene para cada uno, con el fin de optimizar la operación de los agentes y mejorar la experiencia de los clientes.

JUSTIFICACIÓN

Este proyecto se justifica por la urgente necesidad de humanizar y optimizar el sistema financiero peruano, específicamente para proteger a grupos vulnerables como adultos mayores, pensionistas y beneficiarios de Agente Digital Perú, quienes enfrentan diariamente largas colas y tiempos de espera excesivos debido a procesos manuales de identificación y validación que generan congestión y riesgos físicos. La exposición prolongada en condiciones inadecuadas los hace vulnerables a delitos, agravando su situación. Por ello, la iniciativa surge como una respuesta imperativa para garantizar un servicio eficiente, digno y seguro, eliminando barreras de acceso, reduciendo la exclusión financiera y transformando operacionalmente las entidades mediante la implementación de

un sistema inteligente que automatice procesos, optimice recursos y mitigue riesgos, cumpliendo así con una crucial responsabilidad social y ética.

OBJETIVO GENERAL

Desarrollar un sistema para la gestión segura de colas en la atención al cliente de las entidades bancarias.

OBJETIVOS ESPECÍFICOS

Analizar los requerimientos funcionales y no funcionales necesarios para implementar un sistema de control de colas en agentes bancarios, considerando seguridad, accesibilidad y experiencia del usuario.

Diseñar la arquitectura del sistema que integre el uso de códigos QR como medio de acceso y verificación biométrica para las transacciones u operaciones de los clientes.

Integrar un módulo de inteligencia artificial capaz de detectar en tiempo real la cantidad de usuarios en la fila y calcular el promedio de atención por persona, proporcionando datos para la optimización de recursos.

CRONOGRAMA DE ACTIVIDADES

Figura N°01: Tabla de tareas

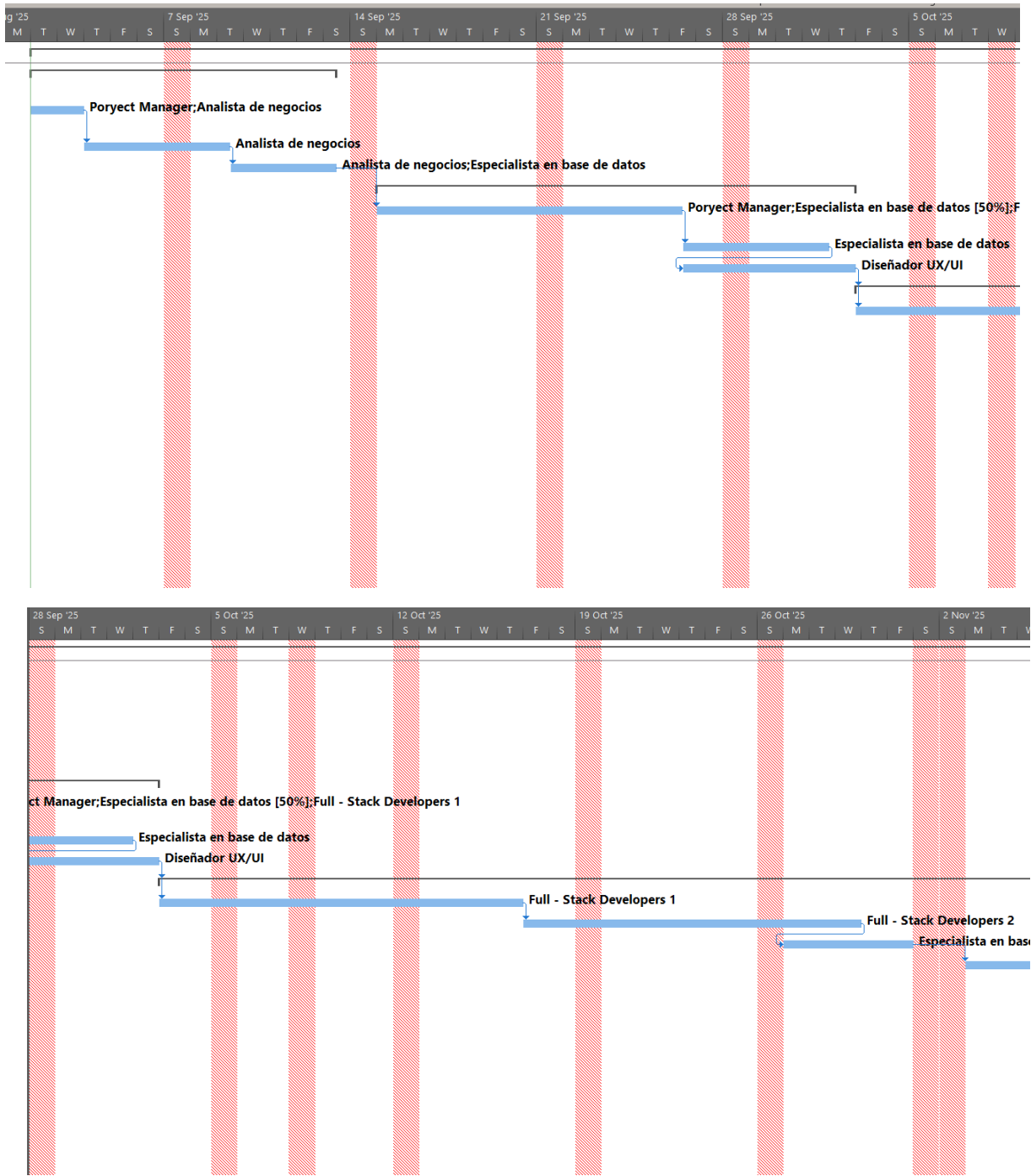
Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
	Proyecto Herramientas	76.5 days	Tue 2/09/25	Thu 11/12/25		
	Etapa 01: Analisis/levantamiento de requisitos	10 days	Tue 2/09/25	Sat 13/09/25		
	Reunion de inicio/ definir alcances	2 days	Tue 2/09/25	Wed 3/09/25		Poryect Manager; Analista de negocios
	Levantamiento de RF/RNF	4 days	Thu 4/09/25	Tue 9/09/25	3	Analista de negocios
	Diseño de flujos	4 days	Tue 9/09/25	Sat 13/09/25	4	Analista de negocios;Especi
	Etapa 02: Diseño del sistema	15 days	Mon 15/09/25	Thu 2/10/25		
	Diseño y arquitectura del sistema	10 days	Mon 15/09/25	Fri 26/09/25	5	Poryect Manager; Especialista en base de
	Diseño de base de datos	4 days	Fri 26/09/25	Wed 1/10/25	7FS-50%	Especialista en base de dato
	Diseño UX/UI	5 days	Fri 26/09/25	Thu 2/10/25	8FS-100%	Diseñador UX/UI
	Etapa 03: Desarrollo del sistema	35 days	Fri 3/10/25	Tue 18/11/25	9	
	Desarrollo de backend	10 days	Fri 3/10/25	Thu 16/10/25	9FS-50%	Full - Stack Developers 1
	Desarrollo de Frontend	10 days	Fri 17/10/25	Wed 29/10/25	11	Full - Stack Developers 2
	Integración de base de datos	5 days	Mon 27/10/25	Fri 31/10/25	12FS-30%	Especialista en base de dato
	Modo de vision por computador	10 days	Mon 3/11/25	Fri 14/11/25	13	Ingeniero de Visión por Cor
	Integración hardware	8 days	Sat 8/11/25	Tue 18/11/25	14FS-50%	Ingeniero Embedded
	Etapa 04:Prueba de validación	10.5 days	Wed 19/11/25	Tue 2/12/25		
	Pruebas de integración	5 days	Wed 19/11/25	Tue 25/11/25	15	Full - Stack Developers 2;Te
	Prubas funcionales	5 days	Tue 25/11/25	Mon 1/12/25	17	Operadores de sorpote 1;T
	Peubas de rendimiento/seguridad	3 days	Fri 28/11/25	Tue 2/12/25	18FS-50%	Ingeniero de Visión por Computador
	Etapa 05: Implementación	6 days	Tue 2/12/25	Thu 11/12/25		
	Implementación piloto	3 days	Tue 2/12/25	Fri 5/12/25	19	Operadores de sorpote 1;O
	Ajustes finales y cierre del proyecto	3 days	Fri 5/12/25	Thu 11/12/25	21	Analista de negocios; Poryect Manager

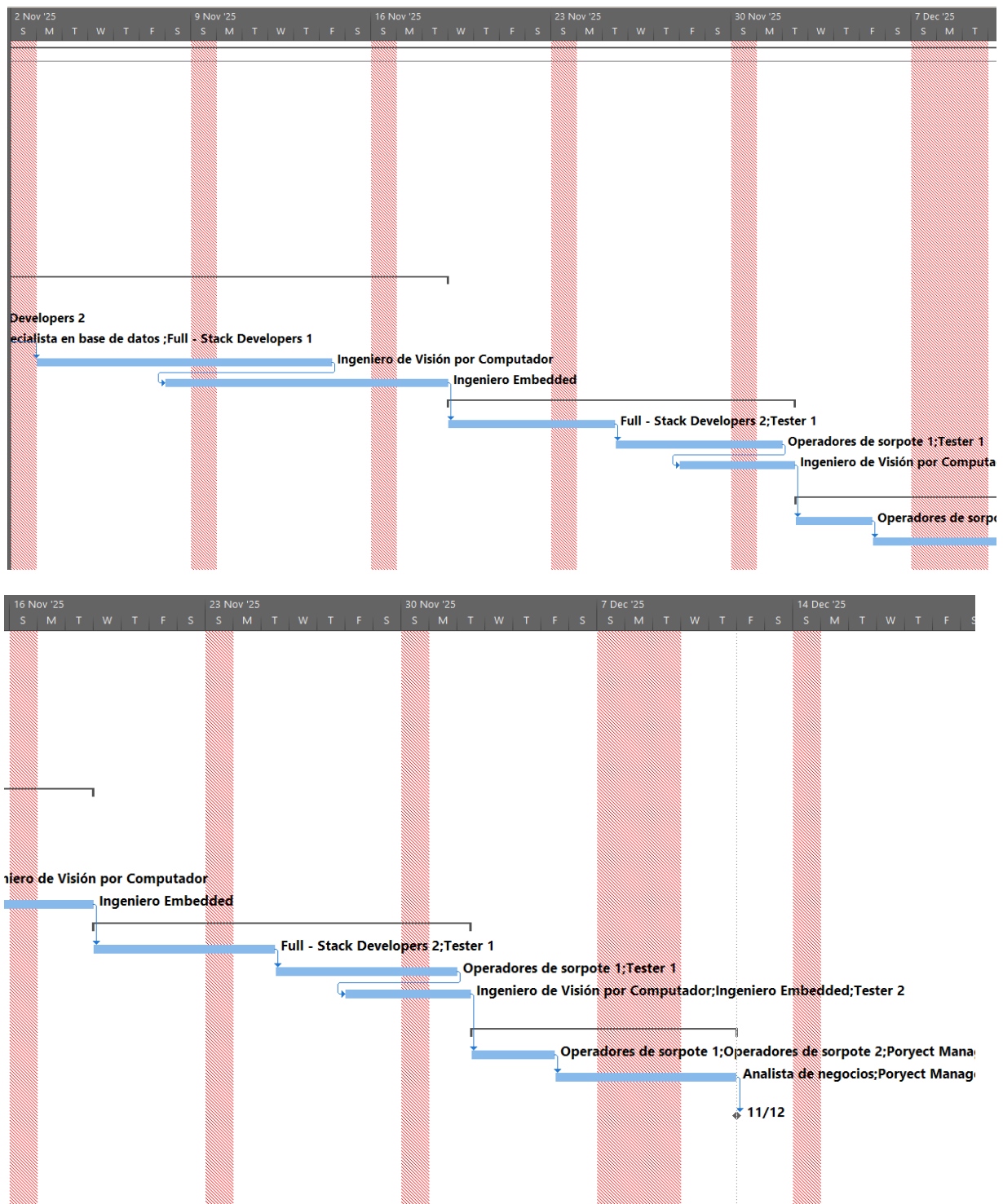
Figura N°02: Hoja de recursos

Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue	Base	Cd
Poryect Manager	Work		PM		100%	S/ 50.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Analista de negocios	Work		AN		100%	S/ 40.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Diseñador UX/UI	Work		UX		100%	S/ 30.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Full - Stack Developers 1	Work		FS		100%	S/ 40.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Full - Stack Developers 2	Work		FS		100%	S/ 40.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Especialista en base de datos	Work		BD		100%	S/ 45.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Ingeniero de Visión por Computador	Work		CV		100%	S/ 50.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Ingeniero Embedded	Work		IOT		100%	S/ 35.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Tester 1	Work		QA		100%	S/ 30.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Tester 2	Work		T		100%	S/ 30.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Operadores de sorpote 1	Work		OP		100%	S/ 20.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	
Operadores de sorpote 2	Work		OP		100%	S/ 20.00/hr	S/ 0.00/hr	S/ 0.00	Prorated	ProyectoHerrai	

Fuente: Elaboración Propia

Figura N°03: Cronograma grafico





Fuente: Elaboración Propia

MARCO TEÓRICO

Git

Según Astigarraga y Cruz-Alonso (2022) es un sistema de control de versiones, permite monitorear el progreso de un proyecto capturando sucesivos estados del mismo. Este mecanismo posibilita rastrear de manera precisa el historial de cambios y el progreso

completo del trabajo.

Git se emplea mediante un ciclo de trabajo donde los cambios se preparan mediante el comando "git add" y luego se guardan instantáneamente con "git commit" dentro de la rama presente, permitiendo desarrollar funcionalidades de forma aislada a la rama principal. Asimismo, sus ventajas incluyen la trazabilidad completa del historial, la capacidad de trabajar en equipo de manera paralela y sin interferencias, la operación local sin depender de internet y la seguridad sobre cada modificación o cambio realizado.

Para el desarrollo del sistema web de gestión segura de colas en entidades bancarias, se utilizará Git con el objetivo de optimizar el trabajo colaborativo, mantener un historial de cambios organizado y garantizar la trazabilidad de todas las modificaciones realizadas.

Github

GitHub es una de las plataformas de desarrollo colaborativo más importantes a nivel global. En ella, los usuarios pueden almacenar y compartir sus proyectos, los cuales superan los 96 millones. La plataforma registra automáticamente las actividades de los usuarios, como la creación de proyectos, la copia de estos o la adición de nuevos archivos, lo que facilita el control de versiones. Además de estas funciones técnicas, GitHub promueve la interacción social al permitir que los usuarios muestren su aprecio por los proyectos o se suscriban para seguir sus actualizaciones (Al-Rubaye y Sukthakar, 2020).

La plataforma Github se utiliza para gestionar el ciclo de vida del software mediante repositorios cloud donde se almacena el código, utilizando ramas para aislar características y pull requests para integrarlas de forma controlada. Esto ofrece ventajas clave como el control preciso de versiones, colaboración simultánea sin conflictos, revisión sistemática de código y despliegue continuo, esenciales para proyectos escalables.

La implementación de GitHub en el desarrollo de nuestro Sistema Web es fundamental, ya que facilitará la colaboración remota del equipo, permitiendo un trabajo simultáneo y organizado mediante el control de versiones, lo que optimizará el ciclo de desarrollo del proyecto.

MVC

Según Enriquez et al., (2023) señala que el MVC es un patrón arquitectónico que plantea dividir el aplicativo en módulos llamados modelo, vista y controlador, con la finalidad de que estén claramente diferenciados y tengan una estructura definida. Asimismo, la separación en módulos ayuda a crear un software fuerte, mantenible y escalable.

El patrón arquitectónico MVC se emplea mediante el Modelo que representa la lógica de negocio y el acceso a datos, la Vista que es la interfaz del usuario (UI) y el Controlador que hace el rol de intermediario entre la Vista y el Modelo. Sus ventajas son: la separación de responsabilidades, mantenimiento del código y el crecimiento del aplicativo.

El uso del patrón MVC en el presente Proyecto Web es primordial para el desarrollo del mismo, porque nos facilitará la implementación de nuevos módulos, facilitar la corrección de errores y facilitar el trabajo en equipo.

JPA

Spring Data JPA es un componente del ecosistema Spring que facilita el acceso y la

gestión de datos persistentes en aplicaciones Java, basándose en la especificación JPA para el mapeo objeto-relacional (Darzu, 2024).

El uso de JPA es para simplificar el acceso y manipulación de datos en aplicaciones Java, a través de repositorios, sin necesidad de escribir consultas SQL complejas. Sus principales ventajas son la reducción significativa de código, una integración fluida con el ecosistema de Spring, consultas derivadas del nombre de los métodos y facilitar el desarrollo de capas de persistencia.

En el sistema web que se desarrollara, se optó por utilizar Spring Data JPA con el propósito de simplificar el acceso y la gestión de datos dentro de la aplicación, aprovechando el mapeo objeto-relacional (ORM) y el uso de repositorios que automatizan las operaciones.

Spring Boot

Spring Boot es un módulo de Spring que permite, de manera sencilla, crear aplicaciones autónomas y listas para producción basadas en Java, eliminando gran parte de la configuración manual.” (Ramírez Pérez, 2020)

Esta cita muestra la ventaja clave de Spring Boot que es contar con una mejor configuración y un mejor arranque en la aplicaciones. En nuestro proyecto tener un framework para realizar las funcionalidades como operaciones, autenticación por QR y monitoreo de colas, es una gran ventaja y nos permite solo enfocarnos en la lógica.

También una tesis de fin de máster analizó Spring Boot como framework de aplicaciones web empresariales y resaltó su impacto positivo en la productividad y uso de librerías incluidas.

MySQL

"MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que se utiliza para almacenar y gestionar datos. Su fiabilidad, rendimiento, escalabilidad y facilidad de uso lo convierten en una opción popular para los desarrolladores.” (Erickson, 2024)

Esta cita nos reafirma nuestra elección de mysql para nuestro proyecto. En el entorno bancario que estamos desarrollando, donde se manejan múltiples clientes, cuentas, transacciones y altos volúmenes de operaciones, necesitamos un sistema que ofrezca consistencia, escalabilidad y buen rendimiento. Mysql cumple perfectamente estas exigencias

En nuestro sistema, MySQL será la capa de gestión de datos donde residirán todas las entidades críticas como cliente, cuenta, tipo de cuenta, tipo de transacción, transacción, empleado y sus relaciones.

Bootstrap

“Bootstrap es un framework de desarrollo web gratuito y de código abierto. Está diseñado para facilitar el proceso de desarrollo de los sitios web responsivos y orientados a los dispositivos móviles, proporcionando una colección de sintaxis para diseños de plantillas.” (Deyimar A., 2025)

Para nuestro sistema bancario la elección de Bootstrap nos parece muy adecuada

porque necesitamos que la aplicación web sea usable en múltiples dispositivos como computadores, tabletas, teléfonos móviles; y que tenga una apariencia profesional y consistente sin tener que diseñar todos los componentes desde cero.

En la nuestra plataforma la usaremos para construir la interfaz de nuestro proyecto. Asimismo con esto podremos asegurar que se adapte a los diferentes tamaños de los dispositivos.

Thymeleaf

“Thymeleaf es un moderno motor de plantillas del lado del servidor para entornos web y autónomos, capaz de procesar HTML, XML, JavaScript, CSS e incluso texto plano.” (Thymeleaf, 2018).

Como lo indica en esta cita Thymeleaf es una herramienta clave porque conecta directamente con el backend como lo puede ser spring boot y nos permite generar vistas HTML dinámicas desde los datos de nuestro sistema.

En nuestro sistema web lo usaremos para gestionar los distintos apartados que tenemos como lo son registro de clientes, operaciones, autenticación, servicios y visualización de los empleados.

Spring Rest APIs

REST o Representational State Transfer es un estilo de Arquitectura al momento de realizar una comunicación entre cliente y servidor (Cecilio Álvarez, 2023) que nos permite acciones adecuadas (GET, POST, PUT, DELETE, y otras), almacenamiento en caché, redirección y envío, seguridad (cifrado y autenticación).

Se utilizará para desarrollar los servicios backend responsables de la comunicación entre los diferentes módulos del sistema. Permitirá gestionar el registro y acceso mediante código QR y validación biométrica, así como el módulo de monitoreo con inteligencia artificial para la detección de personas y cálculo del tiempo de atención.

Ofrece alta escalabilidad y un mantenimiento sencillo. Además, su uso del protocolo HTTP estándar asegura compatibilidad con diferentes lenguajes y plataformas, favoreciendo un ecosistema flexible y moderno.

Spring Security

Spring Security Framework es un marco de Java de código abierto robusto, altamente personalizable, completo y extensible que admite autenticación y autorización (Andreas, 2022).

Este framework nos permitirá proteger los endpoints REST contra accesos no autorizados, implementar roles y privilegios (Administrador, usuario), integrar la autenticación biométrica y el QR como métodos de acceso seguro.

Este proporciona un alto nivel de seguridad mediante una gestión avanzada de autenticación y autorización. Su integración nativa con Spring Boot simplifica el desarrollo y la administración de sesiones. Además, ofrece protección contra ataques comunes como CSRF, XSS o inyección de código, fortaleciendo la defensa general del sistema.

Java

Java es un lenguaje de programación orientado a objetos, multiplataforma ya que se ejecuta en diversos dispositivos (Zuleyka Mesa, 2022), además posee una amplia documentación de manuales en el cual se muestran sus funciones y prestaciones de las diferentes APIS de programación.

Java será el lenguaje principal para el desarrollo del backend. Con él se implementarán los servicios REST y las conexiones con la base de datos

Es un lenguaje robusto y orientado a objetos. A Cuenta con un extenso ecosistema de librerías y frameworks, como Spring Boot, Hibernate y Maven. Su portabilidad, al ejecutarse en cualquier entorno con JVM, lo convierte en una opción versátil y duradera.

Python

Es un lenguaje de programación informático que se utiliza para crear sitios web y software, automatizar tareas y realizar análisis de datos (Coursera Staff, 2023), al ser un lenguaje de propósito general, puede ser utilizado para crear una variedad de programas diferentes y no está especializado en ningún problema en específico.

Python será empleado en el módulo de inteligencia artificial, encargado de procesar y analizar datos provenientes de video e imágenes. Este módulo permitirá detectar el número de personas en la fila mediante técnicas de visión computacional y calcular el tiempo promedio de atención de los usuarios. Los resultados generados serán enviados al backend desarrollado con Spring REST para su almacenamiento y visualización.

Dispone de una amplia gama de librerías para inteligencia artificial y visión por computadora. Su sintaxis sencilla y claridad en el desarrollo facilitan la implementación de algoritmos de análisis. Además, presenta una gran interoperabilidad con otros lenguajes, lo que permite integrarse fácilmente con Java a través de servicios REST.

REQUERIMIENTOS FUNCIONALES

1. El sistema debe permitir al administrador registrar un cliente y generar el código QR con todos sus datos personales y sus montos.
2. El sistema debe validar la identidad del cliente mediante el QR y la contraseña
3. El sistema debe permitir realizar operaciones financieras básicas(depósitos, retiros y transferencias).
4. El sistema debe permitir al cliente consultar el historial de sus operaciones y transacciones realizadas asociadas a su código QR
5. En el sistema se deben visualizar los tiempos de espera por cliente y la cantidad de clientes atendidos.
6. El sistema usa algoritmos para monitorear el tiempo real de atención por cliente y la cantidad de clientes atendidos.
7. El sistema debe generar alertas automáticas si el tiempo promedio de atención excede ciertos umbrales.

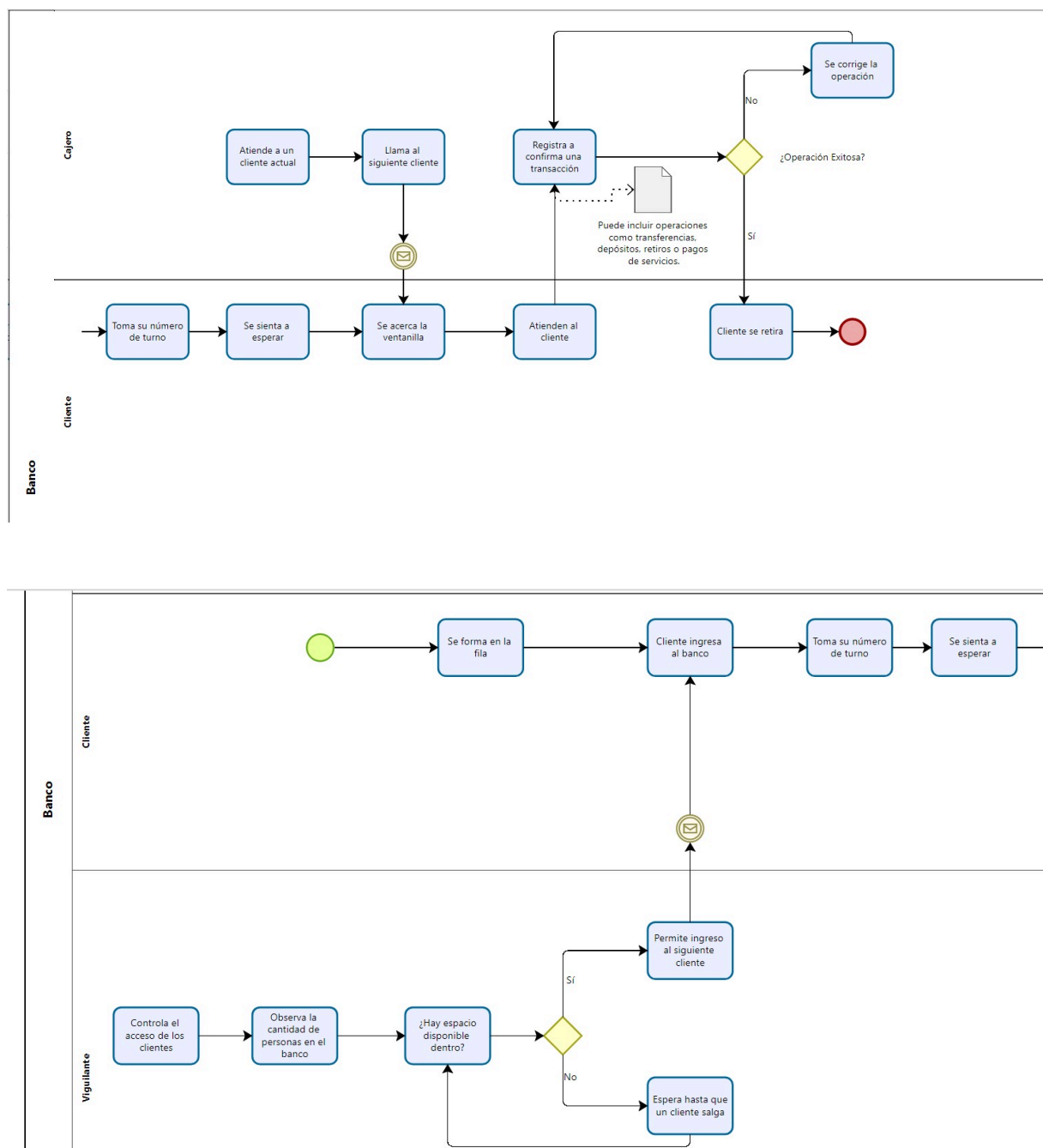
REQUERIMIENTOS NO FUNCIONALES

1. El sistema debe cifrar las contraseñas y datos sensibles.

2. El sistema debe requerir autenticación antes de permitir cualquier transacción financiera.
3. El sistema debe estar disponible al menos el 99% del tiempo operativo.
4. La interfaz del sistema debe ser intuitiva y fácil de usar.
5. El sistema debe ser accesible desde navegadores modernos y dispositivos con pantallas de distintos tamaños.
6. El sistema debe ofrecer mensajes de error claros y orientativos ante entradas inválidas
7. El código debe seguir buenas prácticas de programación

DIAGRAMA DE ACTIVIDADES

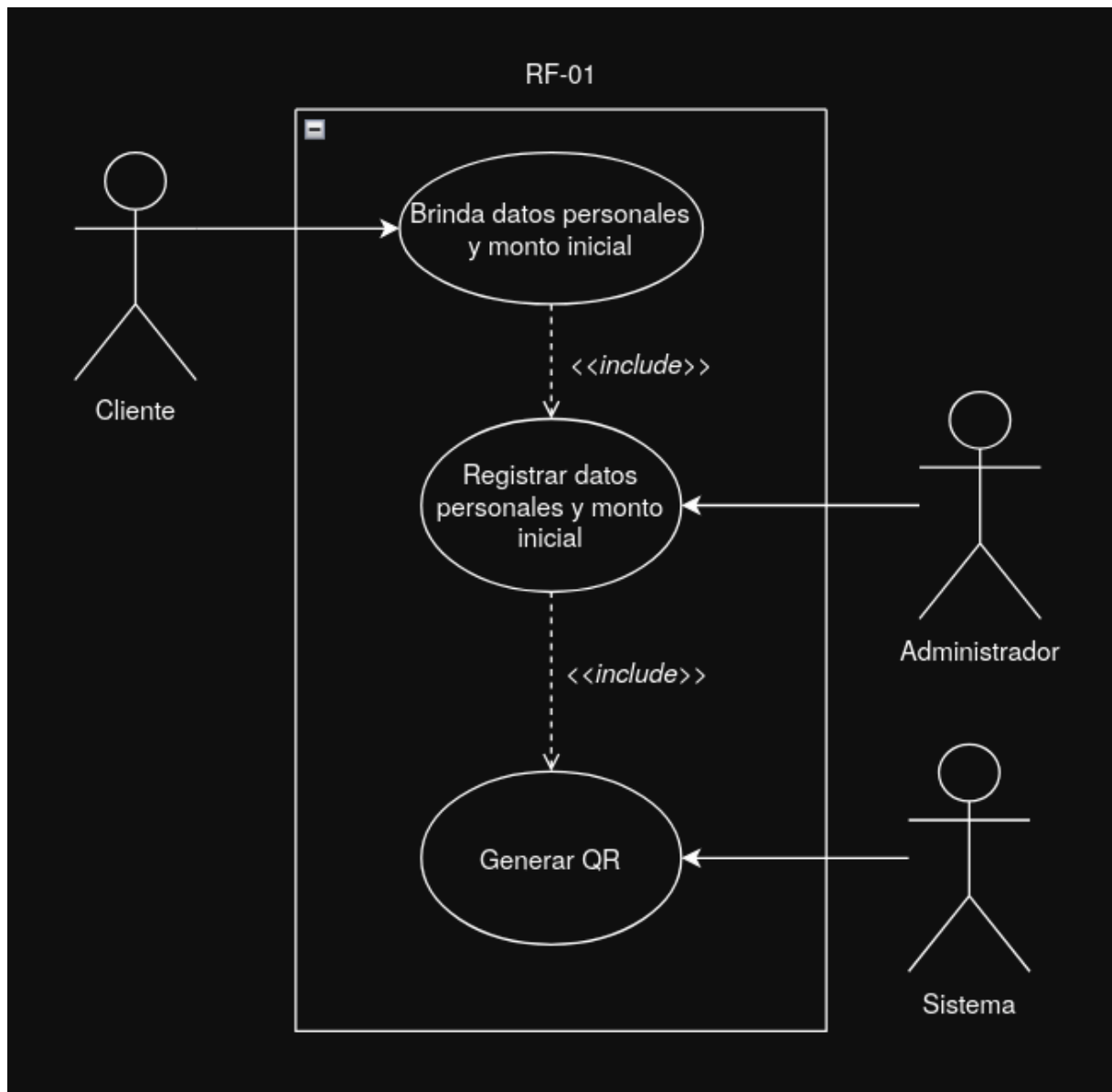
Figura N°04: Diagrama de actividades



Fuente: Elaboración Propia

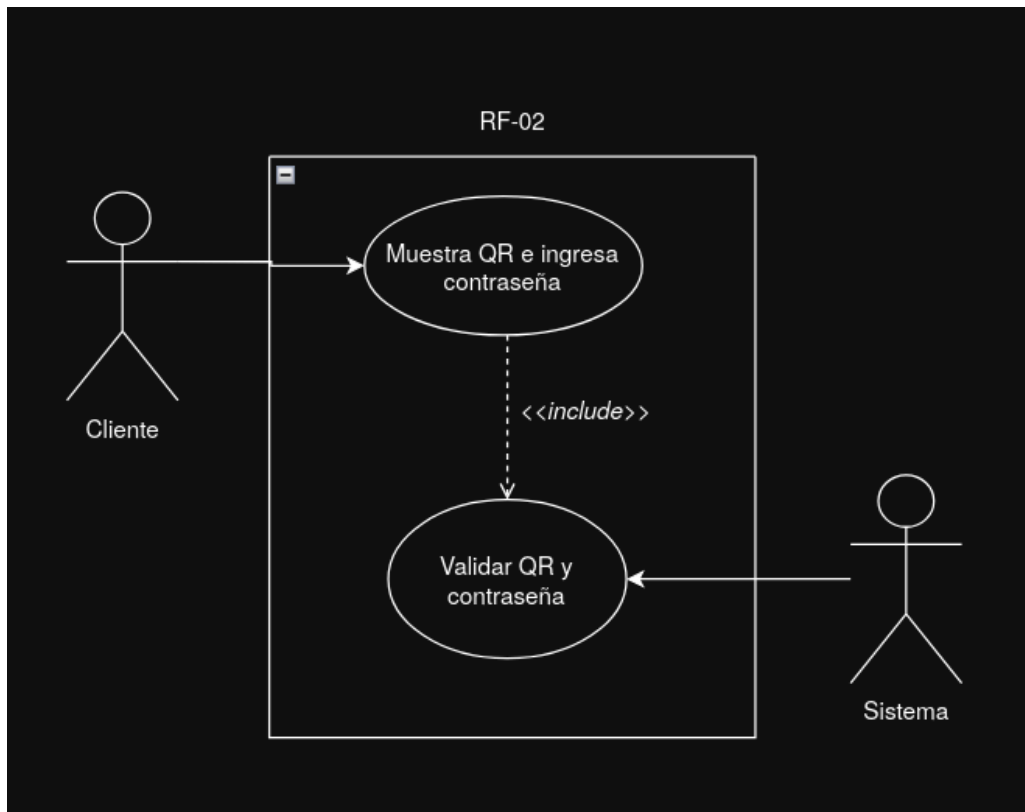
DIAGRAMA DE CASOS DE USO

Figura N°05: DCU - RF:01



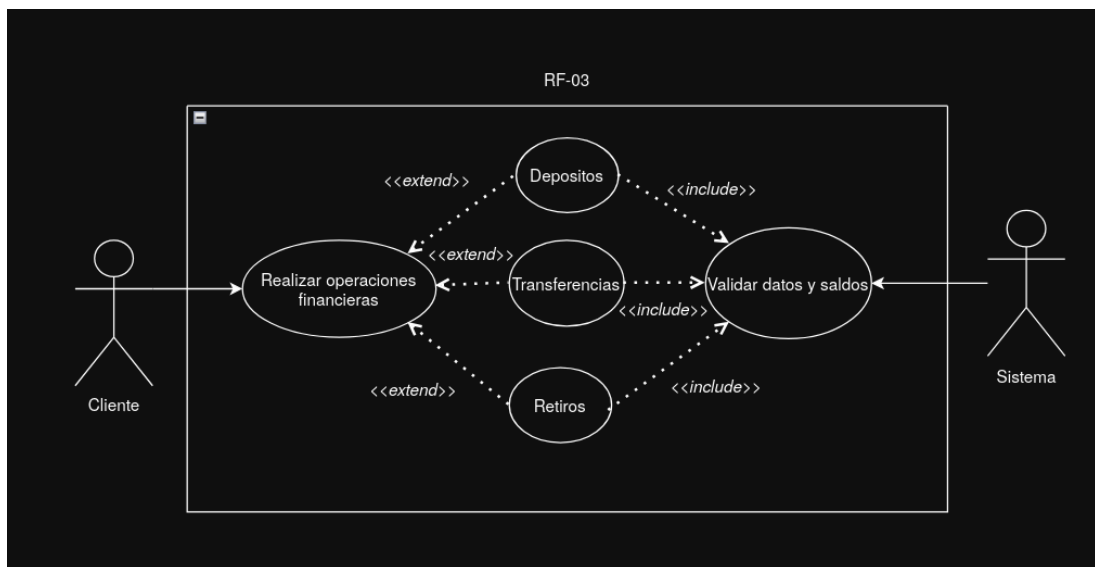
Fuente: Elaboración Propia

Figura N°06: DCU - RF:02



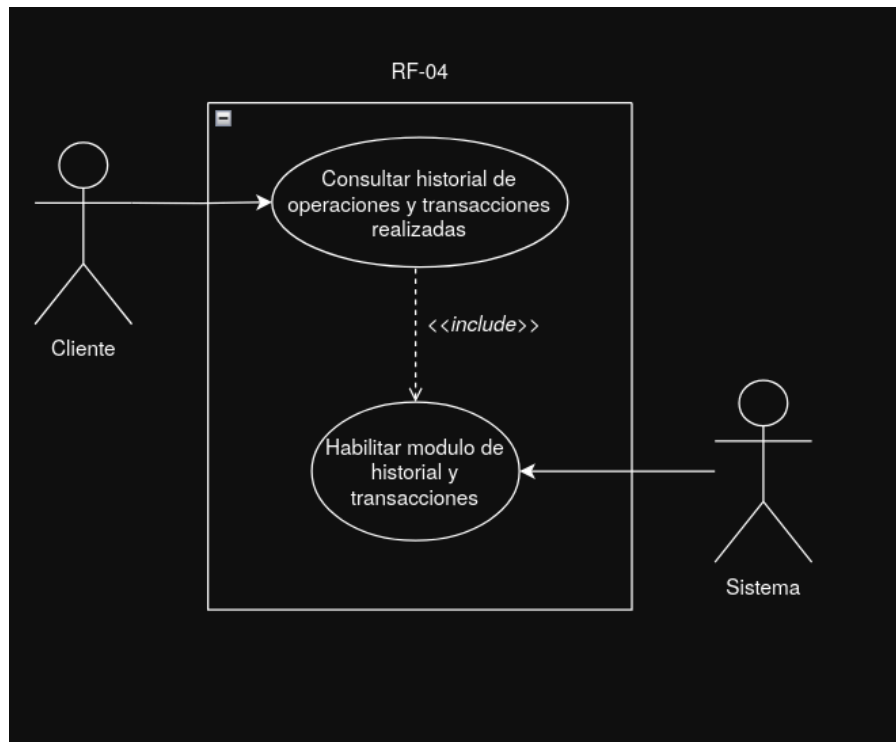
Fuente: Elaboración Propia

Figura N°07: DCU - RF:03



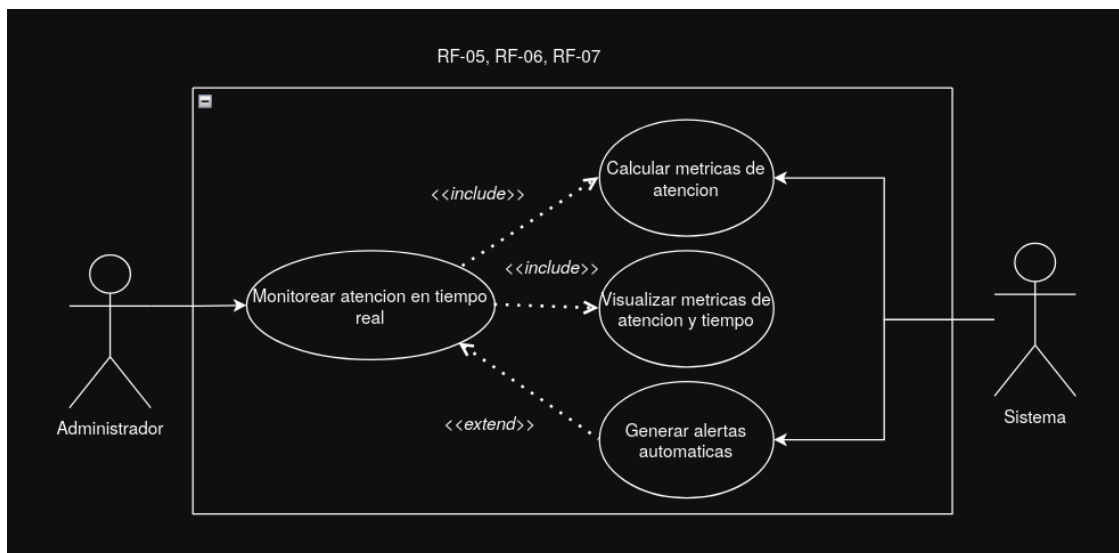
Fuente: Elaboración Propia

Figura N°08: DCU - RF:04



Fuente: Elaboración Propia

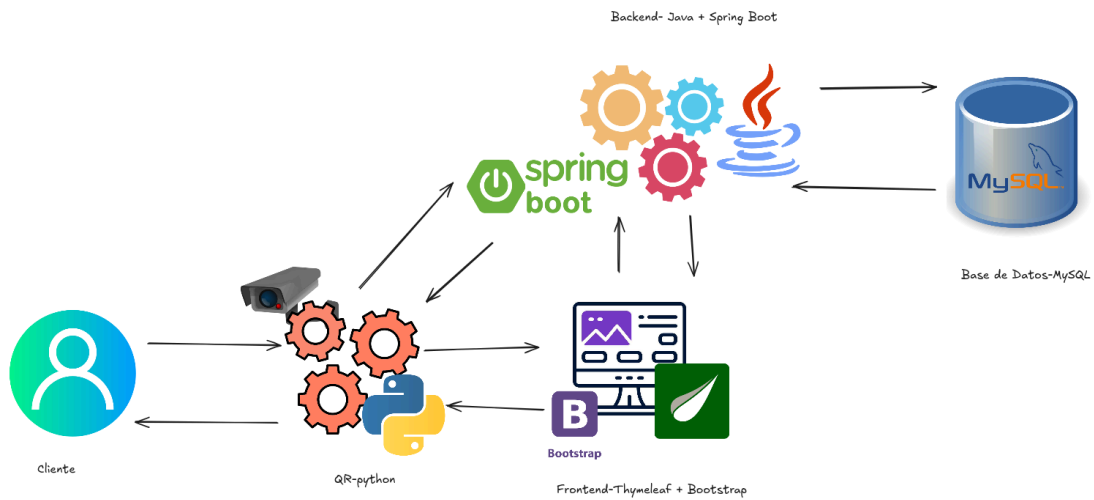
Figura N°09: DCU - RF:05, RF:06, RF, 07



Fuente: Elaboración Propia

ARQUITECTURA DEL SISTEMA

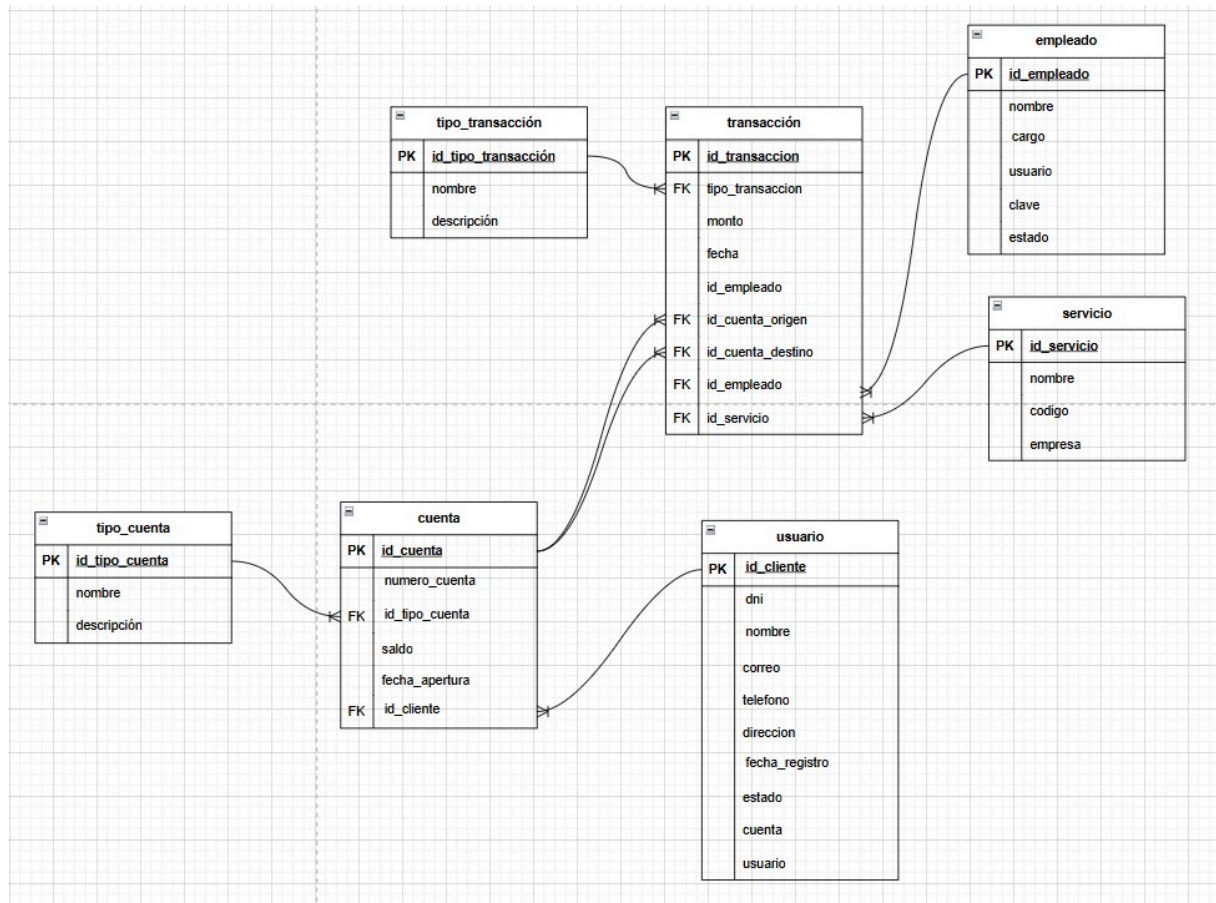
Figura N°10: Arquitectura del sistema



Fuente: Elaboración Propia

BASE DE DATOS

Figura N° 11: Diagrama lógico de Base de Datos



DISEÑO UX/UI

Figura N°12: Prototipo RF-01

El prototipo RF-01 muestra la pantalla de 'Registrar Cliente' para el administrador de Banco X. A la izquierda hay un menú lateral con opciones para el administrador y el usuario. El título principal es 'Registrar Cliente' y el saludo es 'BIENVENIDO ADMINISTRADOR'. El formulario contiene campos para Nombre, Apellido, Fecha de Nacimiento, Contraseña, Sexo y DNI. Hay un botón 'Generar QR' y una imagen de un código QR.

Banco X

ADMINISTRADOR

- Registrar Cliente
- Tiempos de Espera
- Alertas Generadas

USUARIO

- Validar Identidad
- Operaciones
- Historial

Registrar Cliente

BIENVENIDO ADMINISTRADOR

Nombre

Apellido

Fecha de Nacimiento

Contraseña

Sexo

DNI

Generar QR



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1&p=f&t=iLdtPNOUFDmzWyhN-0>

Figura N°15: Prototipo RF-02

El prototipo RF-02 muestra la pantalla de 'Validar Identidad' para el usuario de Banco X. A la izquierda hay un menú lateral con opciones para el administrador y el usuario. El título principal es 'Validar Identidad' y el saludo es 'BIENVENIDO USUARIO'. El formulario contiene un campo para 'Ingrese el QR' (con una imagen de un código QR) y un campo para 'Ingrese su Contraseña'. Hay un botón 'Activar Windows'.

Banco X

ADMINISTRADOR

- Registrar Cliente
- Tiempos de Espera
- Alertas Generadas

USUARIO

- Validar Identidad
- Operaciones
- Historial

Validar Identidad

BIENVENIDO USUARIO

Ingrese el QR



Ingrese su Contraseña

Activar Windows
Ve a Configuración para activar Windows.

Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1>

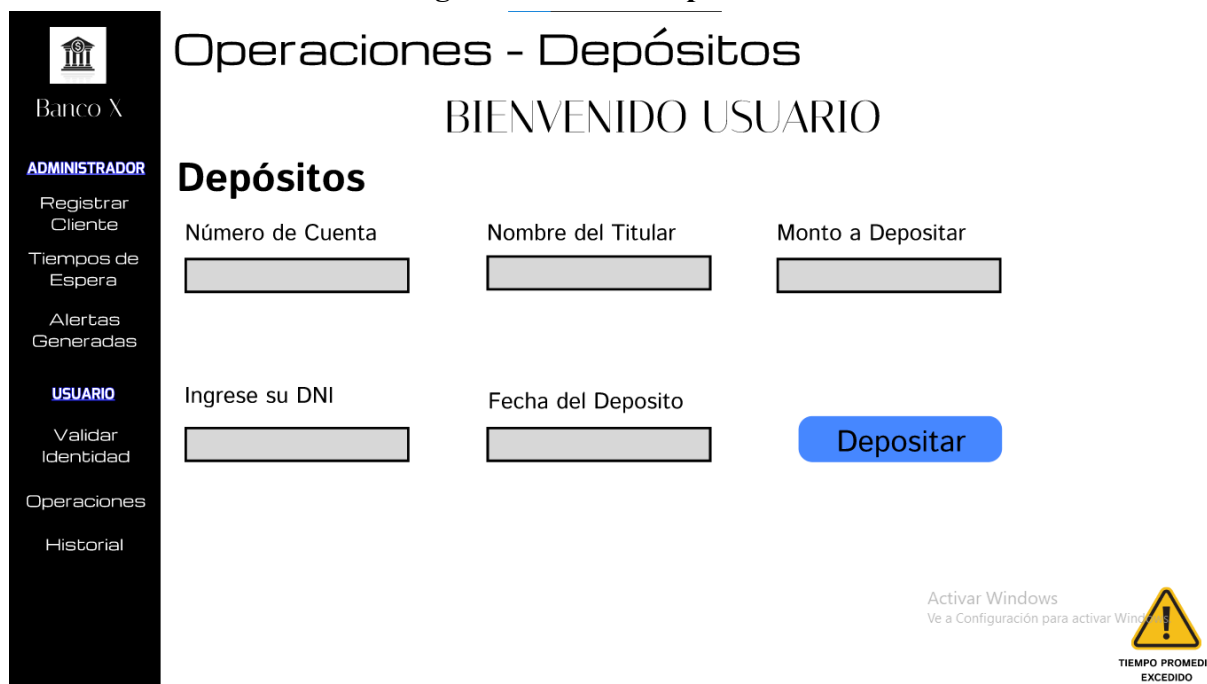
Figura N°16: Prototipo RF-03



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1&p=f&t=iLdtPNOUFDmzWyhN-0>

Figura N°17: Prototipo RF-03



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1>

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1&p=f&t=iLdtPNOUFDmzWyhN-0>

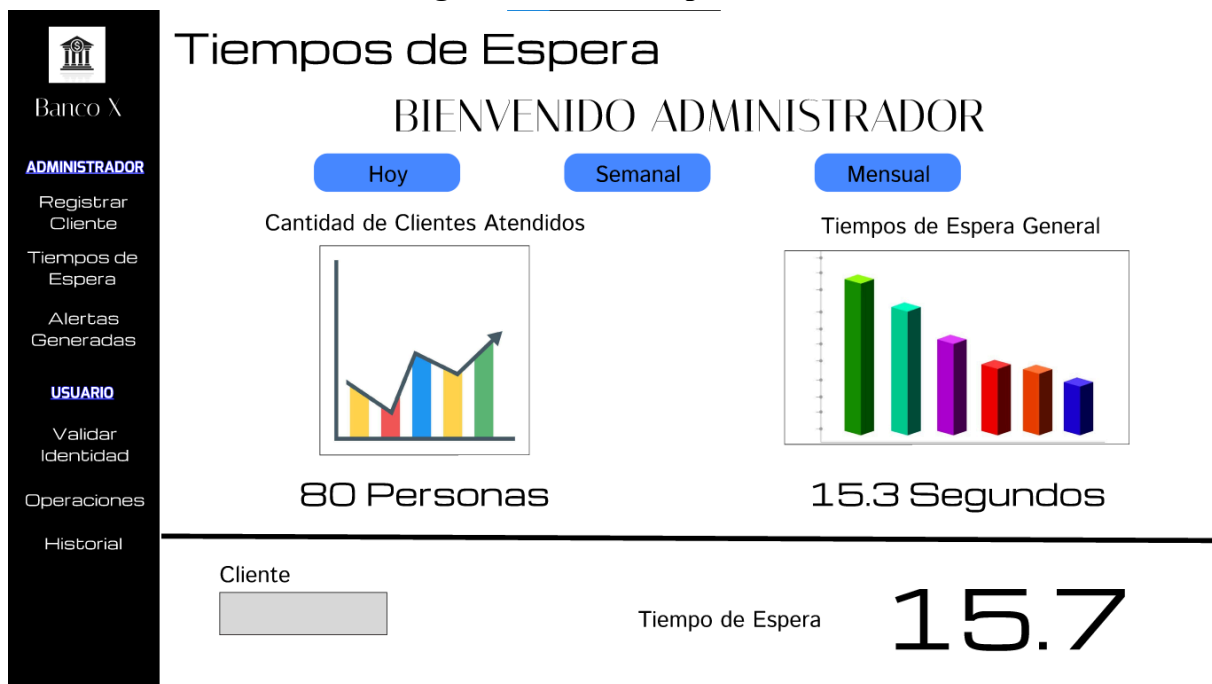
Figura N°19: Prototipo RF-04



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1&p=f&t=iLdtPNOUFDmzWyhN-0>

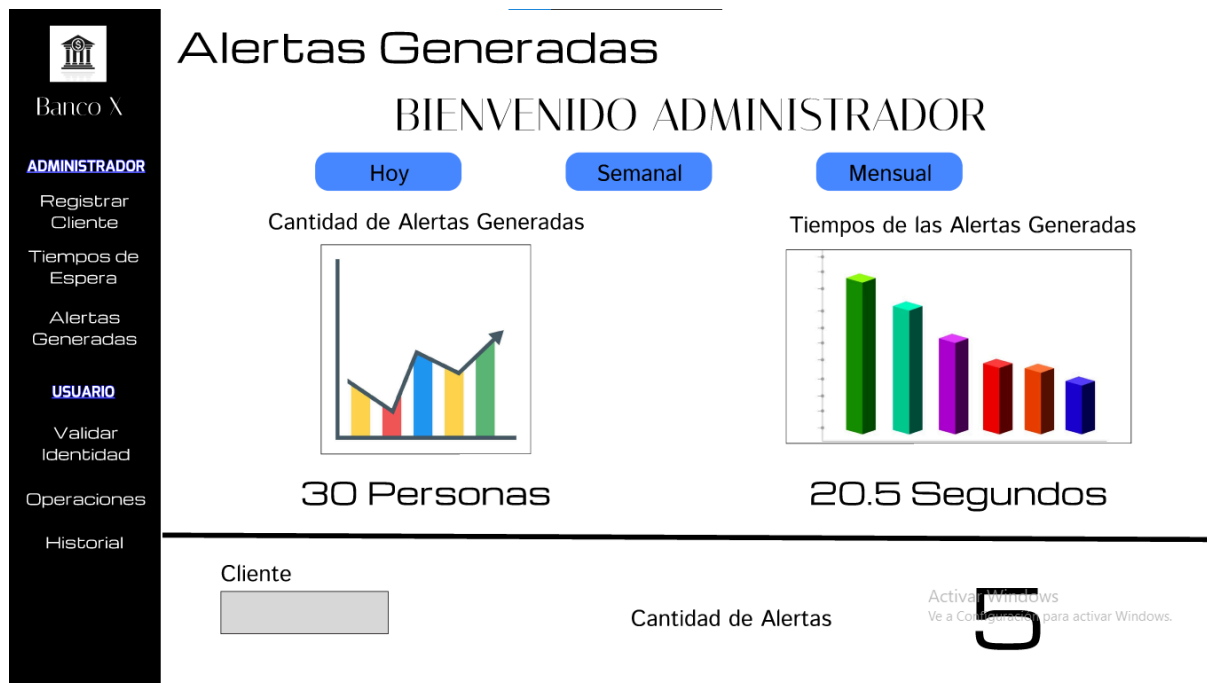
Figura N°13: Prototipo RF-05



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1>

Figura N°14: Prototipo RF-07



Fuente:

<https://www.figma.com/design/GUrugBlTgm3xSUXs3ZuAGg/Prototipado--HD-?node-id=0-1&p=f&t=iLdtPNOUFDmzWyhN-0>

REFERENCIAS BIBLIOGRÁFICAS

- Al-Rubaye, A., & Sukthankar, G. (2020, December). Scoring popularity in GitHub. In 2020 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 217-223). IEEE. <https://arxiv.org/pdf/2011.04865v1>
- Astigarraga, J., & Cruz-Alonso, V. (2022). ¿ Se puede entender cómo funcionan Git y GitHub!. Ecosistemas, 31(1), 2332-2332. <https://www.revistaecosistemas.net/index.php/ecosistemas/article/download/2332/1505>
- Cecilio Álvarez Caules (2023, marzo 07). ¿Que es REST?. <https://www.arquitecturajava.com/que-es-rest/>
- Chatterjee, A., & Prinz, A. (2022). Aplicación de Spring Security Framework con OAuth2 basado en KeyCloak para proteger las API de la arquitectura de microservicios: un estudio de caso. sensors. <https://www.mdpi.com/1424-8220/22/5/1703>
- Coursera Staff (2023). ¿Qué es Python y para qué se usa?. <https://www.coursera.org/mx/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- DARZU, I. B. (2024). Spring Data JPA. <https://repository.utm.md/bitstream/handle/5014/27940/Conf-TehStiint-UTM-StudMastDoct-2024-V1-p626-629.pdf?sequence=1&isAllowed=y>
- Deyimar, A. (2025, julio 10). ¿Qué es Bootstrap? – Una guía para principiantes. Glosario.https://www.hostinger.com/es/tutoriales/que-es-bootstrap?utm_source
- Enríquez, F., Fierro, S., Flores, B., Esparza, D. I., & Michelena, J. (2023). Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo de vida. EASI: Ingeniería y Ciencias Aplicadas en la Industria, 2(1), 11-16. <https://revistas.ug.edu.ec/index.php/easi/en/article/view/821/2033>
- Erickson, J. (2024, agosto 29). MySQL: qué es y cómo se usa. Oracle España – Base de datos HeatWave.https://www.oracle.com/es/mysql/what-is-mysql/?utm_source
- Ramírez Pérez, S. (2020). Estudio del framework Spring, Spring Boot y microservicios [Trabajo de fin de máster, Universidad de Alcalá]. e_Buah Repositorio Institucional. https://ebuah.uah.es/xmlui/handle/10017/45107?utm_source
- Thymeleaf. (2018, octubre 29). Tutorial: Using Thymeleaf (Versión 3.0.11.RELEASE). https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html?utm_source

Zuleyka Calzado (2022). Proyecto de codificación industrial en la gestión de inventarios.
<https://www.redalyc.org/journal/1815/181572159007/181572159007.pdf>