**PONTIFICIA UNIVERSIDAD JAVERIANA**

**INGENIERÍA DE SISTEMAS**

**FUNDAMENTOS DE INGENIERÍA DE SOFTWARE**

**GESTIÓN DE BANCO: PMP**

Imagen que contiene Patrón de fondo

El contenido generado por IA puede ser incorrecto.

**INTEGRANTES:**

**Santiago Andrés Mesa N.**

**Luis Enrique Santos Marulanda**

**Damián Rey Salcedo**

**Juan Diego Barreto Castañeda**

**PROFESOR:**

**Jaime Andrés Pavlich Mariscal.**

# Historial de Cambios

|  |  |  |
| --- | --- | --- |
| Fecha del cambio | Descripción del cambio | Responsable |
| 1/03/2025 | Creación básica del documento | Santiago Mesa |
| 1/03/2025 | Propuesta del proyecto | Santiago Mesa |
| 1/03/2025 | Organigrama | Santiago Mesa |
| 1/03/2025 | Modelo de Ciclo de Vida | Santiago Mesa |
| 1/03/2025 | Ambiente de trabajo | Santiago Mesa |
| 3/03/2025 | Tabla de Contenidos | Damian Rey |
| 3/03/2025 | Inicio introducción | Damian Rey |
| 3/03/2025 | Interfaces Externas | Damian Rey |
| 6/03/2025 | Monitoreo y Control de Progreso | Damian Rey |
| 9/03/2025 | Lenguajes y Herramientas | Luis Santos |
| 12/03/2025 | Correcciones | Luis Santos |
| 12/03/2025 | Administración de Configuración y Documentación | Santiago Mesa |
| 12/03/2025 | Elaboración Introducción | Damian Rey |
| 12/03/2025 | Tabla de Contenidos actualizada | Damian Rey |
|  |  |  |
|  |  |  |

# Tabla de Contenidos

[Historial de Cambios 2](#_Toc215223503)

[1. Introducción 4](#_Toc1668873359)

[2. Propuesta de proyecto 6](#_Toc1200654064)

[3. Administración del Proyecto 8](#_Toc1885530348)

[3.1 Modelo de Ciclo de Vida 8](#_Toc1128160345)

[3.2 Lenguajes y Herramientas 9](#_Toc2093218420)

[3.3 Interfaces Externas 10](#_Toc2014370440)

[3.4 Organigrama y Descripción de Roles 10](#_Toc1617941326)

[3.5 Calendarización y Entregables 12](#_Toc1487658477)

[4. Procesos de Soporte 14](#_Toc1517996865)

[4.1 Ambiente de Trabajo 14](#_Toc996901820)

[4.2 Monitoreo y Control de Progreso 14](#_Toc1292480962)

[4.3 Administración de Configuración y Documentación 15](#_Toc1787149231)

[4.4 Control de Calidad 16](#_Toc1059932594)

[Observaciones 16](#_Toc1790005898)

[Referencias 18](#_Toc1916601664)

[Anexos 19](#_Toc1231464409)

# Introducción

Este documento presenta el desarrollo de un Sistema de Gestión Bancaria, elaborado por un equipo de cuatro integrantes. En él, se detallarán los aspectos clave del proyecto, desde su propuesta inicial hasta su administración, planificación y mecanismos de control de calidad.

En primer lugar, se expondrá la propuesta general del proyecto (2.), describiendo sus objetivos, alcance y la importancia del sistema dentro del contexto bancario.

A continuación, se profundizará en la administración del proyecto (3.), donde se abordarán diversos aspectos fundamentales para su gestión:

- 3.1 Modelo de Ciclo de Vida: Se explicará la metodología adoptada para el desarrollo del sistema, detallando las fases y enfoques que guiarán el proceso.

- 3.2 Lenguajes y Herramientas: Se especificarán las tecnologías seleccionadas para la implementación del sistema, incluyendo lenguajes de programación, frameworks y entornos de desarrollo.

- 3.3 Interfaces Externas: Se describirán los sistemas, servicios o plataformas con los que el sistema bancario deberá interactuar.

- 3.4 Organigrama y Descripción de Roles: Se presentará la estructura organizativa del equipo de desarrollo, detallando las responsabilidades de cada integrante.

- 3.5 Calendarización y Entregables: Se establecerán los plazos de ejecución del proyecto, junto con los productos intermedios y finales esperados.

Posteriormente, se abordarán los procesos de soporte (4.), esenciales para garantizar el éxito del proyecto:

- 4.1 Ambiente de Trabajo: Se definirán las herramientas colaborativas, metodologías de trabajo y entornos técnicos utilizados por el equipo.

- 4.2 Monitoreo y Control de Progreso: Se explicarán los mecanismos implementados para supervisar el avance del desarrollo y la toma de decisiones correctivas.

- 4.3 Administración de Configuración y Documentación: Se describirán las estrategias para la gestión de versiones del software, control de cambios y documentación técnica.

- 4.4 Control de Calidad: Se detallarán los procesos de verificación y validación aplicados para asegurar la fiabilidad, seguridad y desempeño del sistema.

Finalmente, se incluirá una sección de referencias, donde se listarán las fuentes consultadas durante la planificación y ejecución del proyecto.

Este documento servirá como una guía integral para el desarrollo del Sistema de Gestión Bancaria, proporcionando un marco estructurado que garantice su correcta implementación y gestión.

# Propuesta de proyecto

**Propósito del Proyecto**

Este proyecto tiene como objetivo desarrollar un **sistema bancario local** que permita a los usuarios gestionar sus cuentas, realizar transferencias, solicitar préstamos y otras operaciones básicas.

El sistema será diseñado para ejecutarse en el equipo del usuario sin necesidad de conexión a internet, lo que simplifica su implementación y reduce dependencias externas.

Además de centrarse en la parte técnica, el propósito principal del proyecto es proporcionar una experiencia de desarrollo que simule un entorno real de trabajo, aplicando buenas prácticas de ingeniería de software y gestión de proyectos con un cliente simulado.

En términos metodológicos, se utilizarán componentes específicos de Scrum y metodologías ágiles, sin adoptar el marco completo. Esto se debe a que el equipo aún está en proceso de adaptación a estas metodologías, por lo que se priorizará un enfoque que fomente la independencia y la responsabilidad individual.

* Reuniones semanales para revisar avances, resolver bloqueos y ajustar tareas
* Entregables cada dos semanas, permitiendo iteraciones de mejora continua
* Uso de Trello para la asignación y gestión de tareas, asegurando transparencia en el progreso
* Seguimiento del desarrollo en GitHub, garantizando control de versiones y trabajo colaborativo.
* Gestión del backlog, donde se priorizarán las funcionalidades del sistema según su impacto y viabilidad dentro del alcance del proyecto.
* Visualización de avance por medio de una burn down chart.

De esta manera, se mantiene un balance entre una gestión estructurada del proyecto y la flexibilidad necesaria para que los integrantes del equipo puedan trabajar de manera autónoma y desarrollar habilidades de autogestión.

**Alcance del Proyecto**

El sistema incluirá las siguientes funcionalidades principales:

* **Registro e inicio de sesión** con autenticación local.
* **Gestión de cuentas bancarias**, permitiendo apertura y consulta de cuentas de ahorro y corriente.
* **Transferencias entre cuentas**, tanto propias como de terceros.
* **Gestión de tarjetas de débito y crédito**, incluyendo solicitud y bloqueo.
* **Pagos programados** y automatización de pagos recurrentes.
* **Préstamos y créditos**, con simulación de pagos y proceso de aprobación.
* **Inversiones y ahorros**, permitiendo la simulación y apertura de cuentas de inversión.
* **Seguridad básica**, con detección de transacciones sospechosas y configuración de alertas.
* **Reportes financieros**, proporcionando resúmenes de ingresos y gastos.
* **Servicio al cliente**, a través de un módulo de soporte con preguntas frecuentes.

Para evitar complejidad innecesaria, el sistema **no** incluirá:

* Integración con bancos externos o pasarelas de pago.
* Funcionalidad en la nube o acceso remoto.
* Sucursales físicas o soporte para múltiples administradores.
* Soporte para criptomonedas o inversiones avanzadas.

**Ejecución y Plataforma**

El sistema será una aplicación de escritorio que deberá instalarse o ejecutarse localmente en el equipo del usuario. Se explorará la posibilidad de utilizar **mensajes a Telegram** para la verificación de identidad si su implementación resulta sencilla.

# Administración del Proyecto

## Modelo de Ciclo de Vida

Este proyecto sigue el **modelo ágil Scrum**, con un enfoque en la entrega continua de valor a través de iteraciones llamadas **Sprint**.

El proyecto se organiza en **esprints de 1 o 2 semanas**, en los cuales se definen las tareas a realizar y se establece un objetivo claro para cada ciclo. A lo largo de cada sprint, el equipo trabajará en las funcionalidades del sistema bancario, con revisiones periódicas al final de cada sprint para evaluar el progreso.

**Prácticas Específicas Utilizadas en el Proyecto**

* **Gestión de Tareas con Trello**: Todas las tareas serán registradas y gestionadas en Trello, donde se asignarán a los programadores y se establecerán fechas límite. Trello permitirá un seguimiento claro del progreso de cada tarea y su estado (por hacer, en progreso, o completada).
* **Colaboración mediante GitHub**: El desarrollo se realizará de manera colaborativa utilizando GitHub. Los miembros del equipo compartirán y revisarán el código de forma conjunta, asegurando que todas las funcionalidades sean desarrolladas de manera coherente y bajo un control de versiones adecuado.
* **Pruebas Manuales y automatizadas**: Aunque no se implementarán pruebas automatizadas al inicio, se realizarán pruebas manuales al final de cada sprint en los primeros pasos para verificar que las funcionalidades implementadas cumplan con los requisitos y no afecten el funcionamiento de otras partes del sistema, posteriormente cuando el proyecto avance se implementarán las pruebas automatizadas.
* **Comunicación**: La comunicación se gestionará principalmente a través de reuniones en clase y mensajes por WhatsApp para resolver dudas o coordinar tareas fuera del horario de las reuniones formales.

**Seguimiento del Progreso**

El avance del proyecto será monitorizado utilizando herramientas como Trello para ver el estado de las tareas y el cumplimiento de los plazos. Además, se realizarán reuniones diarias de **standup** (breves reuniones) para discutir el progreso y cualquier bloqueo que el equipo pueda estar enfrentando.

## Lenguajes y Herramientas

Para la realización del proyecto se incentiva el uso de lenguajes y herramientas que puedan proveer de un modelado integral del proyecto, que sirvan para la ejecución de este, que ayuden al diseño del proyecto y finalmente que documenten cada detalle de las operaciones que se harán de manera interna. Es por esto que a continuación se identificarán cuáles son los principales aspectos y herramientas que se usarán para el proyecto.

El lenguaje de programación establecido para el desarrollo del proyecto es Java. Esto se debe a dos motivos principales; es un lenguaje de programación extremadamente robusto y confiable, y todos los integrantes que participan en el proyecto cuentan con conocimientos de este. Adicionalmente, Java permite la programación orientada a objetos, paradigma clave para el desarrollo, y la integración de herramientas que se verán más adelante.

Posteriormente se utilizará el IDE Intellij Idea. Con la ventaja de tener un correo universitario, Jet Brains, empresa dueña de Intellij Idea, provee de una licencia profesional para tener acceso a todas las características dentro del IDE. De esta manera se tiene una solución profesional para comenzar con el desarrollo del proyecto, así como un IDE confiable para la ejecución de este, creando un entorno de desarrollo fácil y seguro de utilizar.

Java FX es una herramienta que permite crear y programar interfaces gráficas basadas en el lenguaje de programación JAVA. Su facilidad de uso es clave para el desarrollo del proyecto, por lo que este será la herramienta asignada para la creación de interfaces gráficas. Java FX a su vez, se puede integrar con el IDE anteriormente mencionado para lograr una construcción integral de desarrollo asociado a su lenguaje y los elementos gráficos.

El control de versiones distribuido va a ser clave para el desarrollo completo del proyecto. Tener una trazabilidad de todos los cambios que se van cometiendo a través de una línea de tiempo ayuda a evitar problemas y conflictos en el desarrollo de la solución. Es por esto que se va a utilizar Git como herramienta para control de versiones del proyecto, así como GitHub para un control de versiones remoto en donde todos los integrantes puedan extraer y colocar cambios a medida que el proyecto vaya progresando.

Para la documentación, la suite de office que provee la universidad es suficiente. Herramientas como Microsoft Word, o Word Online serán utilizadas para la redacción y publicación de los documentos de alta importancia que vayan surgiendo durante el desarrollo del proyecto.

Adicionalmente, se requiere de acceder de bases de datos SQL para poder continuar con el almacenamiento y procesamiento de datos. Para esta función se utilizará H2.

## Interfaces Externas

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre Entidad | Descripción | Responsabilidades asociadas al proyecto | Datos de contacto/ Medios de comunicación |
| Jaime Andrés Pavlich | Profesor de la materia | Supervisa y califica el proyecto | Horas de clase comunes/ Correo Electrónico: jpavlich@javeriana.edu.co |
| Stack Overflow | Foro de programación | Ayuda miscelánea en programación | <https://stackoverflow.com/> |
| Profesores externos | Profesores que cumplan un rol de consultor | Dar asesoría sobre aspectos técnicos o de documentación | Horas de clase comunes/ Correo Electrónico |

## Organigrama y Descripción de Roles

Diagrama

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
| Rol | Descripción | Responsabilidades |
| Cliente (Profesor) | Actúa como el cliente del proyecto, proporcionando requisitos y validando entregables. | * Evaluar avances y entregables. * Proporcionar retroalimentación al equipo. |
| Scrum Master (Facilitador) | Coordina la aplicación de la metodología Scrum, asegurando la correcta gestión del proyecto. | * Gestionar sprints y facilitar reuniones. * Apoyar la organización del equipo. * Resolver bloqueos y coordinar con el cliente. * Supervisar el uso de Trello y GitHub. * Líder servicial * Documentación general, revisión y corrección de errores |
| Programadores | Encargados del desarrollo, pruebas y documentación del sistema bancario. | * Implementar funcionalidades del sistema. * Realizar pruebas manuales de las características desarrolladas. * Gestionar código en GitHub y tareas en Trello. * Asistir a reuniones de equipo y sprints. * Documentación especifica |
| Consultores Externos (Profesores) | Brindan orientación técnica y resuelven dudas específicas. | * Responder preguntas técnicas. * Orientar sobre metodologías o herramientas. |

## Calendarización y Entregables

|  |  |  |
| --- | --- | --- |
| Fechas | Descripción | Responsabilidades |
| Cliente (Profesor)   * Evaluar avances y entregables: Cada 3 semanas (Primera revisión: 5 de abril, luego: 26 de abril, 17 de mayo, 7 de junio). * Proporcionar retroalimentación al equipo: Posterior a cada evaluación (6 de abril, 27 de abril, 18 de mayo, 8 de junio). | Actúa como el cliente del proyecto, proporcionando requisitos y validando entregables. | * Evaluar avances y entregables. * Proporcionar retroalimentación al equipo. |
| Scrum Master (Facilitador)   * Gestionar sprints y facilitar reuniones: Sprints de 2 semanas, reuniones semanales (Inicia: 25 de marzo, luego: 1 de abril, 8 de abril, etc.). * Apoyar la organización del equipo: Permanente, con revisiones semanales (Viernes de cada semana). * Resolver bloqueos y coordinar con el cliente: Según sea necesario, revisión cada dos semanas. * Supervisar el uso de Trello y GitHub: Reportes cada dos semanas (31 de marzo, 14 de abril, 28 de abril, etc.). | Coordina la aplicación de la metodología Scrum, asegurando la correcta gestión del proyecto. | * Gestionar sprints y facilitar reuniones. * Apoyar la organización del equipo. * Resolver bloqueos y coordinar con el cliente. * Supervisar el uso de Trello y GitHub. |
| Programadores   * Implementar funcionalidades del sistema: Entregas parciales cada 3 semanas (Primera entrega: 12 de abril, luego: 3 de mayo, 24 de mayo, 14 de junio). * Realizar pruebas manuales de las características desarrolladas: Después de cada entrega (13 de abril, 4 de mayo, 25 de mayo, 15 de junio). * Gestionar código en GitHub y tareas en Trello: Actividad continua con revisiones semanales. * Asistir a reuniones de equipo y sprints: Todos los lunes. | Encargados del desarrollo, pruebas y documentación del sistema bancario. | * Implementar funcionalidades del sistema. * Realizar pruebas manuales de las características desarrolladas. * Gestionar código en GitHub y tareas en Trello. * Asistir a reuniones de equipo y sprints. |
| Consultores Externos (Profesores)   * Responder preguntas técnicas: Según necesidad, revisión mensual (Primer contacto: 15 de abril, luego: 15 de mayo, 15 de junio). * Orientar sobre metodologías o herramientas: Según necesidad, especialmente al inicio del proyecto (Primera sesión: 25 de marzo). | Brindan orientación técnica y resuelven dudas específicas. | * Responder preguntas técnicas. * Orientar sobre metodologías o herramientas. |

# Procesos de Soporte

## Ambiente de Trabajo

**Reglas del Trabajo en Equipo**

* **Asistencia a reuniones**: Todos los integrantes deben asistir a las reuniones programadas, salvo en caso de fuerza mayor.
* **Comunicación**: La comunicación principal del equipo se realizará a través de **WhatsApp**, tanto para coordinación como para resolver dudas rápidas.
* **Resolución de conflictos**: Si surge un desacuerdo, se intentará resolver dentro del equipo. Si no se llega a un acuerdo, se consultará con el profesor para una decisión final.

**Mecanismos para Asegurar el Cumplimiento**

* **Seguimiento semanal**: Se revisará el avance de cada integrante semanalmente para asegurar que todas las tareas asignadas estén en progreso o completadas.
* **Monitoreo en Trello**: Se utilizará **Trello** para asignar y seguir el estado de las tareas.
* **Registro de código en GitHub**: El desarrollo colaborativo se gestionará en **GitHub**, donde se validará el trabajo realizado.
* **Manejo de incumplimientos**: No habrá penalizaciones iniciales, pero si un integrante incumple repetidamente sus responsabilidades sin justificación, podría ser excluido del equipo.

**Responsables y Artefactos de Control**

* **Responsable del cumplimiento**: El **Scrum Master** (santiago) se encargará de verificar que las reglas se cumplan y consultará con el profesor en caso de problemas recurrentes.
* **Herramientas de control**: Se utilizarán **Trello y GitHub** como evidencia del trabajo realizado y para documentar avances.

## Monitoreo y Control de Progreso

**MÉTRICA** utilizada para medir progreso: Para propósitos de medir el progreso del desarrollo del proyecto, se va a usar una métrica basada en las horas faltantes para completar una tarea de acuerdo con las fechas acordadas por el grupo, así se podrá medir cuanto progreso se ha dado en determinados intervalos de tiempo y ayudar a formular posibles acciones correctivas.

**Actividades por realizar**

Para reportar progreso: Habrá un rol de responsable para los reportes de progreso, estos reportes se realizarán aproximadamente 24 horas antes de las fechas establecidas, estos consistirán en un reporte en las mismas palabras de los integrantes del grupo para testificar sus progresos, asimismo se usará la tabla de historial de cambios para corroborar y mantener un registro de los cambios y modificaciones que se hayan hecho.

Acciones correctivas: En caso de que el progreso del proyecto no vaya según lo esperado, ya sea que haya un decremento de progreso en comparación con el intervalo pasado, o se presente una dificultad notable que impida el progreso de varios miembros del grupo, se realizara una reunión entre todos los miembros del grupo en el que se discutirán las causas de las dificultades en el progreso del proyecto y se decidirá entre todos que acciones en concreto se deberían tomar, estas reuniones sucederían en conjunto con los reportes de progreso cotidiano para aclarar todas las posibles dudas y prepararse para la siguiente fecha.

## Administración de Configuración y Documentación

La administración de la configuración del proyecto garantiza el control sobre la evolución del código, la documentación y otros artefactos del sistema.

|  |  |
| --- | --- |
| Ítem | Descripción |
| Código fuente | Contendrá la implementación del sistema bancario. Se gestionará en un repositorio GitHub. |
| Documentación del proyecto | Se almacenará en formato Word en un repositorio compartido (nube de la universidad). Incluye especificaciones de requisitos, manuales de usuario y documentación técnica. |
| Base de datos | Se versionará cualquier cambio en la estructura de la base de datos mediante H2 controlados en GitHub. |
| Diagramas BPMN y ERD | Modelos visuales del sistema, que evolucionarán junto con el desarrollo del proyecto. Se guardarán en formatos editables y exportables. |
| Casos de uso e historias de usuario | Se documentarán y actualizarán conforme avance el desarrollo. |

|  |  |
| --- | --- |
| **Rama** | **Propósito** |
| **main** | Contiene la versión estable y final del sistema. Solo se fusionan cambios una vez aprobados. |
| **develop** | Rama de desarrollo principal donde se integran nuevas funcionalidades antes de ser fusionadas en main. |
| **feature** | Ramas temporales para el desarrollo de nuevas características. Se crean a partir de develop y, una vez terminadas, se fusionan en develop. |
| **bugfix** | Ramas destinadas a corregir errores detectados en develop antes de integrarse nuevamente. |
| **hotfix** | Usadas para corregir errores críticos en main sin necesidad de esperar un ciclo de desarrollo completo. |

**Control de Versiones y Seguimiento**

* Repositorio GitHub: Todos los cambios en el código y la documentación se gestionarán mediante GitHub.
* Reglas de Merge: Antes de integrar cambios en develop o main, los commits deben ser revisados por otro miembro del equipo.
* Historial de cambios: Se mantendrá un registro de modificaciones en la documentación del proyecto.
* Pruebas automatizadas: se harán pruebas automatizadas para verificar la correcta implementación del código, aunque en etapas tempranas no se evaluara de esta forma, según el avance del proyecto, se implementaran las pruebas

## Control de Calidad

Los procesos de control de calidad se aplican en diferentes fases del proyecto para asegurar que el producto desarrollado cumpla con los requisitos establecidos y con los estándares de calidad esperados. A continuación, se describen los procesos identificados:

|  |  |  |
| --- | --- | --- |
| **Nombre del Proceso** | **Momento del Proyecto** | **Responsables** |
| Revisión de Requisitos | Inicio del proyecto, durante la planificación. | Analista de Requisitos, Líder del Proyecto. |
| Revisión de Diseño | Fase de diseño del sistema. | Arquitecto de Software, Desarrolladores Senior. |
| Pruebas Unitarias | Durante el desarrollo de cada módulo. | Desarrolladores. |
| Pruebas de Integración | Al integrar múltiples módulos del sistema. | Desarrolladores, Equipo de Pruebas. |
| Pruebas de Aceptación | Antes de la entrega final al cliente simulado. | Cliente Simulado, Equipo de Pruebas. |
| Revisión de Documentación | Durante todo el ciclo del proyecto. | Líder del Proyecto, Documentadores. |
| Auditorías de Calidad | Al final de cada sprint (en Scrum). | Líder del Proyecto, Equipo de Calidad. |

### Observaciones

El enfoque en la calidad se extiende desde la planificación hasta la entrega final, asegurando que cada módulo desarrollado sea probado y validado adecuadamente antes de la integración completa. Las metodologías ágiles permiten realizar estos controles de calidad de forma iterativa y constante, garantizando mejoras continuas en cada sprint.

# Referencias

[1] Zotero, “Zotero | Your personal research assistant,” *www.zotero.org*. <https://www.zotero.org>

[2] Asana, “Burndown Chart: qué es y cómo usarlo con ejemplos [2022] • Asana,” *Asana*. <https://asana.com/es/resources/burndown-chart>

# Anexos