**PROYECTO APLICACIÓN DE PRESTAMOS**

**LUIS JIMENEZ**

**DILAN MACHADO**

**FACULTAD DE INGENIERIA DE SISTEMAS**

**UNIVERSIDAD DE LA GUAJIRA SEDE MAICAO**

**2024**

**Introducción**

En el ámbito de los préstamos informales, muchas personas que otorgan créditos carecen de herramientas adecuadas para llevar un registro ordenado y seguro de las transacciones realizadas. Este proyecto surge con la intención de cubrir esta necesidad, ofreciendo una solución práctica que facilite la gestión de préstamos a través de una aplicación móvil intuitiva y accesible.

La aplicación desarrollada en Android utiliza Kotlin como lenguaje de programación, y está diseñada específicamente para simplificar el proceso de registrar información clave del solicitante, calcular el monto total a pagar y realizar un seguimiento detallado de los pagos. Al centralizar la información y automatizar algunos cálculos, el sistema busca reducir errores y ofrecer una experiencia clara y sencilla para los usuarios que desean manejar sus transacciones de manera eficiente.

**Objetivo general del proyecto**

Desarrollar una aplicación móvil en Android para facilitar el registro y gestión de préstamos de manera física, permitiendo al usuario ingresar información clave del solicitante y administrar el seguimiento de los pagos de manera efectiva y accesible.

**Público objetivo**

Este proyecto está dirigido a personas que otorgan préstamos de manera informal y desean contar con un sistema claro y accesible para llevar un registro seguro y ordenado de sus transacciones financieras.

**Requerimientos**

**Requerimientos funcionales**

* Registro de solicitantes de préstamos con detalles (nombre, apellidos, dirección, teléfono).
* Función para tomar foto del solicitante.
* Registro de fechas de otorgamiento y límite de pago.
* Cálculo automático del total a pagar (incluyendo un 20% adicional sobre el préstamo).
* Registro de pagos parciales, seguimiento de deudas y saldo restante.
* Actualización del saldo en caso de incumplimiento (nuevo 20% sobre saldo pendiente).
* Búsqueda de solicitantes y administración de deudas.

**Requerimientos no funcionales**

* Plataforma Android.
* Interfaz intuitiva y accesible para usuarios sin experiencia en tecnología.
* Seguridad en el almacenamiento de datos personales.

**Requisitos**

**Requisitos de Interfaz de Usuario:**

**Pantalla de Inicio de Sesión:** Una pantalla donde los usuarios puedan iniciar sesión para proteger el acceso a la aplicación.

* Formulario de Registro de Solicitantes: Un formulario que permita ingresar la información completa del solicitante y tomar su foto.
* Pantalla de Resumen de Préstamos: Un resumen que muestre el estado de los préstamos actuales y el monto restante de cada solicitante.
* Pantalla de Detalle del Solicitante: Una pantalla donde se pueda ver el historial de pagos y la deuda pendiente de cada solicitante.

**Requisitos de Rendimiento y Escalabilidad:**

**Tiempo de Respuesta:** La aplicación debe cargar y procesar la información en menos de 3 segundos en promedio.

**Capacidad de Almacenamiento:** Poder almacenar y gestionar información de al menos 100 solicitantes sin que el rendimiento de la aplicación se vea afectado.

**Elementos Scrum para la Aplicación de Gestión de Préstamos**

|  |  |
| --- | --- |
| **Personas** | **Roles** |
| **Luis Jiménez** | **Product owner** |
| **Dilan Machado** | **Scrum master** |
| **Luis Jiménez, Dilan Machado** | **Equipo de desarrollo** |

**Backlog del Producto:**

* Documento que contiene todas las funcionalidades, mejoras y correcciones necesarias para la aplicación. Luis Jiménez, como Product Owner, es responsable de priorizar y mantener este backlog, reflejando las necesidades clave del usuario final.

**Sprint Backlog:**

* Lista de elementos seleccionados del Backlog del Producto que el equipo, compuesto por Luis Jiménez y Dilan Machado, se compromete a completar en el sprint actual.

**Historias de Usuario:**

* Descripciones narrativas de una funcionalidad desde la perspectiva del usuario. Estas historias son creadas y revisadas por Luis Jiménez y representan el trabajo a completar en cada sprint.

**Puntos de Historia:**

* Unidad de medida utilizada para estimar la complejidad de las Historias de Usuario. Luis y Dilan colaboran para realizar la estimación durante la planificación del sprint.

**Reunión de Planificación del Sprint:**

* Sesión donde ambos, Luis y Dilan, planifican las tareas y definen el alcance del sprint, estableciendo los objetivos y el trabajo clave para el período.

**Reuniones Diarias de Scrum:**

* Breves reuniones diarias entre Luis y Dilan para compartir actualizaciones rápidas, discutir cualquier impedimento y coordinar el progreso de las tareas.

**Reunión de Revisión del Sprint:**

* Sesión al final de cada sprint donde Luis y Dilan revisan las funcionalidades completadas, evaluando si se cumplieron los objetivos del sprint y obteniendo feedback para ajustes.

**Reunión de Retrospectiva del Sprint:**

* Sesión posterior al sprint donde ambos reflexionan sobre lo que funcionó bien y las áreas que requieren mejoras, ajustando el proceso para el siguiente sprint.

**Tablero Scrum:**

* Tablero en Jira donde ambos pueden visualizar el progreso del sprint, incluyendo el Burndown Chart y las Historias de Usuario activas.

**Gráfico Burndown:**

* Gráfico en el tablero de Jira que muestra el trabajo restante en el sprint a lo largo del tiempo, ayudando a visualizar el progreso y ajustar la carga de trabajo.

**Incremento de Producto:**

* La versión actualizada y potencialmente entregable de la aplicación al final de cada sprint, que incorpora nuevas funcionalidades o mejoras en la gestión de préstamos.

**Reunión de Grooming de Producto:**

* Sesiones donde Luis y Dilan revisan y refinan el Backlog del Producto para asegurar que esté claro y listo para la planificación del próximo sprint.

**Ceremonia de Liberación:**

* Evento que marca la entrega de una nueva versión completa de la aplicación, generalmente después de varios sprints, con las funcionalidades necesarias para su uso final.

**Herramientas de Integración Continua:**

* Herramientas como Jenkins o GitLab CI son usadas para integrar código y ejecutar pruebas automáticas, asegurando que la aplicación mantenga la calidad en cada entrega.

**Herramientas de Colaboración:**

* Plataformas como Confluence o Microsoft Teams facilitan la comunicación entre Luis y Dilan, asegurando que toda la documentación y los avances estén centralizados y accesibles.

**Planificación Detallada de Sprints para la Aplicación de Gestión de Préstamos en Android**

**Sprint Inicial (Sprint 0) - Configuración y Planificación:**

**Objetivos:**

* Configurar el entorno de desarrollo en Android Studio para el proyecto.
* Realizar una planificación detallada de los primeros sprints.
* Identificar y configurar herramientas y tecnologías necesarias, como bases de datos, herramientas de pruebas y repositorios de código.

**Sprint 1 - Diseño de la Interfaz y Planificación Detallada:**

**Objetivos:**

* Crear y revisar los prototipos de la interfaz de usuario de la aplicación.
* Detallar los requisitos del sistema y descomponer historias de usuario en tareas específicas.
* Configurar el tablero Scrum en Jira para gestionar las tareas de diseño y planificación.

**Sprint 2 - Desarrollo Inicial y Configuración de Pruebas:**

**Objetivos:**

* Iniciar el desarrollo de las funcionalidades básicas, incluyendo el registro de solicitantes y el cálculo del monto total a pagar.
* Configurar casos de prueba para las funcionalidades iniciales.
* Revisar y ajustar el diseño según las necesidades del desarrollo.

**Sprint 3 - Integración y Pruebas Iniciales:**

**Objetivos:**

* Integrar componentes desarrollados, como el registro de datos y el cálculo automático de intereses.
* Realizar pruebas unitarias y de integración para verificar la funcionalidad básica.
* Ajustar el código y diseño de acuerdo a los resultados de las pruebas.

**Sprint 4 - Mejora de la Experiencia del Usuario (UX):**

**Objetivos:**

* Implementar mejoras en la interfaz de usuario para simplificar el registro de solicitantes y el seguimiento de pagos.
* Realizar pruebas de usabilidad para obtener feedback sobre la experiencia del usuario.
* Ajustar el diseño y funcionalidades basándose en el feedback de usabilidad.

**Sprint 5 - Desarrollo de Funciones Avanzadas:**

**Objetivos:**

* Desarrollar funcionalidades avanzadas, como la actualización automática de saldos y alertas de incumplimiento de pagos.
* Configurar pruebas específicas para las nuevas funcionalidades.
* Realizar ajustes según las necesidades emergentes del desarrollo.

**Sprint 6 - Seguridad y Pruebas de Rendimiento:**

**Objetivos:**

* Implementar medidas de seguridad para proteger los datos de los solicitantes.
* Realizar pruebas de rendimiento para garantizar una respuesta rápida de la aplicación.
* Ajustar el código y la configuración según los resultados de las pruebas.

**Sprint 7 - Integración Continua y Despliegue (CI/CD):**

**Objetivos:**

* Implementar prácticas de integración continua y despliegue automático para optimizar el flujo de desarrollo.
* Configurar el despliegue en un entorno de prueba.
* Ajustar el proceso de CI/CD con base en los aprendizajes obtenidos durante el sprint.

**Sprint 8 - Iteración Basada en Feedback del Usuario:**

**Objetivos:**

* Evaluar el feedback de los usuarios sobre las funcionalidades implementadas hasta el momento.
* Realizar mejoras en el diseño y añadir nuevas funcionalidades según las necesidades emergentes del usuario.

**Sprint 9 en Adelante - Ciclos de Mejora Continua:**

**Objetivos:**

* Iterar en mejoras continúas basadas en la retroalimentación de los usuarios y del equipo.
* Mantener un ciclo de desarrollo ágil y adaptable en Jira, ajustando funcionalidades según nuevas necesidades o desafíos identificados.

**Presupuesto Ajustado para la Aplicación de Gestión de Préstamos (en COP)**

**Mano de Obra (Desarrollo)**

Desarrollador de Software (1 persona, 8 semanas): $4,500,000

(Costo total por 2 meses de trabajo)

**Diseño de Interfaz de Usuario (UI/UX)**

Diseño básico de pantallas (solo las necesarias): $600,000

**Licencias y Herramientas de Desarrollo**

Jira y Confluence (uso en versiones gratuitas o plan básico): $300,000

Costo total de herramientas (8 semanas): $300,000

**Pruebas y Calidad**

Pruebas de Usuario y Calidad (testers informales): $500,000

Pruebas de Seguridad Básica: $400,000

Costo total de pruebas: $900,000

**Gastos Administrativos y Contingencia**

Gastos varios (contingencias mínimas): $500,000

**Resultado total**

**Mano de Obra:** $4,500,000

**Diseño UI/UX:** $600,000

**Licencias y Herramientas:** $300,000

**Pruebas y Calidad:** $900,000

**Gastos Administrativos:** $500,000

**Total:** 6,800,000

**Mockups**:

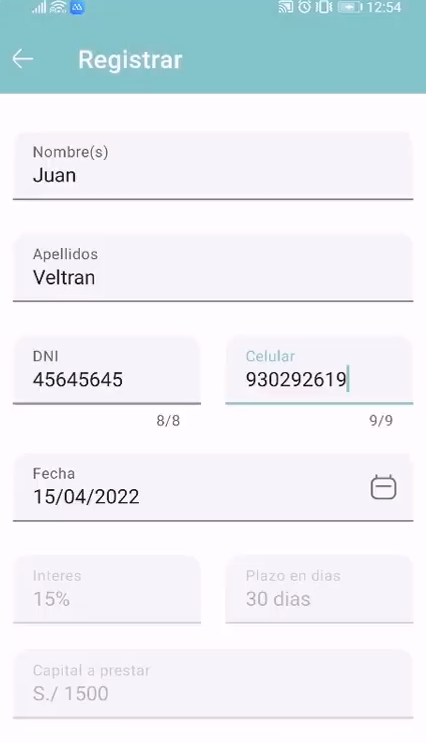
**Login:**

****

**Registro de préstamos:**

****

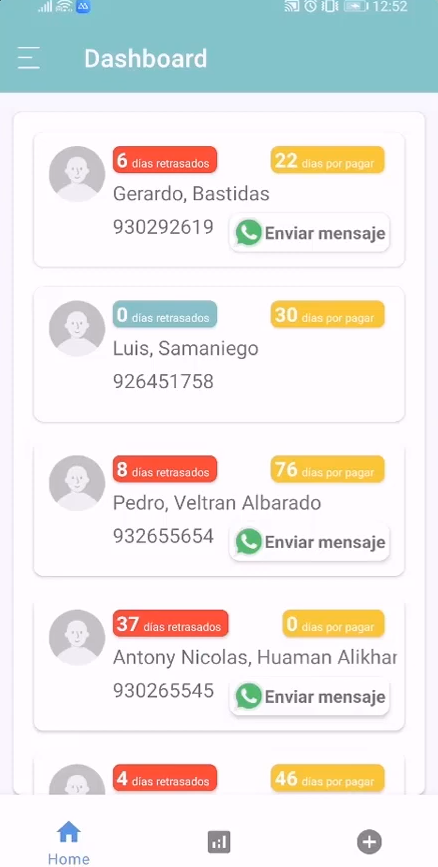
**Registro de cliente:**

****

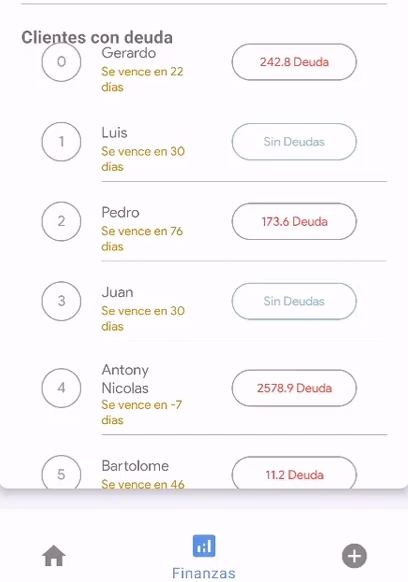
**Información de préstamo del deudor:**

****

**Información de pagos (Días por pagar y días retrasados):**

****

**Clientes con deudas:**

****