

Para esta tarea, necesitas crear un documento que resuma las etapas clave del Ciclo de Vida del Desarrollo de Software (SDLC). Utiliza tus experiencias personales y proporciona ejemplos específicos extraídos del desarrollo e implementación de tu proyecto "Capstone" durante el curso "Desarrollo de Software 2". Asegúrate de ofrecer una visión completa de las etapas importantes del SDLC, destacando los desafíos encontrados, las soluciones implementadas y las lecciones aprendidas durante el proceso de desarrollo del proyecto.

## Introducción

El Ciclo de Vida del Desarrollo de Software (SDLC) es un marco estructurado para gestionar proyectos de software. Durante el curso "Desarrollo de Software 2", implementamos un sistema bancario centrado en el módulo de tarjetas de crédito, aplicando las etapas clave del SDLC.

Las 4P's en la Planificación del Proyecto

**Personas:** Equipo: 5 desarrolladores, 1 líder técnico y stakeholder (Practitioner simulando como entidad bancarias).

**Producto:** Sistema de gestión de tarjetas de crédito

- Emisión de tarjetas.
- Cancelación de tarjetas.
- Cálculo de intereses.
- Calculo de cuotas.

Proceso: Metodología ágil (Scrum) con sprints de 1 semana.

Herramientas: Trello para gestión de tareas, Git y GitLab para control de versiones así como la wiki de Gitlab para la documentación.

**Proyecto:** Cronograma de 8 meses, con hitos como integración de APIs y pruebas de carga.

## Ciclo de Vida del Desarrollo de Software (SDLC)

### 1. Análisis de Requerimientos

Entrevistas con stakeholder (Practitioner) para definir funcionalidades críticas (ejemplo: requisitos de seguridad, tipos de tarjeta, UI).

User stories: "Como usuario, quiero bloquear mi tarjeta desde la app para prevenir fraudes".

Desafíos: Requerimientos ambiguos (ejemplo: "detección de fraudes en tiempo real").

### 2. Diseño del Sistema

Arquitectura: Tres capas (presentación, lógica de negocio, base de datos).

Microservicios para procesamiento de transacciones y notificaciones.

Desafíos: Integrar un sistema de alertas en tiempo real. Integrar lectura de pdf para analizar el dni/recibo de sueldo

### **3. Implementación (Buenas Prácticas de Codificación)**

Tecnologías: Java (Spring Boot), PostgreSQL

Code reviews semanales (ejemplo: revisión del algoritmo de cálculo de intereses).

Estándares de código: Convenciones de nombres (ejemplo: CreditCardService).

Desafíos: Errores en la lógica de pagos mínimos.

### **4. Pruebas de Desarrollo**

Tipos de pruebas: Unitarias: Validación del cálculo de intereses con JUnit.

Integración: Pruebas de APIs con Postman.

### **5. Documentación del Proyecto**

Técnica: Gitlab Wiki, usamos la wiki como documentación, donde teníamos varias pestañas donde se diferenciaban las POC (pruebas de concepto), las implementaciones, la guía de instalación tanto del programa como de la base de datos.