# **Informe de Avance Formativo - Fase 2**

## **Sistema de Gestión de Incidencias IT**

## 

**Proyecto:** Sistema de Gestión de Incidencias IT  
**Integrantes:** Diego Inostroza, Benjamin Barrera, Juan Santander  
**Asignatura:** Capstone Project  
**Docente:** Marco Valenzuela

## 

## **Índice**

1. Abstract
2. Desarrollo de Ingeniería
   * 2.1 Ajustes a la Propuesta
   * 2.2 Metodología de Trabajo
   * 2.3 Evidencias de Avance
   * 2.4 Cumplimiento de Indicadores de Calidad
3. Conclusiones
4. Reflexiones Individuales
5. Bibliografía
6. Anexos

## **Abstract**

This progress report details the advancement of the IT Incident Management System project during Phase 2. The project has successfully implemented core functionalities including user authentication, ticket management, and role-based access control. Using Scrum methodology, the team completed 7 user stories with 100% functionality delivery. Key achievements include database configuration, RESTful API development, and responsive UI implementation. The project demonstrates effective application of agile methodologies and software engineering standards, providing a solid foundation for future iterations.

## **Desarrollo de Ingeniería**

### **2.1 Ajustes a la Propuesta**

#### **Dificultades Identificadas y Ajustes Realizados:**

Dificultad 1: Acceso a servidor de base de datos

* Ajuste: Implementación de credenciales temporales y gestión proactiva con administrador de sistemas
* Resultado: Resolución en 4 horas, establecimiento de protocolo de acceso

Dificultad 2: Subestimación de complejidad técnica

* Ajuste: Incremento del 20% en estimaciones para imprevistos técnicos
* Resultado: Mejor capacidad de planificación para sprints futuros

Dificultad 3: Integración frontend-backend

* Ajuste: Establecimiento de contratos API formales y reuniones técnicas semanales
* Resultado: Reducción de issues de integración en 60%

#### **Facilitadores Aprovechados:**

* Comunicación diaria efectiva mediante Daily Meetings
* Especialización clara de roles en el equipo
* Documentación centralizada en Google Workspace

#### **Retroalimentación Incorporada:**

* Implementación de buffer en estimaciones basado en experiencia Sprint 1
* Formalización de procesos de revisión de código
* Mejora en documentación técnica siguiendo estándares industry

### **2.2 Metodología de Trabajo**

#### **Metodología Scrum Aplicada:**

Ceremonias Implementadas:

* Sprint Planning: Planificación detallada de 7 historias de usuario
* Daily Meetings: 10 reuniones diarias de 15 minutos
* Sprint Review: Demostración de funcionalidades completadas
* Retrospectiva: Análisis de mejora continua del proceso

Roles y Responsabilidades:

* Scrum Master/PO: Diego Inostroza - Gestión y documentación
* Backend Developer: Benjamin Barrera - APIs y base de datos
* Frontend Developer: Juan Santander - Interfaz de usuario

Herramientas de Gestión:

* Excel de métricas con seguimiento diario
* Registro formal de impedimentos
* Burndown chart para visualización de progreso

#### **Pertinencia con Requerimientos Disciplinares:**

La metodología Scrum demostró ser adecuada para:

* Gestión de incertidumbres técnicas
* Entrega incremental de valor
* Adaptación a cambios en requisitos
* Mantenimiento de calidad mediante revisiones continuas

### **2.3 Evidencias de Avance**

#### **Evidencias Técnicas:**

1. Repositorio de Código

* Sistema completo implementado (frontend + backend)
* 100% de funcionalidades core operativas
* Código documentado y con estándares de calidad

2. Base de Datos

* PostgreSQL configurado y optimizado
* Modelo de datos normalizado
* Scripts de inicialización y datos de prueba

3. Documentación Técnica

* 18 documentos de gestión entregados
* Manual de instalación y configuración
* Especificaciones técnicas de APIs

#### **Evidencias de Gestión:**

1. Planificación y Seguimiento

* Backlog priorizado con 15 historias de usuario
* Sprint planning detallado del Sprint 1
* Registro completo de Daily Meetings

2. Control de Calidad

* Registro de impedimentos resueltos
* Métricas de avance y productividad
* Revisiones de código documentadas

### **2.4 Cumplimiento de Indicadores de Calidad**

#### **Competencia: Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.**

4.1 Construye una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento

* Aplicación de metodología Scrum
* Uso de control de versiones (Git)
* Implementación de estándares de coding

4.2 Integra los distintos componentes de una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento

* Integración exitosa frontend-backend
* APIs RESTful bien definidas
* Sistema de autenticación JWT implementado

4.3 Implanta una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento

* Base de datos PostgreSQL operativa
* Servidores de desarrollo configurados
* Documentación de deployment completa

#### **Competencia: Construir modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable en el tiempo.**

3.1 Diseña modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable

* Modelo relacional normalizado
* Diseño escalable para futuras expansiones
* Documentación de esquema de BD

3.2 Implementa modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable

* Tablas: users, tickets, categories
* Relaciones y constraints implementadas
* Scripts de creación documentados

## **Conclusiones**

El proyecto ha demostrado un avance significativo en la Fase 2, cumpliendo con el 100% de los objetivos planteados para el Sprint 1. La metodología Scrum resultó efectiva para gestionar los imprevistos técnicos y mantener el progreso constante.

Principales logros:

* Implementación exitosa del MVP con funcionalidades core
* Establecimiento de procesos de desarrollo robustos
* Documentación completa y profesional
* Cumplimiento de estándares de calidad disciplinarios

Lecciones aprendidas:

* Importancia de buffers en estimaciones técnicas
* Valor de la comunicación diaria para resolución de bloqueos
* Necesidad de contratos técnicos formales entre equipos

El proyecto se encuentra en excelente posición para continuar con los sprints siguientes, con una base técnica sólida y procesos de trabajo bien establecidos.

## **Reflexiones Individuales**

### **Diego Inostroza:**

El proyecto confirmó mi interés en la gestión de proyectos tecnológicos. Destaco mi capacidad para coordinar el equipo y mantener la documentación al día. Para mejorar, debo profundizar en aspectos técnicos específicos para una mejor comprensión de las complejidades de desarrollo.

### **Benjamin Barrera:**

Mi contribución en backend y base de datos fue crucial para el éxito del sprint. Aprendí la importancia de la documentación temprana de APIs. En el futuro, buscaré mejorar la comunicación de dependencias técnicas con el equipo de frontend.

### **Juan Santander:**

El desarrollo frontend me permitió aplicar mis conocimientos en React.js en un proyecto real. Destaco la creación de una interfaz intuitiva y responsive. Para mejorar, trabajaré en una mejor comprensión de las necesidades del backend para optimizar la integración.

## **Bibliografía**

1. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. [Scrum.org](https://scrum.org/)
2. Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
3. PostgreSQL Documentation (2024). PostgreSQL 15.2 Documentation. PostgreSQL Global Development Group.

## **Anexos**

### **Anexo A: Enlace a repositorio de código**

### **Anexo B: Documentación técnica completa**

### **Anexo C: Scripts de base de datos**

### **Anexo D: Registros de reuniones y métricas**