
Informe sobre sistema de control escolar

NOMBRE: Carlos Muñoz
CARRERA: Ingeniería en informática
ASIGNATURA: Ingeniería de software
PROFESOR: Vedran Hrvoj Tomicic Cantizano
FECHA: 31/10/2025

Índice

1. Introducción
2. Metodología de Desarrollo de Software
3. Levantamiento de Requerimientos
4. Casos de uso
5. Diagramas UML
6. Arquitectura del Software
7. Estándares de Calidad Aplicados
8. Costos empleados para el proyecto
9. Conclusión

Introducción

El presente informe tiene como propósito desarrollar un Sistema de Control de Asistencia Escolar orientado a la gestión digital del registro de asistencia en establecimientos educativos. El proyecto se ejecuta considerando las fases de análisis, levantamiento de requerimientos y diseño del software, aplicando buenas prácticas de ingeniería de software y respetando las normas APA versión 7. El cliente es un profesor de enseñanza básica que requiere optimizar el proceso de control de asistencia de sus estudiantes mediante una plataforma web accesible y eficiente.

El objetivo general del proyecto es diseñar una solución informática que permita registrar, almacenar y consultar de manera rápida la asistencia diaria de los estudiantes. Los objetivos específicos se centran en identificar los requerimientos funcionales y no funcionales, seleccionar una metodología adecuada, representar el sistema mediante diagramas UML, y establecer los estándares de calidad que garanticen la fiabilidad y mantenibilidad del software.

Metodología de Desarrollo de Software

Para el desarrollo del Sistema de Control de Asistencia Escolar se empleó una metodología de desarrollo iterativa basada en prototipos, propia del trabajo con tecnologías web como HTML, CSS, JavaScript y el uso del editor Visual Studio Code. Este enfoque permite construir la interfaz y las funcionalidades del sistema de manera progresiva, generando versiones preliminares (prototipos) que son evaluadas, ajustadas y refinadas de forma continua.

El prototipado evolutivo favorece la visualización temprana **del producto**, permitiendo identificar mejoras de diseño, usabilidad y estructura antes de implementar la versión final. Además, ofrece gran flexibilidad para realizar modificaciones durante cualquier etapa del proceso, lo cual resulta especialmente útil en proyectos educativos donde el usuario (profesor) puede revisar el avance y sugerir ajustes sobre la apariencia, navegación o funcionalidad del sistema.

Las fases consideradas en este ciclo de vida fueron: planificación inicial, levantamiento de requerimientos, diseño preliminar, desarrollo de prototipos visuales, implementación funcional, pruebas de interfaz y usabilidad, y ajustes finales. Cada fase se alimentó de los resultados de la anterior, permitiendo un proceso de desarrollo dinámico, adaptable y orientado a la mejora continua del producto final.

Levantamiento de Requerimientos

El levantamiento de requerimientos se realizó mediante entrevistas y observación directa con el profesor cliente, con el fin de comprender las necesidades específicas del proceso de registro de asistencia. Se identificaron los siguientes requerimientos funcionales y no funcionales:

- Registrar asistencia diaria por estudiante y curso.
- Visualizar reportes mensuales de asistencia.
- Editar o corregir registros en caso de error.
- Acceso restringido mediante autenticación de usuario.

- La aplicación debe ser compatible con navegadores modernos y dispositivos móviles.
- Los datos deben almacenarse de forma segura en una base de datos relacional.

Todos los requerimientos fueron validados con el cliente mediante una matriz de verificación, asegurando su comprensión, prioridad y factibilidad técnica.

Casos de uso empleados en el software

1-Iniciar sesión en el sistema

Actor: Docente / Administrador

Descripción: El usuario ingresa su nombre de usuario y contraseña para acceder al sistema de control de asistencia.

Objetivo: Proteger el acceso a la información y permitir acciones solo a usuarios autorizados.

2-Gestionar cursos

Actor: Administrador

Descripción: El administrador puede **crear, editar o eliminar cursos** (por ejemplo, 1°A, 2°B, etc.).

Objetivo: Mantener actualizada la oferta de cursos sobre los cuales se registrará asistencia.

3-Gestionar estudiantes

Actor: Administrador

Descripción: El administrador registra nuevos estudiantes, actualiza sus datos (nombres, apellidos, curso) o los marca como inactivos.

Objetivo: Mantener un listado actualizado de los estudiantes por curso.

4-Registrar asistencia diaria de un curso

Actor: Docente

Descripción: El docente selecciona un curso y una fecha, y luego marca el estado de cada estudiante (presente, ausente, justificado, etc.).

Objetivo: Dejar registro formal de la asistencia de los estudiantes en una determinada jornada.

5-Modificar asistencia registrada

Actor: Docente / Administrador

Descripción: El usuario puede corregir una asistencia ya registrada (por ejemplo, cambiar de “ausente” a “justificado”).

Objetivo: Corregir errores o actualizar información según nueva evidencia (licencias, justificativos, etc.).

6-Consultar asistencia por curso

Actor: Docente / Directivo

Descripción: El usuario selecciona un curso y un rango de fechas para visualizar el resumen de asistencia (quién faltó, cuántas veces, etc.).

Objetivo: Monitorear la asistencia grupal y detectar patrones de inasistencia.

7-Consultar historial de asistencia por estudiante

Actor: Docente / Directivo

Descripción: Se busca a un estudiante y se despliega su historial de asistencias, con fechas y estados.

Objetivo: Analizar la trayectoria de asistencia de un estudiante específico.

8-Generar reportes de asistencia

Actor: Docente / Directivo / Administrador

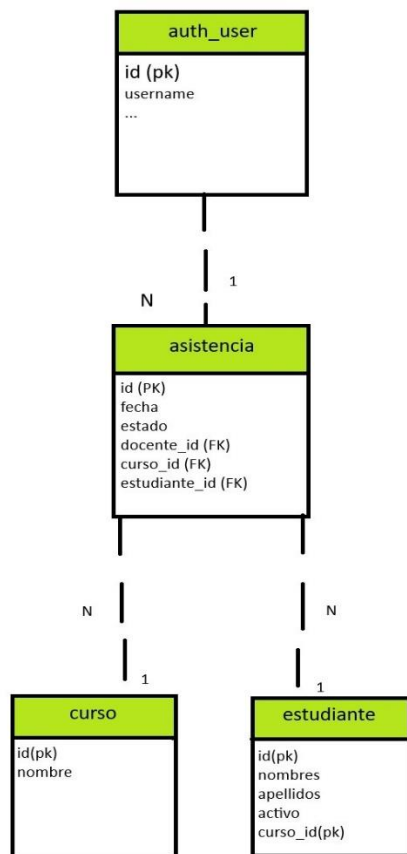
Descripción: El sistema entrega resúmenes o reportes (por curso o por estudiante) que muestran porcentajes de asistencia e inasistencia.

Objetivo: Apoyar la toma de decisiones pedagógicas, derivaciones o comunicación con apoderados.

Diagramas UML

Se elaboraron representaciones estructurales del sistema a través de un diagrama SQL relacional, el cual describe de manera formal las tablas y relaciones que componen la base de datos del Sistema de Control de Asistencia Escolar. Este diagrama identifica las entidades principales —Curso, Estudiante, Asistencia, Usuario (docente) y — junto con sus claves primarias, claves foráneas y relaciones uno a muchos.

El modelo relacional permite visualizar cómo se organiza la información en el sistema y cómo interactúan las distintas tablas que conforman la arquitectura de datos.



Arquitectura del Software

El sistema se diseñó bajo una arquitectura de tres capas: presentación, lógica de negocio y acceso a datos. En la capa de presentación, el usuario interactúa con una interfaz web intuitiva desarrollada en HTML, CSS y JavaScript. La capa lógica, implementada en un framework como Django o Flask, gestiona las reglas de negocio y las validaciones. Finalmente, la capa de datos utiliza una base de datos relacional como MySQL para garantizar integridad y consistencia en el almacenamiento.

Estándares de Calidad Aplicados

Para garantizar la calidad del software, se aplican los estándares ISO/IEC 25010 e IEEE 830. El primero orienta sobre las características de calidad como usabilidad, fiabilidad y mantenibilidad. El estándar IEEE 830 establece las directrices para la correcta documentación de los requerimientos del sistema, asegurando precisión y verificabilidad. Además, se adoptaron prácticas de control de versiones y revisiones continuas para mantener la trazabilidad y control de cambios durante el desarrollo.

Costos empleados para el proyecto

En términos de costos, considerando el uso de tecnologías de código abierto (Django, Python, VS Code y bases de datos gratuitas), el sistema no requiere licencias de software pagadas. Los principales gastos se asocian a la infraestructura de despliegue, como el **dominio web** (aproximadamente \$12.000 CLP anuales) y un **servicio de hosting compartido** para alojar la aplicación (en torno a \$8.000 CLP mensuales, es decir, cerca de \$96.000 CLP anuales). A esto se puede sumar un costo referencial de **correos institucionales y configuración inicial** cercano a \$30.000 CLP.

De este modo, el costo estimado de operación e implementación del sistema se sitúa aproximadamente en **\$140.000 pesos chilenos durante el primer año**, sin considerar las horas profesionales de desarrollo. Si se incorporara el trabajo del desarrollador como prestación de servicios, el valor total podría situarse entre **\$800.000 y \$1.000.000 de pesos chilenos**, dependiendo de la cantidad de horas y la tarifa utilizada.

Conclusión

El proyecto permitió comprender la importancia del análisis de requerimientos y la correcta selección de metodologías ágiles en el desarrollo de sistemas educativos. El Sistema de Control de Asistencia Escolar constituye una herramienta tecnológica que optimiza el registro de asistencia, reduce errores humanos y mejora la gestión del tiempo docente. A través de la aplicación de estándares de calidad, diagramas UML y una arquitectura modular, se garantiza un producto final eficiente, adaptable y de fácil mantenimiento.