

Checklist		
Código:	Versión:	Fecha:

Documento de Especificación de Arquitectura

Realizado por: **María Fernanda Rios Rodríguez**
Ivan Camilo Aguilar
Alejandro Roncancio Roldan

Checklist		
Código:	Versión:	Fecha:

HISTORIAL DE REVISIONES

Fecha	Versión	Autor	Descripción	Revisado Por

Checklist		
Código:	Versión:	Fecha:

Contenido

1.	Documento de Arquitectura de Software	4
1.1.	Introducción	4
1.2.	Propósito	4
1.3.	Alcance	4
1.4.	Referencias	5
1.5.	Definiciones acrónimos y abreviaciones	5
2.	Generalidades del Proyecto	6
2.1.	Problema a Resolver	6
2.2.	Descripción General del Sistema a Desarrollar	6
2.3.	Identificación de los Stakeholders y sus responsabilidades	6
3.	Vistas de la arquitectura	7
3.1.	Vista de Casos de Uso	7
3.2.	Vista de Procesos	8
3.3.	Vista Lógica	8
3.4.	Vista de Implementación	9
3.5.	Vista de Despliegue	9
4.	Arquitectura en capas	10
5.	Vista de Datos	10
5.1.	Modelo Relacional	10
6.	Definición de Interfaces de Usuario.....	11
7.	Características Generales de Calidad	12
8.	Stack Tecnológico	14

Checklist		
Código:	Versión:	Fecha:

1. Documento de Arquitectura de Software

1.1. Introducción

La gestión de la asistencia en entornos educativos es una tarea fundamental para el seguimiento académico y el control administrativo de los estudiantes. En muchos centros educativos, este proceso es realizado de manera manual, lo que puede resultar ineficiente, propenso a errores y demandante en términos de tiempo. Ante esta problemática, este proyecto presenta un sistema de control de asistencia automatizado mediante el uso de carnets de identificación. Al ingresar a clase, cada estudiante debe acercar su carnet a un lector, lo que permite que el sistema registre automáticamente su asistencia en tiempo real. Este enfoque no solo reduce el tiempo y el esfuerzo en la toma de asistencia, sino que también garantiza una mayor precisión y seguridad en el manejo de los registros. Además, proporciona una base de datos centralizada que facilita el acceso y la gestión de la información por parte de docentes y personal administrativo.

1.2. Propósito

El propósito de este proyecto es desarrollar un sistema automatizado de control de asistencia en instituciones educativas, utilizando la tecnología de identificación mediante carnets. Este sistema busca optimizar el proceso de registro de asistencia al minimizar el tiempo requerido para el ingreso de datos y reducir errores humanos. Al implementar una solución eficiente y precisa, se pretende facilitar a los docentes y al personal administrativo el seguimiento y gestión de la asistencia estudiantil, mejorar la puntualidad en el inicio de las clases, y ofrecer una herramienta confiable para la generación de reportes de asistencia que apoyen la toma de decisiones académicas y administrativas.

1.3. Alcance

El sistema de control de asistencia automatizado mediante carnets está diseñado para ser implementado en entornos educativos, específicamente en universidades y centros de formación. Este sistema permitirá registrar la asistencia de los estudiantes al momento de ingresar al aula, mediante la aproximación de su carnet a un lector, el cual capturará y almacenará los datos en una base centralizada.

El alcance del proyecto incluye las siguientes funciones:

Checklist		
Código:	Versión:	Fecha:

1. Registro de asistencia: Captura automática de la asistencia del estudiante al acercar su carnet al lector al ingresar a la clase.
2. Base de datos centralizada: Almacenamiento de registros de asistencia en una base de datos accesible para el personal autorizado.
3. Generación de reportes: Creación de reportes de asistencia que puedan ser visualizados y exportados para análisis y auditoría.
4. Interfaz de usuario amigable: Desarrollo de una interfaz de usuario intuitiva que permita a los administradores y docentes acceder a la información de asistencia de manera rápida y sencilla.
5. Control de acceso: Implementación de roles y permisos para garantizar que solo el personal autorizado pueda acceder o modificar los registros de asistencia.

El proyecto abarcará el diseño, desarrollo, pruebas e implementación del sistema dentro de un entorno controlado, con vistas a una futura expansión y adaptación según las necesidades de la institución educativa.

1.4. Referencias

1. Documento de Especificación de Requerimientos no funcionales.
2. Documento de Visión del Proyecto.
3. Plan de Proyecto del Sistema

1.5. Definiciones acrónimos y abreviaciones

ARQUITECTURA DE SOFTWARE: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al Comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

DESCRIPCION DE ARQUITECTURA: colección de productos de documentación.

VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

TIPOS DE VISTAS: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

STAKEHOLDER: Individuo, equipo u organización con intereses relativos al sistema.

Checklist		
Código:	Versión:	Fecha:

2. Generalidades del Proyecto

2.1. Problema a Resolver

El registro manual de asistencia en la universidad genera retrasos, errores y dificulta la gestión administrativa. Además, la falta de un sistema centralizado impide un monitoreo eficiente y permite la falsificación de registros. Se necesita una solución automatizada para optimizar el proceso, garantizar la precisión y mejorar la seguridad en la toma de asistencia de estudiantes por medio del carnet con lector de código.

2.2. Descripción General del Sistema a Desarrollar

El sistema permitirá que los estudiantes registren su asistencia acercando su carnet a un lector RFID. La información se almacenará en una base de datos y estará disponible para consultas, generación de reportes y análisis estadístico.

2.3. Identificación de los Stakeholders y sus responsabilidades

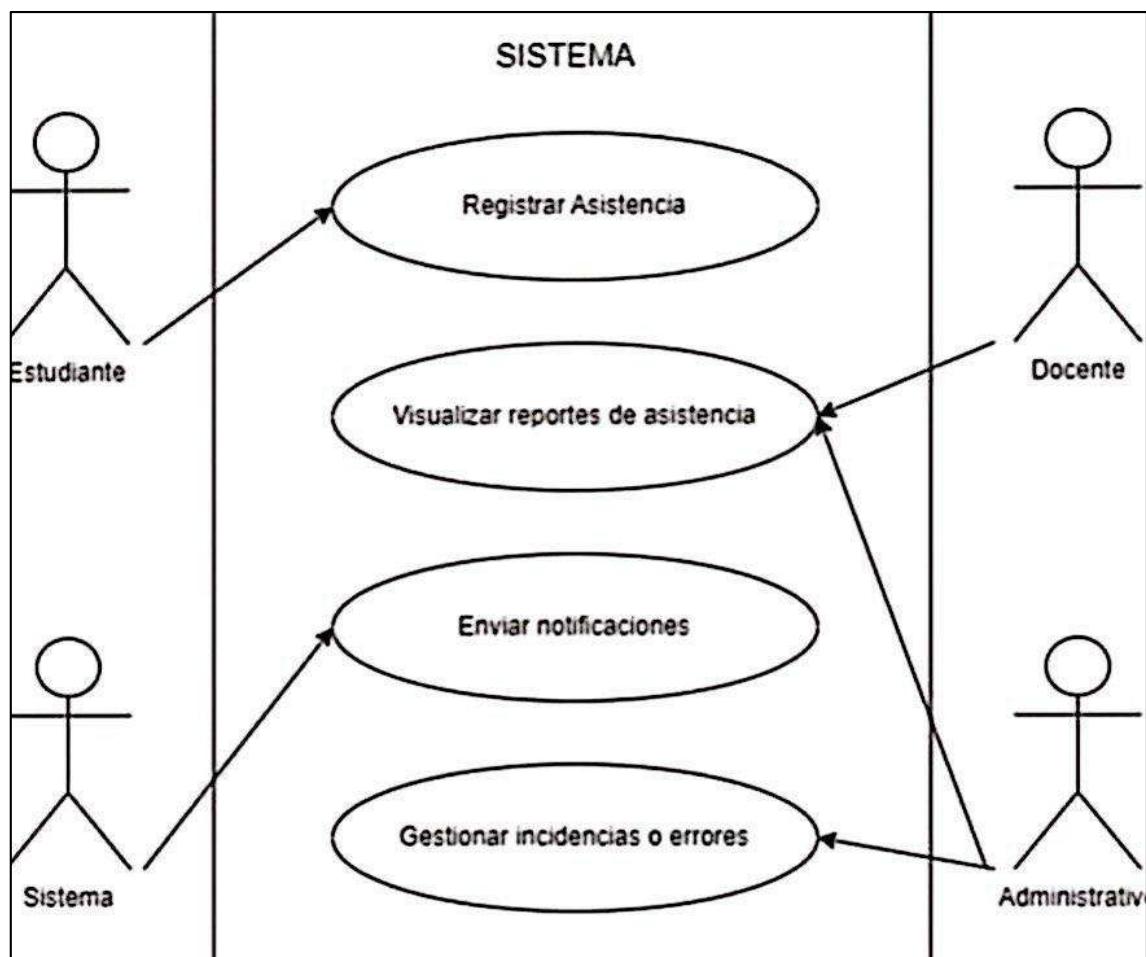
STAKEHOLDER	DESCRIPCIÓN	ESCENARIO	Caso de Uso
Administrador del sistema	Responsable de la gestión del sistema y usuarios.	• Escenario de Administración	<ul style="list-style-type: none"> • CU_Admin_Gestionar_Usuario • CU_Admin_Generar_Reportes
Docente	Puede consultar y analizar la asistencia de sus estudiantes.	• Escenario académico	<ul style="list-style-type: none"> • CU_Docente_Consultar_Asistencia
Estudiante	Usuario que registra su asistencia con su carnet	• Escenario de usuario	<ul style="list-style-type: none"> • CU_Estudiante_Registrar_Asistencia

Checklist		
Código:	Versión:	Fecha:

Personal mantenimiento del sistema	Encargado del mantenimiento y soporte del sistema.	<ul style="list-style-type: none"> • Escenario técnico 	<ul style="list-style-type: none"> • CU_TI_Mantenimiento_Sistema
Directivos Académicos	Utilizan los reportes generados para toma de decisiones.	<ul style="list-style-type: none"> • Escenario administrativo y académico 	<ul style="list-style-type: none"> • CU_Directivos_Analizar_Reportes

3. Vistas de la arquitectura

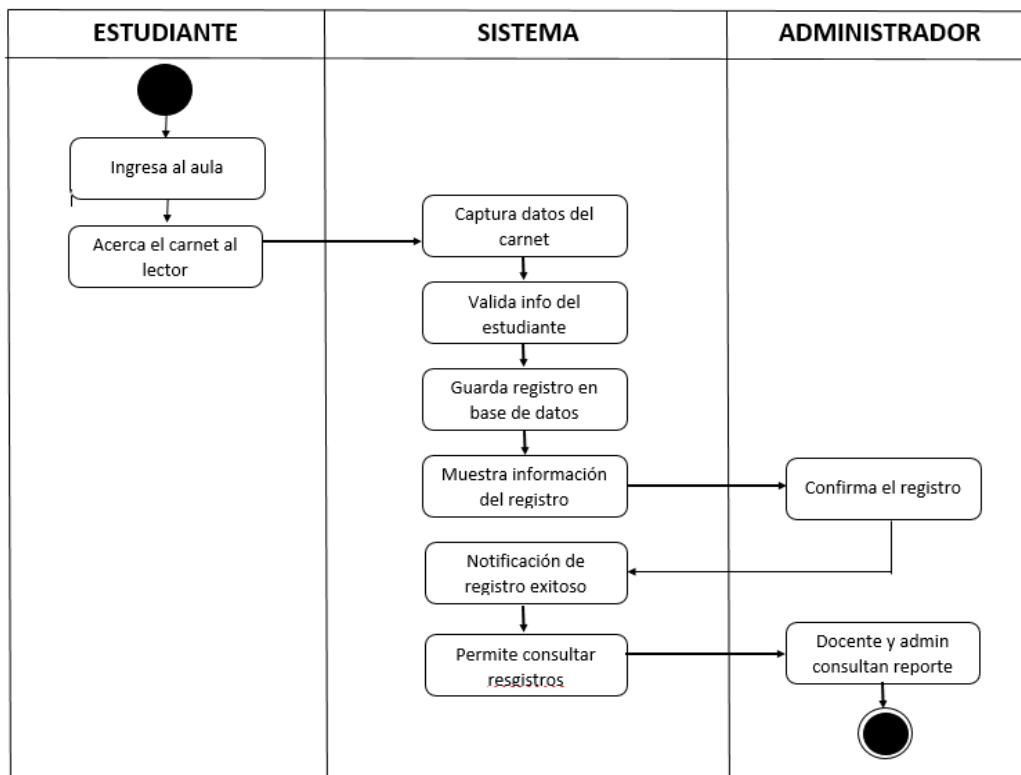
3.1. Vista de Casos de Uso



Checklist		
Código:	Versión:	Fecha:

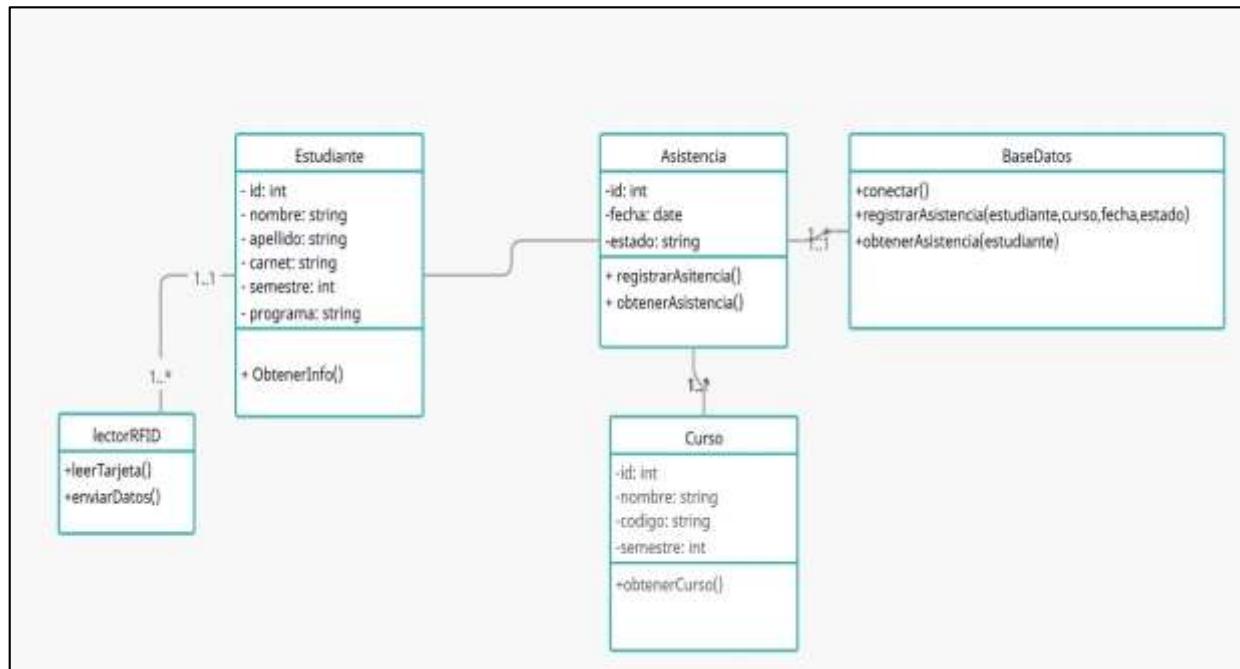
3.2. Vista de Procesos

3.2.1. Diagrama de Actividades



3.3. Vista Lógica

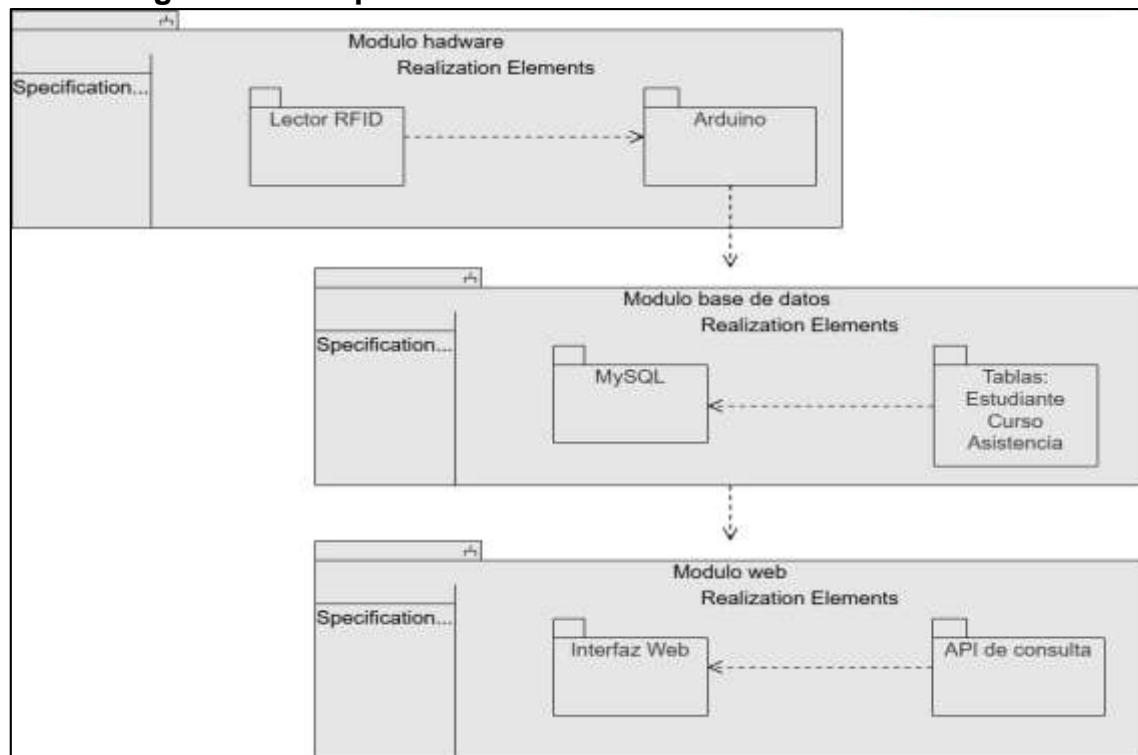
3.3.1. Diagramas – Clases



Checklist		
Código:	Versión:	Fecha:

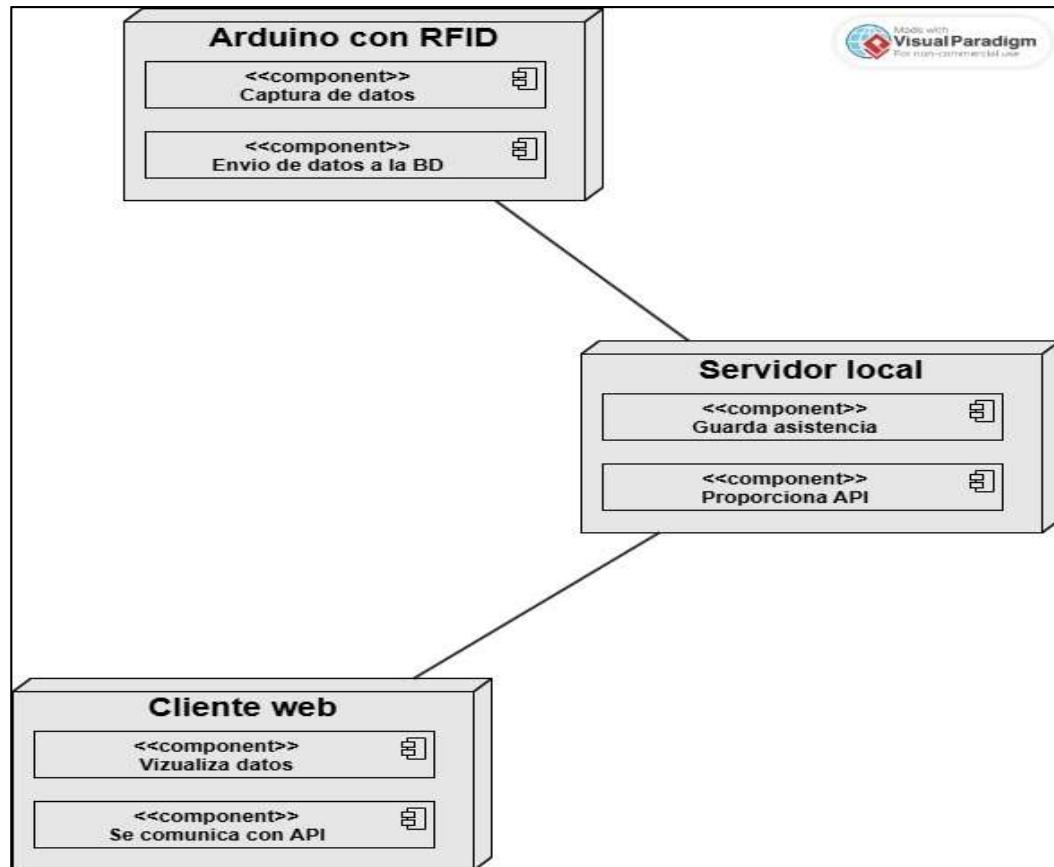
3.4. Vista de Implementación

3.4.1. Diagrama de Paquetes



3.5. Vista de Despliegue

3.5.1. Diagrama de despliegue



Checklist		
Código:	Versión:	Fecha:

4. Arquitectura en capas

El sistema de control de asistencia basado en tecnología RFID estará diseñado bajo una arquitectura en capas para mejorar su mantenibilidad, escalabilidad y seguridad. A continuación, se describen las principales capas del sistema:

1. Capa de Presentación (Frontend):

- Interfaz gráfica para la interacción de los usuarios (estudiantes, docentes y administradores).
- Comunicación con el backend mediante API REST.

2. Capa de Negocio (Lógica de Aplicación):

- Procesa las reglas de negocio del sistema, como la validación de asistencia y gestión de reportes.
- Implementada en Node.js con Express para manejar peticiones del frontend y acceso a datos.
- Control de autenticación y autorización basado en roles.

3. Capa de Datos:

- Base de datos relacional MySQL para almacenar registros de asistencia, usuarios y cursos.
- Optimización de consultas mediante índices y relaciones eficientes.
- Implementación de copias de seguridad y medidas de seguridad en el almacenamiento de datos.

4. Capa de Integración:

- API REST para la comunicación entre el backend y el frontend.
- Conexión con lectores RFID para la captura de asistencia en tiempo real.
- Posibilidad de integración con otros sistemas de gestión académica de la universidad.

5. Vista de Datos

5.1. Modelo Relacional

Checklist		
Código:	Versión:	Fecha:

6. Definición de Interfaces de Usuario

Mapeo de navegación:

1. Pantalla de Inicio de Sesión

- Campos de usuario y contraseña.
- Opción de recuperación de contraseña.
- Acceso según el rol (Docente, Administrador).

2. Panel de Control (Dashboard)

- Estadísticas de asistencia en tiempo real.
- Accesos directos a las funcionalidades principales según el rol del usuario.

3. Registro de Asistencia (Solo para Estudiantes)

- Detección automática del carnet RFID.
- Historial de asistencias personales.

4. Consulta de Asistencia (Docentes y Administradores)

- Búsqueda por estudiante, curso o fecha.
- Filtros avanzados para análisis detallado.
- Exportación de reportes en formatos CSV y PDF.

5. Gestión de Usuarios (Administradores)

- Creación y edición de usuarios.
- Asignación de roles y permisos.
- Eliminación o suspensión de cuentas.

6. Notificaciones y Alertas

- Aviso de inasistencias recurrentes.
- Notificación automática a estudiantes y docentes.
- Personalización de parámetros de notificación.

Demostración de las interfaces:

1. Pantalla de Inicio de Sesión

- Diseño minimalista con campos de usuario y contraseña.

Checklist		
Código:	Versión:	Fecha:

- Mensajes de error en caso de credenciales incorrectas.

2. Panel de Control

- Gráficos estadísticos sobre asistencia general.
- Resumen de clases con mayor número de inasistencias.

3. Registro de Asistencia

- Confirmación en pantalla tras lectura del carnet.
- Historial detallado con fechas y horarios de asistencia.

4. Consulta de Asistencia

- Tabla con información filtrable y exportable.
- Indicadores visuales para asistencia puntual, tardía o ausente.

5. Gestión de Usuarios

- Listado de usuarios con opciones de edición y eliminación.
- Asignación de permisos según el rol.

Este diseño de interfaces busca garantizar una navegación eficiente, optimizando la interacción entre los diferentes tipos de usuarios y asegurando una gestión fluida del sistema de asistencia.

7. Características Generales de Calidad

1. **Disponibilidad:** El sistema debe estar operativo 24/7 para evitar interrupciones en el registro de asistencia y garantizar el acceso en cualquier momento.
2. **Escalabilidad:** La arquitectura debe permitir la incorporación de más estudiantes, aulas y dispositivos sin afectar el rendimiento.

Checklist		
Código:	Versión:	Fecha:

3. **Seguridad:** Se deben aplicar mecanismos de autenticación, encriptación y control de acceso para proteger los datos de asistencia contra accesos no autorizados.
4. **Rendimiento:** El sistema debe procesar los registros en tiempo real sin demoras perceptibles, asegurando un tiempo de respuesta óptimo.
5. **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar para docentes, estudiantes y administradores, minimizando la curva de aprendizaje.
6. **Mantenibilidad:** Se debe garantizar un código modular y bien documentado que facilite actualizaciones y correcciones sin afectar la operatividad del código.
7. **Fiabilidad:** La precisión en el registro de asistencia debe ser alta, asegurando que no haya pérdidas o alteraciones en los datos ya que podría generar errores.
8. **Interoperabilidad:** Capacidad para integrarse con otros sistemas académicos y administrativos utilizados por la universidad San Buenaventura.
9. **Auditabilidad:** El sistema debe generar registros detallados que permitan la verificación y auditoría de la información almacenada.
10. **Portabilidad:** Debe ser posible desplegar el sistema en diferentes infraestructuras, ya sea en servidores en la nube o en entornos locales.

Checklist		
Código:	Versión:	Fecha:

8. Stack Tecnológico

El sistema de control de asistencia utilizará una infraestructura basada en tecnología moderna para garantizar eficiencia, seguridad y escalabilidad. El backend estará desarrollado en **JavaScript** utilizando **Node.js con Express**, proporcionando una API REST que facilitará la comunicación con el frontend y la base de datos.

Para el almacenamiento de la información, se usará **MySQL**, una base de datos relacional optimizada para consultas rápidas y seguras asegurando una interfaz de usuario moderna y fácil de usar. En cuanto al hardware, el sistema contará con **lectores RFID** para la captura automática de asistencia y servidores alojados en la nube, garantizando alta disponibilidad, balanceo de carga y monitoreo en tiempo real.

Este stack tecnológico permitirá una integración fluida con otros sistemas académicos de la universidad, asegurando una experiencia eficiente y confiable para todos los usuarios.