



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

**Proyecto “Aplicación Móvil para la Gestión y
Justificación de Inasistencias de la Universidad
Privada de Tacna”**

Curso: Tópicos de Base de Datos I

Docente: Ing. Patrick Cuadros Quiroga

Integrantes:

Chávez Linares, César Fabián	(2019063854)
Cristian Aldair Quispe Levano	(2018000590)
Neira Machaca, Javier André	(2017057984)
Delgado Castillo, Jesús Ángel	(2018000491)

**Tacna – Perú
2024**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	CCL	PCQ	PCQ	24/11/2024	Versión Original

**Aplicación Móvil para la Gestión y Justificación de
Inasistencias de la Universidad Privada de Tacna
Documento de Arquitectura de Software**

Versión 1.0

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	CCL	PCQ	PCQ	24/11/2024	Versión Original

INDICE GENERAL

Contenido

1.	5	
1.1.	5	
1.2.	5	
1.3.	5	
1.4.	5	
2.	5	
2.1.1.	6	
2.1.2.	6	
3.	7	
3.1.	7	
3.1.1.	9	
3.2.	10	
3.2.1.	10	
3.2.2.	11	
3.2.3.	11	
3.2.4.	11	
3.2.5.	12	
3.2.6.	12	
3.3.	12	
3.3.1.	12	
3.3.2.	13	
3.4.	13	
3.4.1.	13	
3.5.	14	
3.5.1.	14	
4.	15	
Escenario de Funcionalidad		8
Escenario de Usabilidad		8
Escenario de confiabilidad		9

Escenario de rendimiento	9
Escenario de mantenibilidad	9
Otros Escenarios	9

1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)

El propósito del sistema es ofrecer un sistema de gestión de asistencia para los usuarios mediante una aplicación web desarrollada en React. El sistema ofrece la funcionalidad de autenticación, gestión de asistencia y otros servicios como la visualización de historial de asistencia y la justificación de ausencias. El diagrama 4+1 de la arquitectura proporcionará una visión holística que cubra las vistas de usuario, interacción con el sistema, la arquitectura y la infraestructura física.

1.2. Alcance

El sistema tiene como objetivo proporcionar una plataforma para la gestión de asistencia de los usuarios, permitiendo iniciar sesión, registrar su asistencia y justificar inasistencias. También proporciona acceso a los administradores y organizadores para monitorear la asistencia.

1.3. Definición, siglas y abreviaturas

API: Interfaz de Programación de Aplicaciones.

JWT: JSON Web Token, utilizado para autenticación.

CRUD: Crear, Leer, Actualizar, Eliminar, operaciones básicas en bases de datos.

1.4. Organización del documento

El documento se organiza en las siguientes secciones principales:

1. Introducción: Contexto y propósito del sistema.
2. Objetivos y restricciones arquitectónicas: Requerimientos y limitaciones.
3. Representación de la arquitectura: Diagramas de arquitectura.
4. Atributos de calidad del software: Escenarios de calidad.

2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS

Los requerimientos funcionales describen las funciones específicas que el sistema debe ser capaz de realizar. En este sistema de asistencia, los requerimientos funcionales son los siguientes:

2.1. Priorización de requerimientos

ID	Descripción	Prioridad

1.1.1. Requerimientos Funcionales

ID	Descripción	Prioridad
RF-001	Autenticación de Usuario: Permitir que los usuarios inicien sesión con correo y contraseña.	Alta
RF-002	Sincronización de Datos: Sincronizar el código, horario y asistencias del usuario.	Alta
RF-003	Visualización de Asistencia: Mostrar el historial de asistencias por curso y fecha.	Alta
RF-004	Visualización de Horario: Mostrar el horario de clases de lunes a domingo.	Media
RF-005	Justificación de ausencias: Permitir enviar un formulario con motivo de ausencia.	Media
RF-006	Historial de Justificaciones: Consultar el historial de justificaciones enviadas.	Baja
RF-007	Notificación de Estado: Mostrar notificaciones sobre el estado de acciones, como errores o éxito.	Baja

1.1.2. Requerimientos No Funcionales – Atributos de Calidad

ID	Descripción	Prioridad
RNF-01	Seguridad: Implementar autenticación segura y protección del token de autenticación.	Alta
RNF-02	Rendimiento: Asegurar tiempos de respuesta rápidos (menos de 2 segundos) en la mayoría de operaciones.	Alta
RNF-03	Escalabilidad: El sistema debe poder soportar un número creciente de usuarios sin afectar el rendimiento.	Alta
RNF-04	Usabilidad: La interfaz debe ser clara, fácil de usar y responsiva en dispositivos móviles y de escritorio.	Alta
RNF-05	Mantenibilidad: El sistema debe estar diseñado para facilitar su mantenimiento y expansión.	Media
RNF-06	Disponibilidad: El sistema debe estar disponible el 99.9% del tiempo.	Media
RNF-07	Compatibilidad: El sistema debe ser compatible con los principales navegadores web.	Baja
RNF-08	Confiabilidad: Asegurar que los datos sean consistentes y que el sistema maneje errores adecuadamente.	Baja

2.2. Restricciones

Las restricciones son aquellas limitaciones que pueden afectar el desarrollo y funcionamiento del sistema. Las principales restricciones de este proyecto son las siguientes:

1. Restricciones Técnicas:
 - El sistema debe ser desarrollado utilizando tecnologías web como React para el frontend y Node.js/Express para el backend.
 - La comunicación entre el frontend y el backend debe realizarse a través de APIs RESTful, utilizando JSON como formato de intercambio de datos.
 - El sistema debe usar HTTPS para garantizar la seguridad de las comunicaciones.
 - La base de datos debe estar optimizada para consultas rápidas y debe manejar grandes volúmenes de datos sin afectar el rendimiento.
2. Restricciones de Tiempo:
 - El sistema debe estar completamente funcional en un plazo de 3 meses desde el inicio del desarrollo.
 - Las funcionalidades más críticas (como la autenticación y la sincronización de datos) deben estar implementadas en las primeras 4 semanas.
3. Restricciones de Seguridad:
 - El sistema debe cumplir con los estándares de seguridad más recientes para proteger la información personal de los usuarios, incluyendo medidas como el cifrado de contraseñas y la protección de tokens de sesión.
 - El acceso a la API debe estar asegurado mediante autenticación basada en token (JWT).
4. Restricciones de Compatibilidad:
 - El sistema debe ser accesible a través de los navegadores más utilizados, como Google Chrome, Mozilla Firefox, Safari, y Microsoft Edge.
 - Debe ser compatible con dispositivos móviles y de escritorio, adaptándose a pantallas pequeñas sin perder funcionalidad.

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

3.1. Vista de Caso de uso

En esta sección, se presenta una descripción de los casos de uso del sistema de asistencia, identificando las funcionalidades clave, los actores involucrados y los flujos principales. Se incluyen diagramas y explicaciones para validar el diseño arquitectónico.

Casos de Uso del Sistema

Actor Principal: Usuario

Descripción General: El usuario puede autenticarse, sincronizar sus datos, consultar asistencias y horarios, justificar ausencias y consultar su historial de justificaciones.

Casos de Uso Principales:

Inicio de Sesión

Actor: Usuario

Descripción: El usuario ingresa su correo y contraseña para autenticarse en el sistema.

Flujo de Eventos:

El usuario accede a la vista de login.

Proporciona las credenciales requeridas.

El sistema valida las credenciales y genera un token de autenticación.

El usuario es redirigido al panel de asistencia.

Sincronización de Datos

Actor: Usuario

Descripción: El usuario sincroniza su código y contraseña, así como sus horarios y asistencias.

Flujo de Eventos:

El usuario selecciona la opción de sincronizar.

El sistema solicita los datos necesarios y valida la información.

Los datos se sincronizan con el servidor.

Consulta de Asistencias

Actor: Usuario

Descripción: El usuario consulta sus asistencias por curso y fecha.

Flujo de Eventos:

El usuario accede al panel de asistencia.

El sistema recupera los registros de asistencia del servidor.

Se presentan los datos organizados por curso y fecha.

Justificación de Ausencias

Actor: Usuario

Descripción: El usuario envía un formulario para justificar una ausencia.

Flujo de Eventos:

El usuario completa el formulario con asunto, descripción, fecha y archivo adjunto.

El sistema valida la información y guarda la justificación.

La justificación queda registrada en el historial.

Consulta del Historial de Justificaciones

Actor: Usuario

Descripción: El usuario consulta el historial de justificaciones enviadas.

Flujo de Eventos:

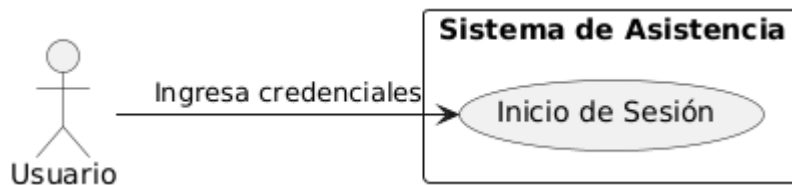
El usuario accede a la vista de historial de justificaciones.

El sistema recupera las justificaciones almacenadas.

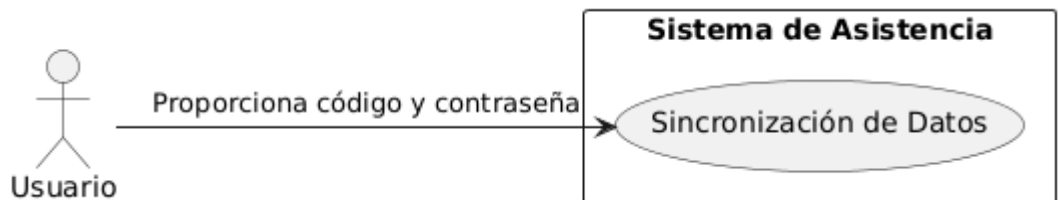
Se presentan los datos al usuario.

1.1.3. Diagramas de Casos de uso

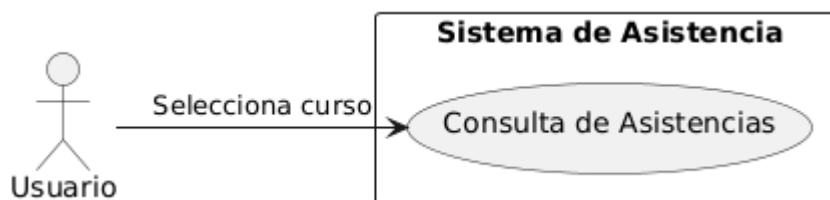
Inicio de Sesión



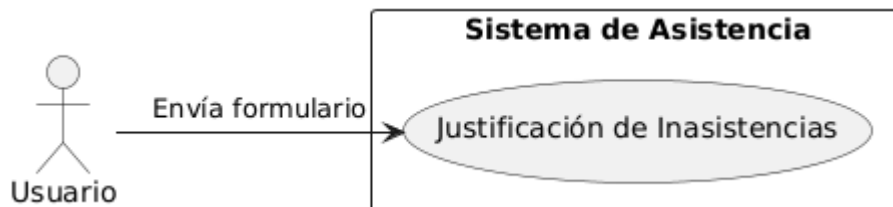
Sincronización de Datos



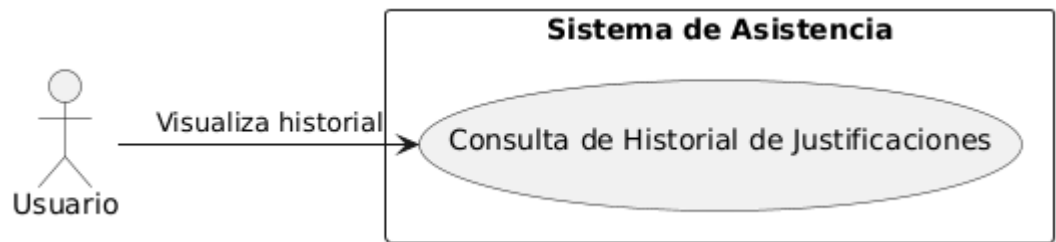
Consulta de Asistencias



Justificación de Inasistencia

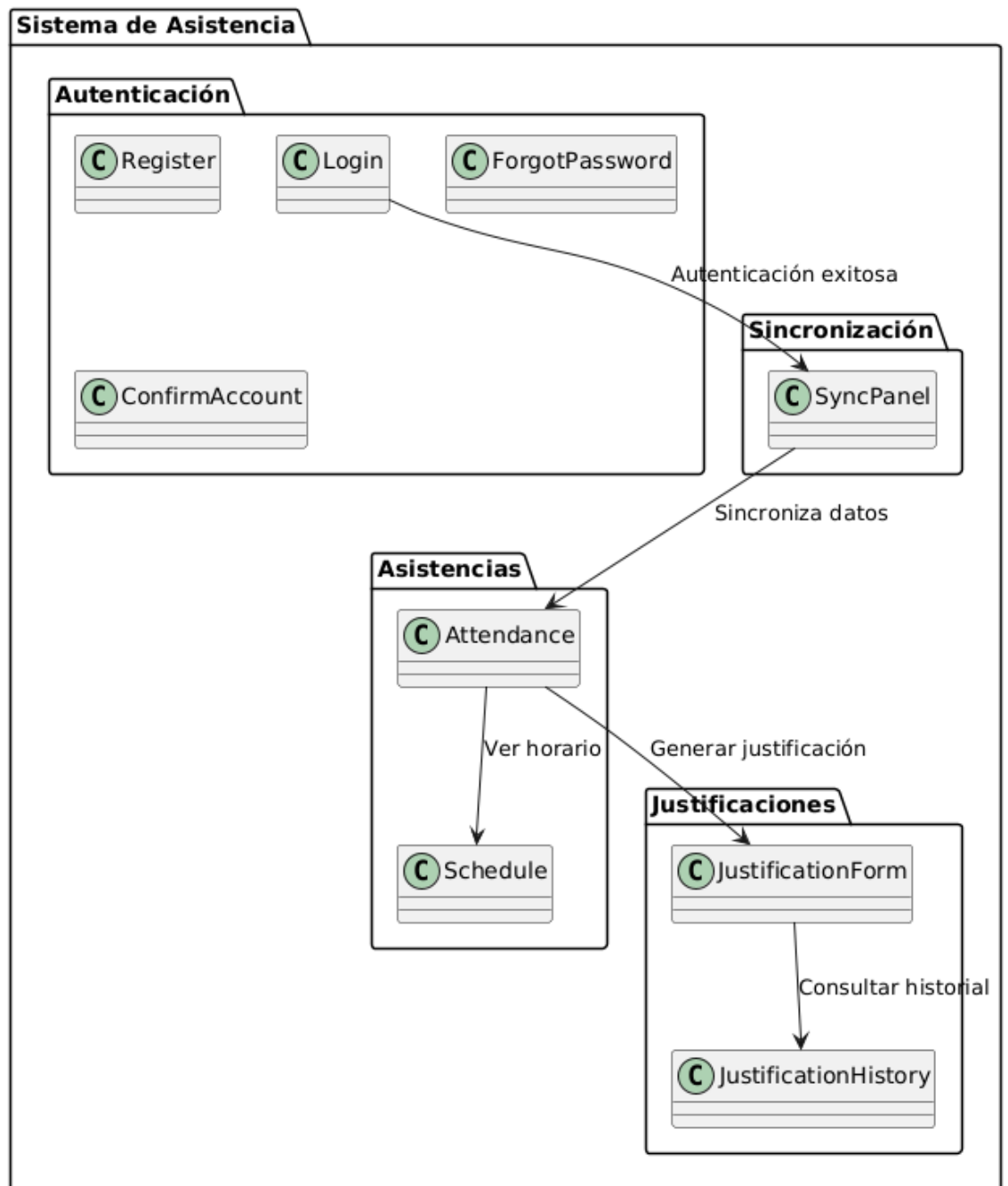


Consulta Historial de Justificaciones

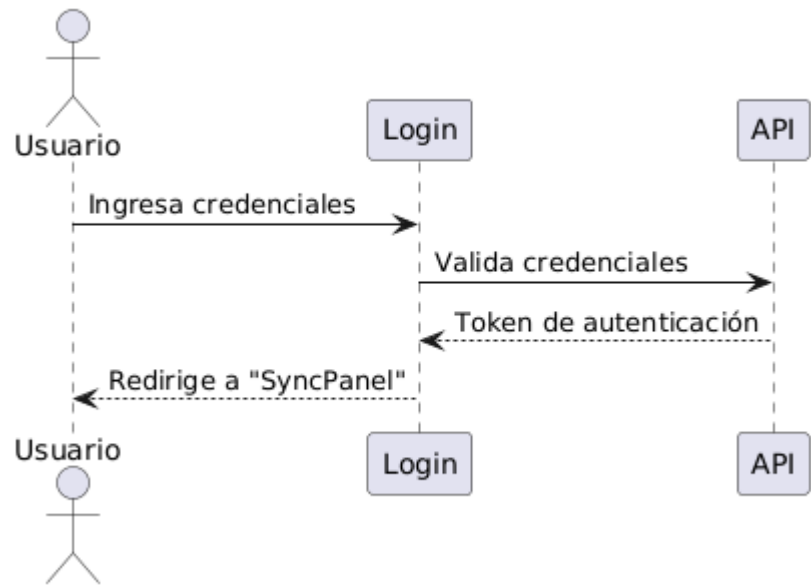


3.2. Vista Lógica

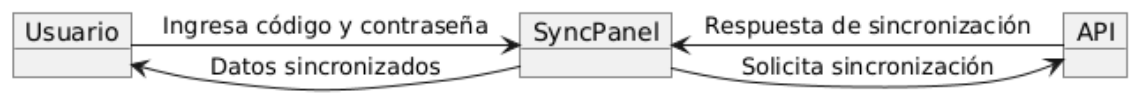
3.2.1. Diagrama de Subsistemas (paquetes)



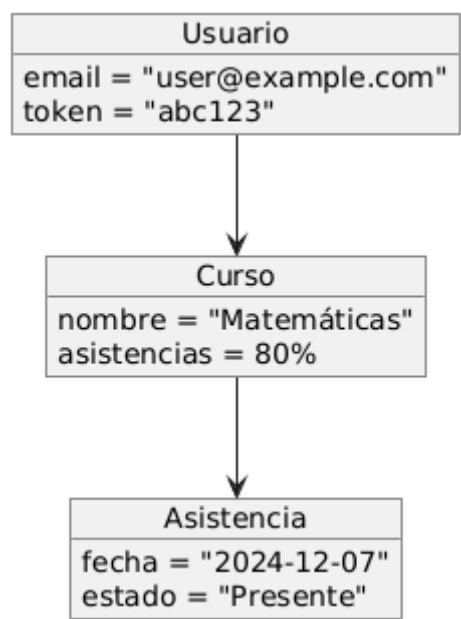
3.2.2. Diagrama de Secuencia (vista de diseño)



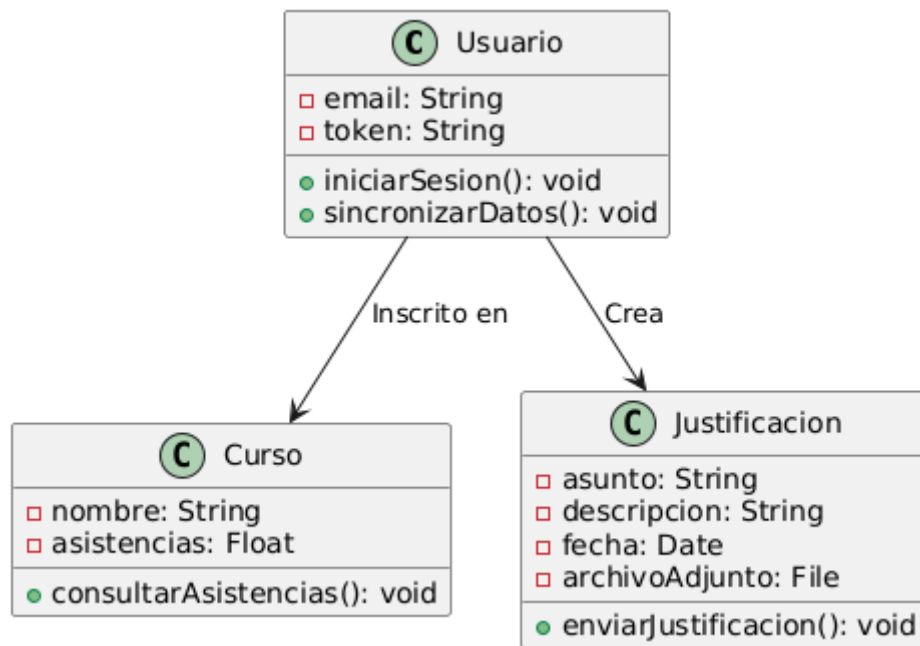
3.2.3. Diagrama de Colaboración (vista de diseño)



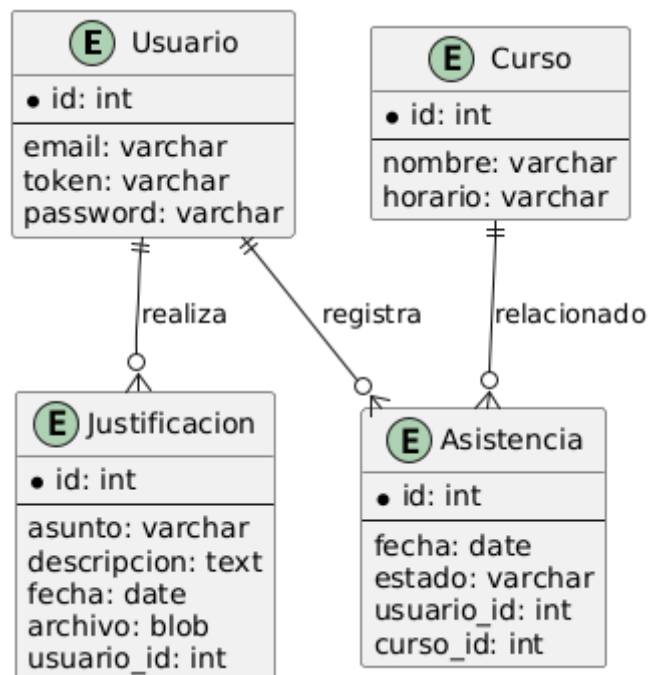
3.2.4. Diagrama de Objetos



3.2.5. Diagrama de Clases

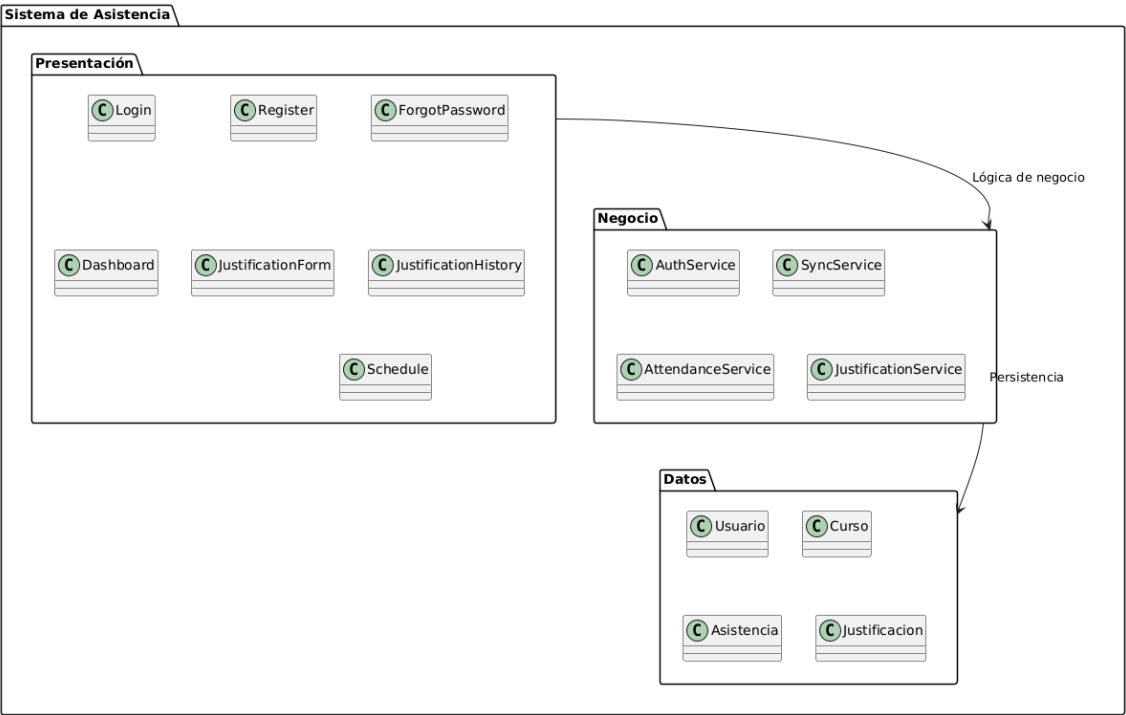


3.2.6. Diagrama de Base de datos (relacional o no relacional)

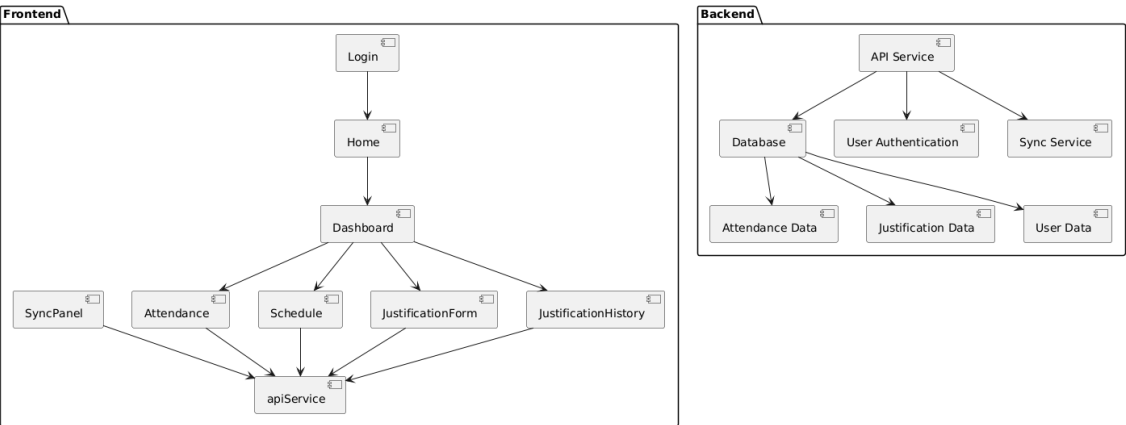


3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)

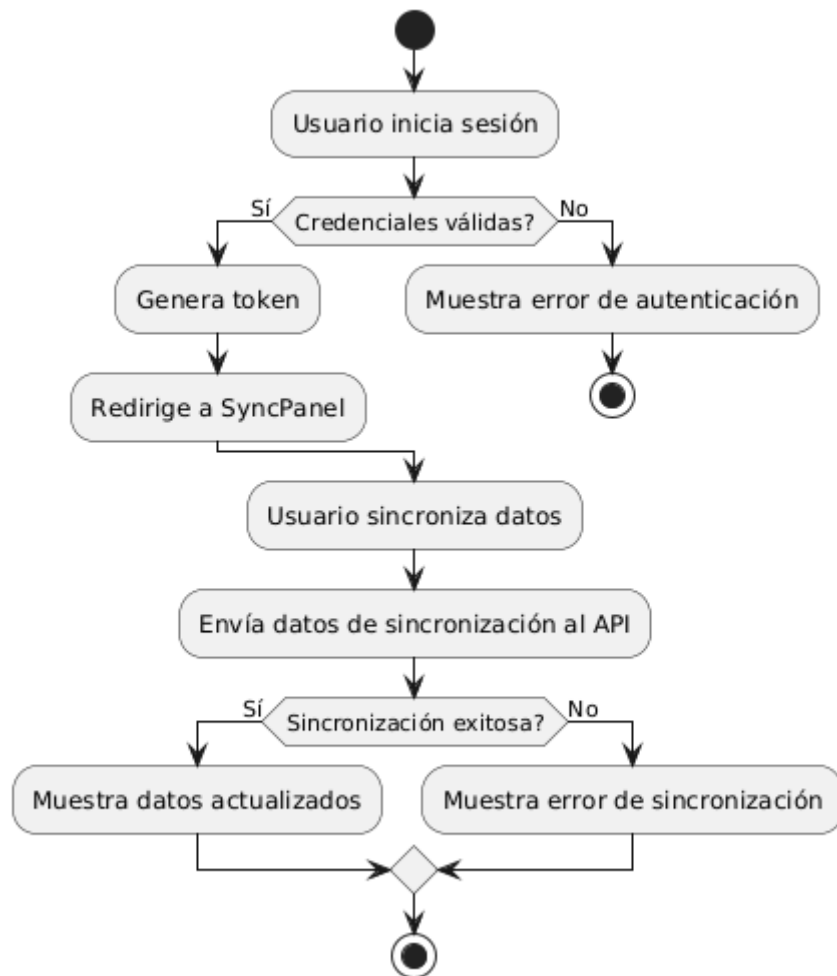


3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)



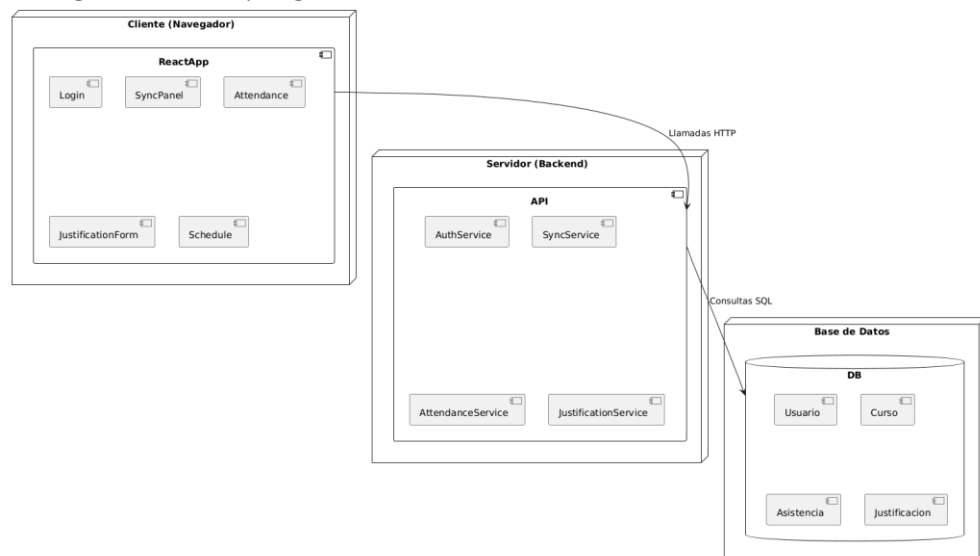
3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

En esta sección se describen los atributos de calidad del sistema de asistencia, definidos de acuerdo con escenarios específicos para garantizar que el sistema cumpla con los requisitos no funcionales establecidos.

Escenario de Funcionalidad

Descripción: Se enfoca en las características y capacidades del sistema para cumplir con sus propósitos funcionales, asegurando la seguridad general en la ejecución de las operaciones.

Ejemplo:

Acción: Un usuario inicia sesión con sus credenciales.

Respuesta esperada: El sistema valida las credenciales de forma segura y redirige al usuario al panel correspondiente sin exponer información confidencial.

Métrica de calidad: Todas las funcionalidades deben operar sin errores en un 95% de las pruebas funcionales realizadas.

Escenario de Usabilidad

Descripción: Considera la facilidad de aprendizaje, eficiencia en el uso y satisfacción del usuario durante la interacción con el sistema.

Ejemplo:

Acción: Un nuevo usuario navega por las funcionalidades principales (sincronización de datos, consulta de horarios y asistencias).

Respuesta esperada: El usuario debe comprender las operaciones básicas en menos de 5 minutos sin asistencia externa.

Métrica de calidad: El 90% de los usuarios deben calificar el sistema como "intuitivo" en pruebas de usabilidad.

Escenario de confiabilidad

Descripción: Garantiza la integridad, disponibilidad y seguridad de los datos y procesos del sistema.

Ejemplo:

Acción: Un usuario envía una justificación con un archivo adjunto.

Respuesta esperada: El archivo se sube de manera segura y está disponible para futuras consultas sin pérdida de datos.

Métrica de calidad: La disponibilidad del sistema debe ser del 99.9%, con un tiempo máximo de recuperación ante fallos de 30 minutos.

Escenario de rendimiento

Descripción: Evalúa la velocidad de procesamiento, tiempo de respuesta y uso eficiente de recursos.

Ejemplo:

Acción: Un usuario consulta su historial de asistencias.

Respuesta esperada: El tiempo de respuesta para mostrar los datos no debe exceder 2 segundos, incluso con 500 usuarios simultáneos.

Métrica de calidad: El tiempo medio de respuesta del sistema debe ser menor a 1 segundo en un 95% de las solicitudes.

Escenario de mantenibilidad

Descripción: Evalúa la capacidad del sistema para ser ampliado o modificado con facilidad, adaptándose a nuevas necesidades o cambios en el entorno.

Ejemplo:

Acción: Se solicita agregar un nuevo módulo para la consulta de reportes avanzados.

Respuesta esperada: Los cambios deben implementarse en menos de 10 días hábiles, sin impactar las funcionalidades existentes.

Métrica de calidad: El tiempo promedio para realizar modificaciones debe ser inferior a 5 días para cambios menores y 20 días para cambios mayores.

Otros Escenarios

Performance

Descripción: Evalúa la capacidad del sistema para manejar cargas de trabajo y responder a eventos en tiempo real.

Ejemplo:

Acción: 1,000 usuarios acceden simultáneamente al sistema para sincronizar sus horarios.

Respuesta esperada: El sistema maneja las solicitudes sin interrupciones ni degradación significativa en el tiempo de respuesta.

Métrica de calidad: El sistema debe soportar hasta 2,000 solicitudes por minuto con un tiempo de respuesta inferior a 1.5 segundos en el 90% de los casos.