

GestiHorarios

Documento de Arquitectura de Software

*Mateo Cadena
Andres Martinez
Miguel Llantén*

Arquitectura y diseño de software
Facultad de ingeniería
Institucion educativa Colegio Mayor del Cauca
2024

Historial de Revisión

Versión	Autores	Fecha
0.1	<i>Mateo Cadena, Andres Martinez, Miguel Llantén</i>	Mayo 2023
0.31	<i>Francisco Obando</i>	Mayo 2023

Tabla de Contenido

	Página
1 Introducción	4
2 Descripción del problema	4
3 Metas y Restricciones de la Arquitectura	5
4 Vista de Casos de Uso y Escenarios de Calidad	8
4.1 Modelo de Casos de Uso	9
4.2 Especificación de los Casos de Uso Relevantes	10
4.3 Especificación de los Escenarios de calidad Relevantes	13
5 Vista Lógica	16
5.1 Parte Estructural: descomposición modular	16
5.1.1 Primer Nivel de Refinamiento	16
5.1.2 Segundo Nivel de Refinamiento	18
5.2 Parte Dinámica (Comportamiento)	23
5.2.1 View Packet Negocio Ventas	23
5.2.2 View Packet Negocio Producto	26
5.2.3 View Packet Negocio Envío	28
5.2.4 View Packet Negocio Financiero	30
6 Vista de Despliegue	32
6.1 Tecnología requerida	33
7 Rationale	33

GestiHorarios

Software Architecture Document - SAD

1 Introducción

El presente documento tiene como objetivo describir la arquitectura del sistema GestiHorarios, una aplicación web enfocada en la gestión y asignación de horarios académicos. Este documento servirá como guía para el diseño, desarrollo, implementación y mantenimiento del sistema, proporcionando una visión detallada de los componentes arquitectónicos, patrones de diseño, tecnologías empleadas y consideraciones relevantes.

GestiHorarios es una solución integral diseñada para simplificar y optimizar el proceso de gestión de horarios académicos en instituciones educativas. Permite a los administradores, coordinadores y personal docente asignar eficientemente los recursos disponibles, como aulas, profesores y materias, de acuerdo con las necesidades y restricciones específicas de cada institución.

El propósito principal de este documento es proporcionar una descripción clara y concisa de la arquitectura del sistema, lo que facilitará la comprensión de los aspectos técnicos y funcionales por parte de los diferentes actores involucrados en el proyecto. Además, servirá como referencia para futuras modificaciones, ampliaciones o mantenimiento del sistema.

Este documento está dirigido a una audiencia diversa, incluyendo desarrolladores, arquitectos de software, diseñadores y cualquier persona interesada en comprender la arquitectura de GestiHorarios. Se asume un conocimiento básico de conceptos de arquitectura de software, patrones de diseño y tecnologías web.

2 Descripción del problema

GestiHorarios busca resolver los desafíos asociados con la creación de horarios académicos en instituciones educativas, como la Institución Universitaria Colegio Mayor. Actualmente, los secretarios académicos enfrentan dificultades para asignar eficientemente los espacios y recursos disponibles, considerando los requisitos y restricciones de las asignaturas, docentes y espacios.

Contexto y antecedentes: La Institución Universitaria Colegio Mayor ha estado utilizando un proceso manual para la creación de horarios académicos, lo cual ha resultado en dificultades para asignar eficientemente los espacios y recursos disponibles, considerando los requisitos y restricciones de las asignaturas, docentes y espacios. Este proceso manual es propenso a errores y consume una cantidad considerable de tiempo y recursos.

Necesidades y objetivos: Optimizar el proceso de creación de horarios académicos, facilitar la asignación de espacios y recursos de manera efectiva, evitar conflictos en la programación de actividades académicas, ahorrar tiempo y recursos en la gestión de horarios, proporcionar una herramienta eficiente y fácil de usar para los secretarios académicos.

Problemas y desafíos actuales: Proceso manual y propenso a errores para la asignación de espacios, dificultad para considerar todos los requisitos y restricciones de asignaturas, docentes y espacios, posibles conflictos en la programación de actividades académicas, consumo excesivo de tiempo y recursos en la creación de horarios.

Limitaciones y restricciones: GestiHorarios debe ser desarrollado como una aplicación web utilizando tecnologías estándar, se debe utilizar un framework de desarrollo web popular y ampliamente adoptado la base de datos debe ser relacional y compatible con el estándar SQL.

Impacto del problema: El proceso manual actual para la creación de horarios académicos en la Institución Universitaria Colegio Mayor ha resultado en ineficiencias, conflictos en la programación y un consumo excesivo de tiempo y recursos. Esto afecta negativamente la productividad de los secretarios académicos y puede generar insatisfacción entre los docentes y estudiantes debido a conflictos de horarios o asignaciones inadecuadas de espacios.



Figura 1. Diagrama de contexto

3 Representación de la Arquitectura

La arquitectura de GestiHorarios se expone mediante una amalgama de elementos que incluyen diagramas, modelos arquitectónicos, vistas y documentación textual. Este enfoque permite una comprensión del sistema, permitiendo visualizar, entender y analizar los elementos clave de la arquitectura. El propósito fundamental de esta representación en el documento de arquitectura de software (SAD) es proporcionar una visión general de los diferentes artefactos y técnicas empleadas para comunicar la arquitectura de manera efectiva. Algunos de los elementos esenciales de esta representación incluyen:

Diagramas de arquitectura: Se pueden incluir diagramas que representen la estructura, los componentes y las interacciones del sistema de software. Esto puede incluir diagramas de bloques, diagramas de componentes, diagramas de despliegue, diagramas de flujo de datos, diagramas de secuencia, entre otros. Estos diagramas ayudan a visualizar y comprender la arquitectura del sistema desde diferentes perspectivas.

Modelos arquitectónicos: Se pueden utilizar modelos arquitectónicos estándar, como el modelo en capas, el modelo cliente-servidor o el modelo de microservicios, para representar la estructura y las relaciones entre los componentes del sistema. Estos modelos brindan un marco conceptual para comprender la arquitectura y ayudan a comunicar la organización y la funcionalidad del sistema.

Vistas arquitectónicas: Se pueden utilizar vistas arquitectónicas para representar diferentes aspectos de la arquitectura, como la vista de diseño, la vista de implementación, la vista de

rendimiento o la vista de seguridad. Cada vista se enfoca en aspectos específicos y proporciona detalles adicionales sobre cómo se abordan esos aspectos en la arquitectura.

Documentación textual: Además de los diagramas y modelos, es importante incluir una descripción textual de la arquitectura. Esto puede ser en forma de narrativa, explicando los principios arquitectónicos, las decisiones de diseño y las relaciones entre los componentes. También puede incluir descripciones detalladas de los patrones arquitectónicos utilizados y las justificaciones detrás de esas decisiones.

Herramientas y notaciones: Se pueden mencionar las herramientas utilizadas para crear y documentar los diagramas de arquitectura, como herramientas de modelado UML(Unified Modeling Language) o herramientas de representación visual. Además, se pueden especificar las notaciones y convenciones utilizadas para garantizar la consistencia y la comprensión común en toda la documentación.

EJEMPLO:

La arquitectura del SGT está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado y del método ADD. Las vistas incluidas en esta versión del documento son:

- Vista de Casos de Uso y Escenarios de Calidad: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
- Vista de Metas Y Restricciones: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.
- Vista Lógica: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el viewtype Componentes y Conectores para representar el comportamiento.
- Vista de Procesos: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
- * Vista de Implementación: Describe los componentes de deployment construidos y sus dependencias.

Las vistas presentadas forman en su conjunto una especificación aún incompleta del sistema (en desarrollo). La relación entre estas vistas se describe en la figura 1.

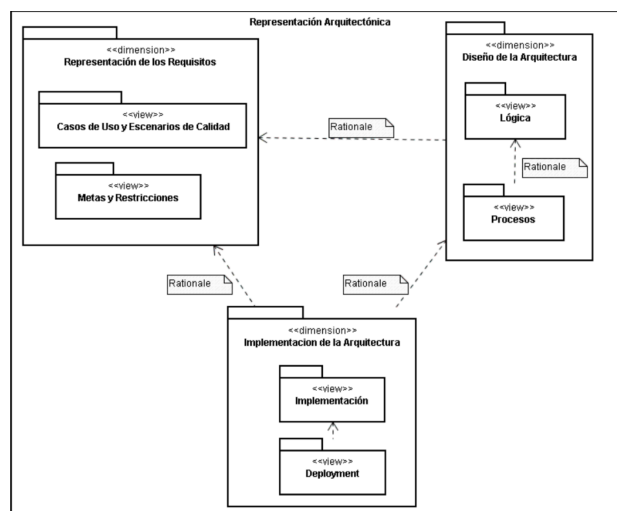


Figura 1. Representación de la arquitectura

3.1 Metas y Restricciones de la Arquitectura

Para definir las metas y restricciones de la arquitectura de GestiHorarios, se llevó a cabo una reunión con los participantes interesados están orientados al ámbito universitario y educativo, y fueron los siguientes:

Coordinador académico: Representado por Mateo Cadena, este rol se centra en garantizar que el sistema cumpla con las necesidades y expectativas del personal docente y los estudiantes. Tiene un punto de vista integral que incluye la capacidad de ver los periodos académicos y desarrollar una visión clara de las necesidades del coordinador académico. Sus intereses se orientan hacia una asignación de horarios justa y eficiente, que optimice los recursos disponibles y satisfaga tanto a profesores como a estudiantes.

Desarrollador de software: Representado por Miguel Llantén, este rol aboga por los intereses del programador encargado de crear y mantener el sistema GestiHorarios. Se enfoca en asegurar la viabilidad del sistema desde los puntos de vista de tiempo, técnica y economía. Su objetivo es desarrollar una solución que sea técnicamente sólida, dentro del presupuesto y que pueda ser implementada en un marco temporal adecuado.

Docente Académico: Este rol fue representado por Andrés Martínez, quien aporta información para el correcto comportamiento del sistema.

De esta reunión surgieron las principales metas y restricciones iniciales de la arquitectura que son:

Eficiencia en la asignación de horarios: El sistema debe ser capaz de asignar horarios de manera eficiente, optimizando el uso de recursos como aulas, profesores y espacios, para asegurar una distribución equitativa y satisfactoria para todas las partes involucradas.

Escalabilidad: El sistema debe ser escalable para adaptarse al crecimiento futuro de la institución educativa y las posibles expansiones, permitiendo gestionar un mayor número de cursos, profesores y estudiantes sin comprometer su rendimiento.

Tolerancia a fallos: El sistema debe ser robusto y capaz de tolerar posibles fallos, como interrupciones de servicio o errores de procesamiento, sin afectar negativamente la continuidad y calidad del proceso de gestión de horarios.

Flexibilidad y adaptabilidad: El sistema debe ser flexible y adaptable para soportar cambios en los procesos y políticas de asignación de horarios, así como para integrarse con sistemas existentes en la institución educativa, garantizando su evolución y reutilización en diferentes contextos.

Desde la perspectiva del negocio, existen restricciones de tiempo y costos que deben ser consideradas en las decisiones arquitectónicas:

- El proyecto de desarrollo de GestiHorarios cuenta con un plazo estimado de entrega de 2 meses, incluyendo la fase de implementación.
- Se dispone de un equipo de 3 personas para el desarrollo y mantenimiento del sistema.
- Además, se deben tener en cuenta las siguientes consideraciones específicas del contexto universitario:
 - El sistema debe cumplir con las regulaciones y políticas de seguridad de datos en el ámbito educativo, garantizando la confidencialidad y privacidad de la información de los usuarios.
 - Se espera que GestiHorarios pueda integrarse con otros sistemas y plataformas utilizadas en la institución educativa, como sistemas de gestión académica y plataformas de aprendizaje en línea, para facilitar la interoperabilidad y el intercambio de datos.

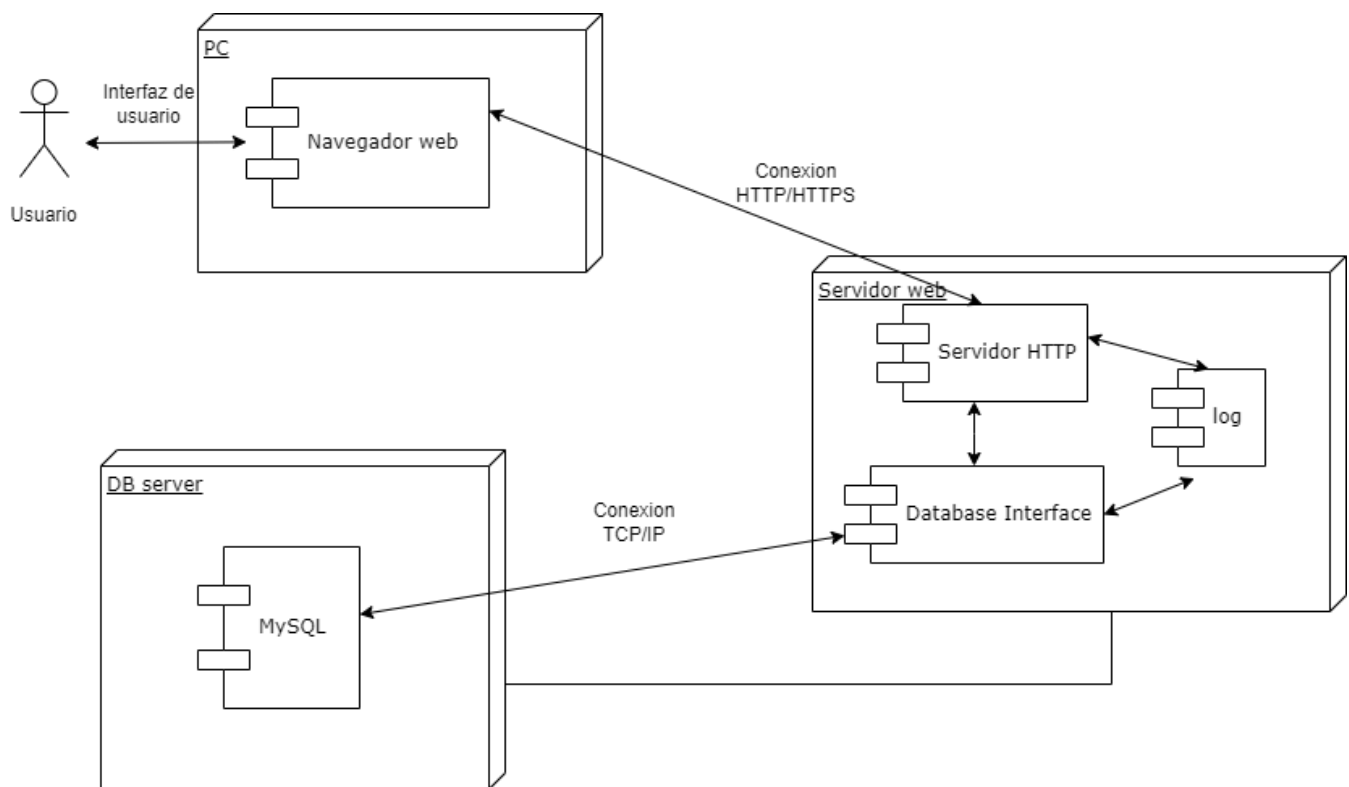


Figura 2. Diagrama de despliegue preliminar

4. Vista de Casos de Uso y Escenarios de Calidad

Esta parte describe en detalle el conjunto de escenarios funcionales y no funcionales, se presenta y describe un modelos de casos de uso dividido en cinco partes, así como los escenarios en que uno o más atributos de calidad se ven involucrados.

4.1 Modelo de Casos de Uso

El modelo de casos de uso de la aplicación web “GestiHorarios” se presenta en las Figuras 3,4,5,6,7. En él se pueden visualizar los actores principales del sistema, así como los principales usos que estos darán al sistema.

Este modelo de casos de uso se ha dividido en cinco parte para una mejor comprensión y organización:

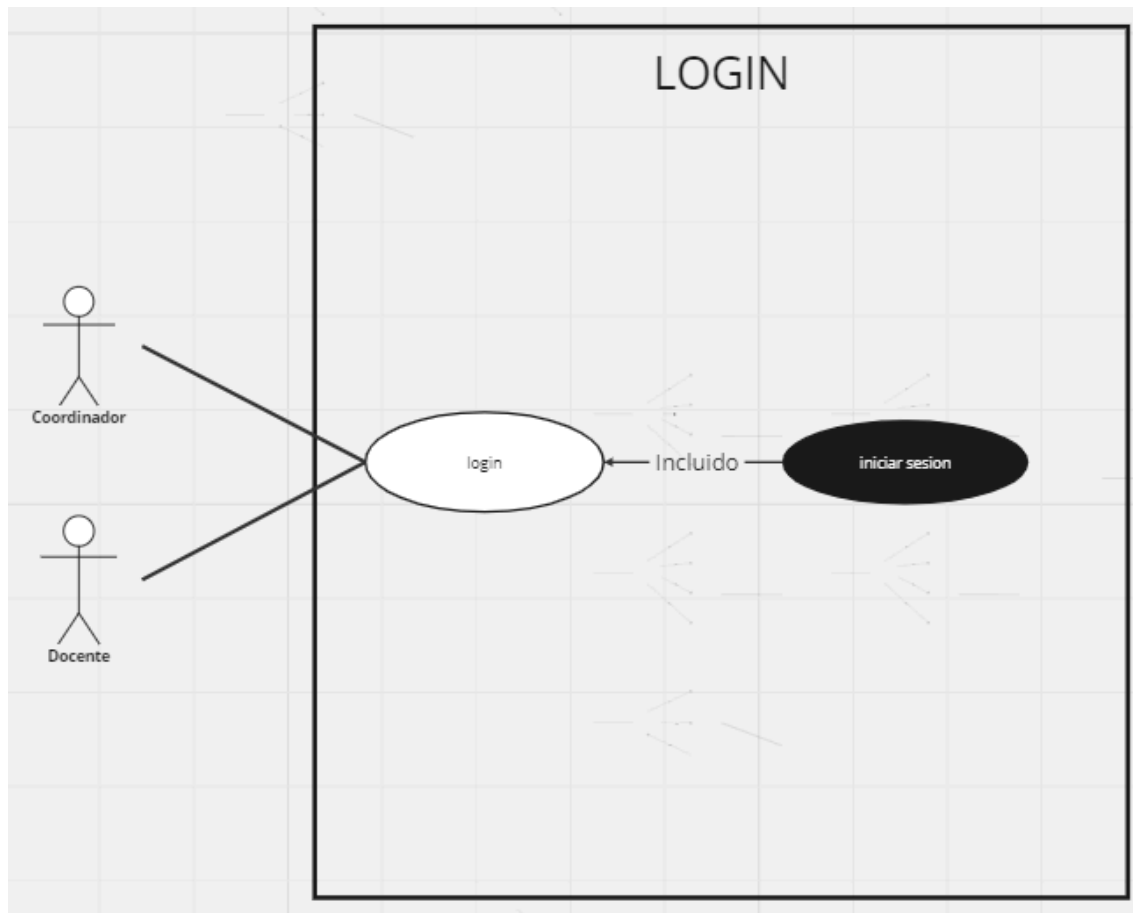


Figura 3. Modelo de Casos de uso del login

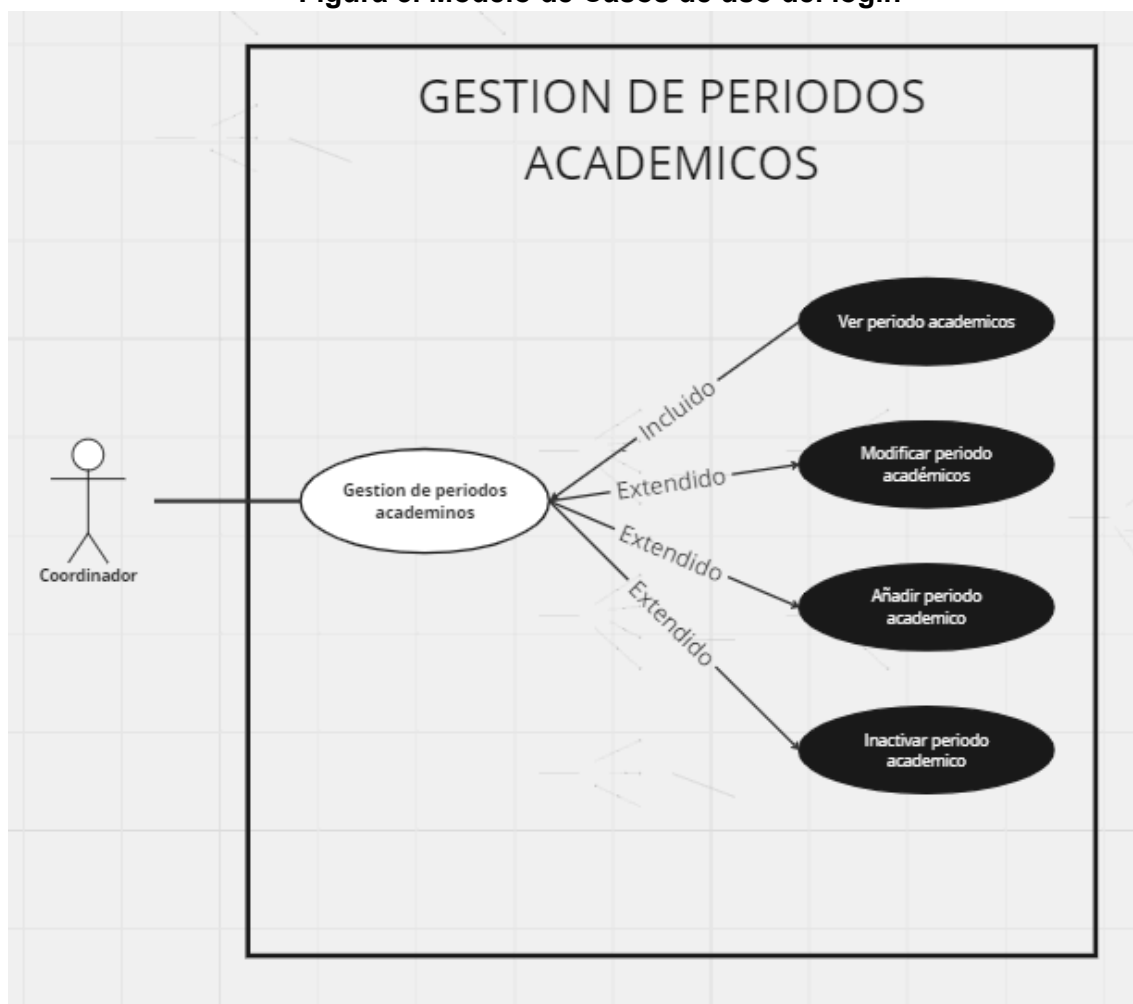


Figura 4. Modelo de casos de uso de gestionar periodos académicos

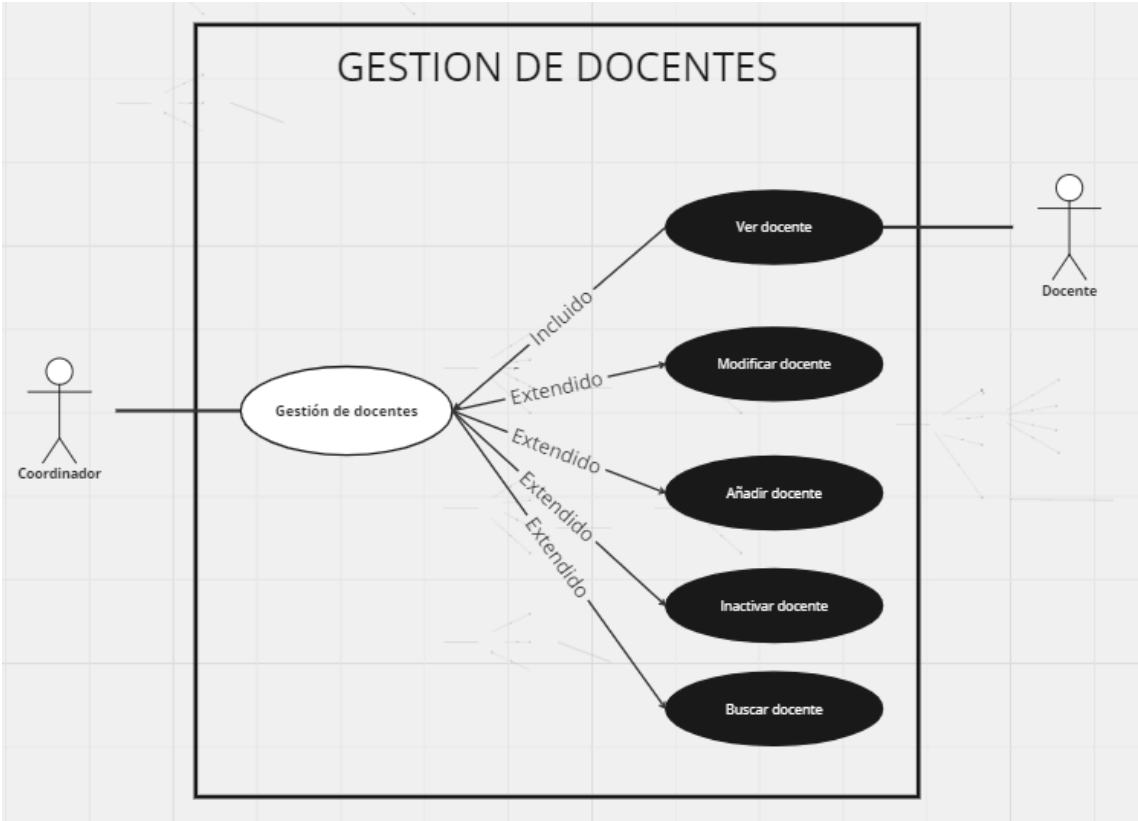


Figura 5. Modelo de casos de uso de gestionar docentes

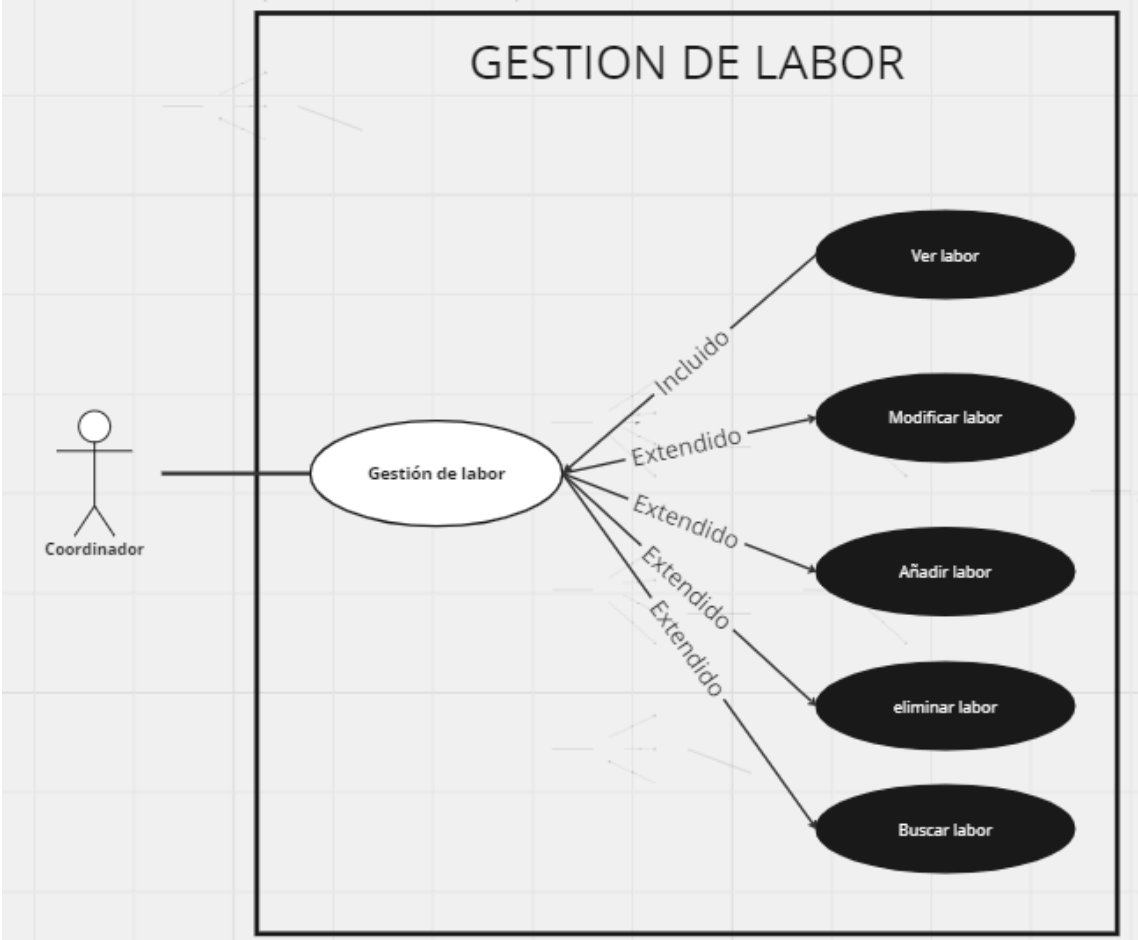


Figura 6. Modelo de casos de uso de gestionar labor

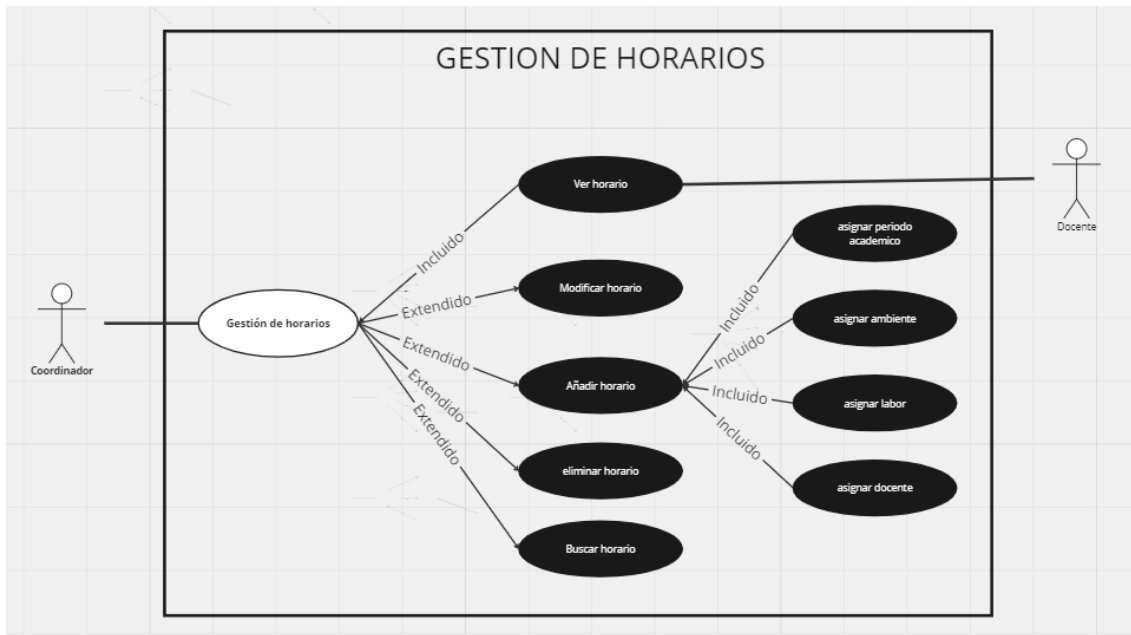


Figura 7. Modelo de casos de uso del gestionar horario

Los actores principales del sistema se describen a continuación:

- **Coordinador:** Actor encargado de gestionar y asignar los horarios académicos en la institución educativa Colegio Mayor del Cauca. Es responsable de configurar los parámetros y restricciones del sistema, así como de ingresar y organizar la información necesaria para la asignación de horarios,, los ambientes, profesores, la labor y el periodo académico.
- **Docente:** usuario que tiene diferentes tipos de labores y utiliza los horarios asignados por el sistema. Su principal interacción con el sistema es la consulta y revisión de sus horarios asignados.

El modelo de casos de uso se describe de manera general partiendo de sus procesos de negocio de la siguiente forma:

Proceso de Inicio de Sesión (Figura 3): Este caso de uso muestra el proceso inicial de inicio de sesión donde un coordinador o un docente se autentica para acceder a las funcionalidades del sistema. La relación "incluido" indica que el inicio de sesión exitoso es requerido antes de acceder a otros casos de uso.

Gestión de Periodos Académicos (Figura 4): Este proceso permite gestionar los períodos académicos o semestres. El coordinador puede ver el período académico actual, modificarlo, extenderlo, agregar un nuevo período o inicializar un período. Esta funcionalidad es crucial para programar cursos, inscripciones y actividades académicas durante el año.

Gestión de Docentes (Figura 5): Este caso de uso permite la gestión de los miembros de la facultad o instructores. El coordinador puede ver una lista de instructores, modificar su información, agregar nuevos instructores o eliminar los existentes. Este proceso es esencial para asignar instructores a cursos y gestionar su carga de trabajo.

Gestión de Labor (Figura 6): Este proceso implica la gestión de aspectos relacionados con el trabajo, como ver detalles laborales, modificar información laboral, agregar nuevos registros laborales, eliminar registros laborales o buscar registros laborales específicos. Esta funcionalidad podría estar relacionada con la gestión de contratos de trabajo, nómina o regulaciones laborales del personal docente o administrativo.

Gestión de Horarios (Figura 7): Este caso de uso cubre la gestión de horarios o calendarios. El coordinador puede ver horarios, modificar los existentes, agregar nuevos horarios o eliminar horarios. Además, este proceso permite asignar períodos académicos, espacios físicos o aulas, cursos, instructores y registros laborales a los horarios. Esta funcionalidad es crucial para organizar y coordinar clases, conferencias y otras actividades académicas.

En general, estos modelos de casos de uso representan los procesos principales necesarios para gestionar períodos académicos, personal docente, aspectos laborales y horarios dentro de una institución educativa. Proporcionan una visión general de alto nivel de la funcionalidad del sistema y las interacciones entre el coordinador (probablemente un rol administrativo) y los diversos procesos involucrados en la operación de la institución.

4.2 Especificación de los Casos de Uso Relevantes

Caso de Uso	Criterio
Crear horario	Este caso de uso se selecciona con el propósito principal de nuestra aplicación: la creación de horarios. Es el núcleo funcional del sistema y requiere una adecuada arquitectura para garantizar su correcto funcionamiento.
Visualizar horario	La visualización del horario generado es fundamental para que los usuarios puedan revisar y validar la asignación de espacios y recursos.
Crear docente	Se selecciona este caso de uso porque es necesario contar con docentes previamente registrados para elaborar el horario de manera adecuada. La gestión de docentes es un requisito previo para la funcionalidad principal del sistema.

Tabla 1. Casos de Uso más relevantes y criterios de selección

A continuación se describen los casos de uso seleccionados:

ID	CU1
Nombre	Crear horario
Actores	Coordinador
Sinopsis	El coordinador ingresa al sistema para registrar un nuevo horario. El sistema almacenará la información del nuevo horario.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> 1. El coordinador está en el sistema. 2. El sistema solicita los detalles del nuevo horario, como el periodo académico, labor, docente y ambiente. 3. El coordinador ingresa los detalles solicitados. 4. El sistema verifica la disponibilidad de los recursos(docente, ambiente, labor) . 5. El sistema registra un nuevo horario. 6. El sistema confirma al coordinador que el horario ha sido creado con éxito.
Extensiones	
	<ol style="list-style-type: none"> 1.a. El coordinador ingresa información inválida o incompleta <ol style="list-style-type: none"> 1.a.1 El sistema notifica al coordinador sobre los campos inválidos o incompletos. 1.a.2 El coordinador completa o corrige la información. 1.a.3 El sistema continúa con el paso 5 del curso típico. 2.a. El sistema no almacena la información.
Prioridad AQ	Alta

ID	CU2
Nombre	Visualizar horario
Actores	Coordinador, Docente
Sinopsis	El coordinador o el docente ingresa al sistema para visualizar los horarios asignados. El sistema muestra la información detallada de los horarios correspondientes.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> 1. El usuario (coordinador o docente) ingresa al sistema. 2. El usuario selecciona la opción para visualizar horarios. 3. El sistema muestra los horarios que previamente han sido ingresados. 4. El usuario puede visualizar los detalles de cada horario, como el docente, el ambiente, el periodo académico, la fecha, etc...
Extensiones	
	<ol style="list-style-type: none"> 1.a. Se presenta un inconveniente con la visualización de la información. 2.a No se encuentra creado ningún horario.
Prioridad AQ	Alta

ID	CU3
Nombre	Crear docente
Actores	Coordinador
Sinopsis	El coordinador ingresa al sistema para registrar un nuevo docente. El sistema almacenará la información del nuevo docente.
Curso Típico de Eventos	
	<ol style="list-style-type: none"> 1. El coordinador está en el sistema. 2. El coordinador selecciona la opción para registrar un nuevo docente. 3. El sistema solicita los detalles del nuevo docente, como el nombre y correo, etc.. 4. El coordinador ingresa los detalles solicitados. 5. El sistema verifica que la información sea válida y completa. 6. El sistema registra al nuevo docente en la base de datos. 7. El sistema confirma al coordinador que el docente ha sido creado exitosamente.
Extensiones	
	<ol style="list-style-type: none"> 1.a. El coordinador ingresa información incompleta o repetida: <ol style="list-style-type: none"> 1.a.1 El sistema notifica al coordinador sobre los campos incompletos o repetidos. 1.a.2 El coordinador completa o corrige la información. 1.a.3 El sistema continúa con el paso 5 del curso típico.
Prioridad AQ	Alta

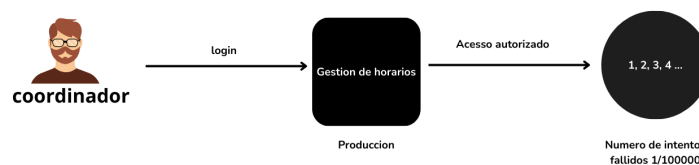
4.3 Especificación de los Escenarios de calidad Relevantes

ID: QS1

- **Nombre:** Seguridad en la gestión de horarios
- **Sinopsis:** Garantiza la integridad y confidencialidad de la información de los horarios

académicos.

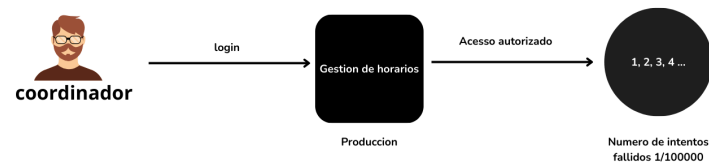
- **Entorno:** GestiHorarios está operativo y los usuarios autorizados acceden al sistema.
- **Cambio en el entorno:** Intento de acceso no autorizado o manipulación de datos por parte de un usuario no autorizado.
- **Comportamiento esperado:** El sistema debe ser capaz de detectar y prevenir intentos de acceso no autorizados o manipulación de datos, registrando cualquier intento y notificando a los administradores del sistema.
- **Medida:** Ningún usuario no autorizado debe ser capaz de acceder al sistema o modificar los datos de los horarios.
- **Prioridad Arquitectónica:** Alta



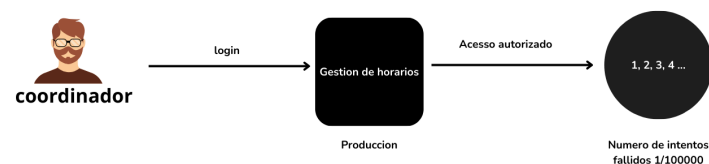
- **Aplicación:** Global

ID: QS2

- **Nombre:** Eficiencia en la generación de horarios
- **Sinopsis:** Asegura que el sistema genere horarios académicos de manera eficiente y oportuna.
- **Entorno:** GestiHorarios está en funcionamiento y los usuarios están generando horarios.
- **Cambio en el entorno:** Aumento de la carga de trabajo o demanda de generación de horarios.
- **Comportamiento esperado:** El sistema debe ser capaz de generar horarios en un tiempo razonable, incluso bajo condiciones de alta carga de trabajo, garantizando que los horarios cumplan con las restricciones y requisitos especificados.
- **Medida:** El tiempo de generación de horarios no debe exceder un límite definido, incluso en momentos de alta demanda.
- **Prioridad Arquitectónica:** Alta
- **Aplicación:** Global



- **Nombre:** Fiabilidad en la asignación de recursos
- **Sinopsis:** Asegurar que los recursos, como aulas y profesores, se asignen de manera confiable y precisa en los horarios académicos.
- **Entorno:** GestHorarios está en funcionamiento y los usuarios están asignando recursos a los horarios.
- **Cambio en el entorno:** Cambio en la disponibilidad de recursos o restricciones adicionales.
- **Comportamiento esperado:** El sistema debe asignar recursos de manera precisa, evitando conflictos y asegurando que los recursos se utilicen de manera eficiente según las



- necesidades y restricciones específicas.
- **Medida:** La asignación de recursos debe ser precisa y sin conflictos, asegurando que todas las clases tengan los recursos necesarios asignados.
 - **Prioridad Arquitectónica:** Alta
 - **Aplicación:** Global

5 Vista Lógica

La vista lógica en un documento de arquitectura de software (SAD) para GestHorarios

se centra en la representación de la estructura y organización interna del sistema desde una perspectiva lógica o conceptual. Esta vista proporciona una visión detallada de los componentes, módulos y sus interacciones, y cómo se organizan para cumplir con los requisitos funcionales del sistema. Para el desarrollo de esta vista se utilizó el método ADD (Attribute-Driven Design). De acuerdo con este método, se identificaron los drivers de la arquitectura, así como las estrategias arquitectónicas asociadas. Esta información se presenta en el ítem de Justificación (Rationale) de la Arquitectura. Según el Rationale, para esta vista se consideraron los patrones de descomposición, generalización y el uso del estilo de módulos. Además, se bosquejó la aplicación del estilo C&C (Componentes y Conectores), teniendo en cuenta el patrón Cliente-Servidor y los tiers, lo cual será de mayor utilidad en la vista de procesos.

Elementos de la Vista Lógica en el SAD

1. Diagrama de Componentes:

- Descripción: Se muestra la estructura de componentes del sistema y sus relaciones. Los componentes representan unidades lógicas de funcionalidad que pueden ser clases, módulos, bibliotecas u otras entidades coherentes.
- Contenido:
 - Componentes Principales: Módulo de gestión de horarios, módulo de gestión de docentes, módulo de gestión de aulas, módulo de autenticación y autorización.
 - Relaciones: Cómo se relacionan entre sí y se agrupan para formar subsistemas o capas lógicas.

2. Diagramas de Clases:

- Descripción: Representan la estructura interna de los componentes, mostrando las clases y sus relaciones. Describen las clases, sus atributos, métodos y asociaciones.
- Contenido:
 - Clases: Horario, Docente, Aula, Usuario, Periodo Académico.
 - Relaciones: Asociaciones entre clases, como relaciones de herencia, agregación y composición.

3. Diagramas de Secuencia:

- Descripción: Representan la interacción entre los componentes o clases durante la ejecución de un escenario o flujo de eventos específico.
- Contenido:
 - Escenarios Clave: Creación de horarios, consulta de horarios, registro de docentes.
 - Interacciones: Orden y mensajes intercambiados entre las entidades participantes.

4. Diagramas de Estado:

- Descripción: Representan los estados y las transiciones de un componente o

entidad en respuesta a eventos o condiciones.

- Contenido:
 - Estados: Estados posibles de un horario (borrador, publicado, modificado).
 - Transiciones: Cómo los componentes o clases cambian de estado en diferentes situaciones.

5. Organización de los Componentes:

- Descripción: Cómo se organiza y estructura el sistema en términos de capas lógicas, subsistemas o módulos.
- Contenido:
 - Capas Lógicas: Presentación, lógica de negocio, acceso a datos.
 - Responsabilidades: Funciones específicas de cada componente, así como las interfaces y contratos entre ellos.

6. Reglas de Negocio y Lógica de Dominio:

- Descripción: La lógica de negocio y las reglas específicas que rigen el comportamiento y las decisiones del sistema.
- Contenido:
 - Reglas: Algoritmos de asignación de horarios, validaciones de disponibilidad de recursos.
 - Lógica: Cálculos y procesos relacionados con la funcionalidad central del sistema.

Attribute-Driven Design (ADD)

El Attribute-Driven Design (ADD) es un enfoque de diseño arquitectónico que se centra en los atributos de calidad del sistema, también conocidos como drivers de la arquitectura. Estos drivers incluyen requisitos de rendimiento, disponibilidad, seguridad, usabilidad, mantenibilidad, y escalabilidad.

1. Identificación de los Drivers de la Arquitectura:

- Descripción: Identificación y documentación de los atributos de calidad clave que el sistema debe cumplir.
- Contenido:
 - Atributos de Calidad: Rendimiento, disponibilidad, seguridad, usabilidad, mantenibilidad, escalabilidad.
 - Origen: Requisitos del proyecto, necesidades de los usuarios, restricciones tecnológicas.

2. Priorización de los Drivers:

- Descripción: Evaluación y priorización de cada driver en función de su importancia y el impacto en el sistema.
- Contenido:

- Importancia: Criticidad de cada driver.
- Impacto: Cómo cada driver afecta las decisiones de diseño.

3. Análisis de los Drivers:

- Descripción: Análisis detallado de cada driver para comprender sus implicaciones en el diseño.
- Contenido:
 - Patrones Arquitectónicos: Identificación de patrones adecuados.
 - Tecnologías: Selección de tecnologías apropiadas.
 - Estrategias: Definición de estrategias de implementación.

4. Diseño de la Arquitectura:

- Descripción: Desarrollo del diseño arquitectónico del sistema utilizando los drivers como guía.
- Contenido:
 - Decisiones de Diseño: Decisiones específicas para abordar cada driver.
 - Implementación: Plan de implementación para satisfacer los atributos de calidad.

5. Evaluación de la Arquitectura:

- Descripción: Evaluación de la arquitectura diseñada en función de cómo cumple con los drivers identificados.
- Contenido:
 - Pruebas: Realización de pruebas para garantizar que la arquitectura satisface los atributos de calidad.
 - Revisiones: Revisiones y análisis para validar la arquitectura.

5.1 Parte Estructural: descomposición modular

La descomposición modular es fundamental para dividir el sistema GestiHorarios en módulos o componentes más manejables, lo que facilita su diseño, implementación y mantenimiento.

Identificación de los Módulos:

- Se identifican los principales módulos o componentes del sistema a través del análisis de los requisitos funcionales, la identificación de las funcionalidades clave y la agrupación de características relacionadas.

Definición de Interfaces:

- Se definen las interfaces de comunicación y los contratos entre los módulos, especificando qué funciones y datos pueden ser utilizados por otros módulos y cómo será la interacción entre ellos.

Cohesión y Acoplamiento:

- Se busca una alta cohesión dentro de cada módulo, lo que significa que cada uno debe

ser responsable de una única funcionalidad o tarea específica. Además, se minimiza el acoplamiento entre los módulos, reduciendo las dependencias y las interacciones complejas.

Niveles de Abstracción:

- Se establecen diferentes niveles de abstracción en la descomposición modular, incluyendo módulos de alto nivel que encapsulan funcionalidades complejas y módulos de bajo nivel que implementan funcionalidades más detalladas.

Jerarquía de Módulos:

- Se establece una jerarquía de módulos donde algunos dependen de otros para su funcionamiento, permitiendo una organización estructurada y clara de los componentes del sistema.

Reutilización de Componentes:

- Se identifican oportunidades de reutilización de componentes existentes o de terceros para evitar la duplicación de esfuerzos y mejorar la eficiencia del desarrollo.

Diagramas de Estructura:

- Se utilizan diagramas, como diagramas de componentes o diagramas de paquetes, para visualizar la estructura modular del sistema, lo que ayuda a comprender la organización y las relaciones entre los módulos.

5.1.1 Primer Nivel de Refinamiento

Para comprender inicialmente el sistema GestiHorarios, se presenta un primer nivel de refinamiento, el cual se basa en la descomposición funcional y la distribución física expuesta en la vista de Deployment. Este modelo se visualiza en la figura 8, donde se bosqueja la aparición de un estilo basado en descomposición, uso y generalización. Este nivel inicial de refinamiento proporciona una visión general de los componentes principales del sistema y su interacción funcional y física.

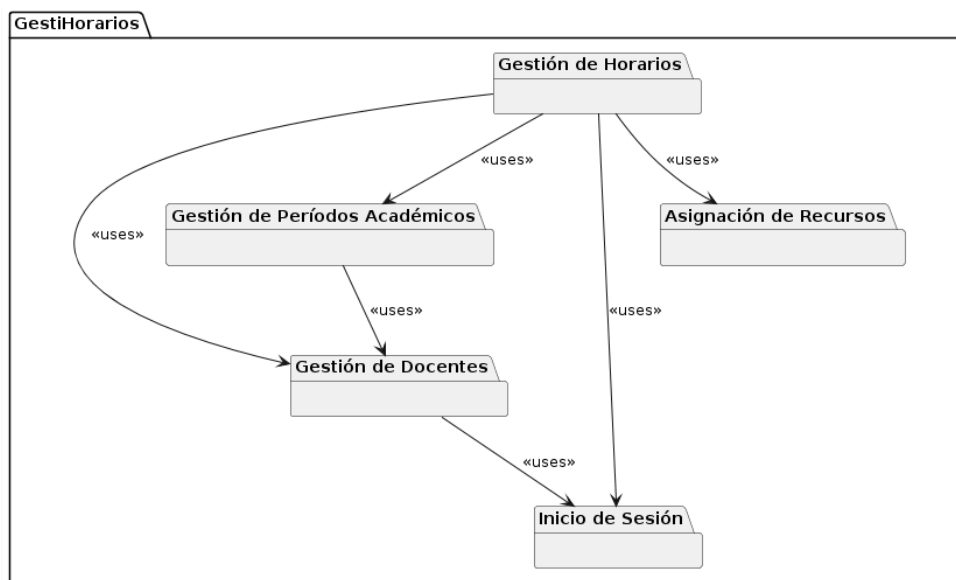


Figura 8. Primer Nivel de Refinamiento

Una descripción de los subsistemas presentados en la Figura 8 se presenta en la tabla junto con la información heredada de la vista de casos de uso.

Módulo	Descripción	Atributos/Escenarios	Casos de uso
Gestión de Períodos Académicos	Administra la información correspondiente a los períodos académicos, como semestres, trimestres, etc.	QS1, QS2, QS3	CU4, CU5, CU6
Gestión de Horarios	Gestiona la información de los horarios académicos, asignando clases, profesores y aulas.	QS2, QS3	CU1, CU3
Gestión de Docentes	Gestiona la información de los docentes, incluyendo asignaciones de horarios y disponibilidad.	QS3	CU2
Asignación de Recursos	Administra la asignación de recursos como aulas, equipos y otros materiales necesarios.	QS1, QS3	CU7, CU8
Inicio de Sesión	Es el módulo que da acceso a los usuarios al sistema, gestionando la autenticación y autorización.	QS1, QS3	Todos los CU que requieran autenticación y autorización

Tabla 1. Localización de requisitos en los subsistemas respectivos

5.1.2 Segundo Nivel de Refinamiento

El segundo nivel de refinamiento implica dividir los componentes de alto nivel identificados en el primer nivel en módulos más detallados y específicos. Estos módulos representan unidades funcionales más pequeñas y con mayor nivel de detalle dentro de cada componente de alto nivel.

EJEMPLO

Para el segundo nivel de refinamiento cada uno de los subsistemas del modelo anterior es descompuesto recursivamente.

5.1.2.1 Módulo "Asignación de Recursos":

Este subsistema ha sido descompuesto, de acuerdo a la Figura 9 siguiendo un estilo de capas sus capas son:

- **"Capa Gestión de Aulas"**: se encarga del registro y mantenimiento de las aulas disponibles y su asignación a los horarios.
- **"Capa Gestión de Horarios"**: se encarga de la creación y programación de los horarios, así como de la validación de los mismos para evitar conflictos.
- **"Capa Asignación de Docentes"**: se encarga de asignar docentes a los horarios y distribuir su carga académica.
- **"Capa Asignación de Materias"**: se encarga de asignar las materias a los horarios y distribuirlas por programa académico.

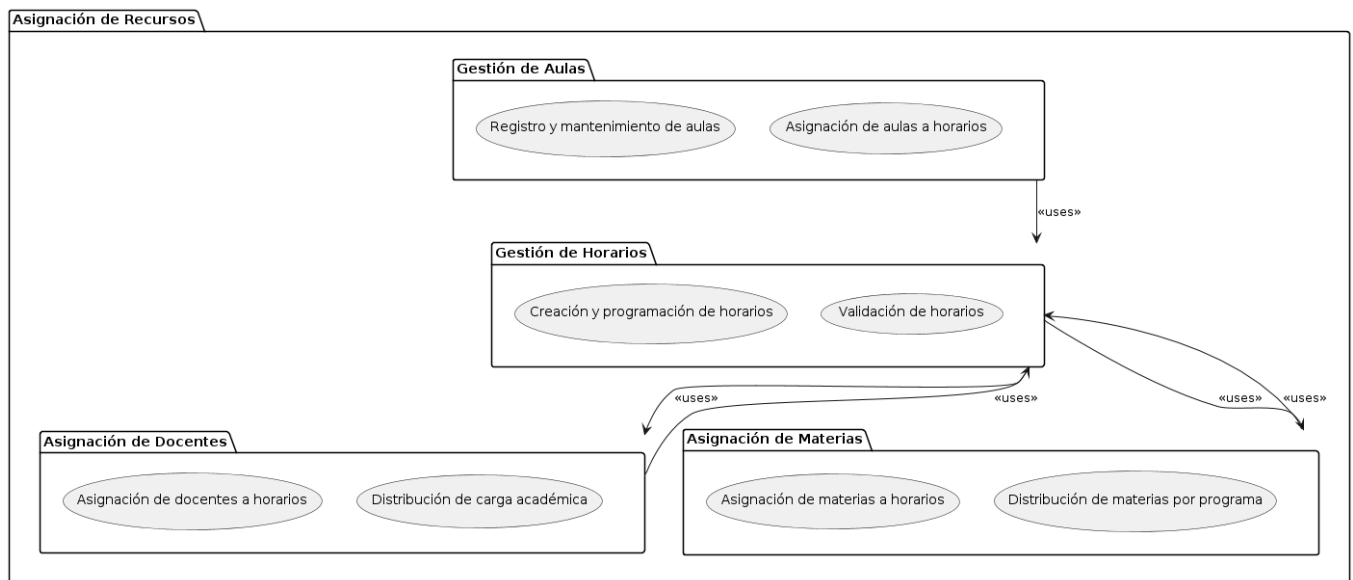


Figura 9. Descripción de subsistema Asignación de Recursos

5.1.2.2 Módulo "Gestión de Horarios":

Este subsistema ha sido descompuesto, de acuerdo a la Figura 10 siguiendo un estilo de capas sus capas son:

- **"Capa Programación de Horarios"**: se encarga de la creación y edición de los horarios, así como de la asignación de recursos (aulas, docentes, materias) a los horarios.
- **"Capa Visualización de Horarios"**: proporciona vistas de los horarios para docentes y estudiantes, con funciones de filtrado y búsqueda.
- **"Capa Reportes de Horarios"**: se encarga de generar reportes de los horarios y de exportarlos en diferentes formatos.
- **"Capa Configuración de Horarios"**: permite configurar los parámetros y definir las

restricciones y reglas aplicables a los horarios.

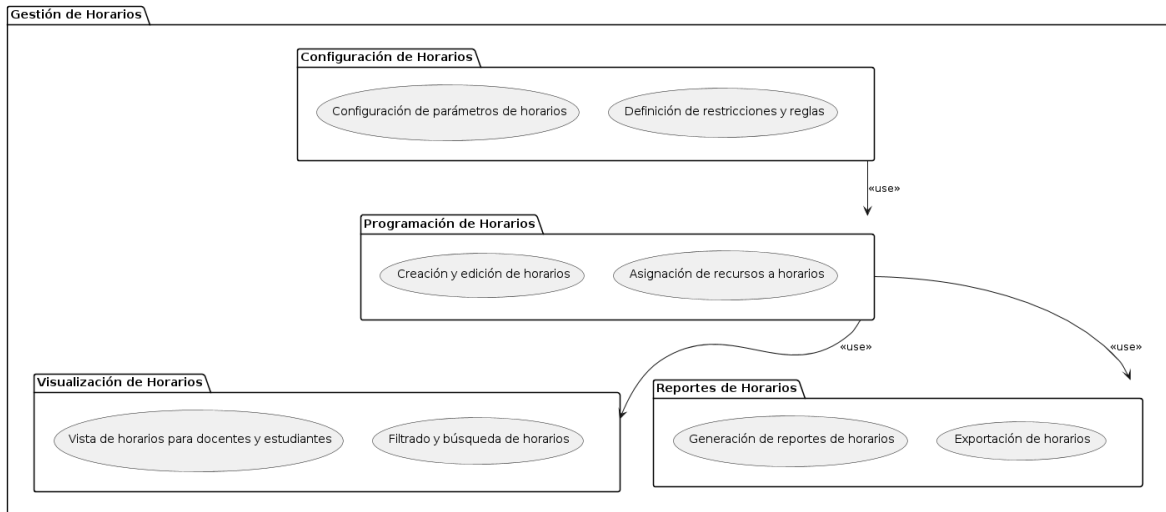


Figura 10. Descripción de subsistema Gestión de Horarios

5.1.2.3 Módulo "Gestión de Períodos Académicos":

Este subsistema ha sido descompuesto, de acuerdo a la Figura 11 siguiendo un estilo de capas sus capas son:

- **"Capa Definición de Períodos":** se encarga de la creación y edición de los períodos académicos, estableciendo sus fechas de inicio y finalización, así como su duración.
- **"Capa Administración de Calendarios":** permite la creación y mantenimiento de calendarios académicos, así como la asignación de eventos a los mismos.
- **"Capa Gestión de Eventos":** se encarga del registro y programación de eventos académicos, además de enviar notificaciones relacionadas con los mismos.
- **"Capa Reportes de Períodos":** genera reportes de los períodos académicos y permite la exportación de los calendarios.

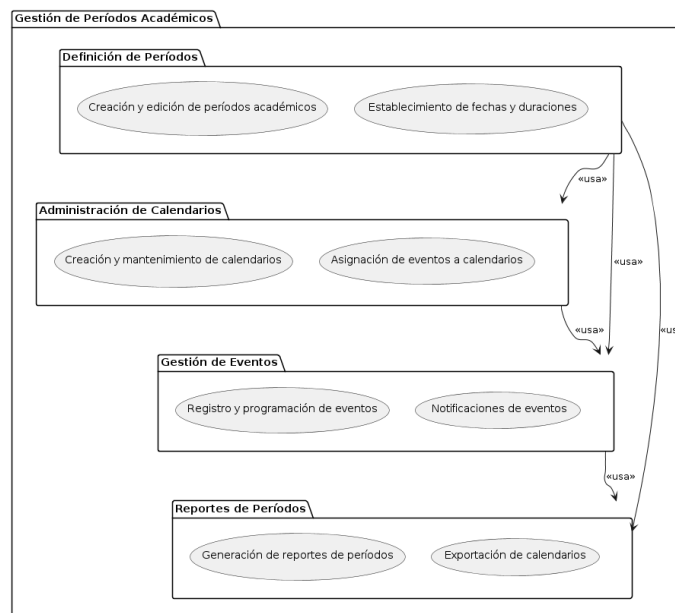


Figura 11. Descripción de subsistema Gestión de Períodos Académicos

5.1.2.4 Módulo "Gestión de Docentes":

Este subsistema ha sido descompuesto, de acuerdo a la Figura 12 siguiendo un estilo de capas sus capas son:

- **"Capa Registro de Docentes"**: se encarga del registro y mantenimiento de los datos de los docentes, así como de la asignación de roles y permisos.
- **"Capa Asignación de Carga Académica"**: asigna las materias a los docentes y calcula su carga académica correspondiente.
- **"Capa Evaluación de Desempeño"**: permite registrar las evaluaciones de desempeño de los docentes y generar informes relacionados.
- **"Capa Reportes de Docentes"**: genera reportes con información sobre los docentes y permite exportar sus datos.

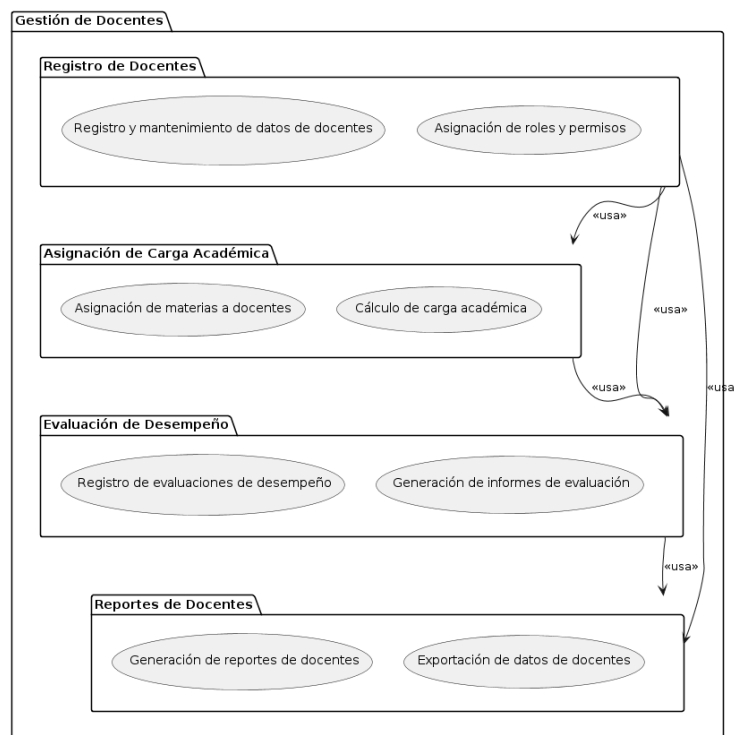


Figura 12. Descripción de subsistema Gestión de Docentes

5.1.2.5 Módulo "Inicio de Sesión":

Este subsistema ha sido descompuesto, de acuerdo a la Figura 13 siguiendo un estilo de capas sus capas son:

- **"Capa Autenticación de Usuarios"**: Verifica la identidad de los usuarios mediante credenciales y maneja el inicio y cierre de sesión.
- **"Capa Gestión de Roles y Permisos"**: Administra los roles y permisos, asignando roles a los usuarios y definiendo sus capacidades dentro del sistema.

- **“Capa Recuperación de Contraseña”**: Proporciona herramientas para que los usuarios recuperen o restablezcan sus contraseñas mediante métodos de verificación y correos electrónicos.

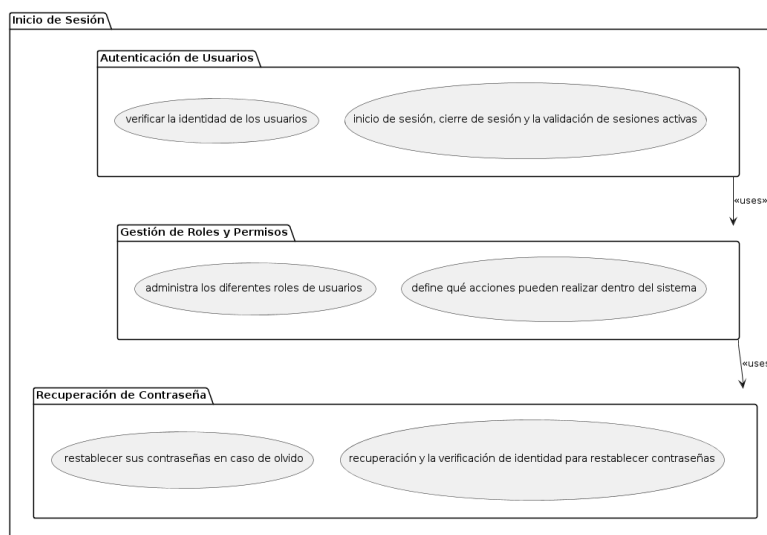


Figura 13. Descripción de subsistema Inicio de Sesión

5.2 Parte Dinámica (Comportamiento)

Esta sección se enfoca en describir cómo interactúan los componentes y módulos del sistema "GestiHorarios" en tiempo de ejecución. El objetivo es comprender y representar el flujo de información, los eventos, los procesos y las interacciones entre los diferentes elementos del sistema.

Escenarios de Uso

Se describen diferentes escenarios de uso que representan las interacciones del sistema con los usuarios, otros sistemas o componentes externos. Estos escenarios ayudan a entender cómo se comportará el sistema en situaciones reales, proporcionando un contexto para diseñar y evaluar la funcionalidad del sistema.

Estados y Transiciones

Se describen los diferentes estados en los que pueden encontrarse los objetos o componentes del sistema, así como las transiciones entre estos estados. Esto es crucial para sistemas con comportamientos complejos o estados de persistencia.

Gestión de Eventos

Se especifica cómo se manejan los eventos y las notificaciones dentro del sistema. Esto incluye cómo se generan, capturan y procesan los eventos, así como cómo se notifican a los componentes relevantes para llevar a cabo acciones específicas.

Hilos de Ejecución

Si el sistema utiliza múltiples hilos de ejecución o procesos concurrentes, se describe cómo se gestionan y sincronizan estos hilos para garantizar un comportamiento correcto y predecible.

Reglas de Negocio

Se describen las reglas de negocio y los procesos empresariales implementados en el sistema. Esto permite comprender cómo se toman decisiones y se aplican las reglas dentro del flujo de trabajo del sistema.

Ejemplo: Modelo de Ingeniería GestiHorarios

La parte dinámica del modelo de ingeniería "GestiHorarios" se modela a través del view type C&C (Componentes y Conectores), siguiendo la notación brindada en la guía y complementada con trabajos académicos.

Primera Descomposición:

Los componentes principales del negocio se estructuran a través de un esquema Cliente/Servidor. La interacción entre las unidades de negocio se realiza a través de un modelo de solicitud/respuesta.

Segunda Descomposición:

Cada subsistema identificado incluye diferentes niveles de funcionalidad, desarrollados, empaquetados e instalados en módulos, componentes y nodos. Estos niveles, que en la vista estructural se organizan en capas, coinciden con una dinámica de n-tiers en la vista dinámica.

5.2.1 View Packet Negocio *Asignación de Recursos*

Esta vista proporciona una representación visual de la estructura de los paquetes o módulos que componen el sistema, mostrando la jerarquía y las relaciones entre los paquetes. Esto ayuda a entender la organización y agrupación lógica de los componentes en unidades coherentes.

Paquetes y Subpaquetes: Los paquetes representan unidades lógicas que agrupan componentes relacionados, y pueden contener subpaquetes para una mayor división y organización.

Dependencias entre Paquetes: Se muestran las relaciones de dependencia entre los componentes de diferentes paquetes, indicando cómo se comunican o interactúan entre sí.

Interfaces y Contratos: Se representan las interfaces y contratos que los paquetes proporcionan y requieren, facilitando la comprensión de cómo se comunican y colaboran entre sí.

Organización Jerárquica: La estructura jerárquica de los paquetes se muestra claramente, permitiendo entender los diferentes niveles de abstracción.

Niveles de Abstracción: Los paquetes pueden representar diferentes niveles de abstracción en el sistema, desde paquetes de alto nivel que encapsulan funcionalidades principales hasta paquetes de nivel inferior con componentes más específicos.

Modularidad y Reutilización: Se identifican oportunidades de modularidad y reutilización de componentes, promoviendo la coherencia interna y la reutilización de funcionalidades en diferentes partes del sistema.

5.2.1.1 Vista de Componentes y Conectores

Esta vista lógica representa la estructura y la interacción de los componentes del sistema, así como los conectores que facilitan la comunicación entre ellos.

Componentes:

Unidades lógicas o funcionales que encapsulan una parte del sistema y realizan tareas específicas (módulos, clases, bibliotecas, servicios).

Conectores:

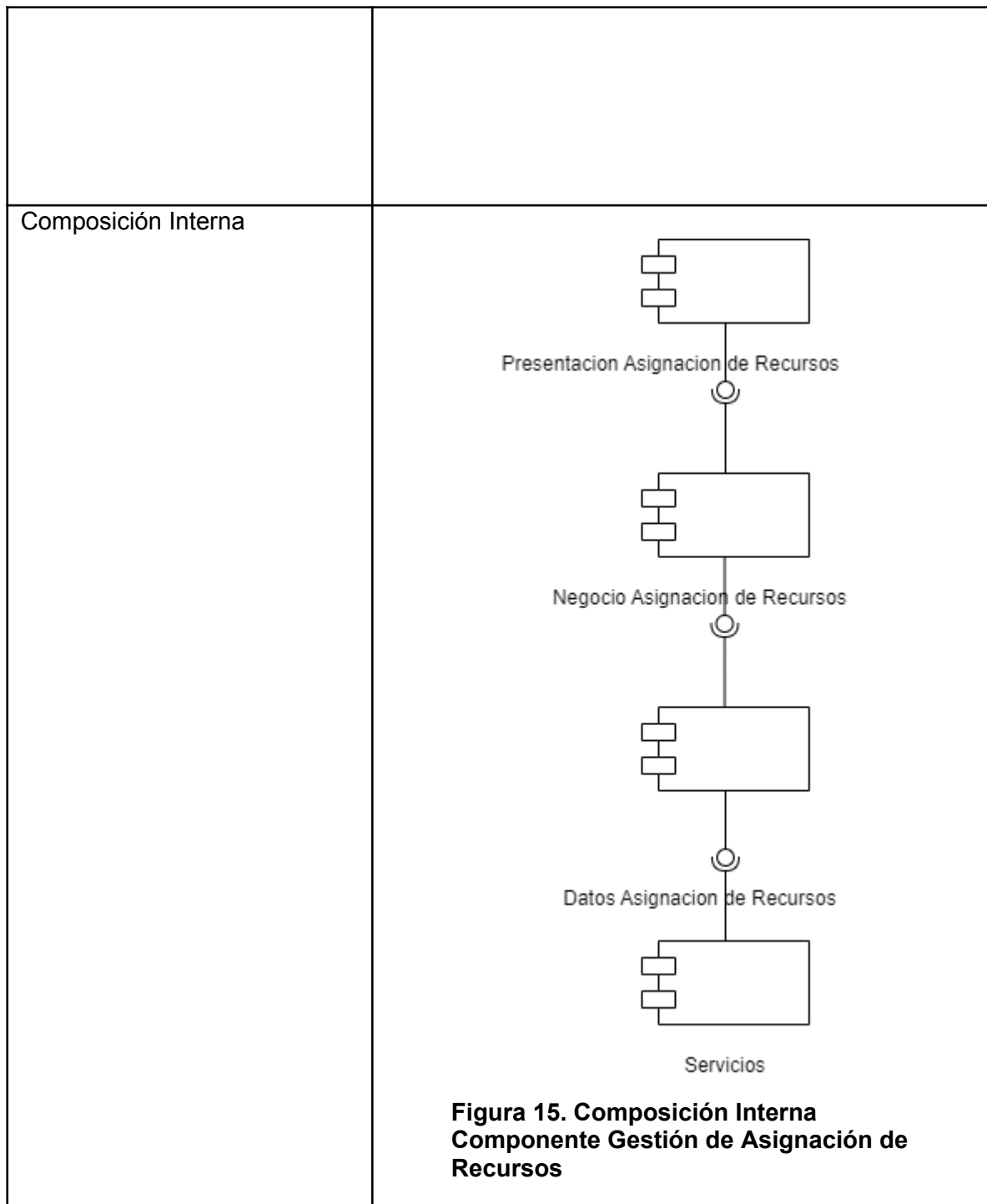
Mecanismos de comunicación o interacción entre los componentes, como protocolos de red,

llamadas a API, colas de mensajes y eventos. Permiten que los componentes se comuniquen, intercambien datos y cooperen entre sí para lograr la funcionalidad del sistema.

Figura 14. C&C View para el View Packet Negocio Asignación de Recursos

5.2.1.2 Catálogo de componentes

Nombre del View Packet	Negocio Asignación de Recursos
Rationale	El patrón arquitectónico Cliente-Servidor se considera adecuado para el sistema GestiHorarios debido a su capacidad de agregar nuevos clientes sin un impacto significativo en la arquitectura. Esta característica es especialmente beneficiosa para la asignación de recursos, ya que permite una escalabilidad eficiente del sistema. Dada la naturaleza del problema, donde múltiples usuarios pueden necesitar acceder simultáneamente para gestionar horarios, aulas, docentes y materias, es crucial contar con una arquitectura que soporte esta demanda. La escalabilidad proporcionada por este patrón garantiza que el sistema pueda manejar un incremento en la cantidad de usuarios y operaciones sin comprometer el rendimiento, asegurando una asignación de recursos efectiva y sin conflictos.
Componente	Gestión de Asignación de Recursos
Descripción:	Este componente se encarga de gestionar los recursos necesarios para la asignación y planificación de horarios, asegurando la disponibilidad de aulas, la correcta distribución de docentes y materias, y la programación eficiente de las actividades académicas.



5.2.1 View Packet Negocio *Gestión de Períodos Académicos*

5.2.1.1 Vista de Componentes y Conectores

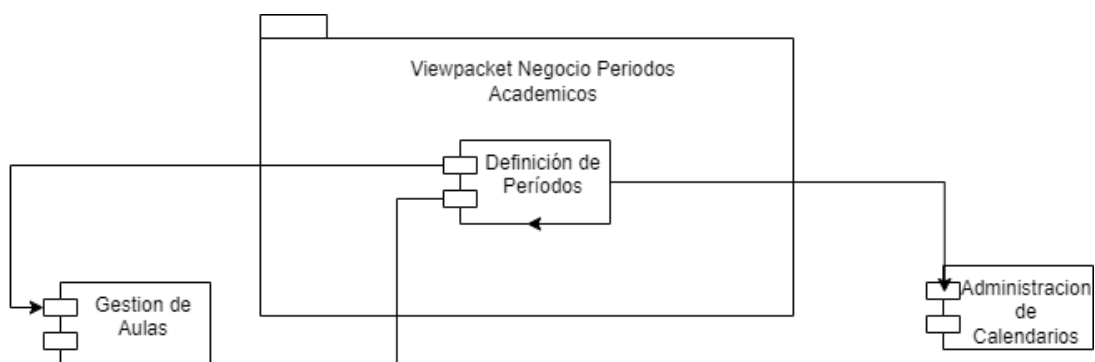
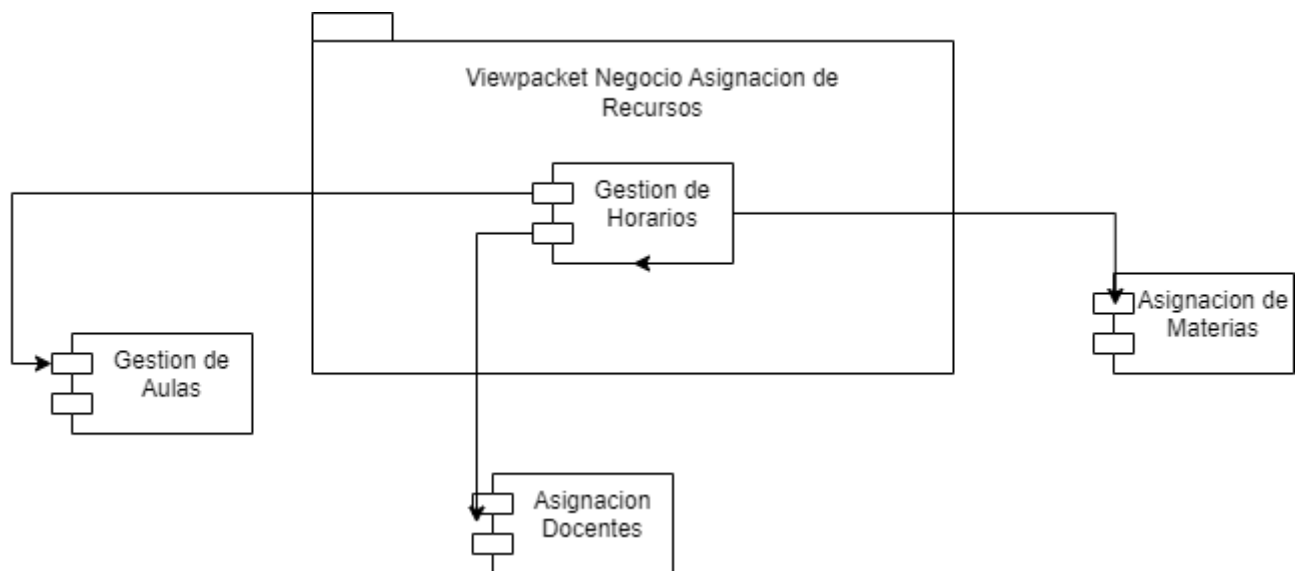
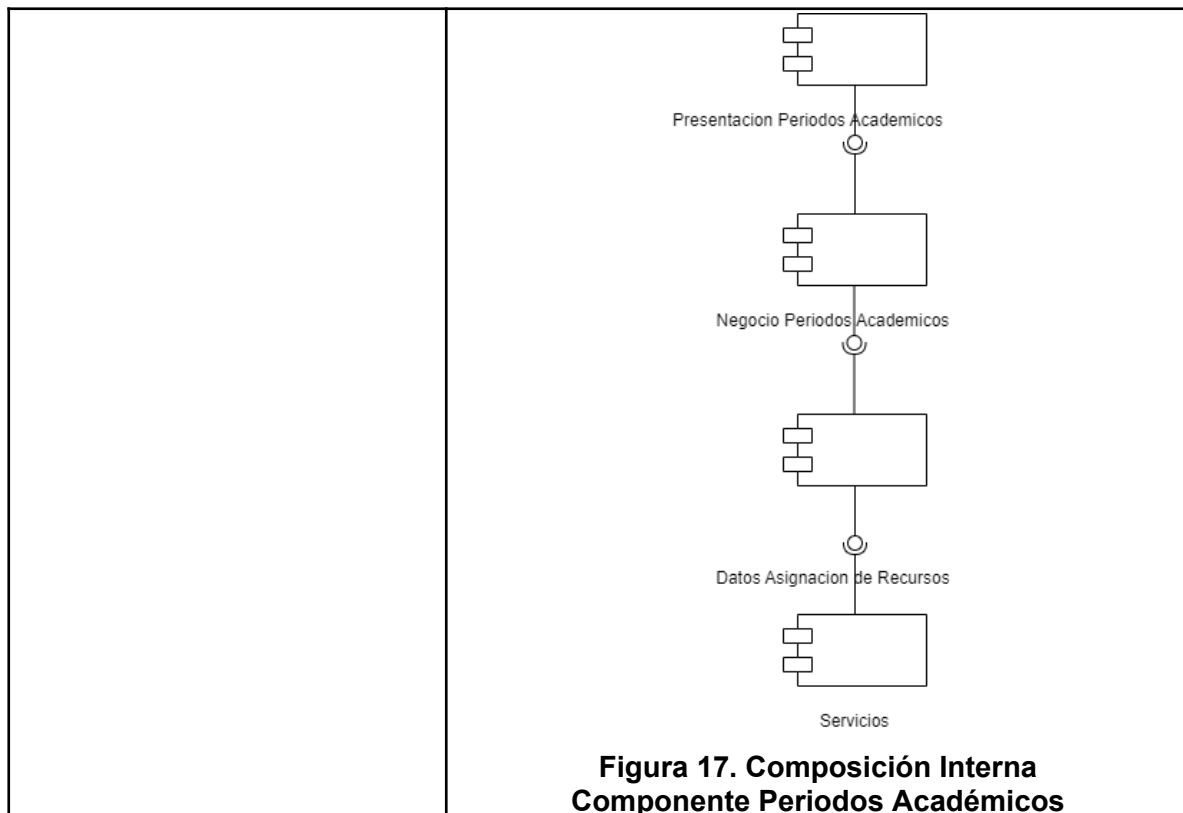


Figura 16. C&C View para el View Packet Negocio *Períodos Académicos*

5.2.1.2 Catálogo de componentes

Nombre del View Packet	Negocio <i>Gestión de Períodos Académicos</i>
Rationale	El patrón arquitectónico Cliente-Servidor se considera adecuado para el sistema GestiHorarios debido a su capacidad de agregar nuevos clientes sin un impacto significativo en la arquitectura. Esta característica es especialmente beneficiosa para la gestión de períodos académicos, ya que permite una escalabilidad eficiente del sistema. Dada la naturaleza del problema, donde múltiples usuarios pueden necesitar acceder simultáneamente para gestionar la configuración y administración de períodos académicos, es crucial contar con una arquitectura que soporte esta demanda. La escalabilidad proporcionada por este patrón garantiza que el sistema pueda manejar un incremento en la cantidad de usuarios y operaciones sin comprometer el rendimiento, asegurando una gestión efectiva y sin conflictos de los períodos académicos.
Componente	Gestión de Periodos Académicos
Descripción:	Este componente Gestión de Períodos Académicos coordina las actividades relacionadas con la creación, configuración y administración de los períodos académicos dentro del sistema. Este módulo es esencial para definir las fechas clave, administrar los ciclos académicos y asegurar que los períodos se gestionen de manera eficiente y sin conflictos.
Composición Interna	



5.2.2 View Packet Negocio *Gestión de Horarios*

5.2.2.1 Vista de Componentes y Conectores

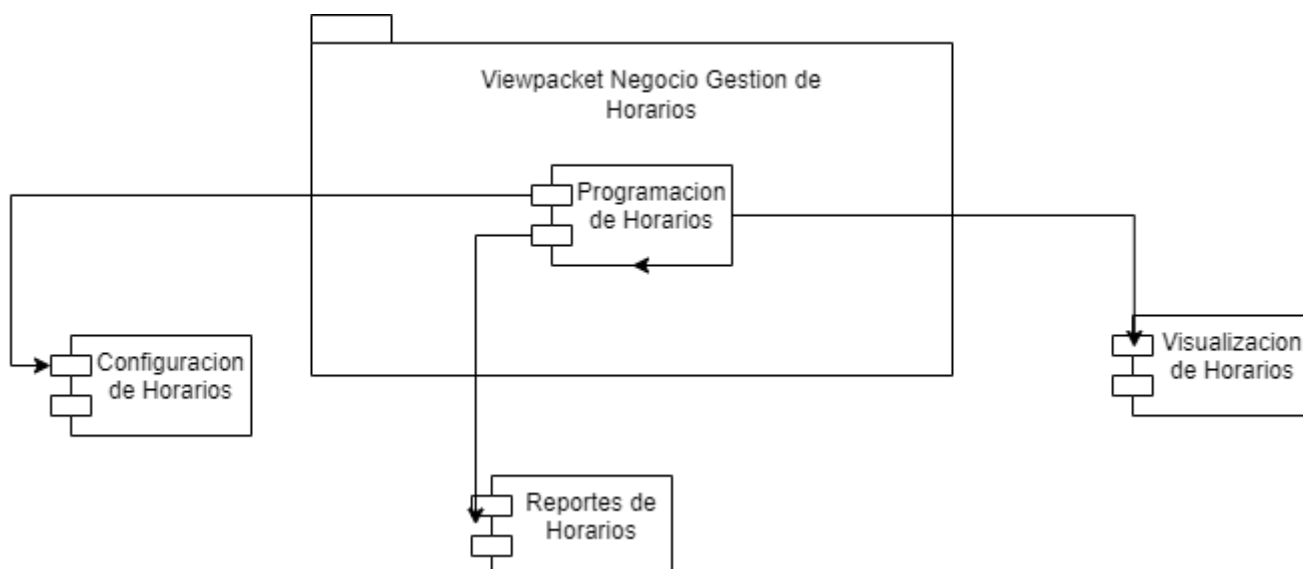
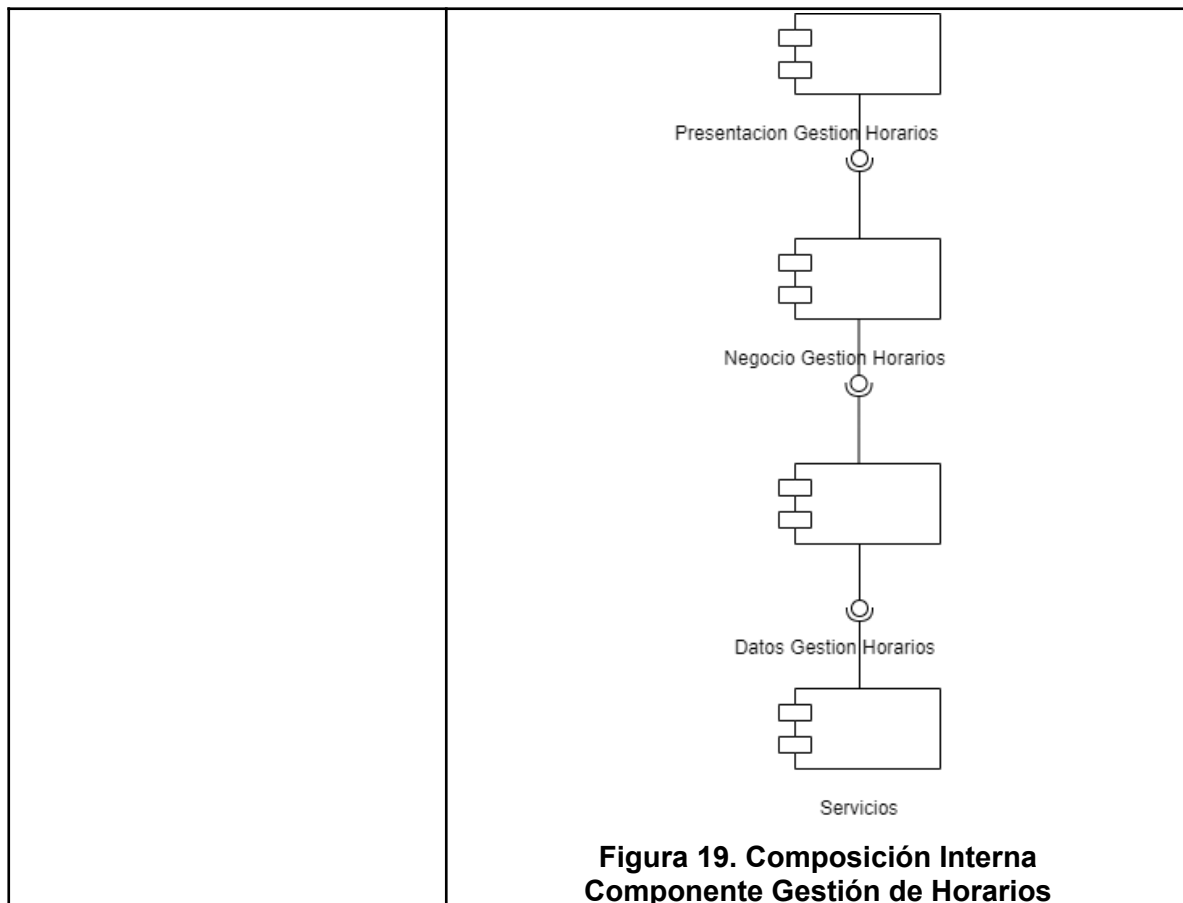


Figura 18. C&C View para el View Packet Negocio *Gestión de Horarios*

5.2.2.2 Catálogo de componentes

Nombre del View Packet	Negocio <i>Gestión de Horarios</i>
Rationale	El patrón arquitectónico Cliente-Servidor se considera adecuado para el sistema GestiHorarios debido a su capacidad de agregar nuevos clientes sin un impacto significativo en la arquitectura. Esta característica es especialmente beneficiosa para la gestión de horarios, ya que permite una escalabilidad eficiente del sistema. Dada la naturaleza del problema, donde múltiples usuarios pueden necesitar acceder simultáneamente para crear, modificar y consultar horarios, es crucial contar con una arquitectura que soporte esta demanda. La escalabilidad proporcionada por este patrón garantiza que el sistema pueda manejar un incremento en la cantidad de usuarios y operaciones sin comprometer el rendimiento, asegurando una gestión de horarios efectiva y sin conflictos.
Componente	Gestión de Horarios
Descripción:	El Componente Gestión de Horarios coordina las actividades relacionadas con la creación, modificación, y validación de los horarios dentro del sistema. Este módulo es esencial para asegurar que los horarios se gestionen de manera eficiente, evitando conflictos y asegurando la disponibilidad de los recursos necesarios como aulas y docentes.
Composición Interna	



5.2.3 View Packet Negocio *Gestión de Docentes*

5.2.3.1 *Vista de Componentes y Conectores*

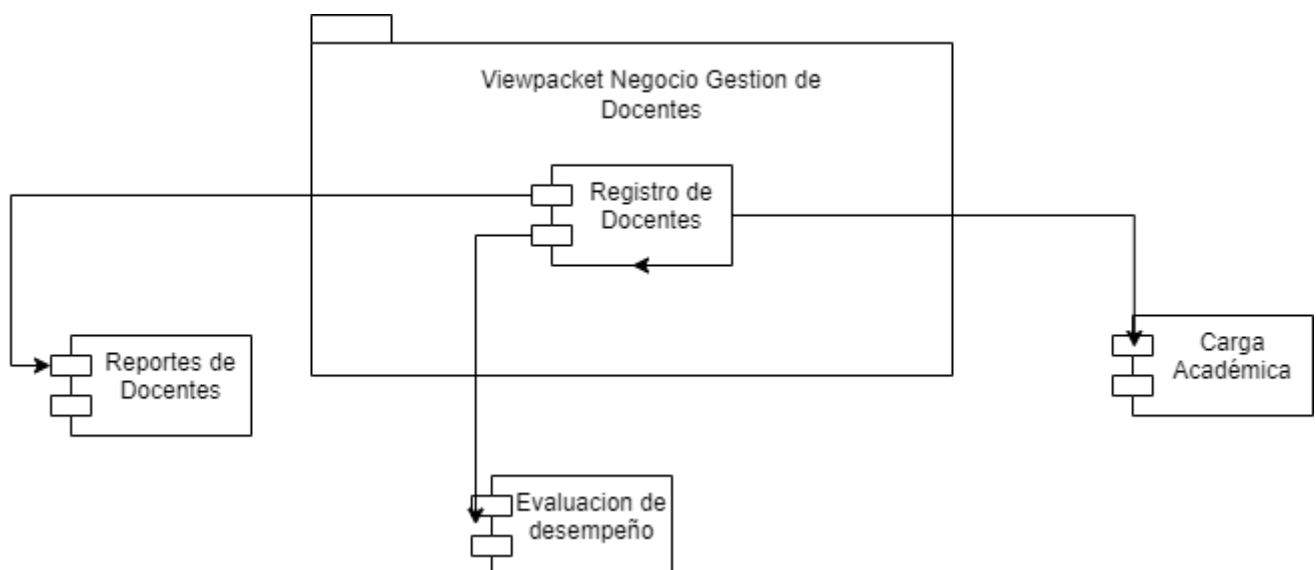
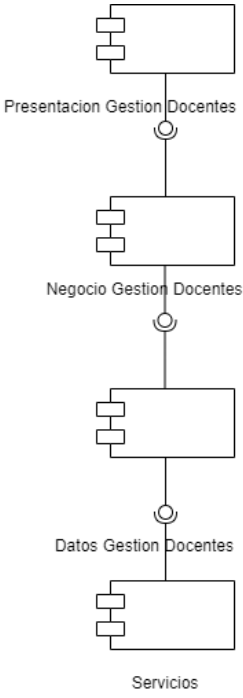


Figura 20. C&C View para el View Packet Negocio *Gestión de Docentes*

5.2.3.2 Catálogo de componentes

Nombre del View Packet	Negocio <i>Gestión de Docentes</i>
Rationale	<p>El patrón arquitectónico Cliente-Servidor se considera adecuado para el sistema GestiHorarios debido a su capacidad de agregar nuevos clientes sin un impacto significativo en la arquitectura. Esta característica es especialmente beneficiosa para la gestión de docentes, ya que permite una escalabilidad eficiente del sistema. Dada la naturaleza del problema, donde múltiples usuarios pueden necesitar acceder simultáneamente para gestionar la información de los docentes, asignar sus horarios y distribuir su carga académica, es crucial contar con una arquitectura que soporte esta demanda. La escalabilidad proporcionada por este patrón garantiza que el sistema pueda manejar un incremento en la cantidad de usuarios y operaciones sin comprometer el rendimiento, asegurando una gestión efectiva y sin conflictos de los docentes.</p>
Componente	Gestión de Docentes
Descripción:	<p>El componente Gestión de Docentes coordina las actividades relacionadas con la administración de la información de los docentes, la asignación de horarios y la distribución de la carga académica. Este módulo es esencial para asegurar que los docentes estén adecuadamente asignados a las clases y que su carga de trabajo se gestione de manera eficiente.</p>
Composición Interna	 <p style="text-align: center;">Figura 21. Composición Interna Componente Gestión de Docentes</p>

5.2.4 View Packet Negocio Inicio de sesión

5.2.4.1 Vista de Componentes y Conectores

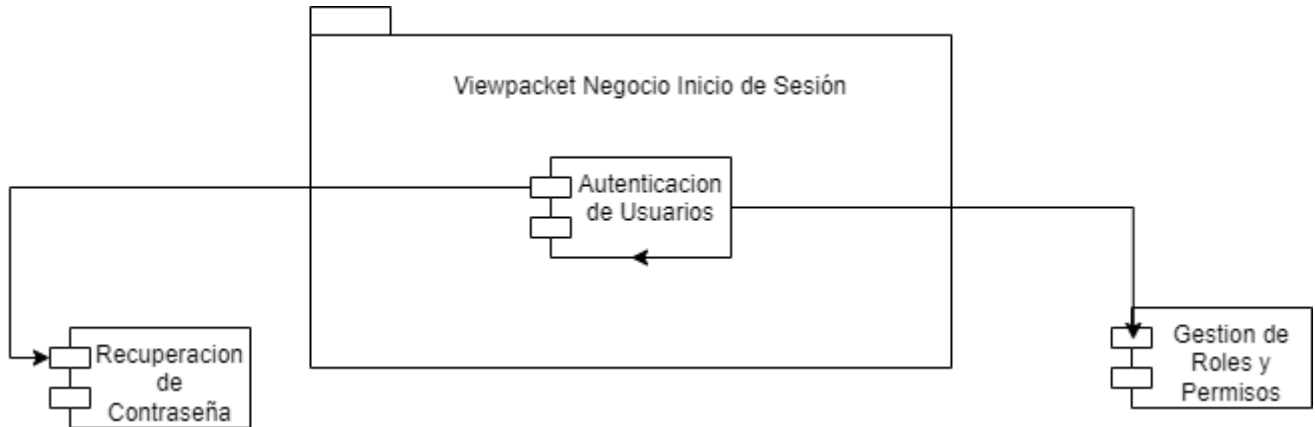
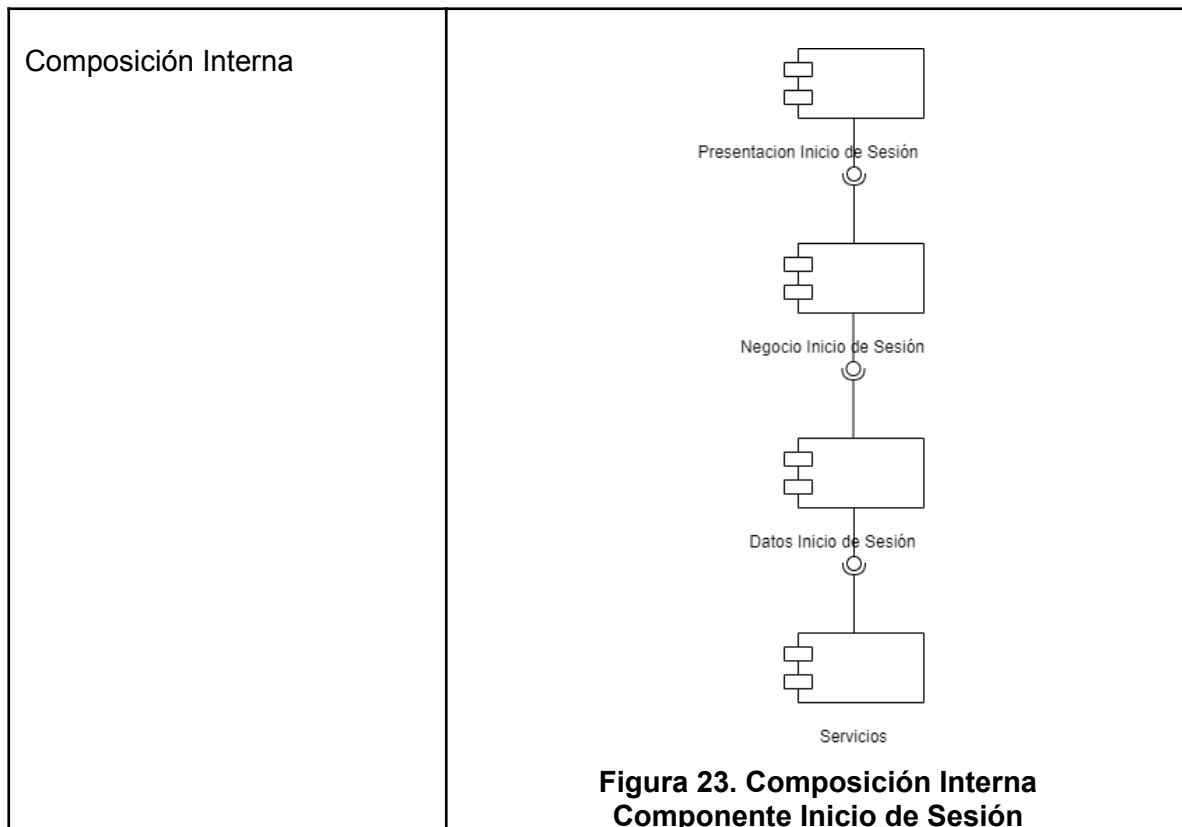


Figura 22. C&C View para el View Packet Negocio Inicio de sesión

5.2.4.2 Catálogo de componentes

Nombre del View Packet	Negocio Inicio de Sesión
Rationale	El patrón arquitectónico Cliente-Servidor se considera adecuado para el sistema GestiHorarios debido a su capacidad de agregar nuevos clientes sin un impacto significativo en la arquitectura. Esta característica es especialmente beneficiosa para el módulo de inicio de sesión, ya que permite una escalabilidad eficiente del sistema. Dada la naturaleza del problema, donde múltiples usuarios pueden necesitar acceder simultáneamente para iniciar sesión y autenticarse, es crucial contar con una arquitectura que soporte esta demanda. La escalabilidad proporcionada por este patrón garantiza que el sistema pueda manejar un incremento en la cantidad de usuarios y solicitudes de autenticación sin comprometer el rendimiento, asegurando una gestión segura y eficiente de las sesiones de usuario.
Componente	Inicio de Sesión
Descripción:	El Componente Inicio de Sesión coordina las actividades relacionadas con la autenticación de usuarios dentro del sistema. Este módulo es esencial para asegurar que solo los usuarios autorizados puedan acceder al sistema, proporcionando seguridad y control de acceso.



6. Vista de despliegue

La vista de despliegue de GestiHorarios representa cómo los componentes del sistema se despliegan y ejecutan en el entorno de implementación. Proporciona una visión de la distribución física de los componentes del sistema en los nodos de hardware, servidores y dispositivos. Algunos aspectos importantes de esta vista incluyen:

Nodos de Hardware:

Se representan los nodos físicos, como servidores, estaciones de trabajo o dispositivos embebidos, en los que se ejecutan los componentes del sistema. Estos nodos pueden estar conectados en una red.

Componentes Desplegados:

Se muestran los componentes del sistema que se despliegan en cada nodo de hardware. Cada componente puede ser un módulo, una aplicación, un servicio o cualquier otra entidad lógica que se ejecute en un nodo específico.

Conectores de Despliegue:

Se representan los conectores que permiten la comunicación y la interacción entre los componentes desplegados en diferentes nodos. Estos conectores pueden ser protocolos de red, servicios web, colas de mensajes, entre otros.

Relaciones de Despliegue:

Se muestran las relaciones entre los componentes y los nodos de hardware. Estas relaciones indican en qué nodos se despliegan los componentes y cómo se conectan entre sí para formar la arquitectura de despliegue del sistema.

Recursos de Hardware:

Se pueden especificar los recursos de hardware utilizados por los nodos, como capacidad de

procesamiento, memoria, almacenamiento, ancho de banda de red, entre otros.

Topología de Red:

Si el sistema se distribuye en una red, se puede mostrar la topología de red que conecta los nodos. Esto incluye la estructura de la red, los enlaces de comunicación y la configuración de la red utilizada para la comunicación entre los nodos.

EJEMPLO:

La vista a continuación presenta una distribución física preliminar de los componentes y la infraestructura (recursos técnicos) necesaria para el sistema GestiHorarios. Dado que se cuentan con componentes ya definidos, la solución presentada puede ir creciendo y escalando a medida que se vaya necesitando más capacidad del sistema.

Esta vista proporciona una representación física detallada de cómo se despliega GestiHorarios en el entorno de implementación, mostrando la distribución de los componentes del sistema y los recursos de hardware necesarios para su ejecución.

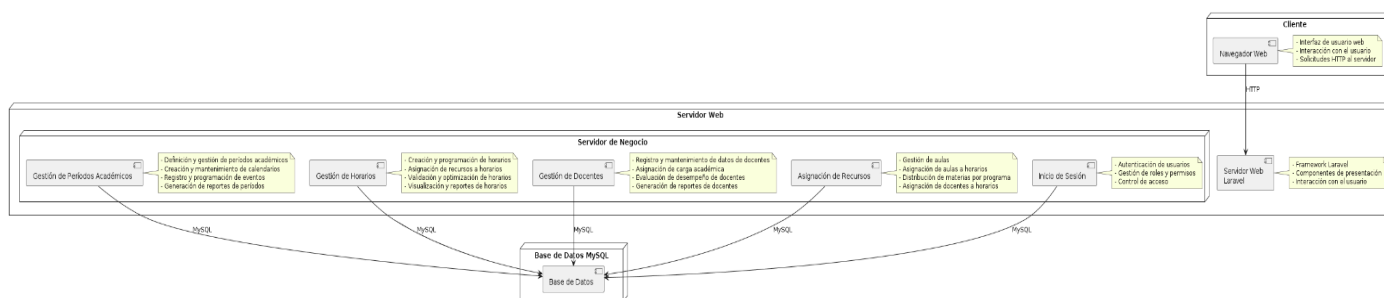


Figura 24. Vista de Despliegue Preliminar GestiHorarios

El manejo general del sistema GestiHorarios se realizará a través de un cliente web, este cliente puede ser un navegador web instalado en un computador (escritorio o portátil) o cualquier navegador web disponible para dispositivos móviles. El balanceador de carga puede ser utilizado o no, y tiene como objetivo asignar la carga entre los servidores web que se desplieguen, en caso de que se requieran varios servidores web por aspectos de escalabilidad. En los servidores web se instalan los componentes de presentación del sistema, como la Gestión de Períodos Académicos, Gestión de Horarios, Gestión de Docentes, Asignación de Recursos e Inicio de Sesión, que interactúan a través de una conexión de intranet con el servidor de negocio, donde se encuentran los componentes de gestión y lógica de negocio del sistema. El tener los componentes definidos a nivel del negocio permite que se pueda tener, en algún caso dado, los componentes de Gestión de Períodos Académicos, Gestión de Horarios, Gestión de Docentes, Asignación de Recursos o Inicio de Sesión residiendo en un servidor aparte, separados de los demás componentes. Lo mismo se puede hacer para cualquiera de los conjuntos de componentes de negocio.

Para el manejo de la persistencia de datos, los componentes de negocio interactúan directamente con el motor de base de datos instalado en el servidor establecido (Base de Datos de Horarios y Base de Datos de Usuarios). También desde el servidor de negocio se podría realizar la conexión con servicios externos ofrecidos por otras empresas, si fuera necesario.

6.1 Tecnología requerida

Describe las tecnologías específicas que se requieren para implementar y ejecutar el sistema. Esta sección proporciona detalles sobre las plataformas, lenguajes de programación, frameworks, bases de datos, herramientas y otros componentes tecnológicos necesarios para

construir y operar el sistema de software.

Plataforma de Hardware:

- Servidores con capacidad suficiente para ejecutar el sistema de software.
- Dispositivos de red para la conectividad.
- Equipos necesarios para el desarrollo y pruebas del sistema.

Sistema Operativo:

- Windows, Linux o macOS para el desarrollo y ejecución del sistema.

Lenguajes de Programación:

- PHP 8.0.3 para el backend.
- JavaScript ECMAScript 6 para el desarrollo del cliente.
- HTML 5 y CSS 3 para la estructura y estilos de las páginas web.

Frameworks y Bibliotecas:

- Laravel 8.0 para el backend.
- Bootstrap 5.2.1 para el diseño web receptivo.
- jQuery 3.6.0 para funciones JavaScript adicionales.
- Popper.js 2.9.3 para gestionar la posición de elementos emergentes.

Bases de Datos:

- MySQL 8.0.23 para almacenar y gestionar los datos del sistema.

Protocolos y Estándares:

- HTTP para la comunicación entre cliente y servidor.
- JSON para el intercambio de datos estructurados.

Herramientas de Desarrollo:

- Visual Studio como IDE principal.
- XAMPP como paquete de software para el desarrollo local, que incluye MySQL, Apache, PHP y Perl.

7 Rationale

El propósito de esta sección es proporcionar una explicación detallada de las decisiones y elecciones arquitectónicas realizadas durante el diseño y desarrollo de GestiHorarios. Aquí se explican las razones y motivaciones detrás de las decisiones arquitectónicas clave, proporcionando una base sólida para entender el diseño y las implicaciones de la arquitectura del software.

Objetivos arquitectónicos:

Los objetivos arquitectónicos para GestiHorarios incluyen escalabilidad, rendimiento, seguridad, modularidad y mantenibilidad. Estos objetivos son fundamentales para asegurar que el sistema pueda crecer, adaptarse y mantenerse eficaz a lo largo del tiempo.

- Escalabilidad: Permite manejar un número creciente de usuarios y horarios a medida que el sistema se expande.
- Rendimiento: Asegura que el sistema responda rápidamente y maneje eficientemente múltiples usuarios simultáneos.
- Seguridad: Protege los datos de los usuarios mediante mecanismos robustos de encriptación, autenticación y autorización.
- Modularidad: Facilita el mantenimiento y la actualización del sistema, permitiendo que los componentes sean modificados o reemplazados independientemente.
- Mantenibilidad: Garantiza que el sistema sea fácil de actualizar y mantener, minimizando

el tiempo y esfuerzo necesarios para realizar cambios y mejoras.

Decisiones de diseño:

Durante el diseño de GestiHorarios, se tomaron decisiones arquitectónicas clave para alinear el sistema con los objetivos definidos.

- **Arquitectura Cliente-Servidor:** Esta elección se basó en la necesidad de agregar nuevos clientes (usuarios) sin impactar significativamente la arquitectura existente. El patrón Cliente-Servidor facilita la gestión de múltiples usuarios accediendo simultáneamente para ver y gestionar horarios e información académica.
- **Patrón de Capas:** Se implementaron capas para separar la lógica de negocio del acceso a datos y de la interfaz de usuario. Esto permite una mayor modularidad y facilita la escalabilidad horizontal y vertical.

Alternativas consideradas:

- **Arquitectura Monolítica:** Fue considerada pero finalmente descartada debido a su falta de flexibilidad y dificultad para escalar. Aunque una arquitectura monolítica podría haber simplificado el desarrollo inicial, no habría soportado bien el crecimiento y la necesidad de cambios rápidos.
- **Arquitectura de Microservicios:** Evaluada por su alta modularidad y escalabilidad. Sin embargo, se consideró demasiado compleja para el alcance actual del proyecto y con un costo de implementación y mantenimiento superior.

Trade-offs (compromisos):

En el diseño de GestiHorarios, se realizaron varios compromisos para equilibrar diferentes objetivos arquitectónicos.

- **Modularidad vs. Rendimiento:** Se prioriza la modularidad y mantenibilidad sobre el rendimiento inmediato, asumiendo que la capacidad de actualizar y mantener el sistema a largo plazo era más beneficiosa.
- **Seguridad vs. Usabilidad:** Se implementaron medidas de seguridad robustas que no comprometen significativamente la facilidad de uso para los usuarios.

Consideraciones del entorno:

Se tuvieron en cuenta varios factores del entorno en el que se desplegará GestiHorarios:

- **Restricciones de hardware:** El sistema se diseñó para funcionar eficientemente en diversas plataformas, considerando limitaciones de hardware.
- **Requisitos de interoperabilidad:** Asegurar que GestiHorarios sea compatible con otros sistemas educativos y administrativos existentes.
- **Políticas de seguridad:** Cumplir con las regulaciones y estándares de seguridad de datos educativos, protegiendo la información sensible de los usuarios.

Decisiones futuras y evolución:

Se mencionan las consideraciones y planes para la evolución futura del sistema. Esto puede incluir aspectos como la incorporación de nuevas características, cambios tecnológicos, actualizaciones de software o mejoras en la arquitectura. Se justifican las decisiones tomadas en función de la flexibilidad y capacidad de adaptación del sistema a futuros cambios.

Ejemplo:

El diseño arquitectural de GestiHorarios se realizó siguiendo el proceso unificado y el método de diseño arquitectural guiado por los atributos de calidad identificados en la etapa anterior. Una de las principales tareas fue documentar y entender el método ADD para posteriormente aplicarlo.

Dado que tuvimos dos niveles de refinamiento, el ADD fue aplicado al sistema para definir la estructura lógica entre los principales subsistemas y fue parcialmente aplicado en el segundo nivel de refinamiento. Antes de descomponer el sistema completo, se revisó que no hicieran falta requisitos relevantes mediante un chequeo y una discusión entre los participantes.

Posteriormente, se priorizaron los atributos de calidad y casos de uso en términos de relevancia arquitectónica y de su localidad/globalidad. Esta información fue agregada a la descripción de los casos de uso y los escenarios de calidad.

Basándonos en esta priorización, se identificaron los drivers arquitectónicos para la primera descomposición. A estos drivers les asociamos las tácticas a tener en cuenta y los patrones que podrían ayudar a implementar las tácticas. De esta manera, se identificó qué patrones permiten implementar las tácticas. Los patrones relevantes fueron:

Desde un view type de módulos:

- Descomposición, uso y generalización (patrones naturalmente aplicados).
- Capas para la separación de lógica y acceso.

Desde un view type de C&C:

- Cliente/Servidor y Tiers para conseguir escalabilidad horizontal y vertical.

La siguiente tabla detalla los drivers arquitectónicos del sistema GestiHorarios, las incumbencias asociadas, las tácticas empleadas y los patrones arquitectónicos aplicados para satisfacer estos drivers.

Driver Candidato	Incumbencias	Tácticas	Patrones
QS1	Seguridad	Encriptación, autenticación y autorización	Cliente-Servidor, capas
QS2	Usabilidad	Separar interfaz de usuario, soporte de usuario	Capas
QS3	Escalabilidad	Independencia modular, ligadura tardía, asincronía	Cliente-Servidor

Tabla 3. Drivers, Tácticas y Patrones

El patrón arquitectónico de Cliente-Servidor se selecciona porque permite agregar nuevos usuarios (estudiantes, profesores, administradores) sin un mayor impacto en la arquitectura. Esto es esencial para GestiHorarios, ya que la plataforma debe manejar múltiples usuarios accediendo simultáneamente para ver y gestionar horarios, información académica y recursos.

Debido a la posible escalabilidad que puede tener el servicio, este patrón es adecuado para manejar el aumento de usuarios y datos sin comprometer el rendimiento. Además, el patrón de capas ayuda a mantener la separación entre los componentes del sistema, lo que permite que en determinado momento estas partes puedan estar en diferentes máquinas, favoreciendo la

escalabilidad horizontal y vertical.

El patrón Cliente-Servidor permite definir claramente el rol de quien expone los servicios y quien los consume. Por motivos de seguridad, es necesario controlar la gestión de acceso a los datos académicos y recursos, usando servicios que aseguren la autenticación y autorización de los usuarios. Esto es crucial para proteger la información sensible y evitar accesos no autorizados.

Adicionalmente, el patrón arquitectónico de Cliente-Servidor facilita la integración entre diferentes sistemas educativos y administrativos, además de que su estructura modular permite la integración de nuevas tecnologías y mejoras en la infraestructura existente. La escalabilidad vertical permitirá mejorar las características del servidor de datos a medida que crezcan las necesidades del sistema.