

**Universidad Nacional de San Antonio Abad del Cusco**  
**Facultad de Ingeniería Eléctrica, Electrónica, Informática y**  
**Mecánica**



**Escuela Profesional de Ingeniería Informática y de Sistemas**

---

SISTEMA WEB PARA LA REVISIÓN DE PORTAFOLIOS DE DOCENTES DEL  
DEPARTAMENTO

---

**Integrantes:**

- Aguilar Sanchez, Nik Antoni
- Calderon Garmendia, Joseph Timothy
- Huisa Nina, Yimy Yohel
- Orcon Diaz, Darcy Omar
- Pari Arriaga, Denilson

**Docente:**

Ing. Willian Zamalloa Paro

Cusco - Perú

2024  
**ÍNDICE**

<b>1. PRESENTACIÓN DE SOLUCIÓN DE PROYECTO.....</b>	<b>3</b>
1.1. Descripción del Proyecto.....	3
1.2 Metodología Utilizada.....	4
1.3. Pasos Seguidos para la Implementación.....	5
1.4. Recomendaciones para la Implementación.....	7
<b>2. Organigrama del proyecto.....</b>	<b>8</b>
<b>3. Plan del proyecto.....</b>	<b>8</b>
3.1. Proceso de asignación de revisores.....	8
3.2. Proceso de revisión de portafolios.....	8
4. Glosario de términos.....	9
<b>5. DOCUMENTO DE REQUERIMIENTOS.....</b>	<b>10</b>
5.1. Requerimientos funcionales.....	10
5.2. Requerimientos no funcionales.....	10
5.3. Matriz de rastreo de los requerimientos (Rastreabilidad).....	10
<b>6. DOCUMENTOS DE DISEÑO.....</b>	<b>11</b>
6.1. Arquitectura funcional del sistema.....	11
6.2. Arquitectura técnica del sistema.....	11
<b>7. CONTRATO DE DESARROLLO DE SOFTWARE.....</b>	<b>15</b>
7.1. Objeto del contrato:.....	15
7.2. Especificaciones del proyecto:.....	15
7.3. Obligaciones del desarrollador (QUIERO APROBAR S.A.C.):.....	16
7.4. Obligaciones del Cliente (DAII - UNSAAC):.....	16
7.5. Plazos de entrega:.....	17
7.6. Precio y forma de pago:.....	17
7.7. Confidencialidad:.....	17
7.8. Garantías:.....	17
7.9. Resolución de controversias:.....	18
7.10. Firma de las partes:.....	18
<b>8. REPORTE.....</b>	<b>18</b>
8.1. Reporte General de Revisión:.....	18
8.2. Reporte Individual de Portafolios:.....	18
8.3. Reporte por Revisor:.....	18
8.4. Reporte Comparativo:.....	19
8.5. Reporte de Observaciones:.....	19
<b>9. REFERENCIAS.....</b>	<b>19</b>



# **1. PRESENTACIÓN DE SOLUCIÓN DE PROYECTO**

## **1.1. Descripción del Proyecto**

Título del Proyecto: Sistema de Revisión de Portafolios Docentes

Antecedentes:

En la actualidad, muchas instituciones educativas, incluida la UNSAAC, han centrado sus esfuerzos en mejorar la calidad educativa, lo que ha generado la necesidad de administrar de manera eficiente la información relevante y contar con medios de evaluación que proporcionen una visión clara del desempeño docente. Estas herramientas no solo evalúan el progreso académico, sino que también fomentan la reflexión y la auto-evaluación de los docentes.

El uso de los portafolios docentes en la UNSAAC surge como un instrumento clave para gestionar el desarrollo profesional y promover evaluaciones más objetivas y estructuradas, contribuyendo directamente a la mejora continua en los procesos educativos.

Problema:

A la fecha, la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) enfrenta desafíos similares a los experimentados por otras instituciones en la gestión de los portafolios docentes, llevando el proceso de evaluación de manera manual. Este proceso implica revisar el portafolio, presentar resultados y organizar los registros según los lineamientos y formatos específicos que la universidad exige. La dificultad de administrar y hacer un seguimiento efectivo de cada portafolio se refleja en la lentitud y falta de precisión en la entrega de los resultados, afectando tanto a docentes como a supervisores.

Para solucionar estas dificultades, el uso de herramientas informáticas permite optimizar el proceso, mejorando los tiempos de respuesta, facilitando la evaluación y promoviendo un mejor control y administración de la información. Este contexto impulsa la necesidad de desarrollar un sistema automatizado de revisión de portafolios, alineado con los lineamientos y formatos establecidos por la UNSAAC, con el fin de modernizar y hacer más eficiente la gestión académica.

Objetivo:

El proyecto busca desarrollar un sistema web para la gestión y evaluación de portafolios docentes. El sistema de revisión de portafolios docentes tiene como objetivo principal facilitar y automatizar el proceso de evaluación de criterios en los contenidos de los portafolios que los docentes suben a la plataforma de Google Drive. Esta solución se enfoca en garantizar que los supervisores puedan revisar, de manera eficiente y estructurada, los criterios predefinidos para cada portafolio, cumpliendo con los lineamientos establecidos por la universidad.

Beneficios:

Automatización de la evaluación de portafolios.

Comunicación fluida entre supervisores y docentes mediante notificaciones.

Generación de informes estructurados y seguimiento del progreso.

## **1.2 Metodología Utilizada**

Metodología de Desarrollo: Scrum

Scrum es una metodología ágil que nos permite desarrollar el sistema de manera iterativa e incremental, enfocado en la entrega continua de valor y la adaptación constante a las necesidades del usuario. Esta metodología se ha elegido por su capacidad para mejorar la colaboración del equipo, permitir la flexibilidad en el desarrollo, y responder a los cambios rápidamente. A continuación, describimos cómo se aplicó Scrum en el desarrollo de nuestro sistema de revisión de portafolios:

Los roles principales son:

Scrum Master y Desarrollador: Lidera el equipo y gestiona las ceremonias de Scrum. (Daily Stand-ups, Sprint Planning, Sprint Review y Retrospective)

Desarrollador Full Stack: Desarrolla tanto el frontend como el backend del sistema.

Desarrollador Frontend: Implementa la interfaz de usuario del sistema.

Desarrollador Backend: Desarrolla la lógica de negocio y la integración con la base de datos.

Diseñador UX/UI: Responsable de la experiencia de usuario y el diseño visual del sistema.

### 1.3. Pasos Seguidos para la Implementación

#### Paso 1: Análisis de Requisitos

Se recopilaron los requerimientos funcionales y no funcionales del sistema. Entre los requerimientos clave se incluyen la capacidad de gestionar usuarios, realizar evaluaciones detalladas y generar reportes automáticos. Esto fue fundamental para estructurar el modelo de la base de datos y definir las relaciones entre los componentes principales del sistema.

#### Paso 2: Diseño de la Base de Datos

Se desarrolló un esquema de base de datos con las tablas principales: Usuarios, Evaluaciones, Criterios de Evaluación, Distribución y Notificaciones. El modelo de datos fue diseñado para garantizar la escalabilidad del sistema y permitir una gestión eficiente de la evaluación de portafolios.

#### Paso 3: Desarrollo del Sistema

El desarrollo del sistema se realizará utilizando Laravel como framework de backend y MySQL como gestor de base de datos. El sistema fue dividido en varios módulos, como:

- Módulo de gestión de usuarios: Este módulo controla el acceso y los permisos de los usuarios dentro del sistema, asegurando que cada usuario tenga los privilegios adecuados según su rol.
- Módulo de Gestión de Carga: Este módulo gestiona las cargas académicas de los docentes, asignando cursos y distribuyendo la carga académica por semestres.
- Módulo de Gestión de Evaluaciones: Este módulo se encarga de la evaluación de los portafolios docentes, permitiendo a los supervisores revisar los contenidos según los criterios establecidos.
- Módulo de Gestión de Informes: Este módulo genera los informes de evaluación y maneja las notificaciones que se envían a los usuarios (docentes y supervisores) al completar una evaluación.

#### Paso 4: Pruebas y Validación

Se llevarán a cabo pruebas internas para verificar que las evaluaciones se registren correctamente, los informes se generen de forma automatizada, y que las notificaciones se envíen con la información adecuada. El sistema pasó por varias iteraciones de prueba antes de estar listo para su implementación.

### **1.4. Recomendaciones para la Implementación**

#### Escalabilidad:

Se recomienda que el sistema sea implementado inicialmente en un entorno de desarrollo controlado, con pocos usuarios, y luego escalado gradualmente según la demanda. Esto evitará sobrecargar el sistema desde su inicio y permitirá ajustar aspectos de rendimiento.

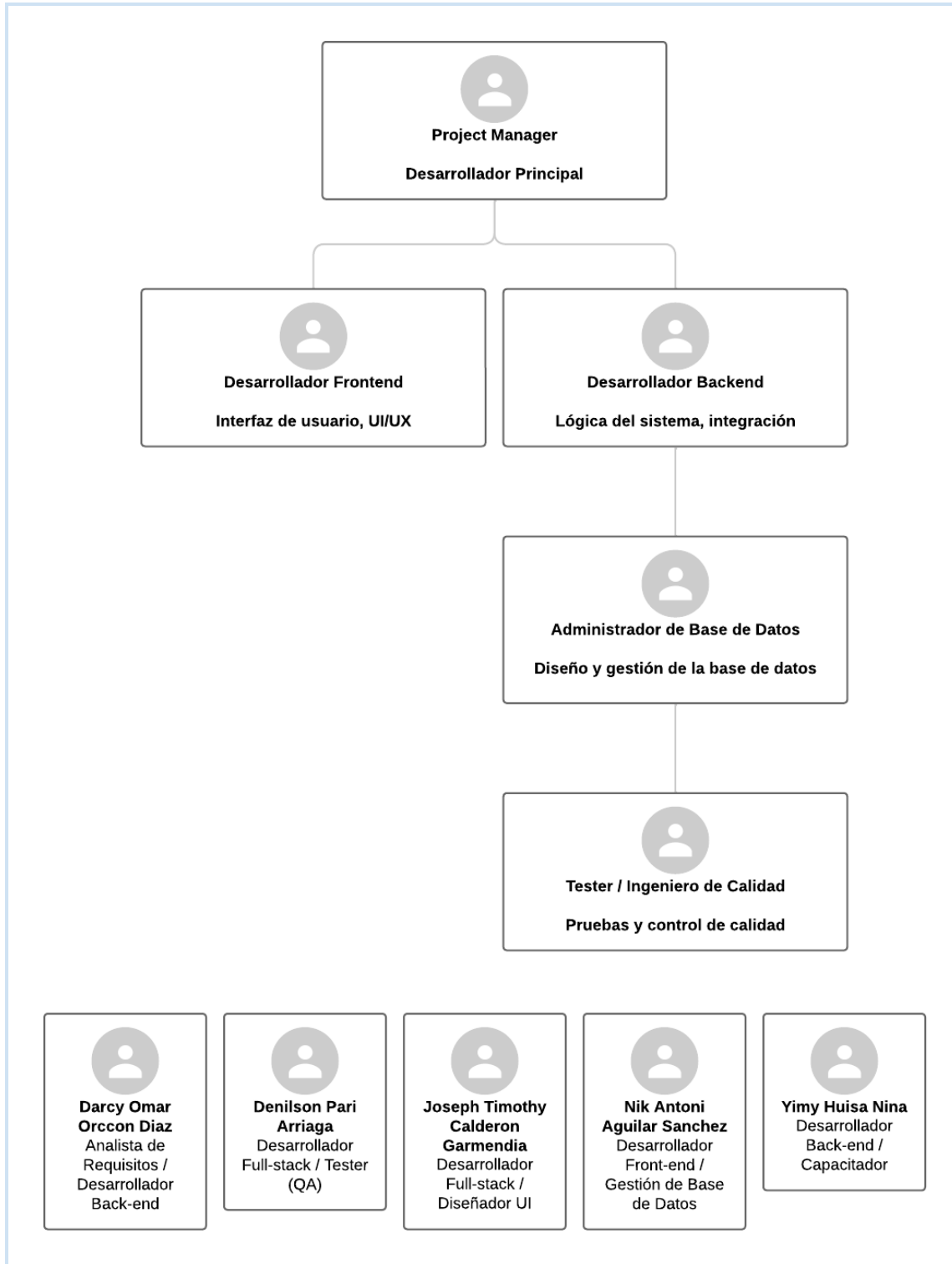
#### Seguridad:

Asegurarse de implementar mecanismos de seguridad en el sistema, como la encriptación de contraseñas, roles de usuario, y control de accesos según privilegios (docente, supervisor, administrador).

#### Mejoras Futuras:

Se sugiere que en futuras versiones del sistema se integre una plataforma de mensajería interna para notificaciones y una interfaz de usuario más amigable y adaptable para los diferentes dispositivos (responsive design).

## 2. Organigrama del proyecto.





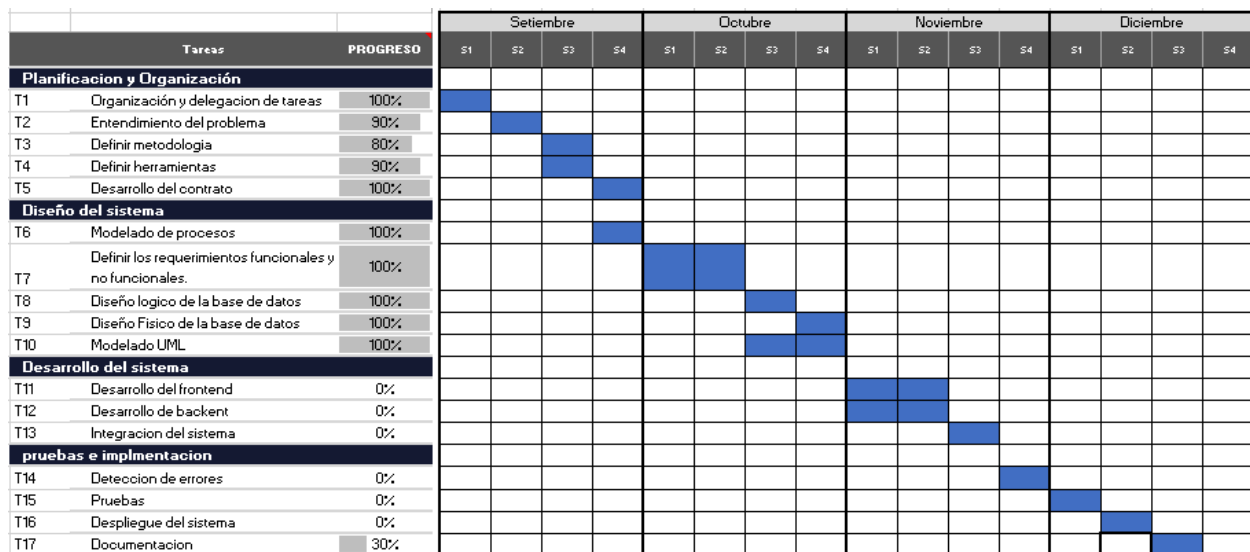
### 3. Plan del proyecto

#### 3.1. Plan del proyecto

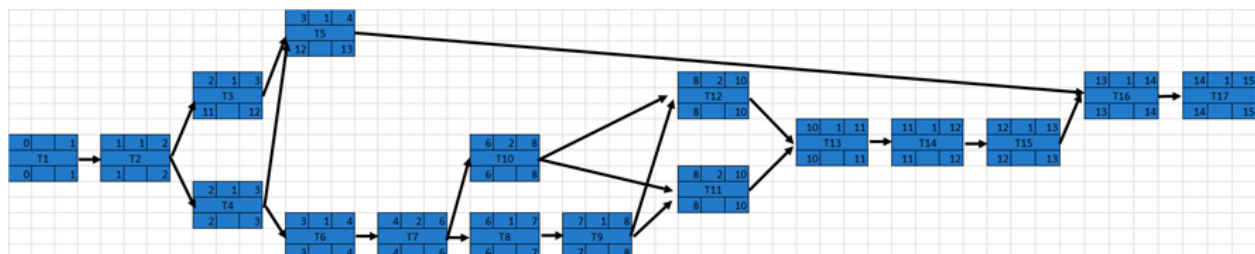
Fase	Tarea	Actividad	Objetivo a alcanzar	Recursos	Tareas necesarias
<b>Planificación y Organización</b>	T1	Organización y delegación de tareas	Establecer un grupo de desarrollo más organizado, con obligaciones claras a cumplir para la primera entrega.	Google meet	
	T2	Entendimiento del problema	Comprender de la mejor manera la asignación, discerniendo las dudas y proponiendo ideas.	Google meet	T1
	T3	Definir la metodología	Establecer una metodología que se adapte mejor a los integrantes del grupo.	Google meet	T2
	T4	Definir herramientas	Se definirá las herramientas a utilizar	Google meet	T2
	T5	Desarrollo del contrato	Desarrollar un contrato en el que estén de acuerdo tanto el cliente como los desarrolladores.	Google docs	T3, T4
<b>Diseño del sistema</b>	T6	Modelado de procesos	Tener un modelo de los procesos que se realizan durante la revisión de portafolios	Bizagi	T4
	T7	Definir los requerimientos funcionales y no funcionales.	Entender el sistema que se busca a desarrollar y conocer los requerimientos que cumplirá el sistema	Google meet, Google Docs	T6
	T8	Diseño lógico de la base de datos	Tener un modelo lógico de una base de datos	draw.io, Google meet	T7
	T9	Diseño Físico de la base de datos	Tener un modelo físico de la base de datos	mysql workbench	T8
	T10	Modelado UML	Modelar los procesos mediante casos de uso, diagramas de secuencia y modelos de objetos.	Lucidchart	T7
<b>Desarrollo del sistema</b>	T11	Desarrollo del frontend	Desarrollar vistas interactivas, dinámicas y atractivas.	Visual studio, laragon	T9, T10
	T12	Desarrollo de backend	Desarrollar una base lógica sólida que garantice eficiencia y estabilidad	Visual studio, laragon	T9, T10

	T13	Integración del sistema	Se busca integrar la base de datos con el frontend y backend.	Visual studio, laragon	T11. T12
<b>Pruebas e implementación</b>	T14	Detección de errores	Encontrar y erradicar errores de código.	Visual studio, laragon	T13
	T15	Pruebas	Obtener un funcionamiento óptimo del sistema	Visual Studio, laragon	T14
	T16	Despliegue del sistema	Sistema web en funcionamiento	AWS, visual studio	T15
	T17	Documentación	Obtener las guías de usuario y la documentación técnica.	Google docs	T16

### 3.2. Diagrama de GANTT



### 3.3. Diagrama de PERT



#### 4. **Glosario de términos**

**Sistema de Revisión de Portafolios Docentes:** Plataforma web diseñada para evaluar portafolios de docentes, permitiendo a supervisores revisar, calificar y emitir informes automáticos.

**Docente:** Usuario del sistema que sube los contenidos de su portafolio a Google Drive. El sistema se utiliza para revisar si esos contenidos cumplen con los criterios establecidos.

**Revisor:** Persona encargada de revisar los contenidos en Google Drive. Los revisores evalúan el cumplimiento de los criterios de evaluación y generan informes sobre el estado de los portafolios.

**Administrador:** Usuario encargado de gestionar la plataforma, asignando revisores a los docentes y controlando el proceso de evaluación y notificaciones.

**Criterios de Evaluación:** Parámetros predefinidos utilizados por los supervisores para calificar el contenido de los portafolios en Drive. Cada criterio tiene un peso y su cumplimiento puede ser obligatorio u opcional.

**Evaluación de Portafolios:** Parámetros predefinidos utilizados por los supervisores para calificar el contenido de los portafolios en Drive. Cada criterio tiene un peso y su cumplimiento puede ser obligatorio u opcional.

**Notificaciones:** Comunicaciones enviadas a los docentes y supervisores para informarles sobre asignaciones, evaluaciones y actualizaciones en el estado de los contenidos.

**Informe de Evaluación:** Documento generado por el sistema que resume el resultado de la evaluación de un portafolio, incluyendo observaciones y sugerencias.

**Historial de Revisiones:** Registro de todas las acciones realizadas sobre un portafolio, que incluye las fechas de revisión, observaciones y el estado de cada evaluación.

**Semestre Activo:** Periodo académico en el cual se realizan las evaluaciones de portafolios. El sistema gestiona portafolios por semestre, permitiendo activar o cerrar ciclos académicos.

**SaaS(Software as a Service) :** Modelo de distribución de software en el que las aplicaciones están alojadas en servidores externos y se acceden a través de internet. En este caso, el sistema de revisión de portafolios se ofrece como un SaaS, permitiendo que los usuarios

**Herramientas informáticas:** accedan desde cualquier lugar sin necesidad de Conjunto de aplicaciones y programas utilizados en el sistema para facilitar la revisión de los portafolios. Esto

incluye el acceso a Google Drive para el almacenamiento de portafolios y el uso de otras tecnologías web para gestionar usuarios, evaluaciones, reportes y notificaciones. Instalar software en sus equipos.

## 5. DOCUMENTO DE REQUERIMIENTOS

### 5.1. Requerimientos funcionales

<b>Autenticación y Autorización</b>	
RF1	El sistema validará las credenciales de los usuarios (docentes, supervisores y administradores) para verificar su identidad.
RF2	El sistema verificará que solo los usuarios autorizados accedan a las funcionalidades correspondientes a su rol (docente, supervisor, administrador)
<b>Asignación de Revisores</b>	
RF3	El sistema podrá asignar revisores automáticamente basándose en la disponibilidad y el semestre activo.
RF4	Los administradores podrán asignar revisores manualmente según las necesidades.
<b>Gestión de Administradores</b>	
RF5	Los administradores podrán gestionar los revisores, incluyendo la asignación de portafolios.
RF6	Los administradores podrán gestionar a los docentes y controlar las evaluaciones de sus portafolios.
<b>Generación de Reportes</b>	
RF7	El sistema generará reportes sobre el progreso de las revisiones, mostrando el cumplimiento de los criterios.

RF8	Los reportes incluirán comentarios de los revisores sobre las evaluaciones.
<b>Notificaciones a Revisores</b>	
RF9	Los revisores recibirán notificaciones cuando se les asignen nuevos portafolios.
RF10	Los revisores serán notificados cuando cambie el estado de una revisión.
<b>Notificaciones a Docentes</b>	
RF11	Los docentes recibirán notificaciones con los detalles de las observaciones realizadas por los revisores.
<b>Listas de Verificación para Revisores</b>	
RF12	El sistema generará automáticamente listas de verificación basadas en los criterios establecidos para los portafolios.
<b>Evaluaciones Basadas en Criterios</b>	
RF13	Los revisores podrán evaluar los portafolios basados en criterios de cumplimiento y peso.
<b>Gestión de Semestres</b>	
RF14	Los semestres tendrán un estado (activo o cerrado) según el momento del ciclo académico.
<b>Historial de Revisiones</b>	
RF15	Se almacenarán las revisiones con fecha, estado de evaluación y observaciones de los revisores.
<b>Acceso a Roles Múltiples</b>	
RF16	Los revisores y administradores tendrán la capacidad de acceder como docentes para revisar sus propios portafolios.

## 5.2. Requerimientos no funcionales

Requerimientos	Descripción
----------------	-------------

Rendimiento y Concurrencia Eficiente	El sistema debe ser capaz de manejar un gran número de usuarios concurrentes (docentes, revisores, administradores) sin afectar el rendimiento. Se deben implementar estrategias de optimización de base de datos y caching para mejorar el rendimiento de las consultas.
Escalabilidad	El sistema debe ser escalable horizontalmente y verticalmente para adaptarse al crecimiento en el número de usuarios y evaluaciones. La arquitectura debe permitir la expansión sin afectar la funcionalidad, y se debe implementar cifrado MD5 en todas las comunicaciones y almacenamiento de credenciales para garantizar la seguridad de los datos
Compatibilidad	El sistema debe ser compatible con navegadores web y dispositivos para asegurar su accesibilidad.
Rendimiento	El sistema debe ofrecer tiempos de respuesta rápidos para las consultas, revisiones y generación de reportes. Se debe optimizar el acceso a la base de datos y minimizar el tiempo de carga, asegurando que todas las operaciones se completan en menos de 2 segundos en condiciones normales de carga.
Usabilidad	El sistema debe contar con una interfaz de usuario intuitiva, fácil de usar y adaptada para revisores, administradores y docentes. Se debe asegurar que las funcionalidades más utilizadas estén accesibles en pocos clics y que los usuarios puedan completar las tareas clave sin dificultad
Accesibilidad	El sistema debe garantizar una alta disponibilidad (al menos un 99.5% de tiempo de actividad). Para lograr esto, debe contar con infraestructura redundante y planes de recuperación ante fallos para asegurar que los usuarios puedan acceder al sistema en cualquier momento.
Seguridad	El sistema validará las credenciales de los usuarios mediante autenticación segura. Todos los datos sensibles, como contraseñas y registros de portafolios, deben estar cifrados tanto en tránsito

	como en reposo.
Mantenibilidad	El código debe seguir estándares de desarrollo bien documentados para facilitar su mantenimiento y actualización. Además, se deben utilizar herramientas de control de versiones (como Git) y documentación técnica para asegurar que el sistema pueda ser modificado fácilmente por cualquier equipo de desarrollo en el futuro.

### 5.3. Matriz de rastreo de los requerimientos (Rastreabilidad)

		lista de Casos de Uso										
		CU1	CU2	CU3	CU4	CU5	CU6	CU7	CU8	CU9	CU10	CU11
Requerimientos	R1	X										
	R2		X									
	R3			X								
	R4				X							
	R5					X						
	R6						X					
	R7							X				
	R8								X			
	R9									X		
	R10										X	
	R11											X

## 6. DOCUMENTOS DE DISEÑO

### 6.1. Arquitectura funcional del sistema

La arquitectura funcional es un esquema que describe cómo se organizan y cómo interactúan los diferentes componentes o módulos del sistema para cumplir con los requerimientos funcionales del proyecto. Este tipo de arquitectura se centra en qué hace el sistema (sus funcionalidades) y cómo se organizan los módulos o componentes para lograr esos objetivos.

#### 1) Elementos a Incluir en la Arquitectura Funcional:

- **Componentes Principales del Sistema**
  - **Gestión de Usuarios:**
    - **Función:** Control de acceso al sistema, autenticación de usuarios (docentes, supervisores, administradores) y asignación de roles.

- **Relación con Módulos:** Este componente pertenece al **módulo de Gestión de Usuarios**, donde se maneja todo lo relacionado con los permisos y la seguridad del sistema.
- **Asignación de Supervisores:**
  - **Función:** Asignación de supervisores a los docentes para la evaluación de los portafolios. Este componente es clave para garantizar que cada docente tenga un supervisor asignado para realizar las evaluaciones.
  - **Relación con Módulos:** Se puede considerar como parte del **módulo de Gestión de Evaluaciones**, ya que permite asignar quién realizará la revisión de los portafolios
- **Evaluación de Criterios:**
  - **Función:** Permitir a los supervisores evaluar los portafolios docentes utilizando un checklist basado en los criterios predefinidos para cada tipo de curso (teoría, práctica).
  - **Relación con Módulos:** Este componente es central en el **módulo de Gestión de Evaluaciones**, ya que realiza la tarea principal del sistema.
- **Generación de Reportes:**
  - **Función:** Automatizar la creación de reportes con los resultados de las evaluaciones de los portafolios. Los reportes detallan si los criterios han sido cumplidos y qué observaciones hizo el supervisor.
  - **Relación con Módulos:** Este componente pertenece al **módulo de Gestión de Informes**, donde se generan y almacenan los reportes.
- **Notificaciones:**
  - **Función:** Enviar notificaciones automáticas a los docentes y supervisores cuando se completa una evaluación y el reporte está disponible. Las notificaciones se envían por correo electrónico.
  - **Relación con Módulos:** Este componente está asociado al **módulo de Gestión de Informes** o podría considerarse un componente transversal que interactúa con varios módulos, ya que envía notificaciones sobre diferentes eventos del sistema.
- **Carga Académica y Distribución:**
  - **Función:** Gestionar la carga académica de los docentes, asignando cursos y semestres, y distribuyendo la carga según los criterios definidos.
  - **Relación con Módulos:** Este componente pertenece al **módulo de Gestión de Carga Académica**, que organiza los cursos, semestres y docentes.

## 2) Descripción de Módulos y Funcionalidades

### a) Módulo de Gestión de Usuarios

- **Descripción:** El módulo de **Gestión de Usuarios** es responsable de controlar el acceso al sistema y la seguridad mediante la autenticación y la
- asignación de roles. Asegura que solo los usuarios autorizados (docentes, supervisores y administradores) puedan acceder al sistema y que cada uno tenga los permisos necesarios para realizar sus tareas.
- **Componentes Relacionados:**
  - **Gestión de Usuarios:** Autenticación y control de acceso.



- **Asignación de Roles:** Define los permisos de los usuarios según sus roles.
- **Funcionalidades Clave:**
  - Validación de credenciales.
  - Asignación de roles (docente, supervisor, administrador).
  - Permisos de acceso diferenciados por rol.

## 2.2. Módulo de Gestión de Carga Académica

- **Descripción:** El módulo de **Gestión de Carga Académica** administra la asignación de cursos y semestres a los docentes. Permite distribuir la carga académica entre los distintos cursos y gestionar la información relacionada con los semestres y la distribución de los docentes en cada curso.
- **Componentes Relacionados:**
  - **Carga Académica y Distribución:** Organización de cursos y semestres.
- **Funcionalidades Clave:**
  - Asignación de cursos a los docentes según los semestres.
  - Registro y distribución de la carga académica.
  - Gestión de semestres académicos.

## 2.3. Módulo de Gestión de Evaluaciones

- **Descripción:** Este módulo gestiona todo el proceso de evaluación de los portafolios docentes. Permite la asignación de supervisores a los docentes y facilita la evaluación de los portafolios utilizando criterios predefinidos. Los supervisores pueden hacer observaciones sobre los portafolios evaluados.
- **Componentes Relacionados:**
  - **Asignación de Supervisores:** Define qué supervisor evaluará a qué docente.
  - **Evaluación de Criterios:** Proceso de evaluación de los portafolios mediante un checklist de criterios.
- **Funcionalidades Clave:**
  - Asignación de supervisores a docentes.
  - Evaluación de portafolios basada en criterios específicos.
  - Observaciones y retroalimentación en el proceso de evaluación.

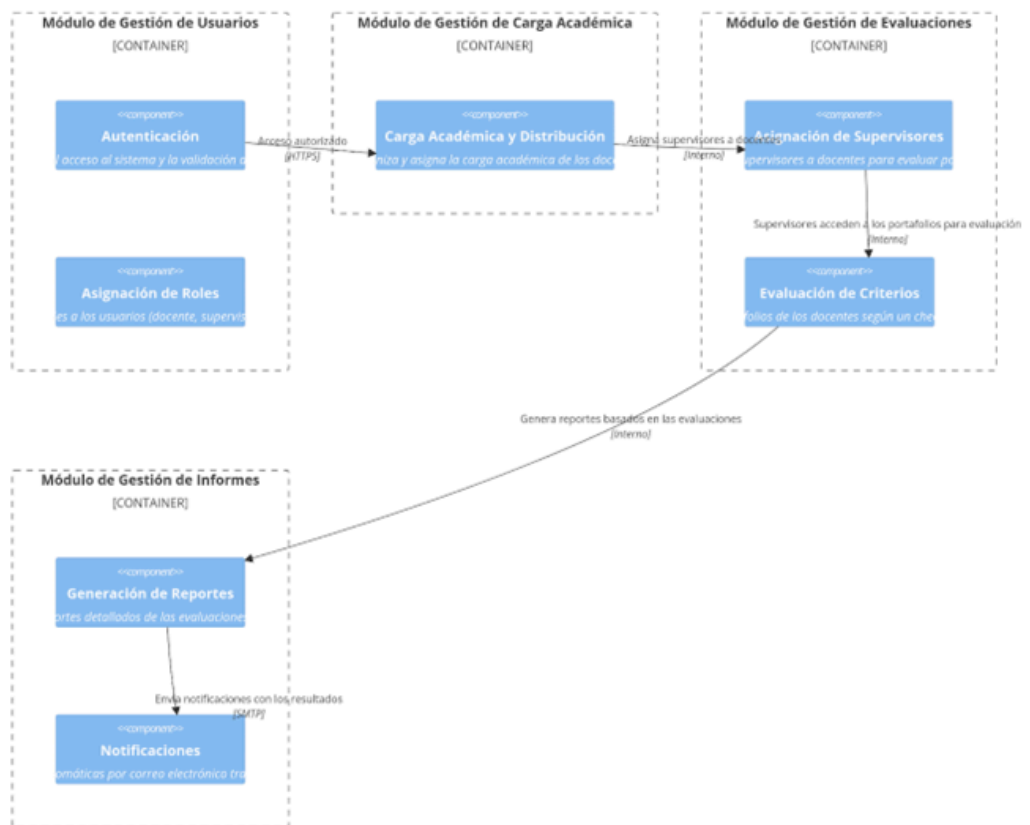
## 2.4. Módulo de Gestión de Informes

- **Descripción:** El módulo de Gestión de Informes genera reportes automáticos basados en las evaluaciones realizadas por los supervisores y envía notificaciones automáticas a los docentes y supervisores al finalizar la evaluación. Los informes incluyen las observaciones y resultados de la evaluación, y se almacenan para referencia futura.

- **Componentes Relacionados:**
    - **Generación de Reportes:** Crea informes detallados de la evaluación de portafolios.
    - **Notificaciones:** Envía alertas y reportes a los usuarios sobre el estado de la evaluación.
  - **Funcionalidades Clave:**
    - Generación automática de informes de evaluación.
    - Almacenamiento y acceso a los reportes.
    - Notificaciones automáticas a docentes y supervisores.
2. **Flujo de Información del Sistema de Revisión de Portafolios Docentes**
1. **Acceso al Sistema y Gestión de Usuarios**
    - **Inicio:** Un usuario (docente, supervisor o administrador) accede al sistema e ingresa sus credenciales.
    - **Acción del Módulo de Gestión de Usuarios:** Este módulo valida las credenciales y determina el rol del usuario, permitiéndole acceso solo a las funcionalidades asignadas.
    - **Resultado:** Los docentes acceden a sus portafolios, y los supervisores y administradores acceden a sus áreas de evaluación y asignación.
  2. **Asignación de Carga Académica**
    - **Inicio:** El administrador asigna la carga académica a cada docente al inicio del semestre, seleccionando los cursos y el semestre correspondiente.
    - **Acción del Módulo de Gestión de Carga Académica:** Este módulo gestiona la distribución de la carga y el registro de los cursos asignados.
    - **Resultado:** La información de carga académica queda registrada y vinculada a cada docente para facilitar la evaluación de sus portafolios.
  3. **Asignación de Supervisores**
    - **Inicio:** El administrador asigna un supervisor a cada docente o grupo de docentes, según los criterios de carga académica y disponibilidad.
    - **Acción del Módulo de Gestión de Evaluaciones:** El módulo registra la asignación de supervisores.
    - **Resultado:** Los supervisores tienen asignados los docentes que deben evaluar en el periodo.
  4. **Evaluación de Portafolios**
    - **Inicio:** El docente sube su portafolio a Google Drive.
    - **Acción del Módulo de Gestión de Evaluaciones:** Los supervisores acceden al checklist de criterios y revisan el portafolio, indicando el cumplimiento de cada criterio.
    - **Resultado:** El sistema almacena las evaluaciones realizadas por el supervisor.
  5. **Generación de Reportes y Notificaciones**
    - **Inicio:** Tras completar la evaluación, el sistema inicia la generación de un informe.

- **Acción del Módulo de Gestión de Informes:**
  - **Generación de Reporte:** Compila la información de la evaluación en un reporte detallado, que se almacena en el sistema.
  - **Notificación:** Envía un correo al docente con un enlace al reporte.
- **Resultado:** El docente recibe la notificación y puede acceder al reporte para ver los detalles de la evaluación

DIAGRAMA DE LA ARQUITECTURA FUNCIONAL DEL SISTEMA DE REVISION DE PORTAFOLIOS



## 6.2. Arquitectura técnica del sistema

### 6.2.1 Topología de software

El patrón de diseño MVC divide una aplicación en tres componentes principales: Modelo, Vista y Controlador. Esta arquitectura es fundamental para separar la

lógica de negocio de la interfaz de usuario, facilitando la escalabilidad y el mantenimiento del sistema.

## 1) Componentes del Desarrollo

### a) Controladores

Los controladores manejan las solicitudes HTTP, permitiendo la comunicación entre el servidor y el cliente. Facilitan la transferencia de datos entre el Modelo y la Vista.

- **Manejo de solicitudes HTTP:** Los controladores reciben las acciones del usuario y las direccionan hacia el modelo o la vista correspondiente.
- **Interacción Vista-Modelo:** Permiten la correcta actualización de la vista con la información del modelo.

### b) Vistas

Las vistas son responsables de mostrar la información del modelo en interfaces amigables y atractivas para el usuario. Se usan módulos visuales ya conocidos.

- **AdminLTE:** Un marco de interfaz que proporciona plantillas visuales listas para la administración de sistemas.
- **Componentes dinámicos:** Las vistas están diseñadas para ser interactivas y responder a las acciones del usuario.

### c) Modelos

Los modelos controlan la lógica de acceso a los datos, interactuando con las bases de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

- **Estructura de base de datos:** Los modelos interactúan con las tablas de la base de datos para realizar consultas y gestionar la información.
- **Organización de datos:** Se asegura que los datos estén estructurados de manera eficiente para su acceso y manipulación.

## 2) Herramientas de Desarrollo

*a) PHP:* Un lenguaje de programación que permite gestionar la lógica de negocio y la interacción entre el Modelo, la Vista y el Controlador, facilitando el desarrollo de aplicaciones web dinámicas.

*b) Laravel:* Un framework de PHP que ofrece una estructura sencilla y organizada para el desarrollo de aplicaciones, permitiendo la rápida implementación del patrón MVC.

*c) MySQL:* Sistema de gestión de bases de datos que proporciona una infraestructura sólida y segura para almacenar y gestionar los datos de la aplicación.

*d) HTML y CSS:* Lenguajes utilizados para la creación de interfaces de usuario. HTML estructura la presentación de la información y CSS se encarga del diseño visual de la aplicación.

### 3) Herramientas adicionales

- a) **ChatGPT:** Asistente de inteligencia artificial que ayuda en el desarrollo tanto del frontend como del backend, proporcionando soluciones rápidas y efectivas.
- b) **AWS EC2:** Servicio de Amazon que permite el despliegue y la administración de aplicaciones en la nube, asegurando un alto rendimiento y escalabilidad.
- c) **Git:** Controlador de versiones que facilita la gestión de cambios en el código fuente, permitiendo la colaboración entre varios desarrolladores y el seguimiento de la evolución del proyecto.

### 6.2.2 Topología de Hardware

En el diseño de un sistema web, como el sistema de revisión de portafolios docentes, es esencial establecer una topología de hardware adecuada. Esta topología es clave para asegurar un rendimiento óptimo, escalabilidad y seguridad del sistema. Incluye los servidores necesarios para alojar la aplicación web, gestionar la base de datos y almacenar documentos, así como las estaciones de trabajo de los usuarios finales que interactúan con el sistema. A continuación, se presentan las principales características de esta topología.

#### 1. Servidor en la Nube:

El sistema se desplegará en plataformas de nube como **AWS**, **Google Cloud** o **Microsoft Azure**, que permitirán alojar tanto el backend de Laravel como la base de datos MySQL. Esta elección de infraestructura ofrece diversas ventajas:

- **Disponibilidad:** Las plataformas en la nube garantizan alta disponibilidad mediante la replicación de servidores en múltiples zonas geográficas y la distribución de la carga de trabajo, minimizando el riesgo de tiempo de inactividad.
- **Escalabilidad:** En caso de un aumento en la carga de usuarios o datos, se podrán escalar los servicios mediante la adición de más instancias o recursos, sin necesidad de realizar modificaciones complejas en la arquitectura base.

#### 2. Base de Datos en la Nube:

La base de datos MySQL se aloja en el mismo entorno en la nube, lo que facilita la comunicación eficiente entre la aplicación y la base de datos. Esta configuración garantiza una baja latencia y un acceso rápido a los datos críticos del sistema.

- **Respaldo Automático:** Se implementarán soluciones de respaldo automático, como instantáneas programadas y replicación de datos, para asegurar la integridad de la información y su recuperación en caso de incidentes, lo que es esencial para la continuidad del sistema.

### 3. Dispositivos del Usuario Final:

Los docentes, revisores y administradores podrán acceder al sistema desde cualquier dispositivo que cuente con conexión a Internet. Dado que se trata de una aplicación web adaptativa, puede ser compatible tanto con computadoras de escritorio como con dispositivos móviles y tablets, garantizando así una experiencia de usuario fluida y accesible en diversas plataformas.

### 4. Optimización y Servicios Adicionales:

- **Redis:** Se evaluará la implementación de Redis para la gestión de caché y la optimización de colas de trabajo. Esto es crucial para gestionar múltiples notificaciones y tareas en segundo plano, mejorando así el rendimiento general del sistema y reduciendo la carga en la base de datos.

## 6.3. Documentos basados en UML

### 6.3.1 Casos de uso

Diagrama de caso de uso 01

CUI	CU01
Nombre del Caso de Uso	Autenticación y Autorización
Actor	Docente, Revisor, Administrador
Tipo	Primario
Descripción	El sistema validará las credenciales de los usuarios mediante autenticación segura, asegurando que solo los usuarios autorizados puedan acceder a las funcionalidades según su rol.

### Diagrama de caso de uso 02

CUI	CU02
Nombre del Caso de Uso	Asignación de Revisores
Actor	Administrador
Tipo	Primario
Descripción	El sistema permitirá asignar revisores responsables de evaluar cada portafolio de manera automática o manual, vinculando a los revisores con los docentes, tomando en cuenta el semestre activo.

### Diagrama de caso de uso 03

<b>CUI</b>	<b>CU03</b>
<b>Nombre del Caso de Uso</b>	Gestión de Administradores
<b>Actor</b>	Administrador
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá asignar administradores que gestionen los revisores y docentes, así como controlar las asignaciones y evaluaciones de cada portafolio.

#### Diagrama de caso de uso 04



CUI	CU04
<b>Nombre del Caso de Uso</b>	Generación de Reportes
<b>Actor</b>	Administrador, Revisor
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema generará reportes actualizados sobre el progreso y estado de las revisiones realizadas. Los reportes incluirán detalles de las evaluaciones, comentarios de los revisores, y el cumplimiento de los criterios de evaluación establecidos.

#### Diagrama de caso de uso 05

CUI	CU05
<b>Nombre del Caso de Uso</b>	Notificaciones a Revisores
<b>Actor</b>	Administrador
<b>Tipo</b>	Secundario
<b>Descripción</b>	El sistema enviará notificaciones automáticas a los revisores cuando se les asignen nuevos portafolios o cuando cambie el estado de una revisión. Estas notificaciones incluirán los detalles del portafolio, las fechas límite y el estado actual.

#### Diagrama de caso de uso 06

<b>CUI</b>	<b>CU06</b>
<b>Nombre del Caso de Uso</b>	Notificaciones a Docentes
<b>Actor</b>	Revisor
<b>Tipo</b>	Secundario
<b>Descripción</b>	El sistema notificará a los docentes sobre las observaciones o faltas encontradas en la revisión de sus portafolios. Las notificaciones incluirán detalles sobre las evaluaciones, criterios incumplidos y comentarios del revisor.

#### Diagrama de caso de uso 07

<b>CUI</b>	<b>CU07</b>
<b>Nombre del Caso de Uso</b>	Listas de Verificación para Revisores
<b>Actor</b>	Revisor
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá que los revisores generen listas de verificación automáticas basadas en los criterios de evaluación definidos para validar si los portafolios solicitados están completos. Estas listas serán generadas según la estructura del portafolio (teoría o práctica).

### Diagrama de caso de uso 08

CUI	CU08
<b>Nombre del Caso de Uso</b>	Evaluaciones Basadas en Criterios
<b>Actor</b>	Revisor
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá que los supervisores realicen evaluaciones basadas en criterios predefinidos. Cada criterio tendrá un peso, y las evaluaciones podrán realizarse en función de su obligatoriedad y cumplimiento.

### Diagrama de caso de uso 09

CUI	CU09
<b>Nombre del Caso de Uso</b>	Gestión de Semestres
<b>Actor</b>	Administrador
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá gestionar la información por semestre académico. Cada semestre tendrá un estado (activo o cerrado), y todos los portafolios, asignaciones y evaluaciones estarán vinculados a un semestre específico.

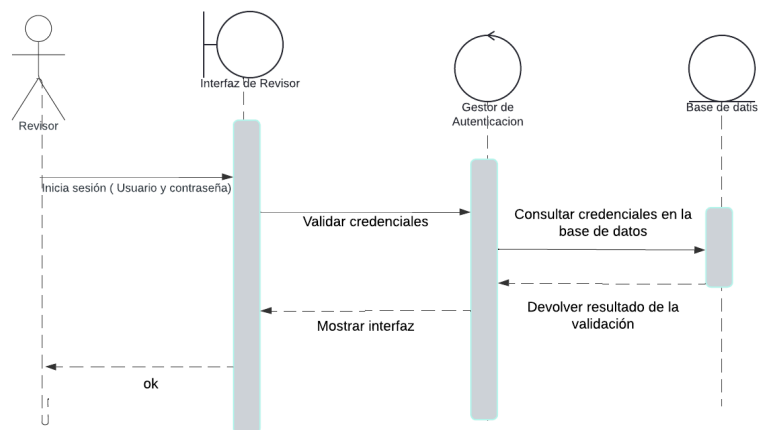
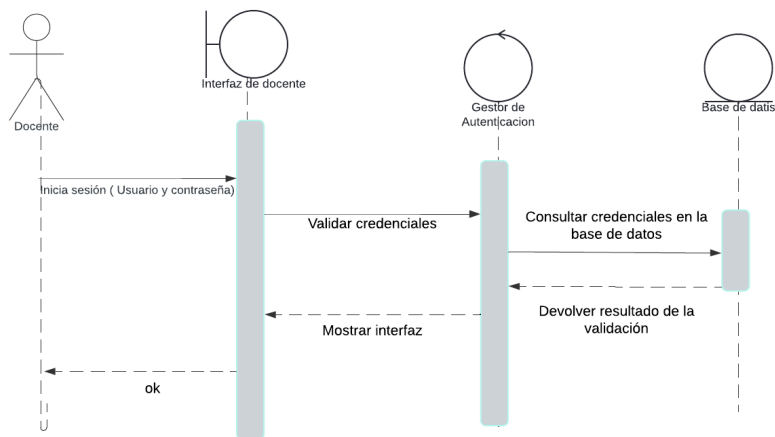
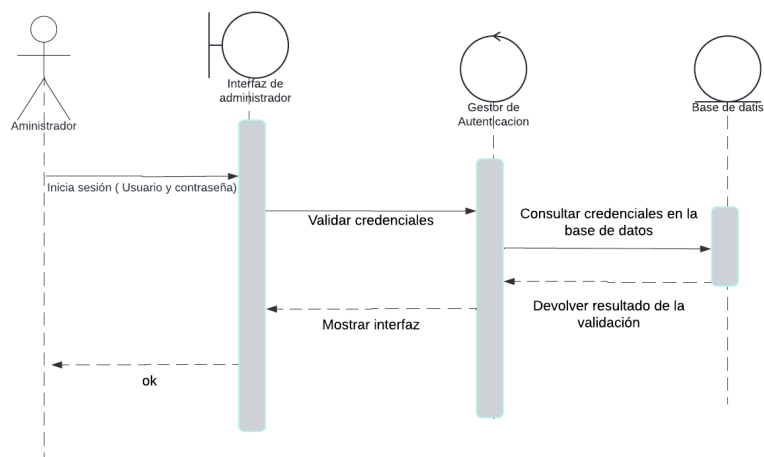
### Diagrama de caso de uso 10

<b>CUI</b>	<b>CU10</b>
<b>Nombre del Caso de Uso</b>	Historial de Revisiones
<b>Actor</b>	Administrador, Revisor
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá a los administradores y revisores consultar el historial de revisiones de los portafolios. Este historial incluirá la fecha de cada revisión, el estado de la evaluación (completo, observado, incompleto) y las observaciones realizadas por los revisores. El usuario podrá acceder a esta información en cualquier momento para verificar los cambios y las evaluaciones anteriores de los portafolios.

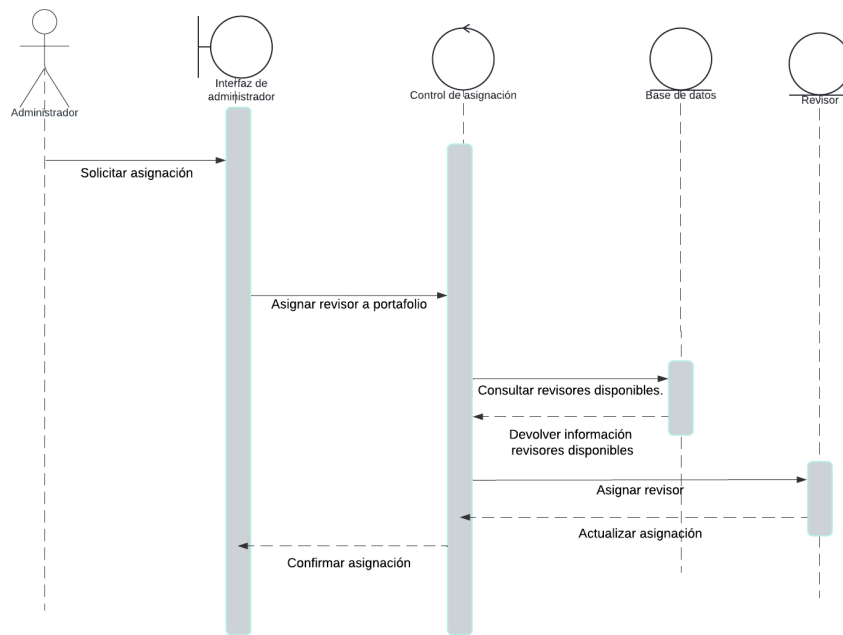
### Diagrama de caso de uso 11

<b>CUI</b>	<b>CU11</b>
<b>Nombre del Caso de Uso</b>	Acceso a Roles Múltiples
<b>Actor</b>	Revisor, Administrador
<b>Tipo</b>	Primario
<b>Descripción</b>	El sistema permitirá que los revisores y administradores también accedan como docentes, gestionando su propio portafolio.

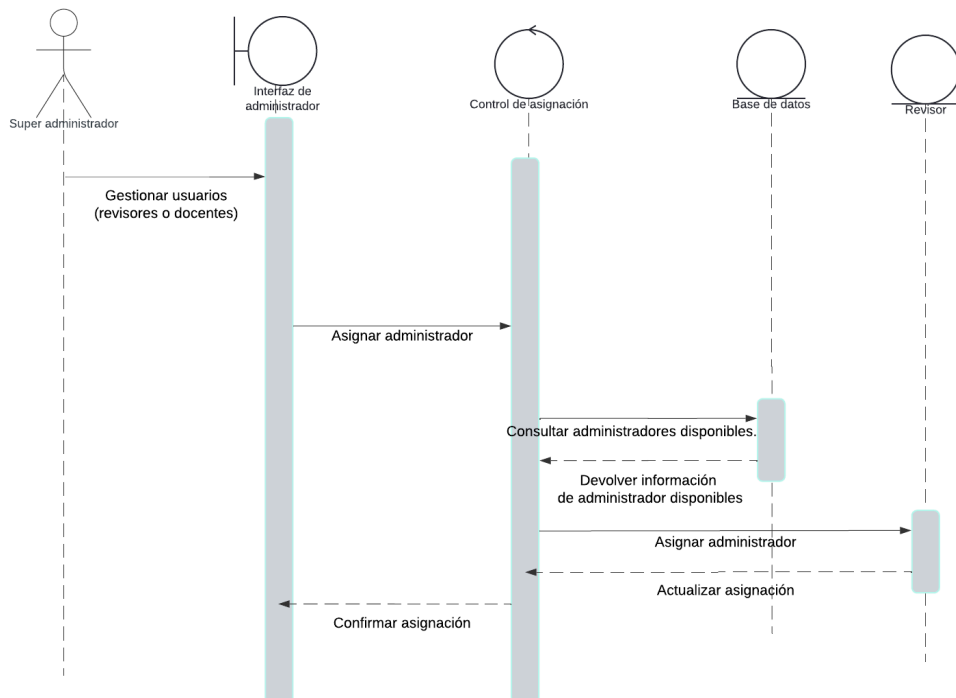
### 6.3.2 Diagrama Secuenciales



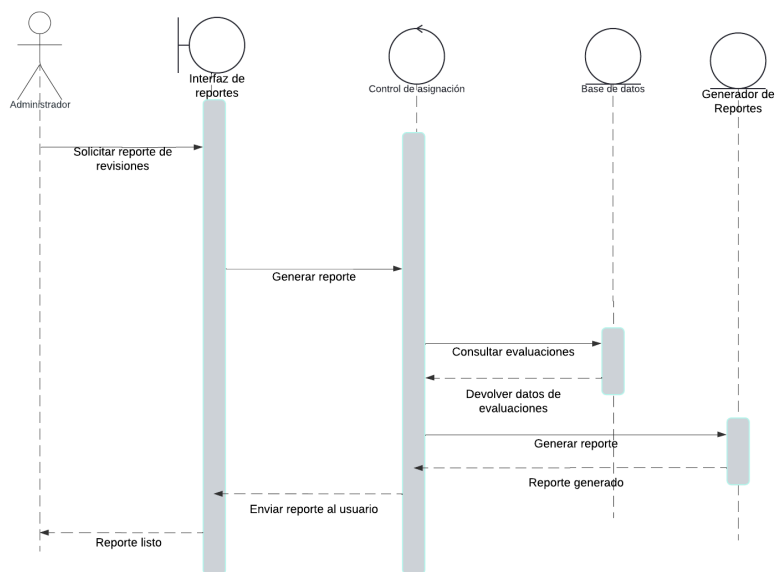
## DS1 Autenticación de usuarios



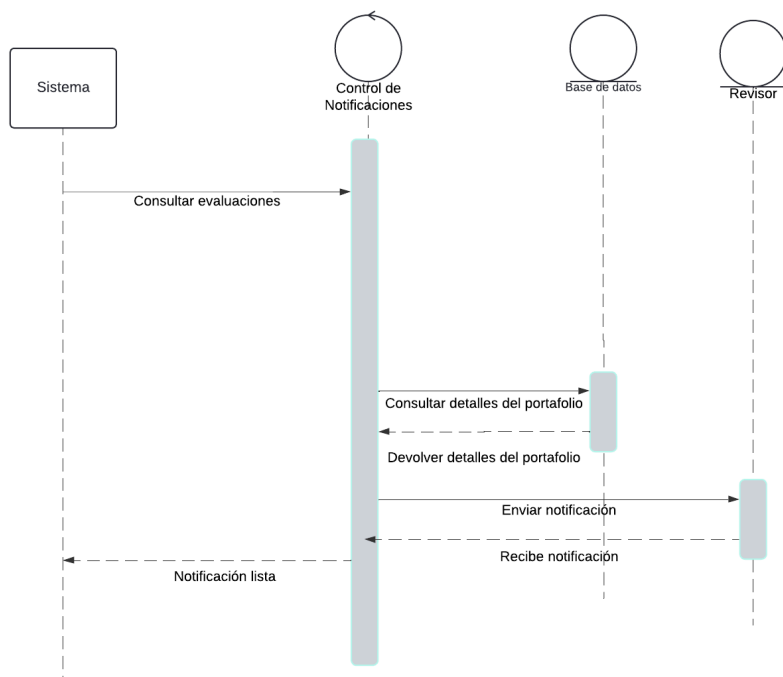
## DS2 Asignación de revisores



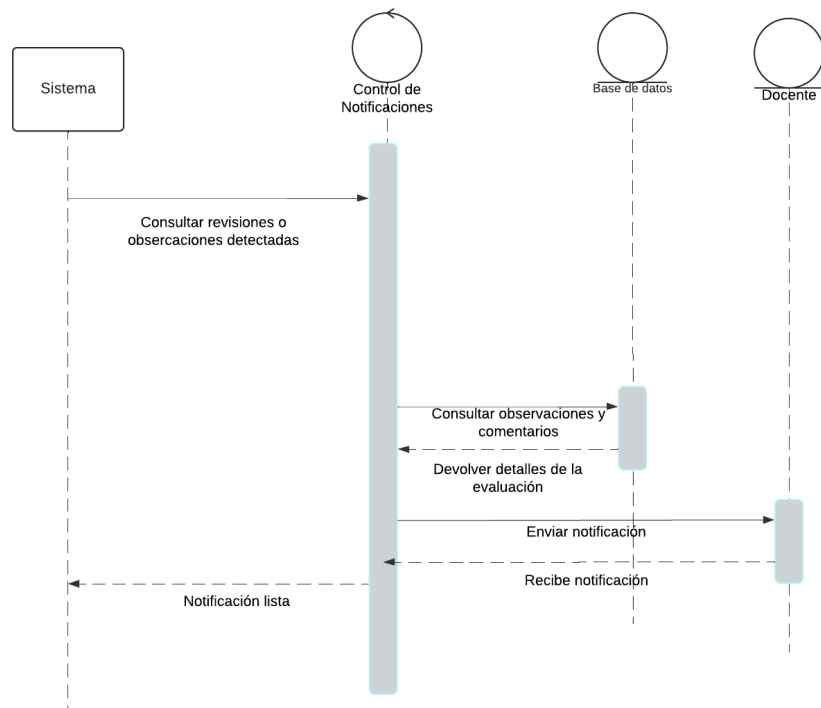
## DS3 Asignación de revisores



## DS4 Generación de reportes

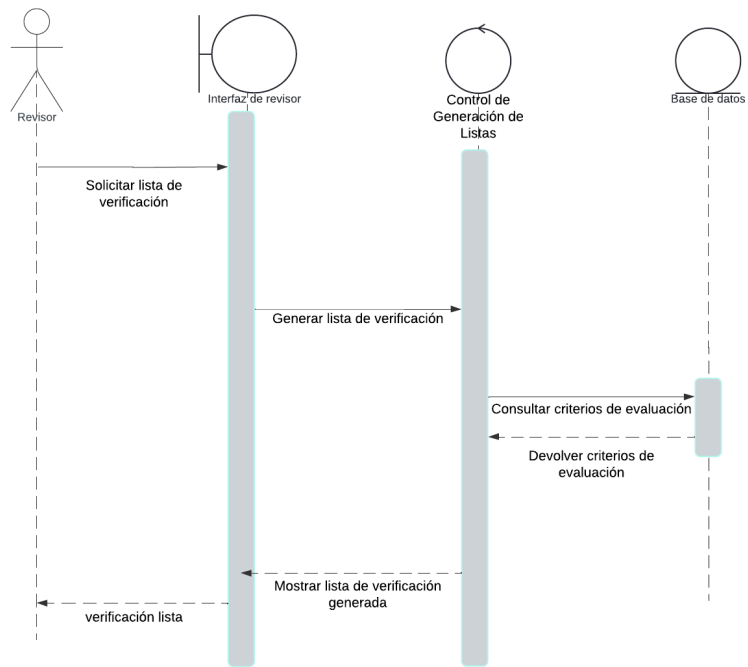


## DS5 Notificación a revisores

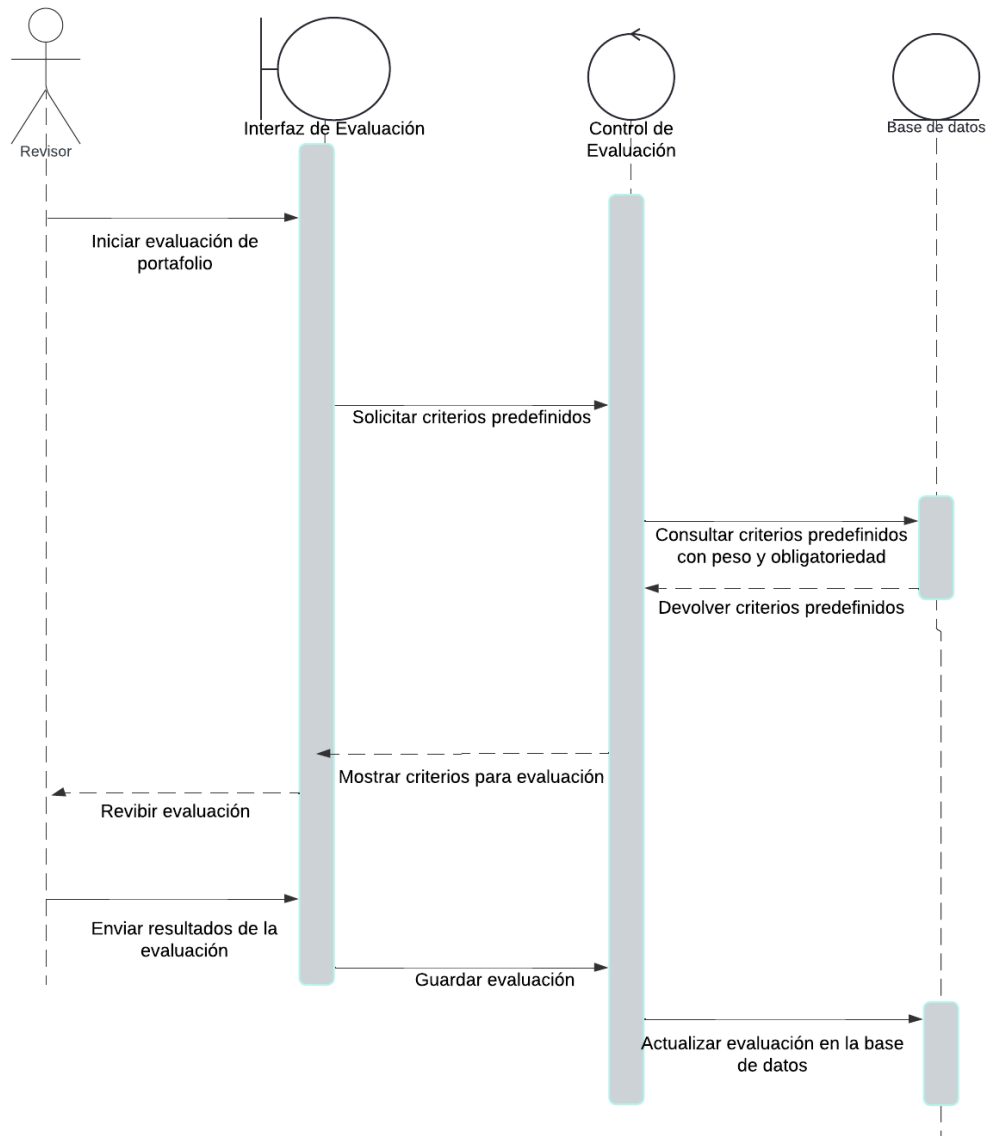


**DS6 Notificación a docentes**

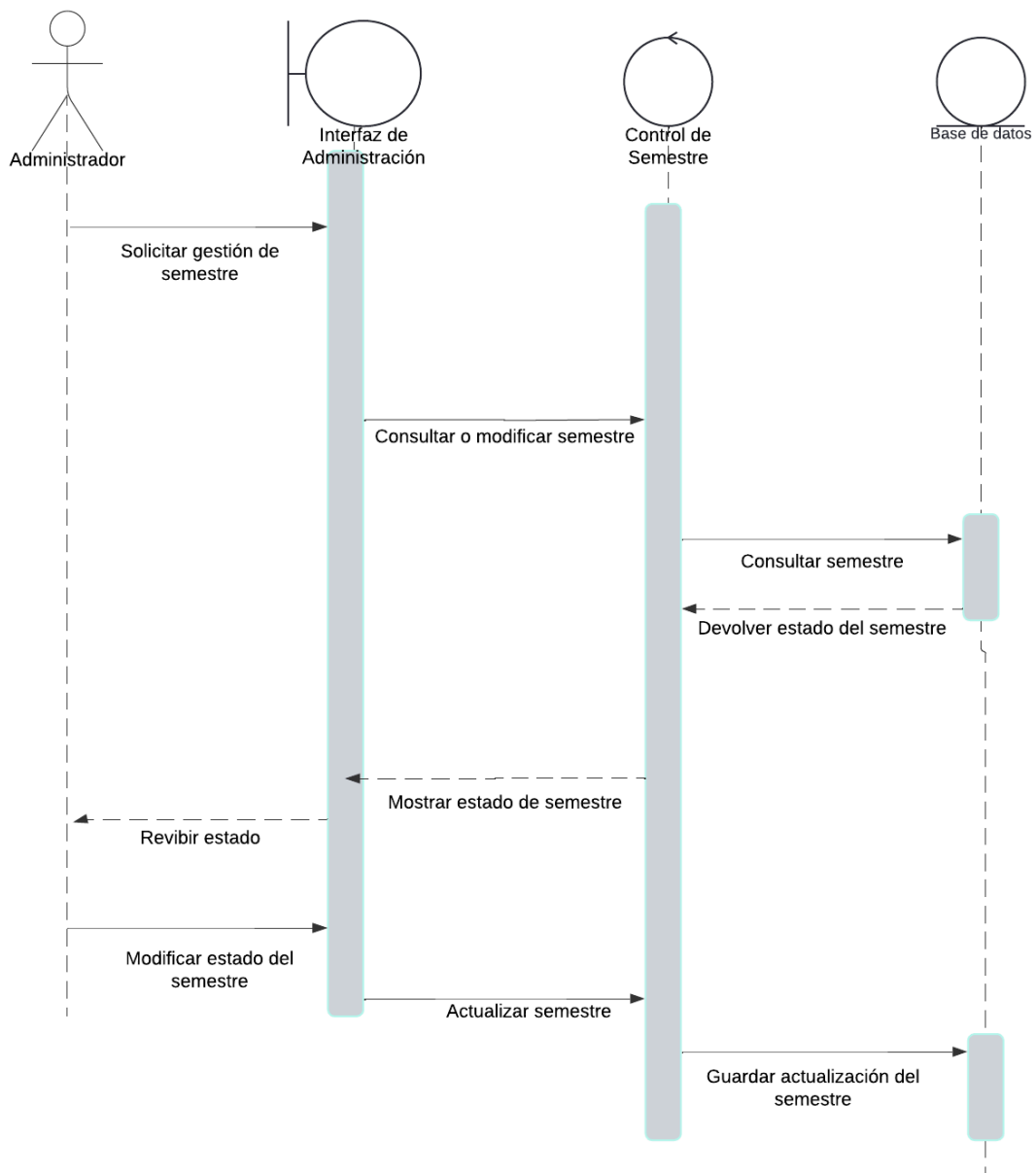




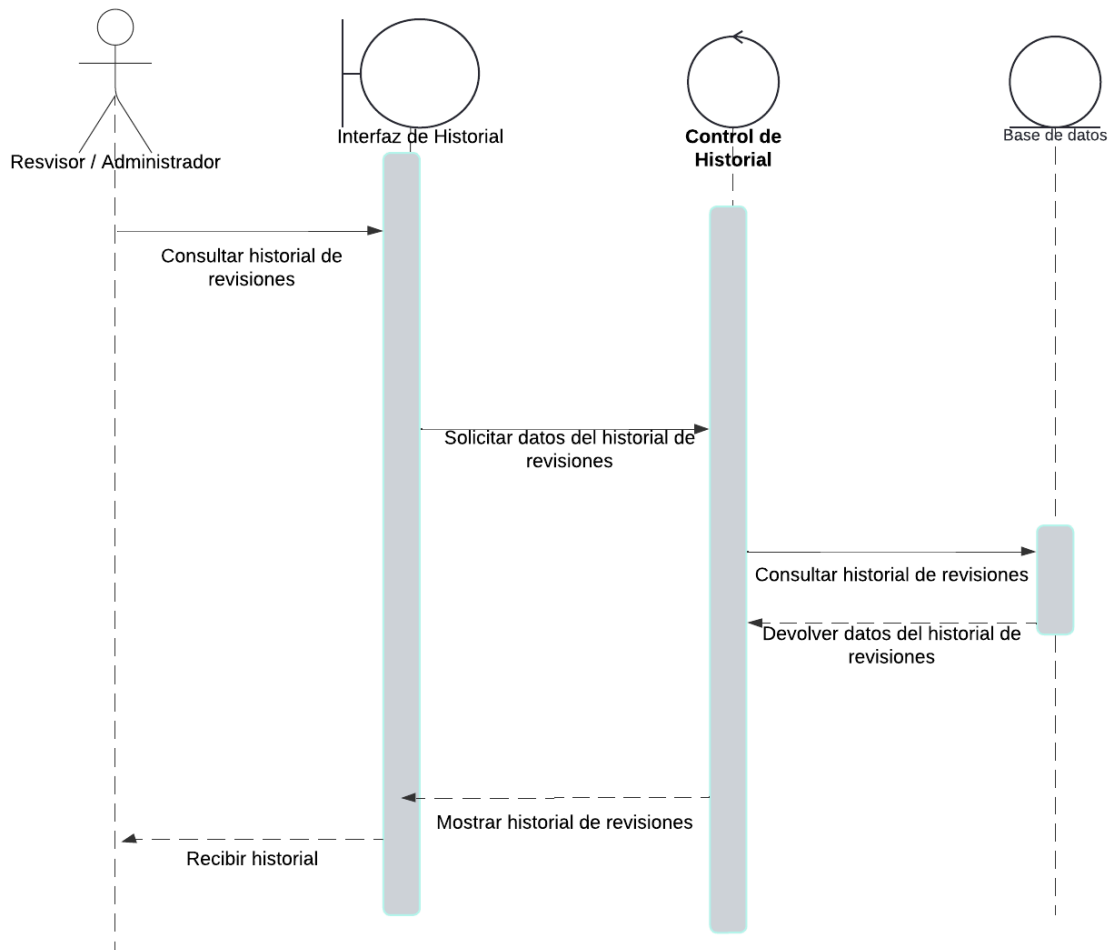
## DS7 Criterios de evaluación



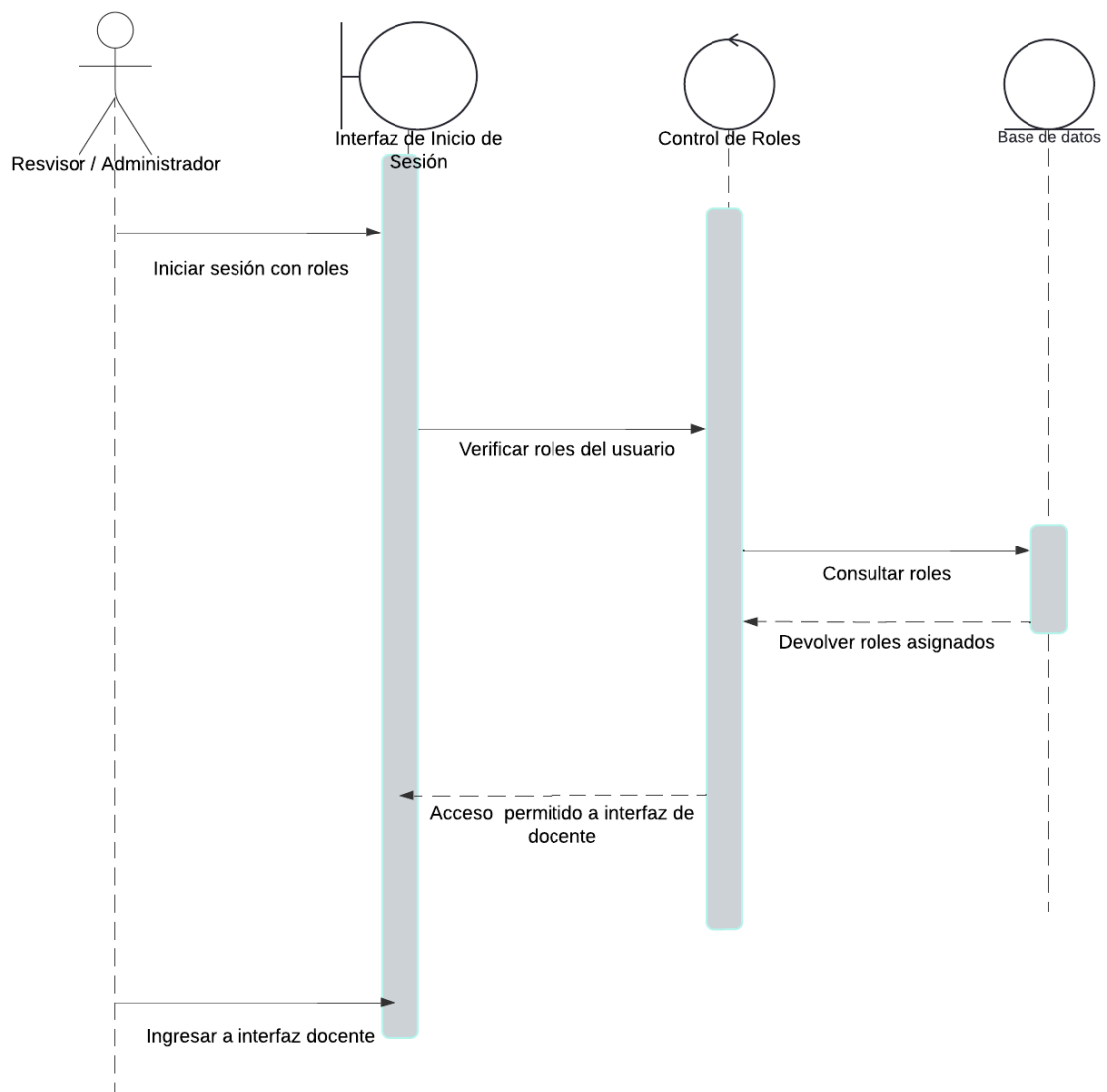
## DS8 Control de evaluaciones



**DS9 Gestión de semestres**



## DS10 Historial



## DS11 Roles

## 6.1. Descripción de la metodología ágil

### Sprint 0: Planificación del proyecto

**Inicio:** 01/09/2024

**Fin:** 15/09/2024

Historia de usuario H01: Organización del Proyecto en Azure DevOps	
Como Scrum Master, quiero tener la idea clara del proyecto y los integrantes del equipo organizados en azure DevOps, para tener una mayor coordinación en el trabajo y realizar tareas en conjunto, así como una mejor documentación.	
<i>Estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>

Historia de usuario H02 : Monitoreo del Avance del Proyecto	
Como Product Owner, quiero estar actualizado en todo lo que se avanza en el proyecto, para poder tener la certeza de que el proyecto se está avanzando en los tiempos previstos.	

<i>estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>

Historia de usuario H03:Uso de Laragon en el Proyecto	
<b>Como Desarrollador full stack, quiero poder usar Laragon, para que nadie en el equipo tenga problemas a la hora de adaptarse al trabajo.</b>	
<i>estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>

Historia de usuario H04: Creación de IU del Proyecto	
<b>Como desarrollador Frontend, quiero usar html, css, javascript y cierta plantilla(puede agrupar algunos lenguajes adicionales), para poder crear la parte visual del trabajo</b>	
<i>estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>

Historia de usuario H05: Desarrollo de la Lógica y Conexiones de Base de Datos	
Como desarrollador BackEnd, quiero usar las herramientas MySQL y php, para poder desarrollar la parte de las relaciones entre diferentes tablas en la base de datos del programa	
<i>estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>

Historia de usuario H06 : Uso de Adobe XD para Diseño UX/UI	
Como diseñador UX/UI quiero usar el software de Adobe XD, puesto que es una herramienta utilizada por una gran mayoría de diseñadores. Aparte de ello, no requiere alto grado de conocimiento de diseño.	
<i>estimación (horas):</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>



**Sprint 1: Presentación (1ra parte) del proyecto**

**Inicio: 16/09/2024**

**Fin: 23/09/2024**

<b>Historia de usuario H07: DOCUMENTACIÓN</b>	
<b>Como Product owner quiero tener completo la primera parte de entrega del proyecto</b>	
<i>estimación (horas):</i>	<i>10</i>
<i>Dependencia</i>	<i>ninguna</i>

<b>Tarea T01: Introducción</b>	
<b>Realizar la introducción de la documentación de manera escrita. Teniendo en cuenta la de los objetivos primordiales a las cuales esta enlazada el proyecto</b>	
<i>Estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>ninguna</i>
<i>Historia de usuario</i>	<i>H07</i>

Tarea T02: Descripción del grupo de trabajo	
Realizar la descripción del grupo de trabajo. Con la intención de tener más clara los roles que tendrá cada persona, puesto que es necesario para designar las diferentes tareas.	
<i>estimación (horas)</i>	<i>1</i>

<i>Dependencia</i>	<i>T01</i>
<i>Historia de usuario</i>	<i>H07</i>

Tarea T03: descripción del proceso	
Realizar la descripción del proceso, de cómo funciona actualmente la revisión de portafolios, usando BPMN puesto que permite el modelado de procesos de forma mas ordenada.	
<i>estimación (horas)</i>	<i>2</i>
<i>Dependencia</i>	<i>T02</i>
<i>Historia de usuario</i>	<i>H07</i>

<b>Tarea T04: <i>problemas del proceso actual</i></b>	
<b>Realizar la descripción escrita de los problemas, que tiene el proceso actual de la revisión de portafolios. Especificar cada uno de estos puntos.</b>	
<b><i>estimación (horas)</i></b>	<b><i>1</i></b>
<b><i>Dependencia</i></b>	<b><i>T03</i></b>
<b><i>Historia de usuario</i></b>	<b><i>H07</i></b>

<b>Tarea T05: <i>justificación</i></b>	
<b>Realizar de manera escrita la justificación del proyecto actual, del porqué es importante desarrollar el proyecto.</b>	
<b><i>estimación (horas)</i></b>	<b><i>1</i></b>
<b><i>Dependencia</i></b>	<b><i>T04</i></b>
<b><i>Historia de usuario</i></b>	<b><i>H07</i></b>

<b>Tarea T06: <i>Factibilidad y viabilidad</i></b>	
<b>Realizar de manera escrita la sección de factibilidad y viabilidad, después de la justificación.</b>	
<i>estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>T05</i>
<i>Historia de usuario</i>	<i>H07</i>

<b>Tarea T07: <i>Contrato</i></b>	
<b>Realizar el contrato entre la directora de escuela profesional y un representante de la compañía encargada de realizar el desarrollo del software. Esto va después de la factibilidad y viabilidad.</b>	
<i>estimación (horas)</i>	<i>2</i>
<i>Dependencia</i>	<i>T06</i>
<i>Historia de usuario</i>	<i>H07</i>

Tarea T08: Reportes	
Realizar la descripción escrita de los reportes de podrá generar el software. Esto va después del contrato.	
<i>estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>T06</i>
<i>Historia de usuario</i>	<i>H07</i>

Historia de usuario H08: Revision y correccion	
Como Product Owner quiero que se realicen las correcciones explicadas en la exposición, para su debida calificación del avance del proyecto.	
<i>estimación (horas):</i>	<i>2</i>
<i>Prioridad:</i>	<i>13</i>
<i>Dependencia</i>	<i>H07</i>

## Sprint 2: Presentación (2da parte) del proyecto

**Inicio:** 25/09/2024

**Fin:** 24/10/2024

Historia de usuario H09: Documentación complementaria	
Como Product Owner quiero que se realice todos los criterios correspondientes a la segunda entrega del proyecto	
<i>estimación (horas):</i>	13
<i>Dependencia</i>	H08

Tarea T09: Documento de Presentación	
Realizar el documento de presentación donde se describe el proyecto y se resumen las metodologías, los pasos seguidos y las recomendaciones hechas para implementar el proyecto.	
<i>estimación (horas)</i>	2
<i>Dependencia</i>	T08
<i>Historia de usuario</i>	H09

Tarea T10: Documento de Presentación	
Realizar el organigrama del proyecto, describiendo sus partes esenciales para incluirlo en la documentación final.	
<i>estimación (horas)</i>	<i>1</i>

<i>Dependencia</i>	<i>T09</i>
<i>Historia de usuario</i>	<i>H09</i>

Tarea T11: Plan del proyecto	
Elaborar el plan del proyecto, creando un cronograma detallado que incluya las tareas a realizar en cada sprint, reflejando el avance del proyecto en cada fase.	
<i>estimación (horas)</i>	<i>2</i>
<i>Dependencia</i>	<i>T10</i>
<i>Historia de usuario</i>	<i>H09</i>

Tarea T12: <i>Glosario de términos</i>	
Elaborar un glosario de términos con definiciones de los términos utilizados a lo largo de la documentación con el fin de dar a entendimiento estos términos al usuario.	
<i>estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>T11</i>
<i>Historia de usuario</i>	<i>H09</i>

Tarea T13: <i>Requerimientos</i>	
Documentar los requerimientos funcionales y no funcionales del sistema, además de desarrollar la matriz de rastreo de requerimientos para asegurar que se cumplan los objetivos del proyecto.	
<i>estimación (horas)</i>	<i>3</i>
<i>Dependencia</i>	<i>T12</i>
<i>Historia de usuario</i>	<i>H09</i>



Tarea T14: Documentos de diseño	
Completar la documentación del diseño de sistema, así como la arquitectura, diagramas UML, Descripción de la metodología ágil, Modelo lógico y físico de la base de datos.	
<i>estimación (horas)</i>	4
<i>Dependencia</i>	T13
<i>Historia de usuario</i>	H09

### Sprint 3: Codificación y Entrega del Proyecto

**Inicio: 20/10/2024**

**Fin: 19/11/2024**

Historia de usuario H10: CODIFICACIÓN Y PRUEBAS	
Como Product Owner, quiero que se realice toda la codificación final del sistema, tomando en cuenta todos los puntos desarrollados hasta el momento.	
<i>estimación (horas):</i>	14
<i>Dependencia</i>	H09

<b>Tarea T15: <i>Implementación general de código</i></b>	
Realizar la implementación del código, asegurando que cumpla con todos los puntos desarrollados y esté correctamente documentado.	
<i>estimación (horas)</i>	6

<i>Dependencia</i>	<i>T14</i>
<i>Historia de usuario</i>	<i>H10</i>

<b>Tarea T16: <i>Pruebas finales</i></b>	
Llevar a cabo pruebas funcionales y de rendimiento para verificar que el sistema funcione según los requerimientos declarados.	
<i>estimación (horas)</i>	3
<i>Dependencia</i>	<i>T15</i>
<i>Historia de usuario</i>	<i>H10</i>

Tarea T17: <i>Documentación de pruebas</i>	
Elaborar un informe de las pruebas realizadas, detallando los casos de prueba, los resultados obtenidos y cualquier hallazgo relevante.	
<i>estimación (horas)</i>	2
<i>Dependencia</i>	T16
<i>Historia de usuario</i>	H10

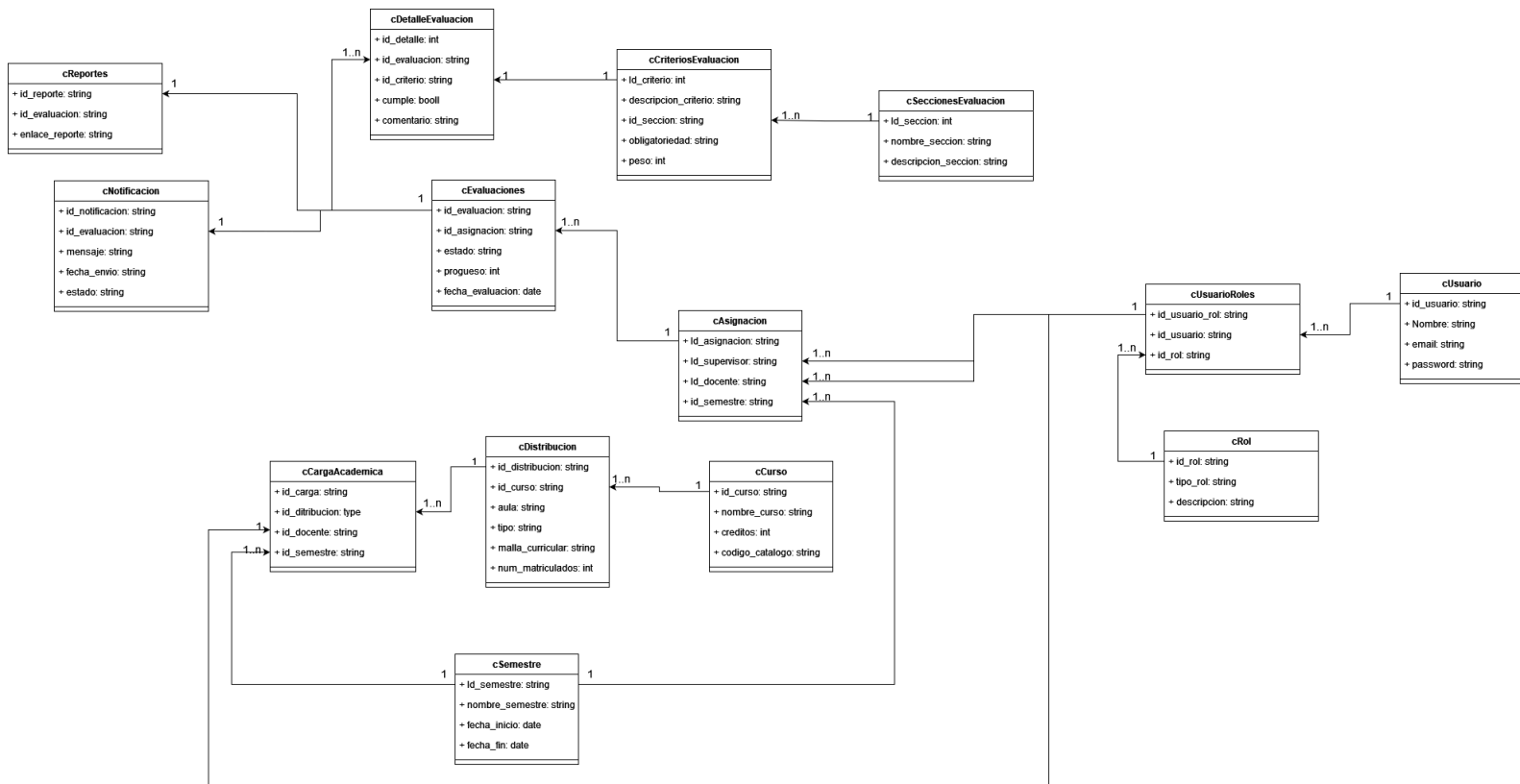
Tarea T18: <i>Preparación de la entrega</i>	
Preparar toda la documentación y los archivos necesarios para la entrega final del proyecto, incluyendo los código fuente y el respectivo manual de usuario.	
<i>estimación (horas)</i>	2
<i>Dependencia</i>	T17
<i>Historia de usuario</i>	H10

<b>Tarea T19: Exposición del proyecto</b>	
<b>Realizar la exposición del proyecto(previa) dando a conocer detalladamente todos los puntos</b>	
<i>estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>T18</i>
<i>Historia de usuario</i>	<i>H10</i>

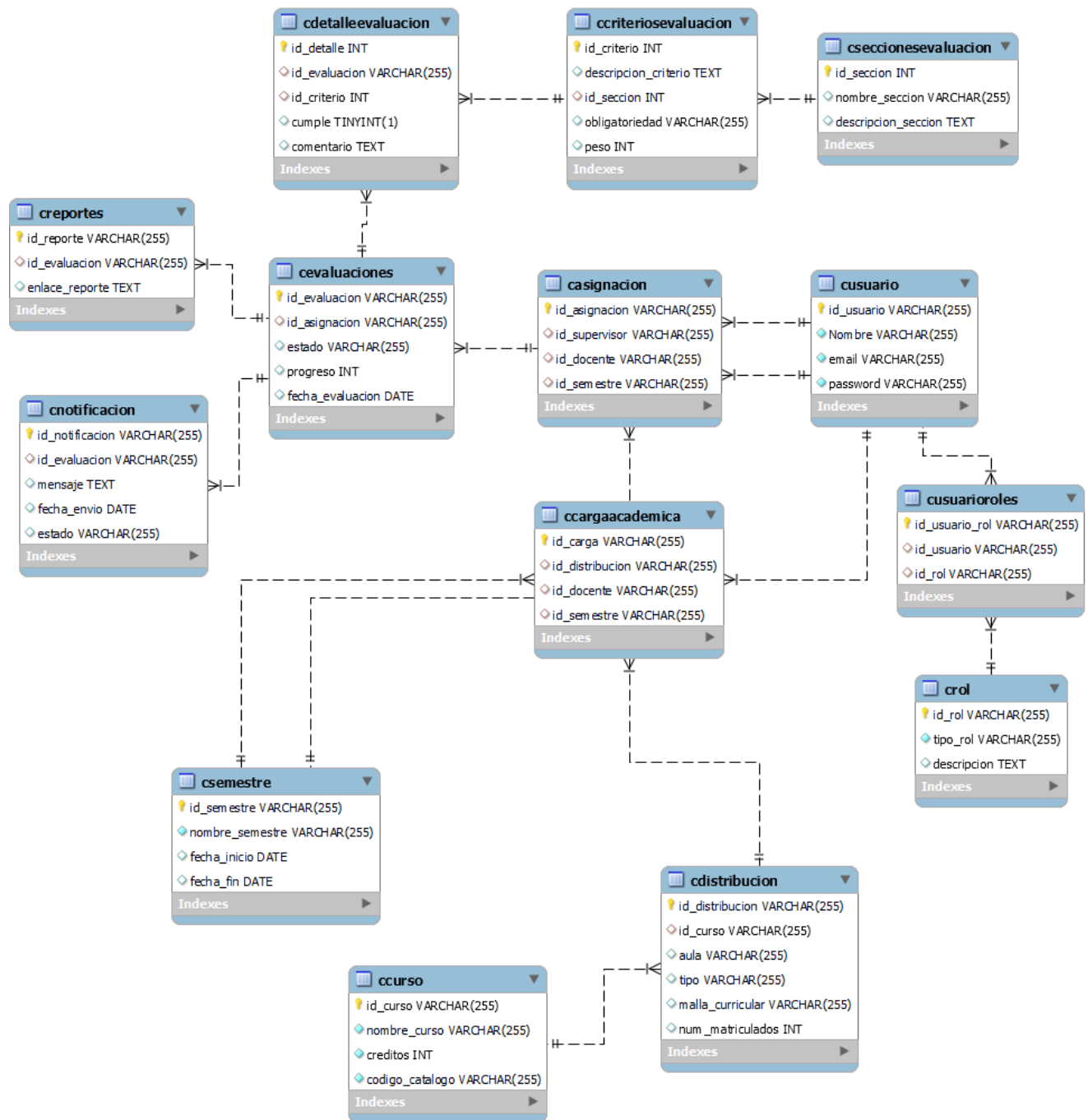
<b>Historia de usuario H11: Documentación de pruebas y Mantenimiento</b>	
<b>Realizar la documentación de acuerdo a las pruebas realizadas , también generar el manual y un video adicional para un mejor uso del sistema</b>	
<i>estimación (horas)</i>	<i>3</i>
<i>Dependencia</i>	<i>T18</i>
<i>Historia de usuario</i>	<i>H10</i>

<b>Historia de usuario H12: <i>PRESENTACIÓN FINAL DEL PROYECTO</i></b>	
<b>Realizar la presentación oficial del proyecto detallando los resultados alcanzados y la eficiencia de esta</b>	
<i>estimación (horas)</i>	<i>1</i>
<i>Dependencia</i>	<i>H11</i>

## 6.4. Modelo lógico de la base de datos



## **6.5. Modelo físico de la base de datos**



Script de base de datos



```

1      -- Crear la base de datos
2      ● CREATE DATABASE bdPortafolioDoc;
3      ● USE bdPortafolioDoc;
4
5      -- Tabla cUsuario
6      ● ○ CREATE TABLE cUsuario (
7          id_usuario VARCHAR(255) PRIMARY KEY,
8          Nombre VARCHAR(255) NOT NULL,
9          email VARCHAR(255) UNIQUE NOT NULL,
10         password VARCHAR(255) NOT NULL
11     );
12
13     -- Tabla cRol
14     ● ○ CREATE TABLE cRol (
15         id_rol VARCHAR(255) PRIMARY KEY,
16         tipo_rol VARCHAR(255) NOT NULL,
17         descripcion TEXT
18     );
19
20     -- Tabla cUsuarioRoles
21     ● ○ CREATE TABLE cUsuarioRoles (
22         id_usuario_rol VARCHAR(255) PRIMARY KEY,
23         id_usuario VARCHAR(255),
24         id_rol VARCHAR(255),
25         FOREIGN KEY (id_usuario) REFERENCES cUsuario(id_usuario) ,
26         FOREIGN KEY (id_rol) REFERENCES cRol(id_rol) ON DELETE CA
27     );
28
29     -- Tabla cSemestre
30     ● ○ CREATE TABLE cSemestre (
31         id_semestre VARCHAR(255) PRIMARY KEY,
32         nombre_semestre VARCHAR(255) NOT NULL,
33         fecha_inicio DATE,
34         fecha_fin DATE
35     );
36
37     -- Tabla cAsignacion
38     ● ○ CREATE TABLE cAsignacion (
39         id_asignacion VARCHAR(255) PRIMARY KEY,
40         id_supervisor VARCHAR(255),
41         id_docente VARCHAR(255),
42         id_semestre VARCHAR(255),
43         FOREIGN KEY (id_supervisor) REFERENCES cUsuario(id_usuario) ON DE
44         FOREIGN KEY (id_docente) REFERENCES cUsuario(id_usuario) ON DELE
45         FOREIGN KEY (id_semestre) REFERENCES cSemestre(id_semestre) ON DE
46     );
47
48     -- Tabla cCurso
49     ● ○ CREATE TABLE cCurso (
50         id_curso VARCHAR(255) PRIMARY KEY,
51         nombre_curso VARCHAR(255) NOT NULL,
52         creditos INT NOT NULL,
53         codigo_catalogo VARCHAR(255) NOT NULL
54     );
55

```

```

88      -- Tabla cNotificacion
89      ● ○ CREATE TABLE cNotificacion (
90          id_notificacion VARCHAR(255) PRIMARY KEY,
91          id_evaluacion VARCHAR(255),
92          mensaje TEXT,
93          fecha_envio DATE,
94          estado VARCHAR(255),
95          FOREIGN KEY (id_evaluacion) REFERENCES cEvaluaciones(id_evaluacion) ON
96      );
97
98      -- Tabla cReportes
99      ● ○ CREATE TABLE cReportes (
100          id_reporte VARCHAR(255) PRIMARY KEY,
101          id_evaluacion VARCHAR(255),
102          enlace_reporte TEXT,
103          FOREIGN KEY (id_evaluacion) REFERENCES cEvaluaciones(id_evaluacion) ON
104      );
105
106      -- Tabla cSeccionesEvaluacion
107      ● ○ CREATE TABLE cSeccionesEvaluacion (
108          id_seccion INT PRIMARY KEY AUTO_INCREMENT,
109          nombre_seccion VARCHAR(255),
110          descripcion_seccion TEXT
111      );
112
113      -- Tabla cCriteriosEvaluacion
114      ● ○ CREATE TABLE cCriteriosEvaluacion (
115          id_criterio INT PRIMARY KEY AUTO_INCREMENT,
116          descripcion_criterio TEXT,
117          id_seccion INT,
118          obligatoriedad VARCHAR(255),

```

```

106      -- Tabla cSeccionesEvaluacion
107      ● ○ CREATE TABLE cSeccionesEvaluacion (
108          id_seccion INT PRIMARY KEY AUTO_INCREMENT,
109          nombre_seccion VARCHAR(255),
110          descripcion_seccion TEXT
111      );
112
113      -- Tabla cCriteriosEvaluacion
114      ● ○ CREATE TABLE cCriteriosEvaluacion (
115          id_criterio INT PRIMARY KEY AUTO_INCREMENT,
116          descripcion_criterio TEXT,
117          id_seccion INT,
118          obligatoriedad VARCHAR(255),
119          peso INT,
120          FOREIGN KEY (id_seccion) REFERENCES cSeccionesEvaluacion(id_seccion) ON DEL
121      );
122
123      -- Tabla cDetalleEvaluacion
124      ● ○ CREATE TABLE cDetalleEvaluacion (
125          id_detalle INT PRIMARY KEY AUTO_INCREMENT,
126          id_evaluacion VARCHAR(255),
127          id_criterio INT,
128          cumple BOOLEAN,
129          comentario TEXT,
130          FOREIGN KEY (id_evaluacion) REFERENCES cEvaluaciones(id_evaluacion) ON DELE
131          FOREIGN KEY (id_criterio) REFERENCES cCriteriosEvaluacion(id_criterio) ON D
132      );
133
134      -- Tabla cDetalleEvaluacion

```

```

55
56      -- Tabla cDistribucion
57      CREATE TABLE cDistribucion (
58          id_distribucion VARCHAR(255) PRIMARY KEY,
59          id_curso VARCHAR(255),
60          aula VARCHAR(255),
61          tipo VARCHAR(255),
62          malla_curricular VARCHAR(255),
63          num_matriculados INT,
64          FOREIGN KEY (id_curso) REFERENCES cCurso(id_curso) ON DELETE CASCADE ON UPDAT
65      );
66
67      -- Tabla cCargaAcademica
68      CREATE TABLE cCargaAcademica (
69          id_carga VARCHAR(255) PRIMARY KEY,
70          id_distribucion VARCHAR(255),
71          id_docente VARCHAR(255),
72          id_semestre VARCHAR(255),
73          FOREIGN KEY (id_distribucion) REFERENCES cDistribucion(id_distribucion) ON DE
74          FOREIGN KEY (id_docente) REFERENCES cUsuario(id_usuario) ON DELETE CASCADE ON
75          FOREIGN KEY (id_semestre) REFERENCES cSemestre(id_semestre) ON DELETE CASCADE
76      );
77
78      -- Tabla cEvaluaciones
79      CREATE TABLE cEvaluaciones (
80          id_evaluacion VARCHAR(255) PRIMARY KEY,
81          id_asignacion VARCHAR(255),
82          estado VARCHAR(255),
83          progreso INT,
84          fecha_evaluacion DATE,
85          FOREIGN KEY (id_asignacion) REFERENCES cAsignacion(id_asignacion) ON DELETE C

```

