

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS



“Proyecto de Carga Académica Docente”

CURSO: INGENIERÍA DE SOFTWARE I

DOCENTE: William Zamalloa Paro

PRESENTADO POR:

- | | |
|--------------------------------------|--------|
| ● CALLAPIÑA RODRIGUEZ JOSUE CRISTIAN | 164235 |
| ● NAJOR JOSUE VALDEIGLESIAS DUEÑAS | 151830 |
| ● JOSE LUIS CRUZ CARRION | 170431 |
| ● CASILLA TTITO EVANDIR SAUL | 164238 |
| ● TRUJILLO TORBISCO LUIS ANDERSON | 083221 |
| ● LOAIZA MONROY BRUNO WALDIR | 153097 |
| ● PUMA MAMANI NILSON MAURIÑO | 151822 |
| ● HUAMAN QUISPE JEMY SANDRO | 194519 |
| ● ASECIO ARQUE JHOEL FELIZ | 163806 |

CUSCO – PERÚ

2023-2

Contenido

Presentación de proyecto.....	4
Organigrama del proyecto.	4
Plan del proyecto.	5
Glosario de términos propios de la implementación.....	1
Laravel.....	1
ORM (Object-Relational Mapping)	1
PHP	1
CRUD (Create, Read, Update, Delete)	1
MySQL.....	1
Servidor web.....	1
UML (Unified Modeling Language).....	1
Front-end	2
Back-end.....	2
Control de Versiones	2
Scrum	2
Escalabilidad.....	2
Framework.....	2
Vistas (Views).....	2
Documento de Requerimientos.....	3
Requerimientos Funcionales	3
Gestión de Usuarios:	3
Gestión de Portafolios:.....	3
Planificación y Asignación:	3
Generación de Informes:.....	3
Seguimiento de Carga Académica:	3
Requerimientos No Funcionales	3
Usabilidad:	3
Seguridad:	4
Escalabilidad:.....	4
Compatibilidad:	4
Matriz de Rastreo de los Requerimientos (Rastreabilidad)	1
Documentos de Diseño.....	1
Arquitectura funcional del sistema	1
Interfaz de usuario (capa de presentación)	1
Lógica de Negocio (Capa de Aplicación).....	1
Persistencia de Datos (Capa de Datos)	1

Arquitectura técnica del sistema.....	1
Topología de software	1
Topología del hardware	2
Documentos basados en UML.....	3
Casos de uso	3
Modelo de Objetos (pendiente).....	¡Error! Marcador no definido.
Diagramas secuenciales	5
Descripción de la metodología ágil utilizada de manera específica (sprints, historias, etc)	7
Roles en SCRUM:	7
Ceremonias de SCRUM:	8
Artefactos de SCRUM:.....	9
Modelo lógico de la base de datos.....	9

Presentación de proyecto

El presente proyecto de Gestión de Carga Académica Docente de la UNSAAC. En el entorno universitario, la eficiente asignación de responsabilidades docentes es esencial para garantizar una educación de calidad. Este proyecto tiene como objetivo principal mejorar la planificación y distribución de la carga académica, maximizando así el rendimiento y el bienestar de nuestros docentes.

Organigrama del proyecto.

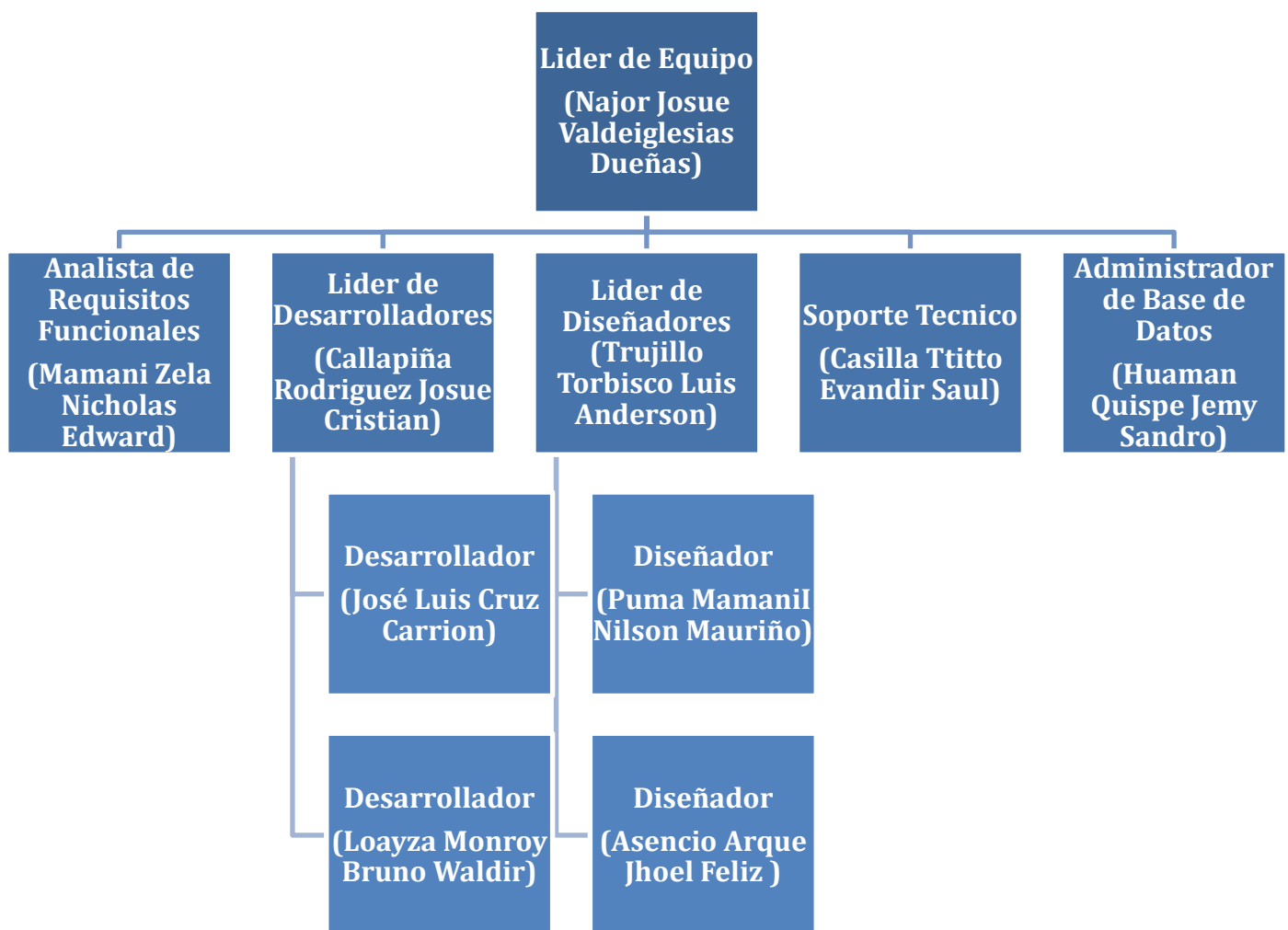


Ilustración 1 Vista del Organigrama del equipo de proyecto

Plan del proyecto.

Semanas 1-2: Definición y Planificación Inicial

Día 1-2: Reunión de inicio del proyecto, definir objetivos y alcance.

Día 3-4: Identificar requisitos del programa de carga académica.

Día 5-7: Desarrollar una estructura inicial del proyecto y asignar roles.

Semana 2: Revisar y ajustar el plan según comentarios del equipo.

Semanas 3-4: Diseño y Arquitectura

Días 1-5: Diseñar la arquitectura del programa.

Días 6-10: Crear prototipos de las interfaces de usuario.

Días 11-12: Revisión y ajuste del diseño.

Días 13-14: Preparar documentación de diseño.

Semanas 5-8: Desarrollo

Días 1-4: Configurar el entorno de desarrollo.

Días 5-9: Desarrollar módulos básicos del sistema.

Días 10-12: Integrar módulos y realizar pruebas preliminares.

Días 13-16: Refinar y ajustar funcionalidades.

Semanas 9-12: Pruebas y Mejoras

Días 1-5: Realizar pruebas exhaustivas del sistema.

Días 6-9: Corregir errores y optimizar el rendimiento.

Días 10-12: Preparar la documentación del usuario.

Días 13-14: Entrenamiento del personal y prueba de aceptación del usuario.

Días 15-16: Preparar para la implementación.

Semanas 13-16: Implementación y Entrega

Días 1-2: Implementar el programa de carga académica en el entorno de producción.

Días 3-7: Monitoreo y ajuste de la implementación implementación.

Días 8-14: Preparar y entregar la documentación final del proyecto.

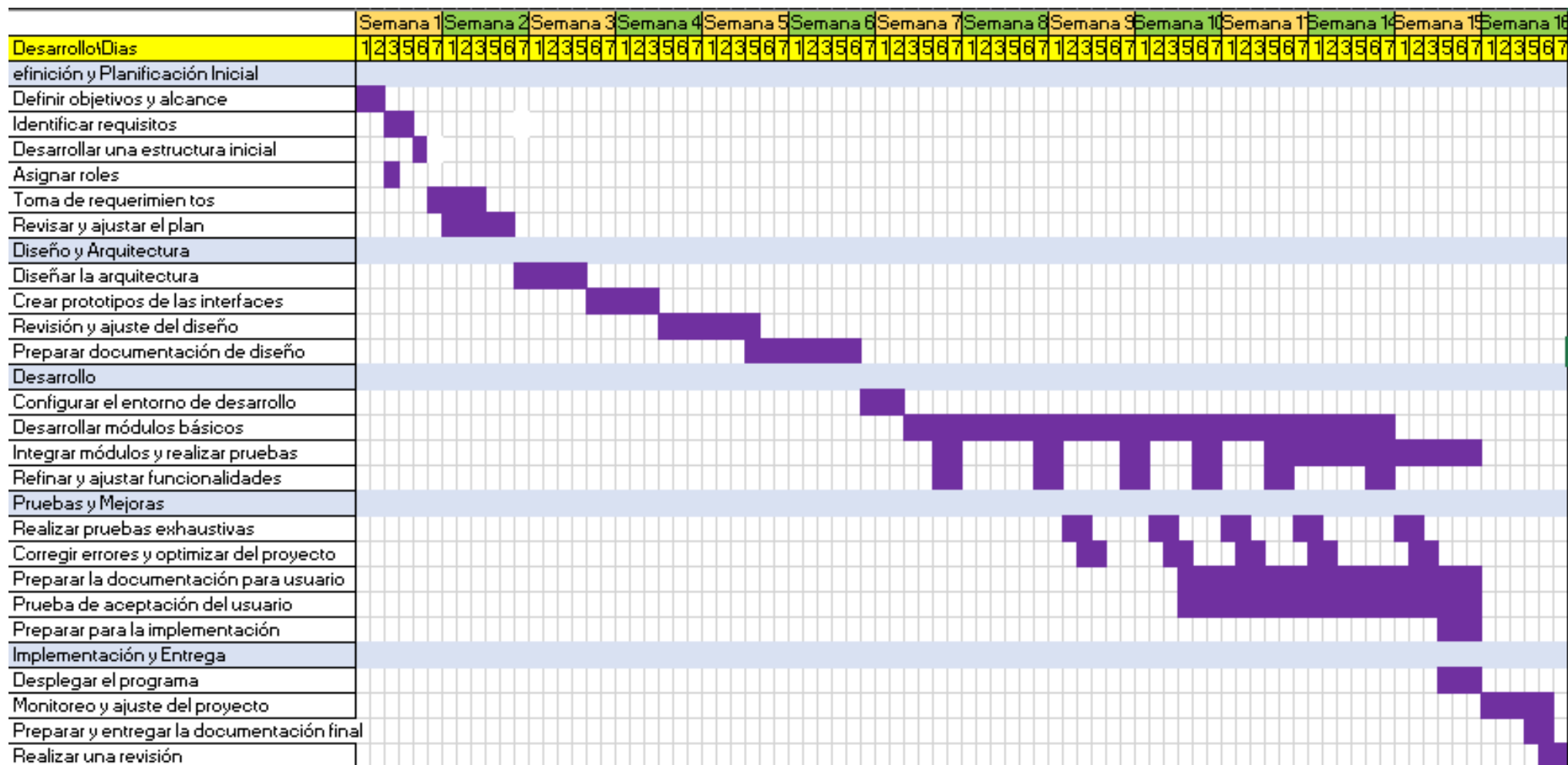


Ilustración 2 Vista del cronograma de Grantt para la vista de tareas de proyecto de evaluación docente

Glosario de términos propios de la implementación

Laravel

Framework de desarrollo web en PHP que sigue el patrón de diseño MVC (Modelo-Vista-Controlador). Ofrece herramientas y funciones para agilizar el proceso de desarrollo web.

ORM (Object-Relational Mapping)

Abstracción que permite manipular datos en la base de datos mediante objetos y clases en lugar de escribir consultas SQL directamente.

PHP

Lenguaje de programación ampliamente utilizado en el desarrollo web. Es especialmente adecuado para crear aplicaciones dinámicas y sitios interactivos, con una comunidad activa y herramientas diversas.

CRUD (Create, Read, Update, Delete)

Acrónimo que se refiere a las operaciones básicas de manipulación de datos en un sistema de gestión de bases de datos o en la persistencia de datos en general.

MySQL

Sistema de gestión de bases de datos relacional de código abierto, utilizado para almacenar, organizar y gestionar datos. Ampliamente usado en aplicaciones web.

Servidor web

Software que se ejecuta en un servidor y maneja las solicitudes de los clientes (navegadores web) enviando recursos solicitados como páginas web, imágenes, archivos, etc. Ejemplos: Apache, Nginx, Microsoft IIS.

UML (Unified Modeling Language)

Lenguaje estándar para modelar sistemas de software. Proporciona notaciones y diagramas que permiten visualizar, especificar, construir y documentar los artefactos de un sistema.

Front-end

La parte del desarrollo web que se centra en la interfaz y la experiencia del usuario. Incluye tecnologías como HTML, CSS y JavaScript para crear la interfaz visual con la que interactúan los usuarios.

Back-end

La parte del desarrollo web que se ocupa de la lógica del servidor y la gestión de datos. Involucra tecnologías como servidores web, bases de datos y aplicaciones que se ejecutan en el servidor.

Control de Versiones

Sistema que registra cambios en un conjunto de archivos a lo largo del tiempo. Git es un ejemplo común. Facilita el trabajo colaborativo, el seguimiento de cambios y la reversión a versiones anteriores.

Scrum

Metodología ágil de gestión de proyectos que se centra en la colaboración, la adaptabilidad y la entrega incremental. Incluye roles como el Scrum Master, el Product Owner y el equipo de desarrollo.

Escalabilidad

Capacidad de un sistema para manejar un aumento en la carga o la demanda sin sacrificar el rendimiento. Puede referirse tanto a la escalabilidad horizontal (añadir más servidores) como a la vertical (mejorar el rendimiento de un solo servidor).

Framework

Conjunto de herramientas, bibliotecas y convenciones que proporciona una estructura para el desarrollo de software. Laravel, mencionado anteriormente, es un ejemplo de framework.

Vistas (Views)

Representación virtual de los datos almacenados en una o más tablas. Las vistas permiten simplificar consultas complejas y proporcionar una capa de abstracción sobre la estructura de la base de datos.

Documento de Requerimientos

Requerimientos Funcionales

Gestión de Usuarios:

- **RF1:** El sistema debe permitir el registro de usuarios con roles específicos: docentes, revisores, presidente de la comisión, y jefe de departamento.
- **RF2:** Los usuarios deben poder iniciar sesión con sus credenciales.

Gestión de Portafolios:

- **RF3:** Los docentes deben poder cargar sus portafolios, seleccionando el formato correspondiente (teórico o práctico).
- **RF4:** El sistema debe permitir la revisión de documentos dentro del portafolio por parte de la comisión revisora.
- **RF5:** Se debe notificar a los docentes sobre observaciones o faltas encontradas durante la revisión.

Planificación y Asignación:

- **RF6:** El sistema debe permitir la planificación de las fechas de revisión.
- **RF7:** Asignar docentes a revisores y especificar roles dentro de la comisión.

Generación de Informes:

- **RF8:** El sistema debe generar informes al concluir cada revisión, indicando el cumplimiento o no según los resultados de la revisión.
- **RF9:** Los informes deben ser enviados automáticamente al jefe de departamento.

Seguimiento de Carga Académica:

- **RF10:** El sistema debe llevar un registro de la carga académica de cada docente en cada semestre.
- **RF11:** Permitir la actualización de la carga académica para reflejar cambios semestrales.

Requerimientos No Funcionales

Usabilidad:

- **RNF1:** La interfaz debe ser intuitiva y fácil de usar para los distintos roles de usuarios.

- **RNF2:** La respuesta del sistema debe ser rápida, manteniendo tiempos de carga mínimos.

Seguridad:

- **RNF3:** La información del portafolio y los informes debe ser confidencial y accesible solo para usuarios autorizados.
- **RNF4:** Se debe implementar un sistema robusto de autenticación y autorización.

Escalabilidad:

- **RNF5:** El sistema debe ser escalable para manejar un aumento en el número de usuarios y documentos.

Compatibilidad:

- **RNF6:** El sistema debe ser compatible con los navegadores web más comunes.

Matriz de Rastreo de los Requerimientos (Rastreabilidad)

ID	Descripción	Requerimientos Asociados	Fuente	Diseño	Prueba
RF1	Registro de Usuarios	-	Usuario, Departamento	Diseño de interfaz de registro	Prueba de registro de usuarios
RF2	Inicio de Sesión de Usuarios	RF1	Usuario	Diseño de interfaz de inicio de sesión	Prueba de inicio de sesión
RF3	Carga de Portafolios por Docentes	RF1	Docente	Diseño de módulo de carga de portafolios	Prueba de carga de portafolios
RF4	Revisión de Documentos en Portafolios	RF3	Comisión Revisora	Diseño de módulo de revisión	Prueba de revisión de documentos
RF5	Notificación de Observaciones a Docentes	RF4	Comisión Revisora	Diseño de sistema de notificaciones	Prueba de notificación de observaciones
RF6	Planificación de Fechas de Revisión	-	-	Diseño de calendario de planificación	Prueba de planificación de fechas
RF7	Asignación de Docentes a Revisores	RF6	Departamento	Diseño de interfaz de asignación	Prueba de asignación de docentes
RF8	Generación de Informes	RF4, RF6	Comisión Revisora	Diseño de generación de informes	Prueba de generación de informes
RF9	Envío Automático de Informes al Jefe de Departamento	RF8	Departamento	Diseño de sistema de envío automático	Prueba de envío automático de informes
RF10	Registro de Carga Académica	-	-	Diseño de módulo de carga académica	Prueba de registro de carga académica
RF11	Actualización de Carga Académica	RF10	Departamento	Diseño de interfaz de actualización	Prueba de actualización de carga académica
RNF1	Interfaz Intuitiva y Fácil de Usar	-	Usuario	Diseño de interfaz de usuario	Prueba de usabilidad
RNF2	Respuesta Rápida del Sistema	-	-	Diseño de optimización de rendimiento	Prueba de respuesta del sistema
RNF3	Confidencialidad de la Información	-	Seguridad	Diseño de medidas de seguridad	Prueba de acceso no autorizado
RNF4	Sistema de Autenticación y Autorización Robusto	-	Seguridad	Diseño de sistema de autenticación y autorización	Prueba de robustez de autenticación
RNF5	Escalabilidad del Sistema	-	-	Diseño de arquitectura escalable	Prueba de carga y rendimiento
RNF6	Compatibilidad con Navegadores Web	-	-	Diseño de compatibilidad con navegadores	Prueba de compatibilidad con navegadores

Ilustración 3 La presente matriz de rastreo ayuda a establecer la relación entre los requerimientos funcionales y no funcionales, permitiendo un seguimiento claro del cumplimiento de cada uno de ellos a lo largo del desarrollo del proyecto de evaluación docente.

Documentos de Diseño

Arquitectura funcional del sistema

Interfaz de usuario (capa de presentación)

- Desarrollo de una interfaz de usuario amigable para que los usuarios (administradores, docentes) gestionen la carga académica.
- Formularios para agregar, editar y eliminar cursos, asignar docentes, y gestionar detalles como horarios, aulas, etc.

Lógica de Negocio (Capa de Aplicación)

- Desarrollo de la lógica para calcular la carga horaria total de los docentes.
- Implementación de reglas de asignación de cursos a docentes, considerando preferencias y restricciones.
- Validaciones para asegurar que los cursos no excedan el límite de aforo y que las fechas y horarios sean coherentes.

Persistencia de Datos (Capa de Datos)

- ❖ Implementación de modelos en Laravel para representar las entidades como cursos, docentes, aulas, etc.
- ❖ Configuración y utilización de Eloquent ORM para interactuar con la base de datos.
- ❖ Desarrollo de migraciones para crear y actualizar la estructura de la base de datos

Arquitectura técnica del sistema

Topología de software

Laravel Framework

1. Utilización de Laravel como el marco de desarrollo principal para PHP.
2. Uso de controladores para gestionar la lógica de negocio.
3. Utilización de middleware para manejar la autenticación y autorización.
4. Uso de Eventos y Listeners para gestionar eventos del sistema.
5. Vistas y plantillas Blade para la presentación de datos.

Base de Datos (MySQL)

- Configuración de una base de datos para almacenar la información sobre cursos, docentes, aulas, etc.
- Implementación de un sistema de respaldo y recuperación.
- Uso de transacciones para garantizar la integridad de los datos.

Eloquent ORM

- Utilización de Eloquent, el ORM de Laravel, para interactuar con la base de datos y realizar operaciones CRUD.

Servidor de Aplicaciones

- Configuración para gestionar la carga y mejorar la seguridad.
- Uso de HTTPS para la comunicación segura entre el cliente y el servidor.

Topología del hardware

Servidor Web

- Configuración de un entorno de equilibrio de carga para manejar grandes cantidades de tráfico.
- Implementación de un plan de monitorización y escalabilidad.

Base de Datos

- Implementación en un servidor de base de datos dedicado con configuración para el rendimiento óptimo.
- Estrategias de respaldo y recuperación para garantizar la disponibilidad.

Entornos de Desarrollo, Pruebas y Producción

- ❖ Desarrollo en entornos locales.
- ❖ Despliegue utilizando servicios como Laravel Forge, Envoyer o soluciones de orquestación de contenedores.

Red

- Configuración de una red segura y eficiente entre el servidor web y la base de datos

Documentos basados en UML

Modelado de clases

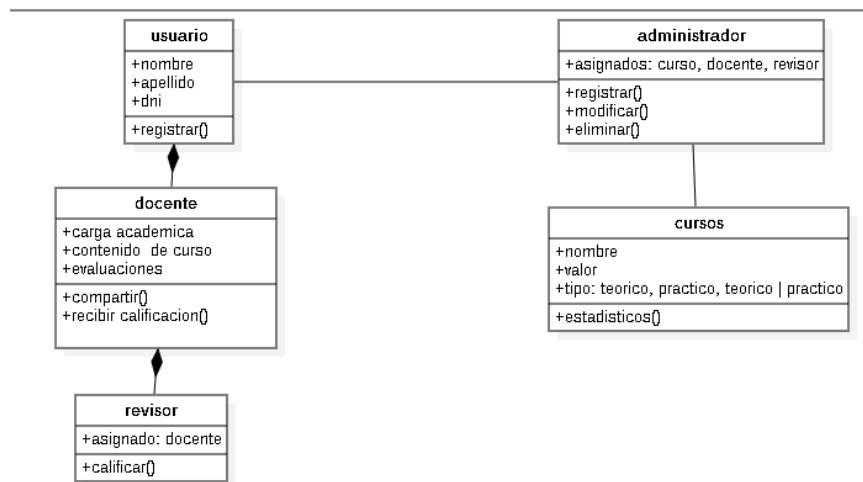


Ilustración 4 Modelado de clases para el sistema de calificación docente

Casos de uso

Caso de uso de revisor de docente del proyecto de carga académica



Ilustración 5 Diagrama de casos de uso de revisor de portafolio docente

Diagrama de caso de uso de docente

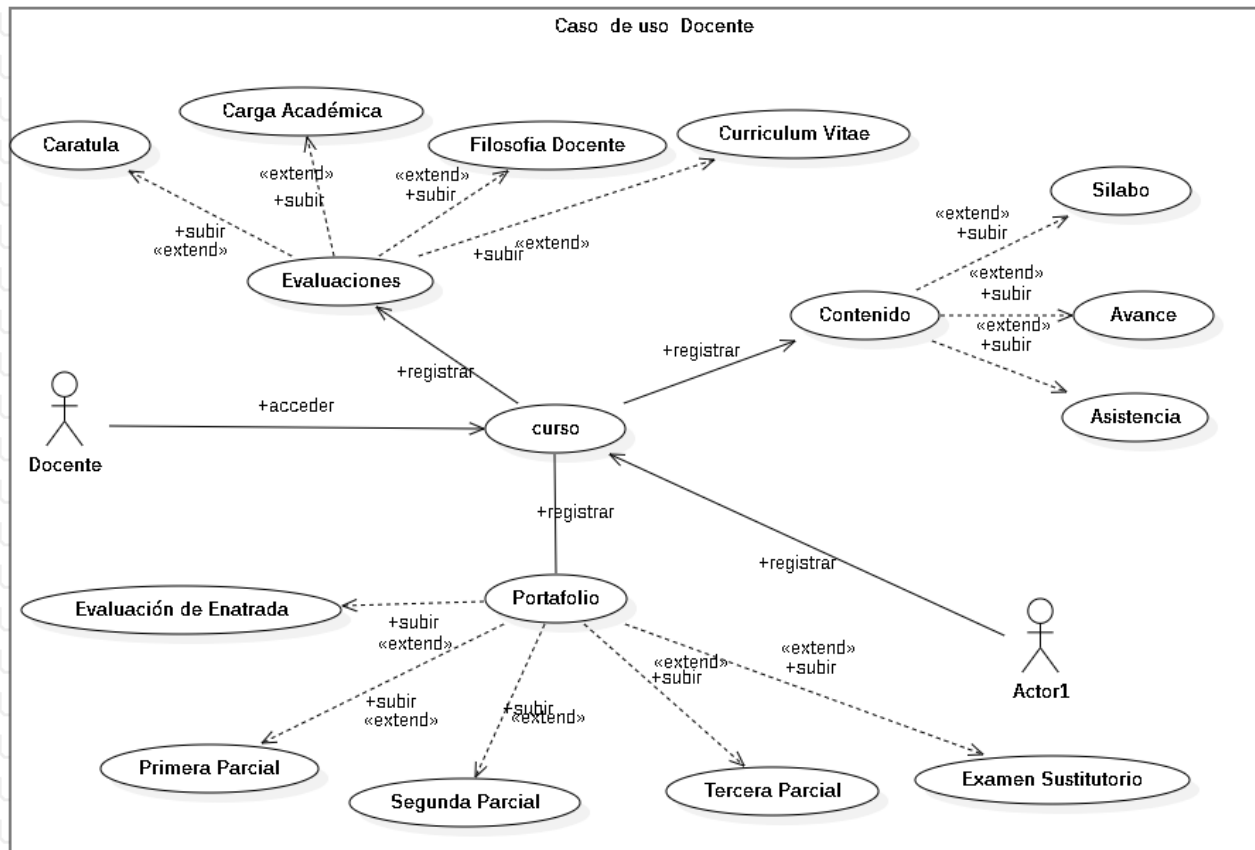


Ilustración 6 Vista del caso de uso de portafolio docente

Diagrama de caso de uso de administrador

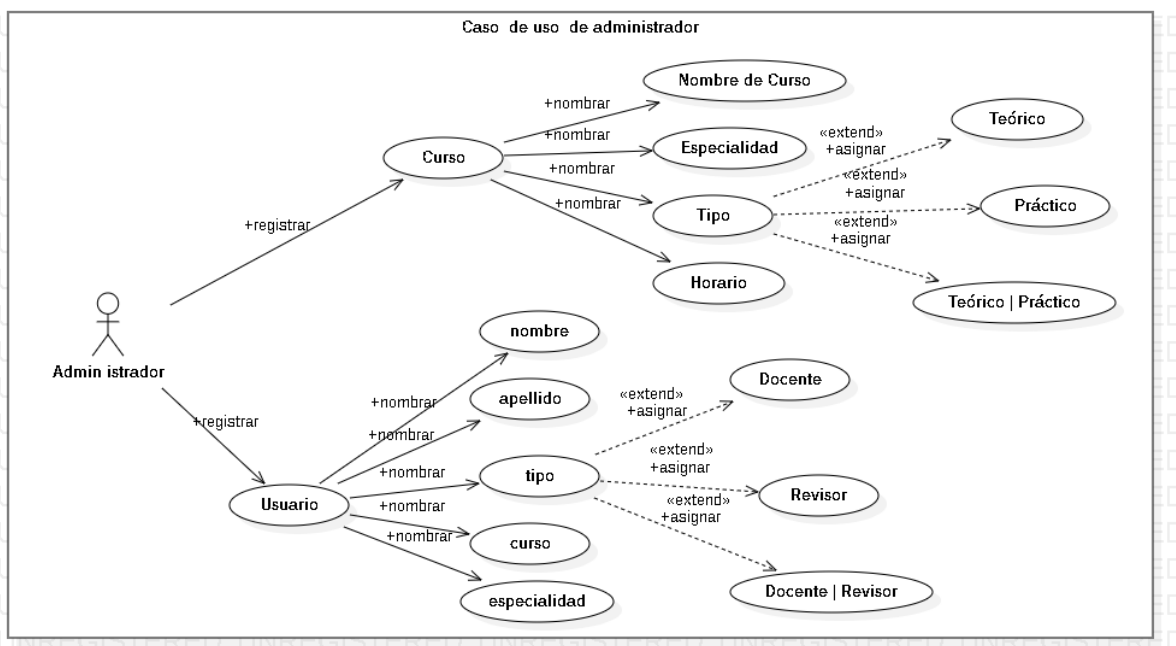


Ilustración 7 Vista del caso de uso del administrador del sistema de evaluación docente

Diagramas secuenciales

Diagrama de secuencias de revisión de portafolio docente

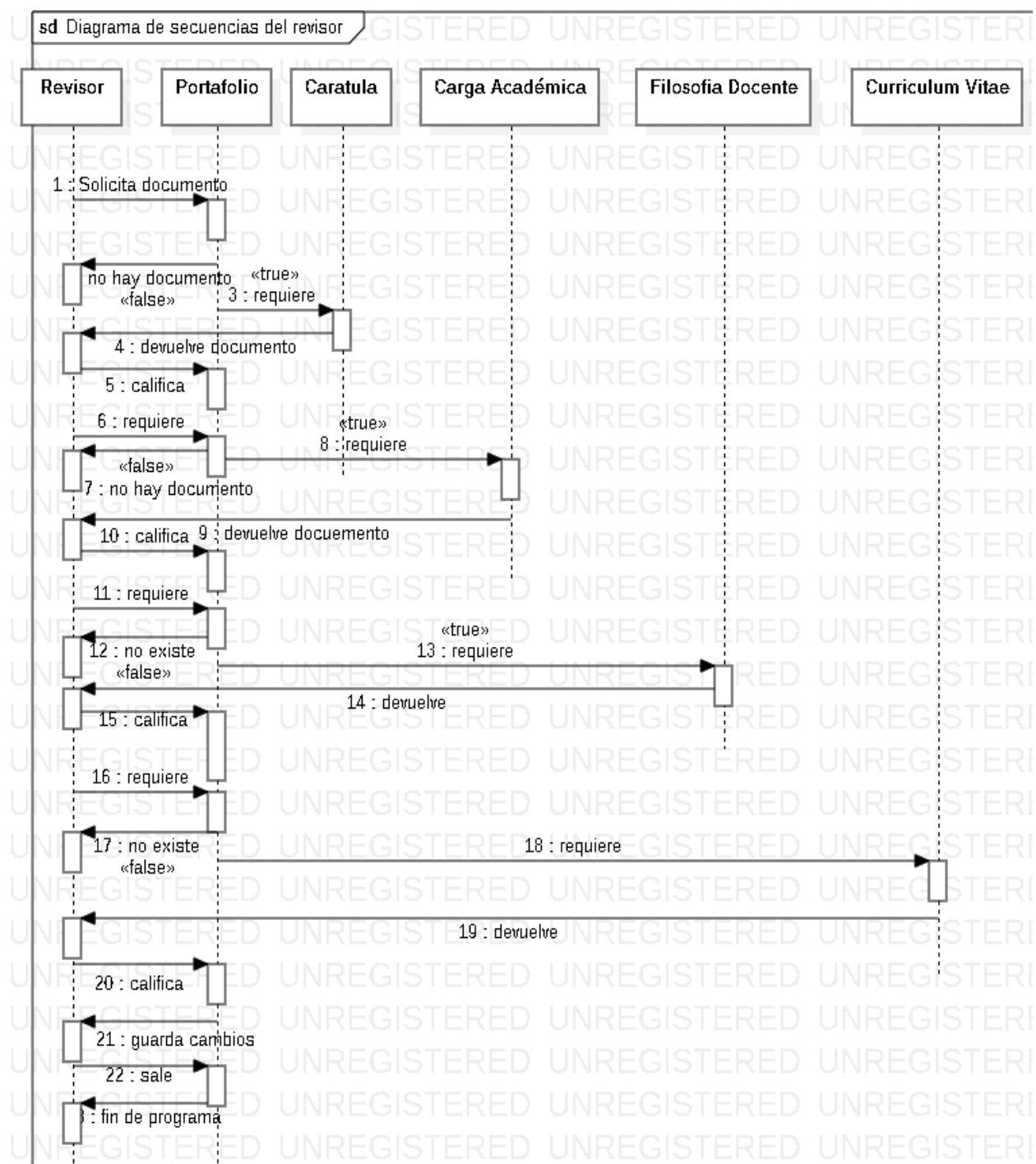


Ilustración 8 Vista del diagrama de secuencias donde el revisor interactúa con el sistema de calificación docente en lo que corresponde a la revisión de portafolio docente

Diagrama de secuencias de revisión de prueba docente

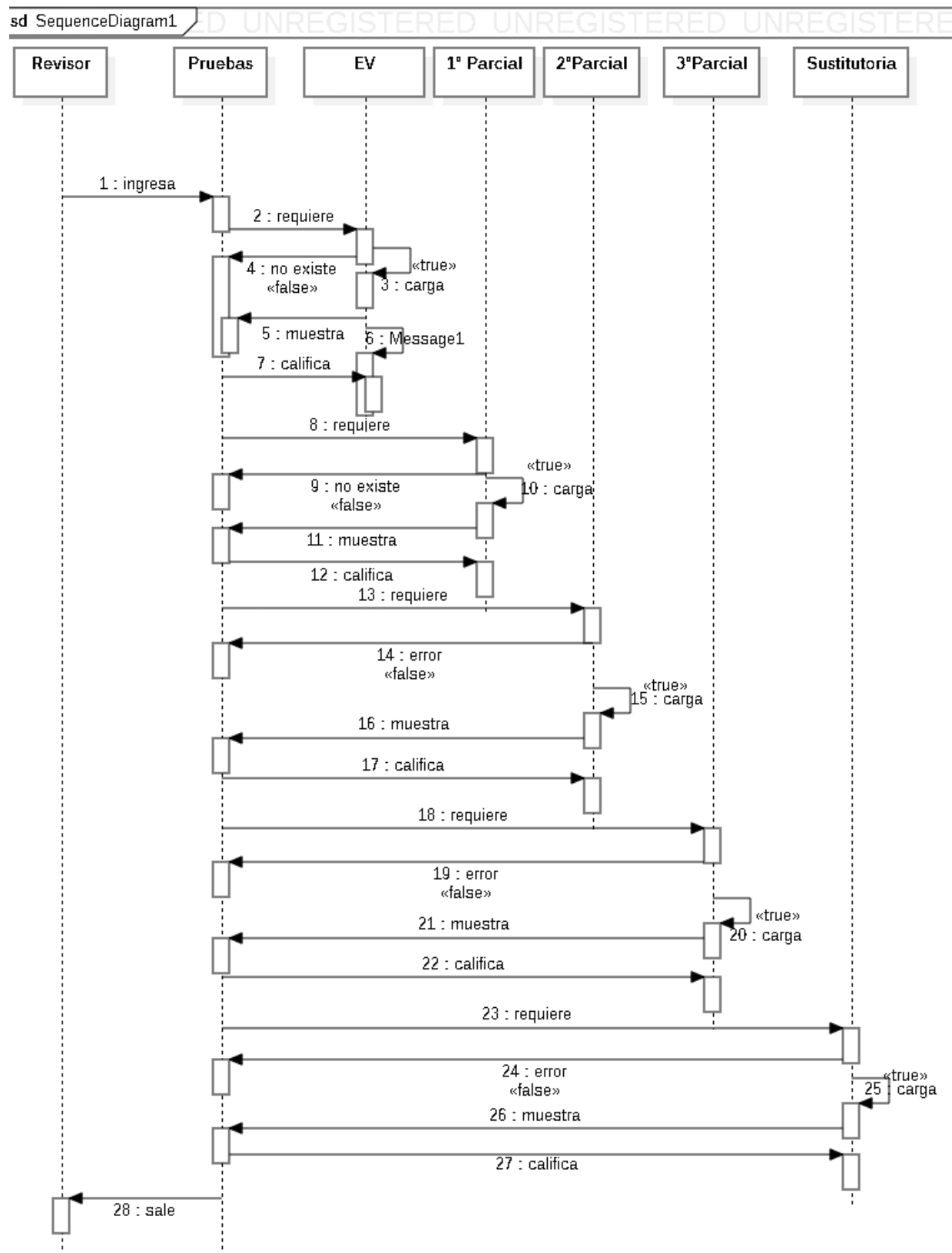


Ilustración 9 Vista del diagrama de secuencias donde el revisor interactúa con el sistema de calificación docente en lo que corresponde a la revisión de prueba docente

Diagrama de secuencias de revisión de contenido docente

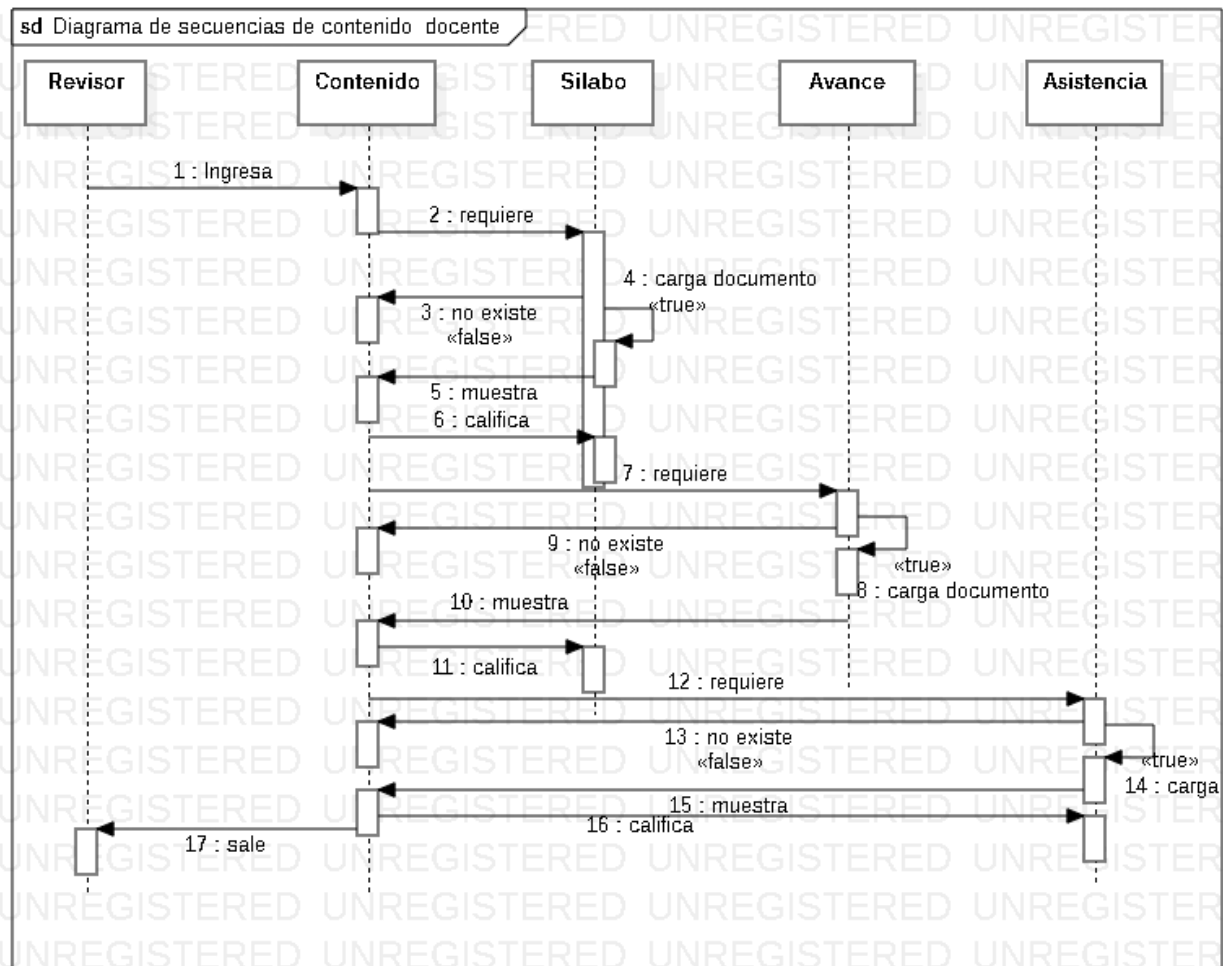


Ilustración 10 Vista del diagrama de secuencias donde el revisor interactúa con el sistema de calificación docente en lo que corresponde a la revisión de contenido docente

Descripción de la metodología ágil utilizada de manera específica (sprints, historias, etc)

Roles en SCRUM:

Product Owner:

- Identifica y prioriza las necesidades y características clave del proyecto en colaboración con los usuarios finales y los administradores académicos.
- Define las historias de usuario y los criterios de aceptación.
- Participa activamente en las reuniones de planificación de sprint y proporciona la visión general del producto.

Scrum Master:

- Facilita las ceremonias de SCRUM y garantiza la adhesión a los principios y prácticas ágiles.
- Elimina obstáculos y fomenta la comunicación efectiva entre los miembros del equipo. Apoya la mejora continua del equipo y promueve un entorno de trabajo colaborativo.

Equipo de Desarrollo:

- Profesionales encargados de diseñar, desarrollar e implementar las funcionalidades del proyecto.
- Se autoorganizan para completar las historias de usuario durante los sprints.
- Participan activamente en la planificación de sprint y las reuniones diarias.

Ceremonias de SCRUM:

Planificación del Sprint:

- El equipo, junto con el Product Owner, planifica las historias de usuario a abordar durante el sprint.
- Se establece la meta del sprint y se determinan las tareas necesarias para alcanzarla.
- La duración típica del sprint puede ser de dos semanas.

Reunión Diaria o Daily Standup:

- Sesiones cortas diarias donde el equipo comparte actualizaciones sobre el progreso,
- los obstáculos encontrados y los planes para el próximo día.
- Fomenta la colaboración y la transparencia, permitiendo a cada miembro del equipo conocer el estado del proyecto.

Revisión de Sprint:

- Al final del sprint, el equipo demuestra las funcionalidades completadas al Product Owner y otras partes interesadas.
- Se recopila la retroalimentación para ajustar y mejorar las futuras iteraciones del proyecto.

Retrospectiva del Sprint:

- El equipo reflexiona sobre el sprint, identifica áreas de mejora y define acciones concretas para implementar cambios positivos en el próximo sprint.
- Incluye una revisión de procesos, herramientas y relaciones interpersonales.

Artefactos de SCRUM:

Backlog del Producto:

- Lista priorizada de todas las características, historias de usuario y mejoras planificadas para el proyecto.
- El Product Owner es responsable de mantener y actualizar continuamente el backlog.

Backlog del Sprint:

- Subset del backlog del producto que se selecciona para abordar durante el sprint.
- Se establece durante la planificación del sprint y se mantiene constante durante el sprint.

Incremento:

- Conjunto de funcionalidades totalmente desarrolladas y probadas al final de cada sprint.
- Representa un avance tangible y utilizable del producto.

Modelo lógico de la base de datos

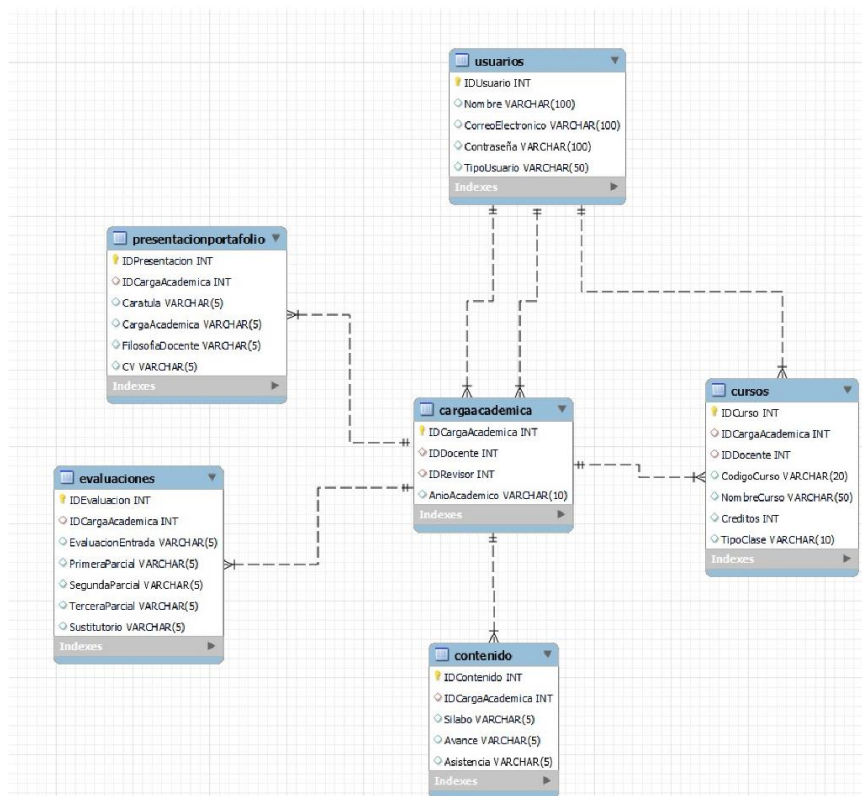


Ilustración 11 Vista del modelo lógico de la base de datos del proyecto de calificación docente

Prototipado