



# UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS

Facultad de Ingeniería

DEPARTAMENTO DE INGENIERÍA EN SISTEMAS

Ingeniería de Software sección IS-802 1100

Proyecto de Ingeniería de Software

Ing. José Manuel Inestroza Murillo

Presentado por Grupo A:

Josué Daniel Mejía Izaguirre 20211031684

Mauricio Alberto Velásquez Díaz 20181008085

Saul Eduardo Torres Vega 20201004758

Dustin Adrian Barahona Lainez 20211020011

Introducción.....	3
Requerimientos Funcionales .....	4
Biblioteca Virtual.....	4
Admisiones.....	4
Estudiantes.....	4
Música .....	4
Docentes .....	5
Jefes de Departamento .....	5
Coordinadores.....	5
Administración .....	5
Requerimientos No Funcionales.....	6
Modelos del Sistema UML.....	7
Biblioteca Virtual.....	7
Diagramas de flujo Biblioteca .....	9
Estudiantes.....	14
Música .....	16
Docentes .....	19
Diagrama de uso de Docente.....	19
Diagrama Entidad Relación.....	21
Mockups .....	25
Glosario.....	29
División de Tareas .....	31
Paletas de Colores.....	32
Backlog.....	32

## Introducción

En el marco del curso IS-802 Ingeniería de Software, se nos ha dado el reto de diseñar y desarrollar el **Sistema de Registro Académico** para la Universidad Nacional Autónoma de Honduras (UNAH). Simulamos una contratación profesional que exige la aplicación rigurosa de principios, metodologías y buenas prácticas propias de la ingeniería de software, con el objetivo de ofrecer una solución tecnológica robusta, eficiente y confiable.

La propuesta contempla el desarrollo de múltiples módulos funcionales, entre ellos: Biblioteca Virtual, Admisiones, Matrícula, Estudiantes, Carrera de Música, Docentes y Administración. Cada uno de ellos está diseñado bajo estándares de calidad que priorizan la mantenibilidad, seguridad, eficiencia y aceptabilidad del usuario final, integrando enfoques éticos y técnicos propios de nuestra formación profesional.

Nuestro equipo de trabajo ha adoptado la metodología ágil **Scrum** para la planificación, ejecución y entrega incremental del proyecto, desempeñando los distintos roles y entregables definidos por el marco metodológico (Ing. José Inestroza). Asimismo, se ha empleado el **patrón de diseño MVC** en conjunto con una **arquitectura orientada a microservicios**, facilitando la escalabilidad y modularidad del sistema.

El desarrollo contempla tecnologías como **PHP** para el backend, **MySQL** como sistema gestor de base de datos, RESTful APIs y lógica asíncrona mediante JavaScript. Se han incluido mecanismos de autenticación con gestión de roles y privilegios, validaciones por medio de expresiones regulares, pruebas en al menos dos módulos, mecanismos de cifrado y procedimientos almacenados que refuerzan la seguridad y la integridad de los datos del sistema.

Este documento presenta la planificación y el diseño del sistema, evidenciando la aplicación real de los conocimientos adquiridos a lo largo de nuestra carrera universitaria y del curso presente.

## Requerimientos Funcionales

### Biblioteca Virtual

- Asignación de permisos a coordinadores y jefes de departamento para subir recursos.
- Visualización de los recursos por parte de estudiantes según sus cursos actuales.
- Registro de metadatos de los recursos: autor(es), título, tags y archivo PDF.
- Visualización de PDF en la pantalla sin opción de descarga
- Acceso controlado por roles

### Admisiones

- Formulario de inscripción con validación de tipo, peso y dimensiones del certificado.
- Asignación automática de solicitudes a revisores y carga balanceada.
- Corrección de datos por parte del estudiante tras rechazo.
- Carga automática de notas desde CSV
- Evaluación automatizada de admisión con base en resultados.
- Envío automático de resultados por correo.
- Generación de archivo CSV para el sistema de registro.
- Soporte para nuevos tipos de exámenes y reglas de clasificación.

### Estudiantes

- Creación automática de usuarios desde CSV generado por Administración.
- Envío de credenciales por correo.
- Recuperación de contraseña.
- Matrícula condicionada a requisitos y materias pendientes.
- Portal exclusivo del estudiante con:
  - Descarga de certificado Académico.
  - Evaluación obligatoria del docente antes de ver notas.
  - Chat con solicitudes, contactos, estados en línea.
  - Solicitudes: cambio de carrera, cancelaciones, reposición, matrícula.
  - Visualización del perfil de los docentes.
  - Cancelación de secciones durante el periodo de matrícula.

### Música

- Plataforma específica con recursos virtuales organizados por formato (PDF, MP3, partituras)
- Visualización en pantalla para PDF y MP3, sin descarga.
- Ingreso de contenido por parte de jefes o coordinadores automáticamente asignados.

## Docentes

- Creación de usuarios por administrador con datos básicos y foto.
- Acceso a sus clases y descarga de listados.
- Carga de materiales introductorios por clase.

## Jefes de Departamento

- Planificación de clases sin traslape de horarios ni aulas.
- Control de cupos, cancelación de secciones con justificación.
- Acceso a:
  - Listas de espera.
  - Historial académico de cualquier estudiante.
  - Evaluaciones docentes.
  - Notas ingresadas por docentes.
  - Estadísticas propuestas por el equipo.
  - Listado de estudiantes matriculados.
  - Reinicio de claves docentes (enlace único con tiempo limitado).
- Gestión de planificación interdepartamental.

## Coordinadores

- Descarga del resumen de carga académica por periodo (PDF o Excel).
- Aprobación/rechazo de cambios de carrera y cancelaciones excepcionales.
- Revisión del historial de los estudiantes.

## Administración

- Alta de nuevos docentes.
- Configuración de fechas y procesos: planificación, matrícula, cancelaciones.

## Requerimientos No Funcionales

### Seguridad:

- Autenticación con roles y privilegios
- El sistema debe garantizar que solo usuarios autorizados puedan subir o modificar los recursos de la biblioteca.
- Los libros en PDF deben estar protegidos contra descargas no autorizadas.
- Los datos de los estudiantes deben almacenarse de manera segura y cumplir con normativas de protección de datos.
- Cifrado obligatorio de campos sensibles (ej. identidades, contraseñas)

### Usabilidad:

- Interfaz responsive compatible con Chrome, Firefox y Edge.
- La interfaz del sistema debe ser intuitiva y fácil de usar tanto para administradores como para estudiantes.
- La búsqueda de libros debe ser rápida y eficiente mediante el uso de filtros por asignatura o palabras clave.

### Rendimiento:

- El sistema debe permitir múltiples accesos simultáneos sin afectar su desempeño.
- La carga y visualización de archivos PDF debe ser rápida y sin demoras.

### Compatibilidad:

- El sistema debe ser accesible desde distintos dispositivos (PC, tabletas y móviles).
- Debe ser compatible con navegadores modernos (Chrome, Firefox, Edge, etc.).

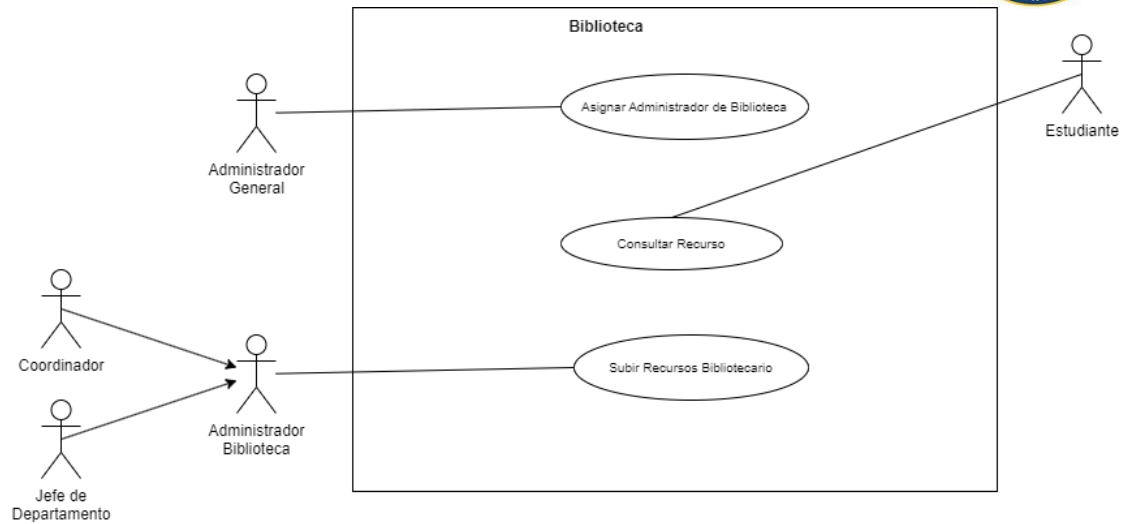
### Mantenimiento y escalabilidad:

- El sistema debe permitir futuras mejoras y actualizaciones sin afectar la funcionalidad existente.
- Debe poder ampliarse para admitir más estudiantes y libros sin comprometer el rendimiento.

## **Modelos del Sistema UML**

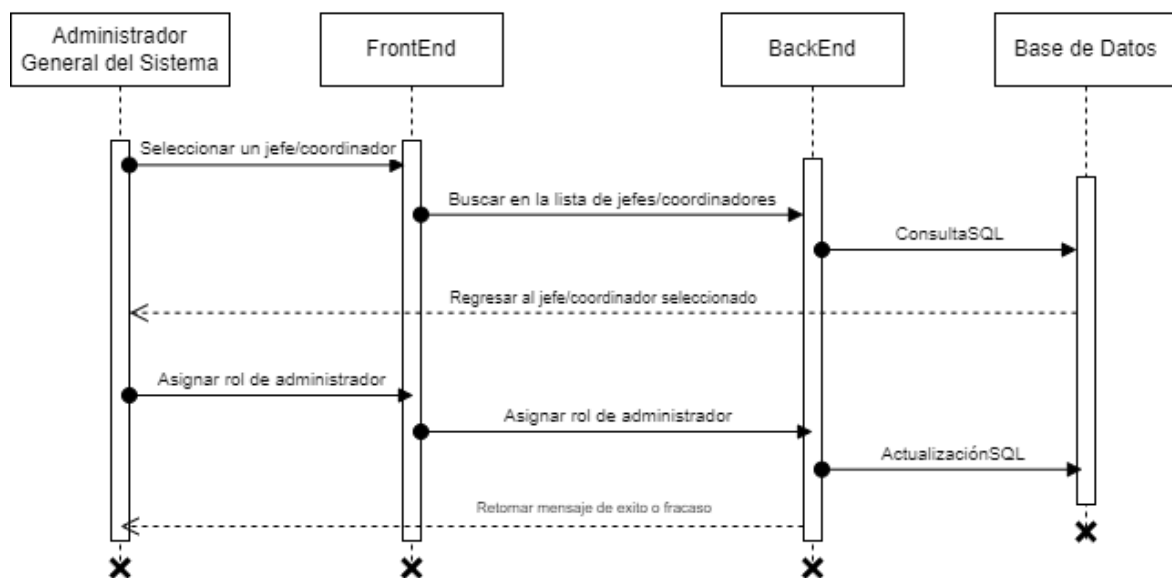
### **Biblioteca Virtual**

#### **Caso de uso Módulo Biblioteca**

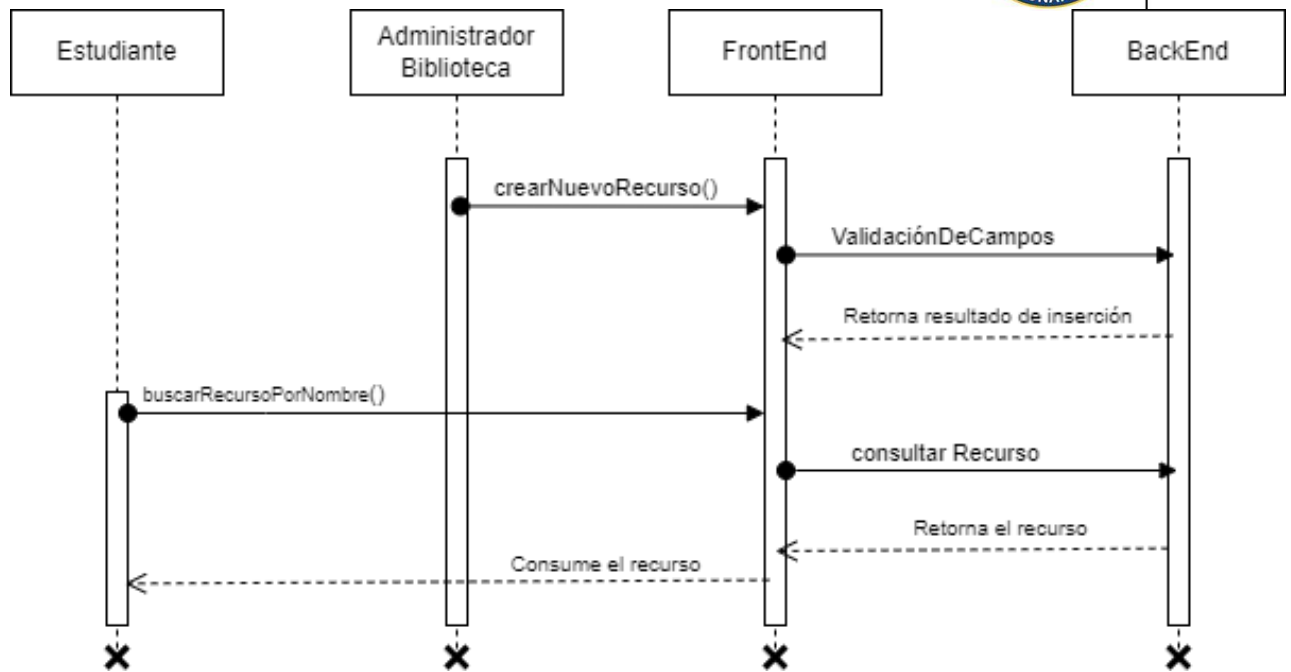


## Diagramas de Secuencia Módulo Biblioteca

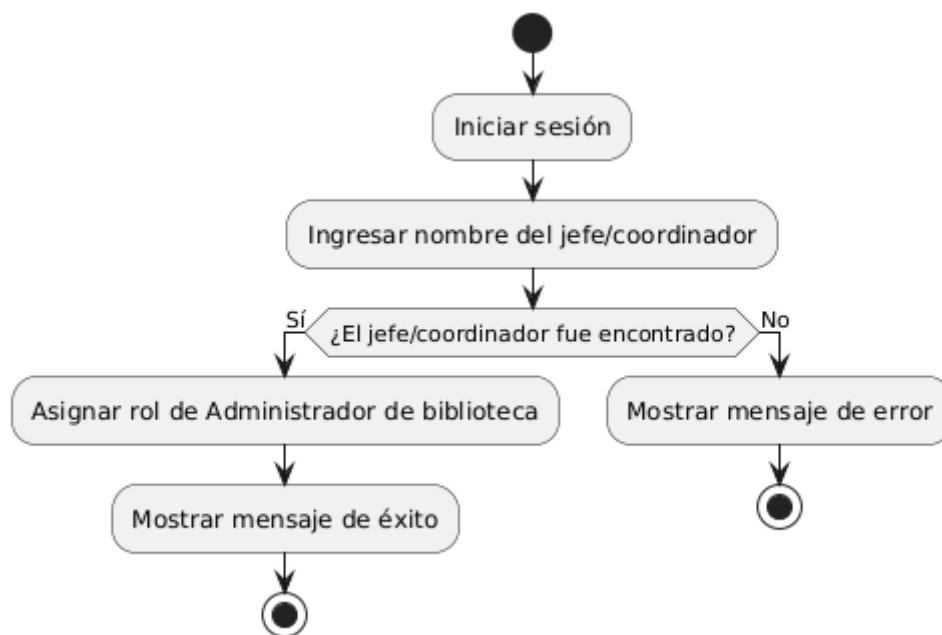
### Asignar Administrador de Biblioteca

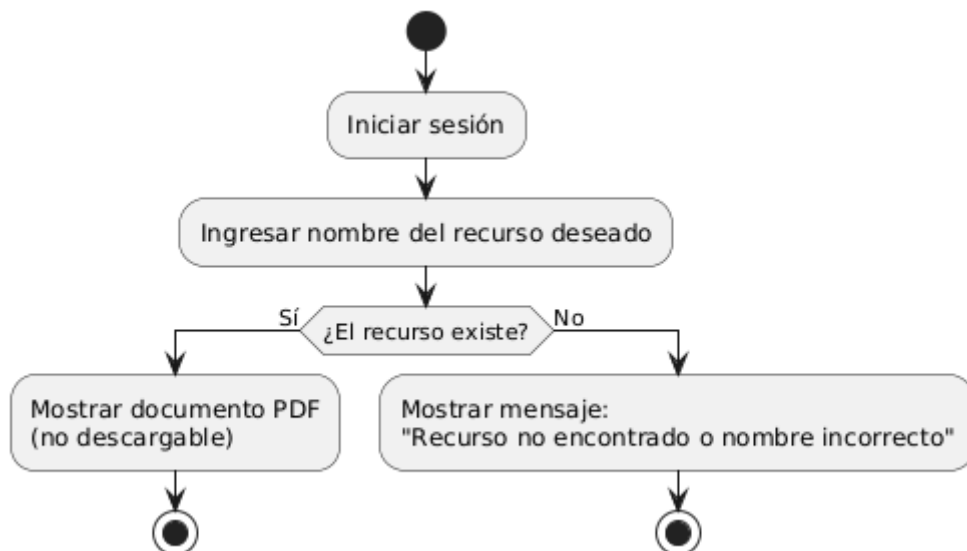
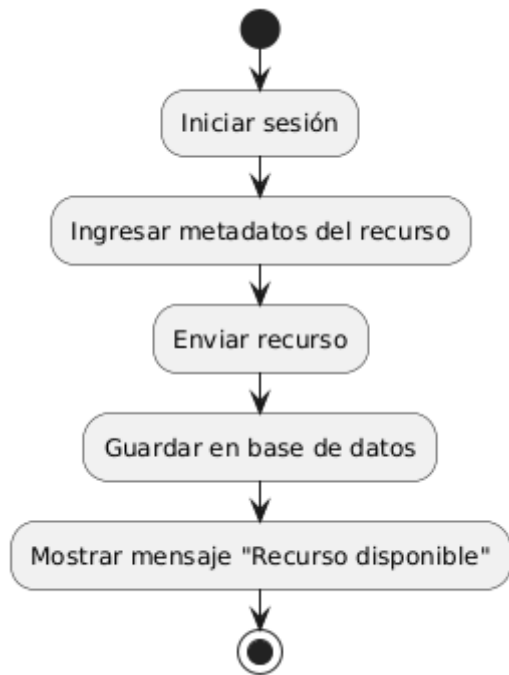






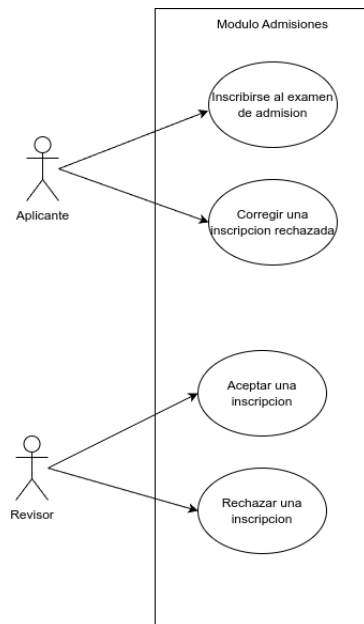
### Diagramas de flujo Biblioteca





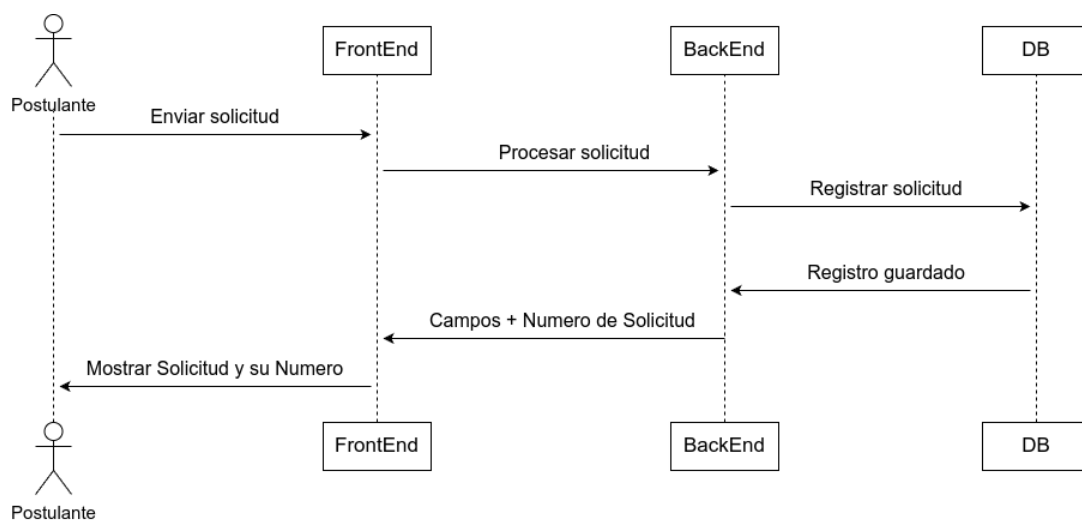
## Admisiones

### Diagrama Caso de Uso

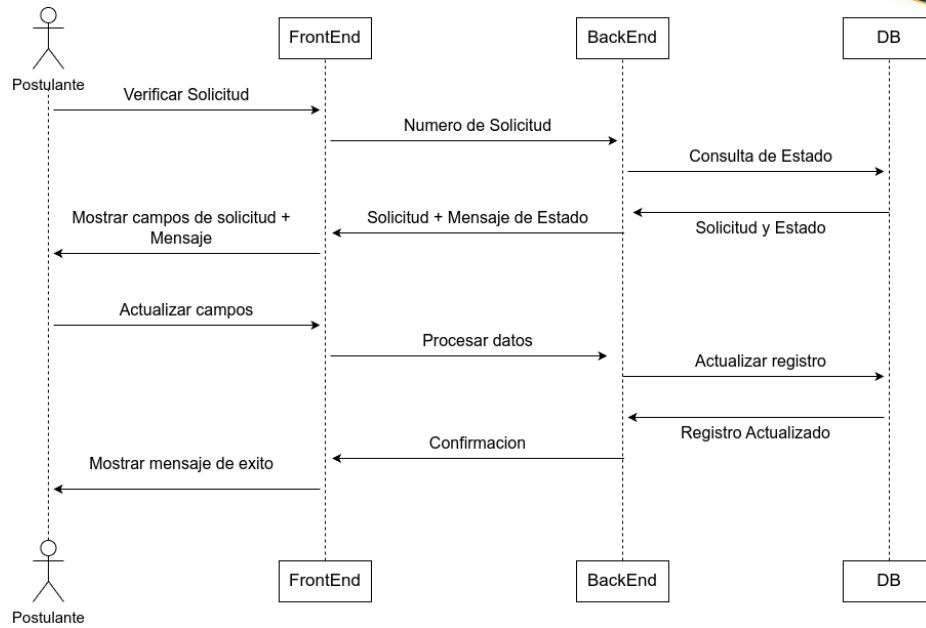


### Diagrama de secuencia

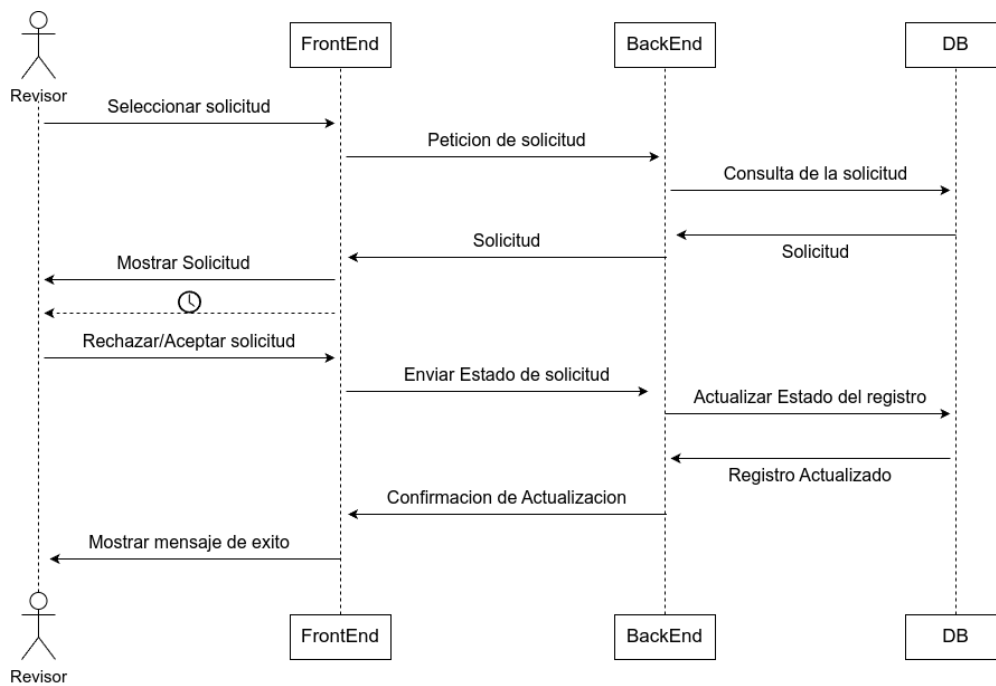
#### Diagrama de Secuencia - Postulante - Enviar Solicitud



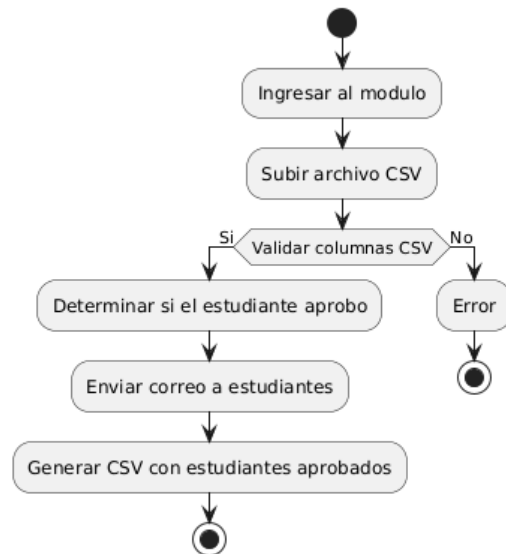
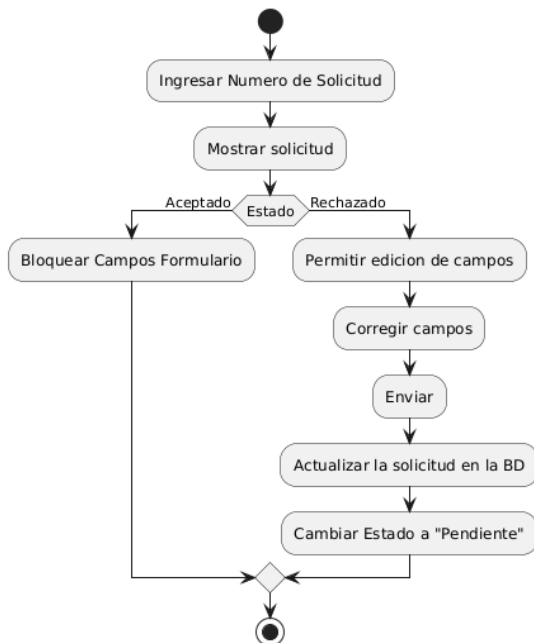
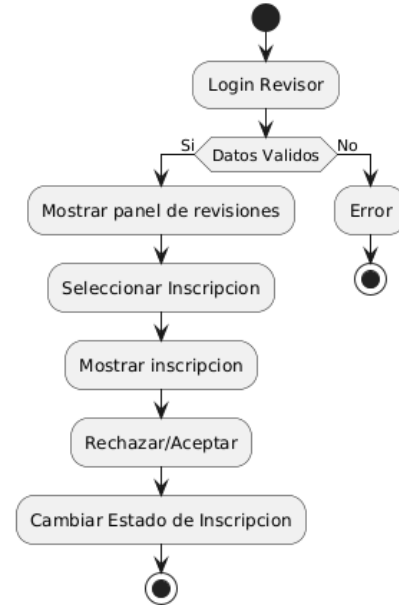
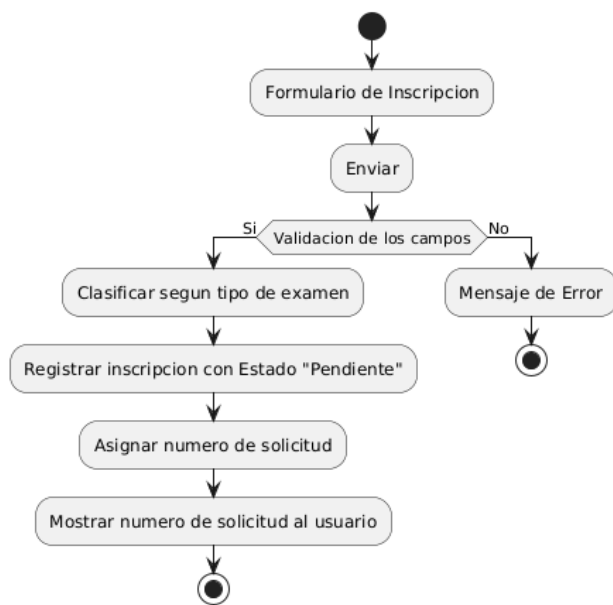
#### Diagrama de Secuencia – Postulante – Corregir Inscripcion



### Diagrama de Secuencia – Revisor – Aceptar/Rechazar Solicitud

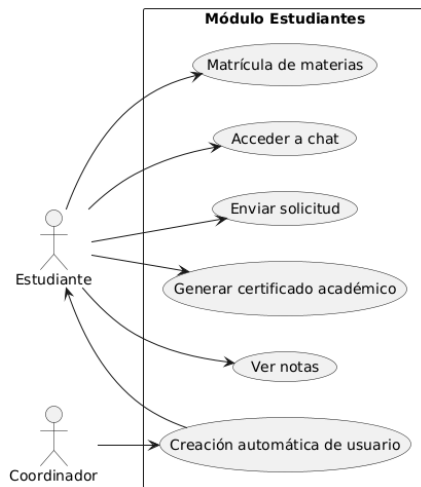


### Diagrama de Flujo



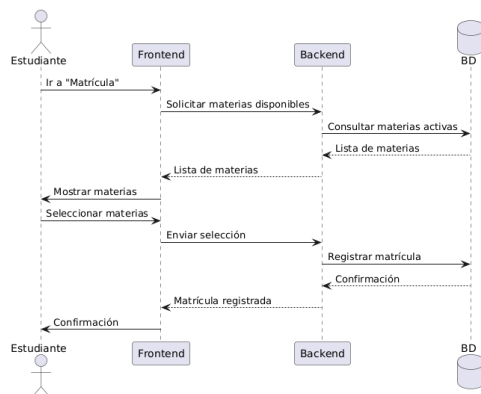
## Estudiantes

### Diagrama Caso de Uso Estudiantes

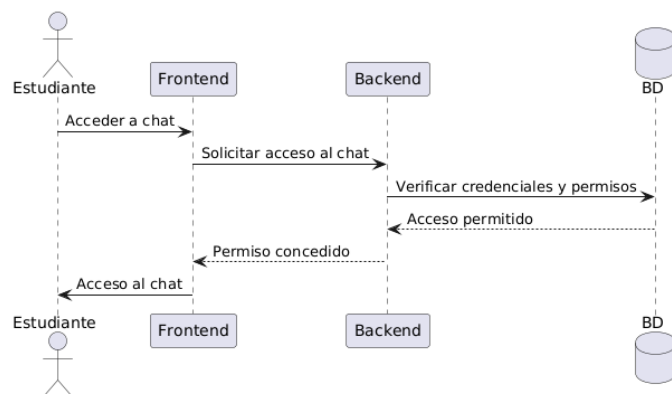


### Diagramas de Secuencia

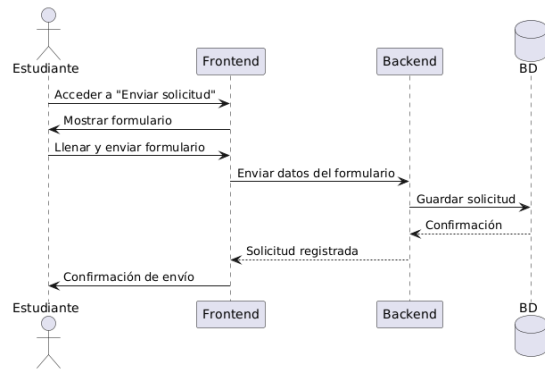
#### Diagrama de flujo de Matricula



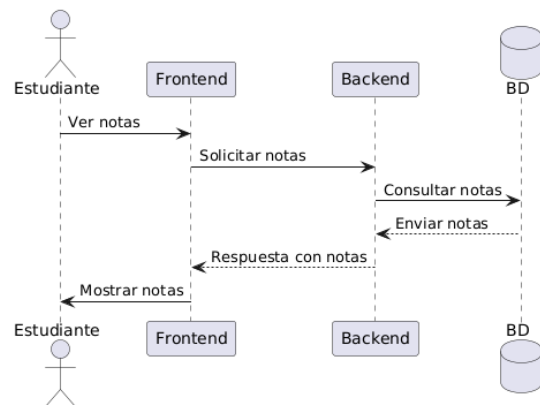
#### Acceder al chat



#### Enviar Solicitud



## Ver Notas



## Generar Certificado

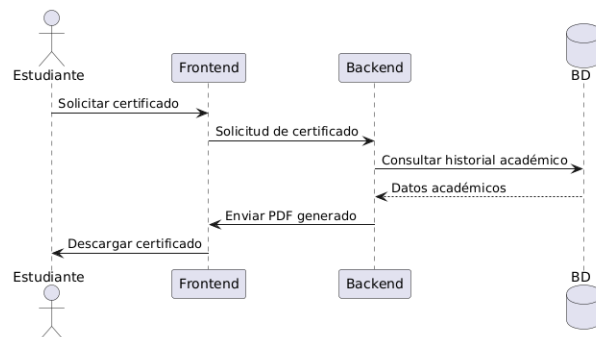
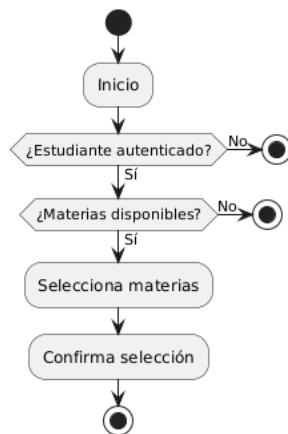
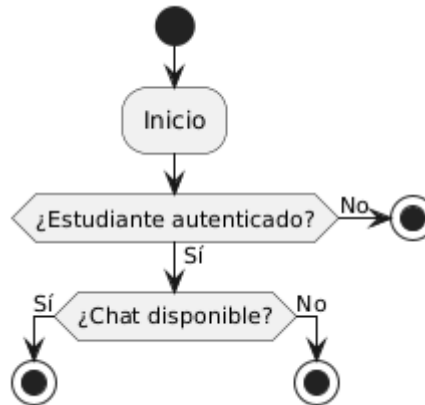


Diagrama de flujo

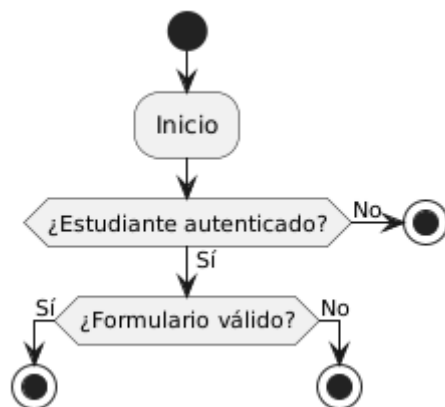
## Matricula



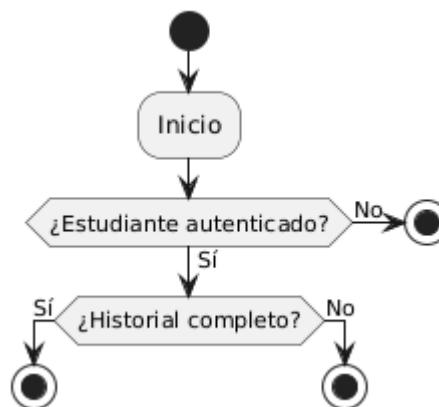
## Chat



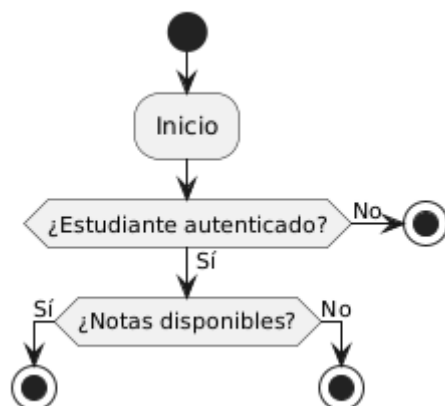
## Solicitud



## Generar Certificado



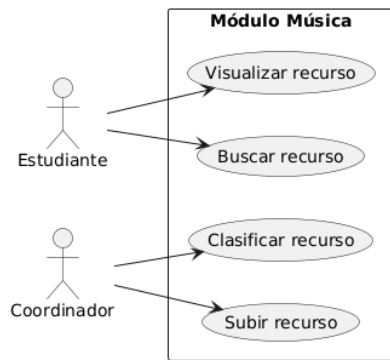
## Ver notas



## Música

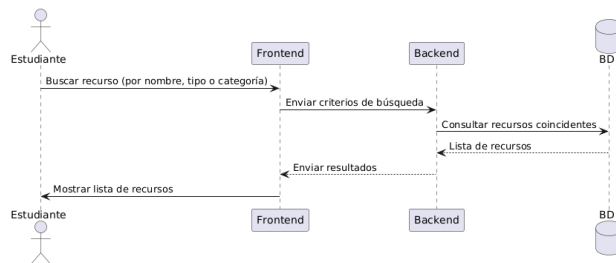
Diagrama Caso de Uso



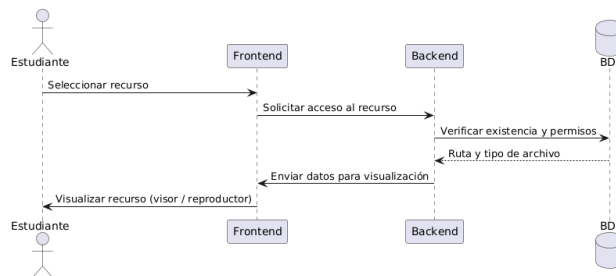


## Diagrama de Secuencia

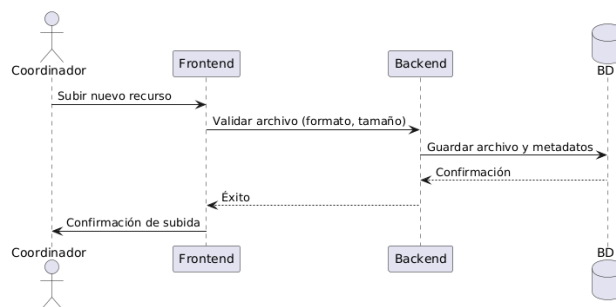
### Buscar recursos



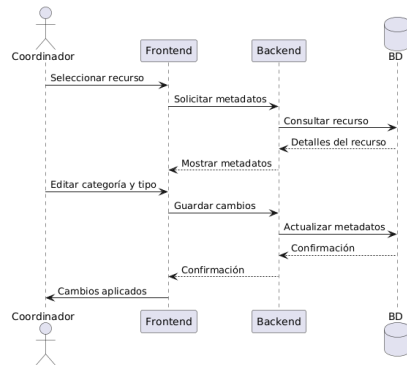
### Visualizar Recursos



### Subir Recurso

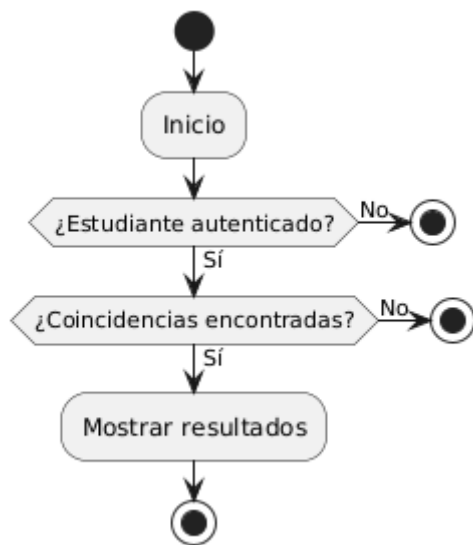


### Clasificar

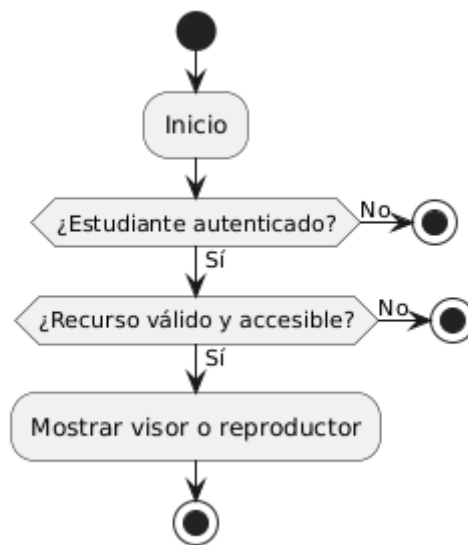


## Diagrama de Flujo

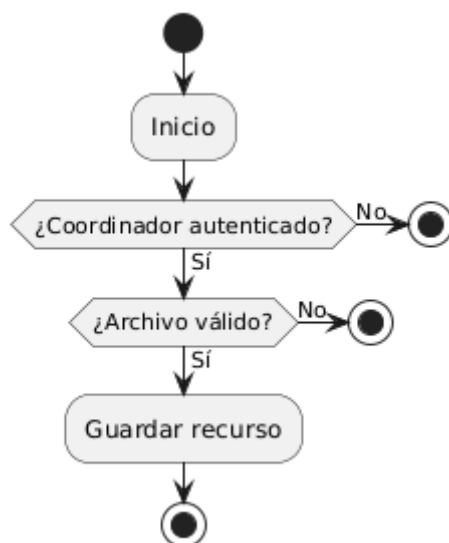
### Buscar recursos



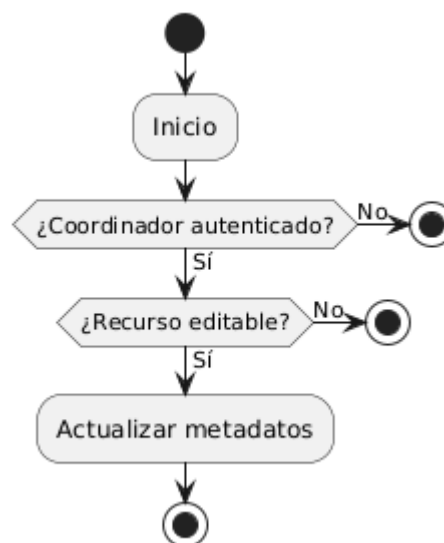
### Visualizar Recurso



### Subir Recurso

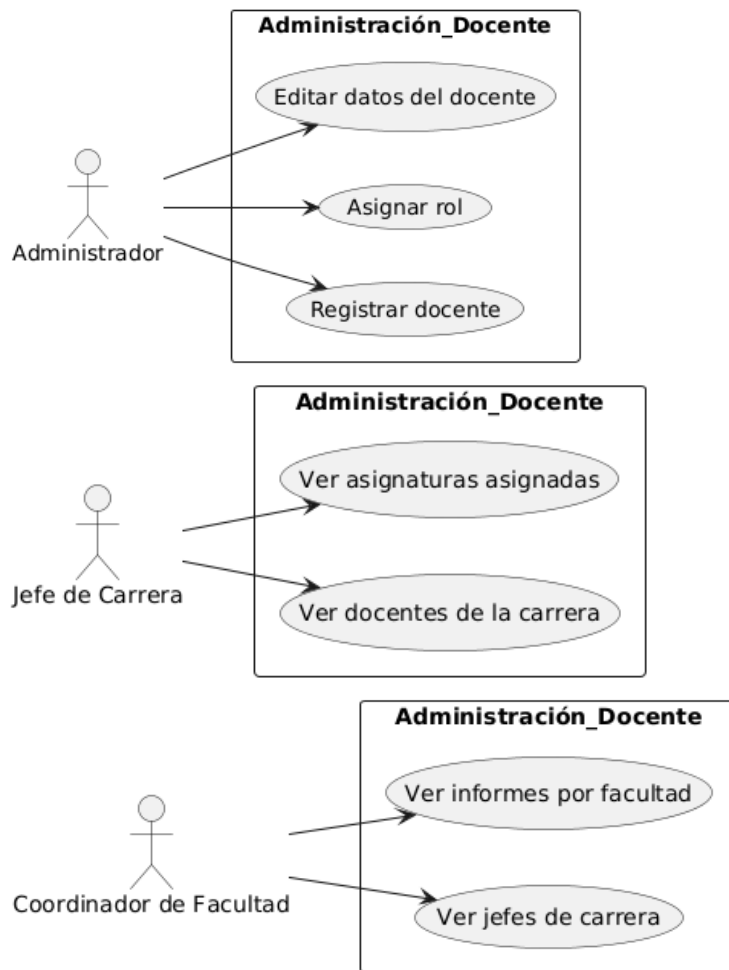


### Clasificar

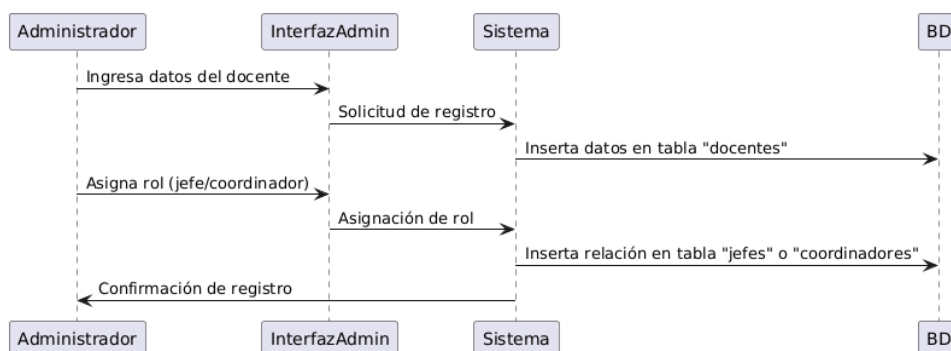


## Docentes

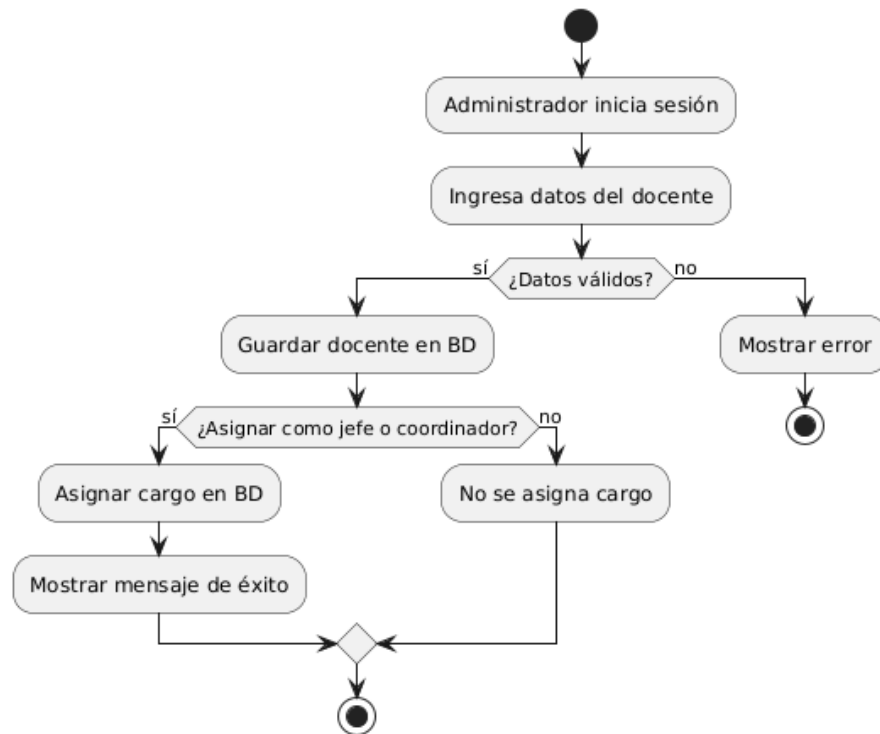
### Diagrama de uso de Docente



### Diagrama de Secuencia de Docentes

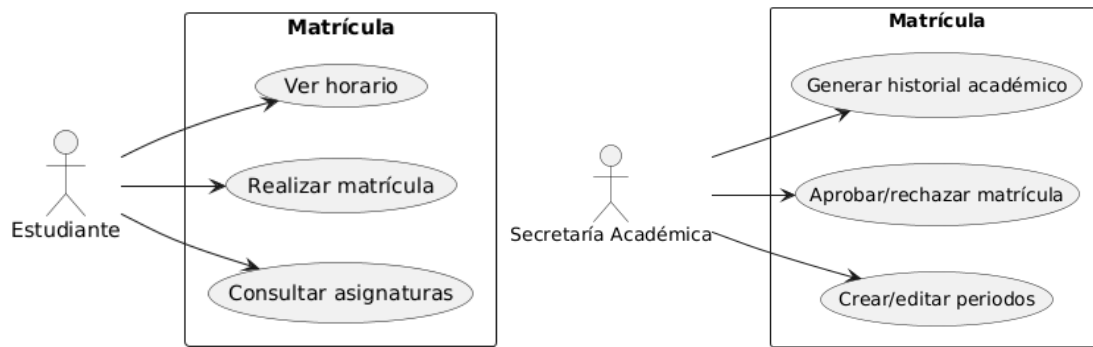


## Diagrama de flujo de Docente

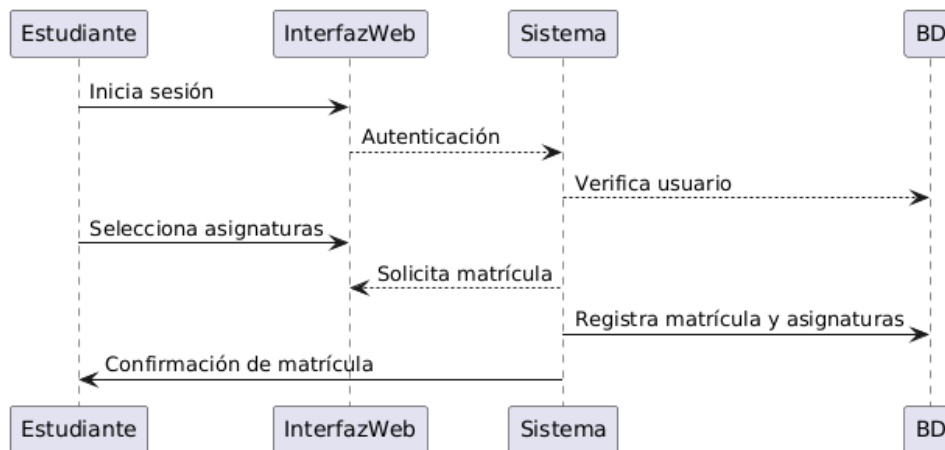


## Matricula

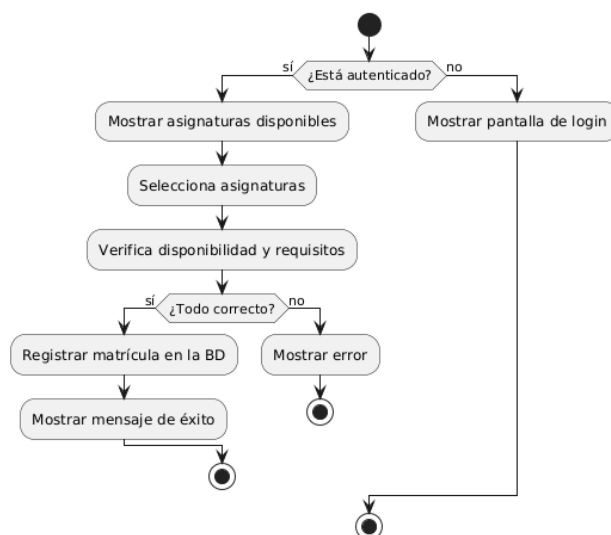
## Diagrama de uso de Matricula



## Diagrama secuencial de Matricula



## Diagrama de flujo de Matricula



## Diagrama Entidad Relación

**Entidad: Usuario**

**Descripción:** identificador para todo usuario del sistema desde aspirante hasta jefe de departamento

**Atributos:**

- UsuarioID (PK): Identificador único.
- PNombre: Primer nombre
- SNombre: Segundo nombre
- PApellido: Primer apellido
- Identidad: Número de identidad.
- CorreoPersonal: Email de contacto.
- CorreoInstitucional: Email proveído por el sistema
- Teléfono: Número de contacto.
- Contraseña: Contraseña del usuario
- NumeroCuenta: Código de empleado del docente.
- Rol: Tipo de usuario (Aspirante, Estudiante, Docente, Administrador).

**Relaciones:**

- Usuario se relaciona con Admisión (1:1, solo si es un aspirante).
- Usuario se relaciona con Estudiante (1:1, solo si es admitido).
- Usuario se relaciona con Docente (1:1, si es profesor).
- Usuario se relaciona con Administrador (1:1, si tiene ese rol).

**Entidad: Admisión**

**Descripción:** Procura el registro de los aspirantes a estudiante

**Atributos:**

- AdmisiónID (PK): Identificador único.
- UsuarioID (FK): Referencia al usuario que solicita la admisión.
- FechaSolicitud: Fecha en que se envió la solicitud.
- Estado: Estado de la solicitud (Pendiente, Aprobada, Rechazada).
- CarreraID (FK): Carrera a la que aplica.
- CarreraAlternativaID (FK): Segunda opción en caso de no ser admitido en la primera.
- CertificadoSecundaria: Archivo con el certificado de estudios.
- Observaciones: Comentarios sobre la solicitud.

**Relaciones:**

- Admisión se relaciona con Usuario (1:1).
- Admisión se relaciona con Carrera (N:1, En ambos campos, CarreraID y CarreraAlternativaID).

### **Entidad: Carrera**

**Descripción:** Representa todas las carreras disponibles dentro de los centros regionales

#### **Atributos:**

- CarreraID(PK): identificador de cada carrera
- NombreCarrera: nombre de la carrera

### **Entidad: Centro Regional**

**Descripción:** Representa todos los centros regionales disponibles

#### **Atributos:**

- CentroID(PK): identificador único para los centros
- NombreCentro: nombre del centro regional

### **Entidad: Secciones**

**Descripción:** Representa todas las clases y seccion de la misma

#### **Atributos:**

- SeccionID(PK): identificador único para cada clase
- Asignatura: nombre de la clase
- Aula: Numero de aula
- Edificio: número de edificio

### **Entidad: Docente**

**Descripción:** Representa a todos los docentes, jefes de departamento y coordinadores

#### **Atributos:**

- DocenteID(PK): Identificador único
- UsuarioID(FK): Referencia al usuario que es docente
- CnetroID: Centro al que pertenece
- DepartamentoID(FK): Referencia al id de los departamentos

### **Entidad: Departamento**

**Descripción:** Se representa a cada uno de los departamentos que existen

### **Atributos:**

- DepartamentoID(PK): Identificador único
- NombreDepartamento: nombre del departamento

### **Entidad: Edificio**

**Descripción:** Representa edificio en el cual se da las clases

### **Atributos:**

- EdificioID(PK): identificador único
- Edificio

### **Entidad: Matricula**

**Descripción:** Representa la matricula que realiza el estudiante

### **Atributos:**

- MatriculaID(PK): identificador único
- EstudianteID(FK): Referencia al estudiante
- DocentesID(FK): Referencia al docente
- SeccionID((FK): Referencia a la asignatura
- EdificioID(FK): Referencia a los edificios
- Aula: número de aula donde se imparte la clase
- Horario: hora en la que se imparte la clase
- CupoMaximo: número máximo de cupos

### **Relaciones:**

- Matricula se relaciona con Docente (N:1)
- Matricula se relaciona con Secciones(1:N)
- Matricula se relaciona con Estudiante(1:N)
- Matricula se relaciona con Edificio(1:N)

### **Entidad: Notas**

**Descripción:** Registra las calificaciones de los estudiantes.

### **Atributos:**

- NotaID (PK): Identificador único.
- EstudianteID (FK): Relación con el estudiante.
- MatriculaID(FK): Relación con la sección.



- Calificación: Puntuación obtenida.

#### **Relaciones:**

- Notas se relaciona con Estudiante (N:1).
- Notas se relaciona con Matricula(N:1)

#### **Entidad: Biblioteca**

**Descripción:** Representara todo libro virtual que esté disponible en la biblioteca

#### **Atributos:**

- LibroID(PK)Identificador único
- Titulo: nombre del libro
- Autor: autor del libro
- CategoriaLibroID: clasificación temática
- ArchivoPDF: URL o ubicación del archivo

#### **Entidad: Musica**

**Descripcion:** Representara toda musica virtual que este disponible en la biblioteca

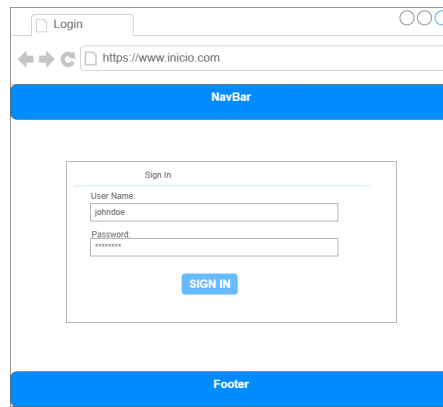
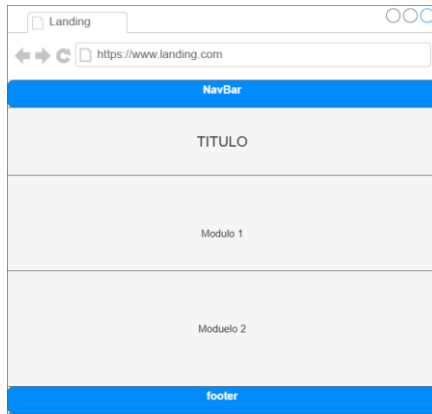
#### **Atributos:**

- MusicalID(PK)Identificador único
- Titulo: nombre del canción
- Autor: autor del canción
- CategoriaCancionID: clasificación temática
- ArchivoPDF: URL o ubicación del archivo

## **Mockups**

Landing page

Login

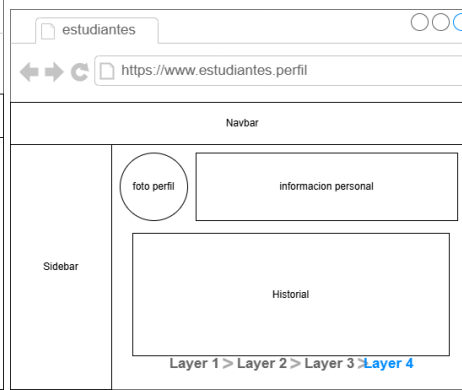


## Estudiantes

### Estructura a nivel global



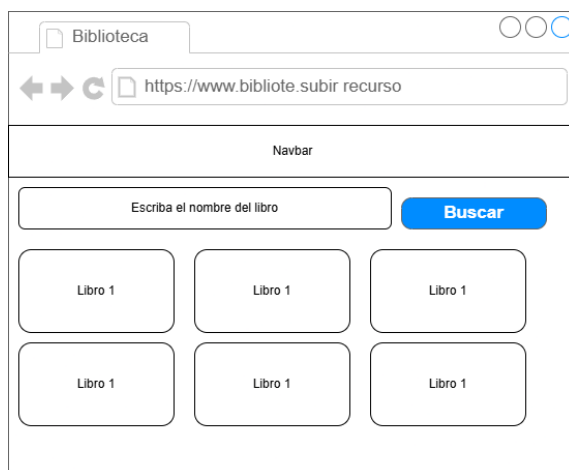
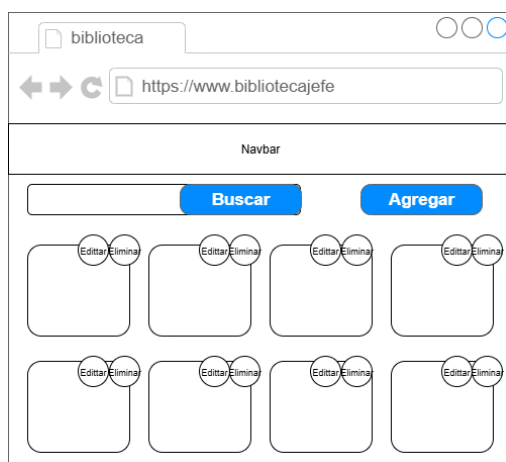
### Estructura mas especifica



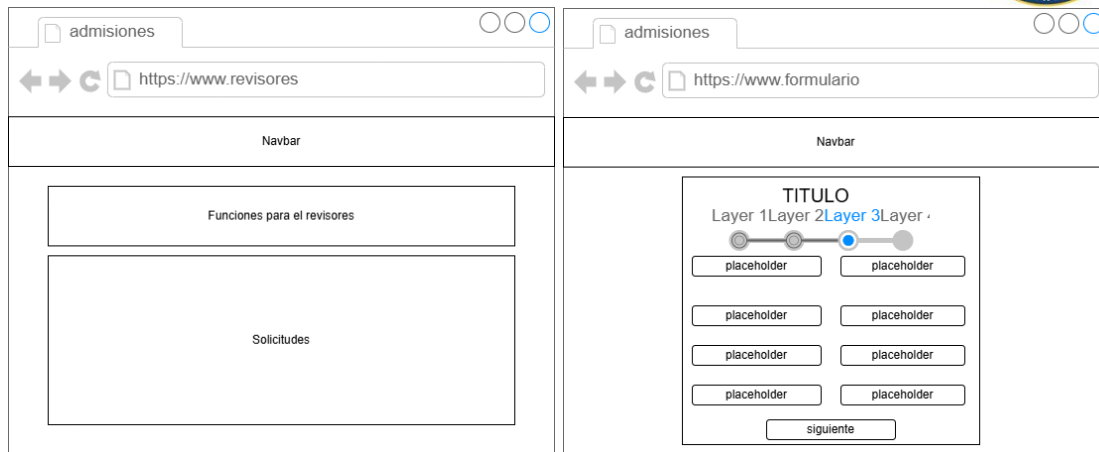
## Música



## Biblioteca

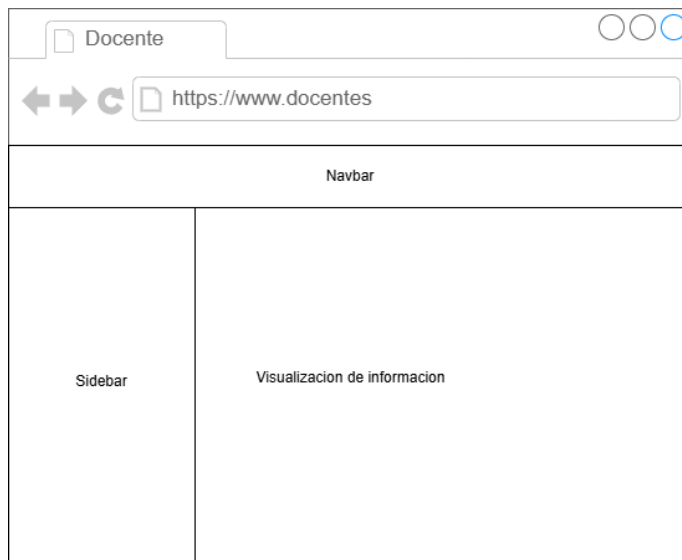


## Admisiones

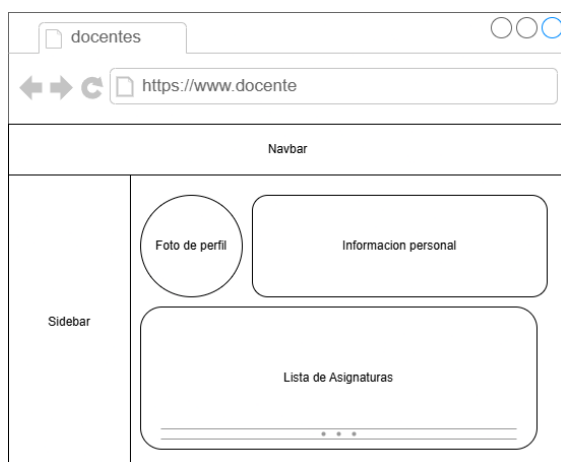


## Docentes

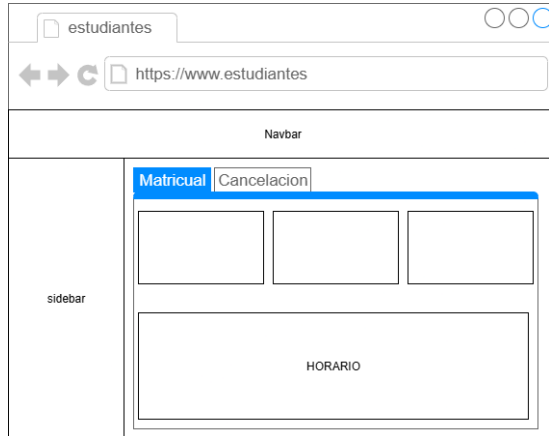
### Concepto amplio



### Concepto enfocado



## Matricula



## Glosario

**API (Interfaz de programación de aplicaciones)**

Una API es un conjunto de reglas y definiciones que permite que diferentes aplicaciones o sistemas se comuniquen entre sí. Se usa para que los desarrolladores puedan integrar funciones de otros servicios sin necesidad de conocer su código interno.

### **Autenticación**

Proceso mediante el cual un sistema verifica la identidad de un usuario antes de permitirle acceso a recursos protegidos.

### **Autorización**

Proceso que define los permisos que tiene un usuario una vez autenticado.

### **Actor**

Un actor es una entidad externa que interactúa con el sistema

### **Backend**

Parte de una aplicación que maneja la lógica, las bases de datos u el procesamiento de información. Es la capa que el usuario no ve.

### **Base de datos**

Un sistema que almacena organiza y gestiona información de manera estructurada para su consulta y manipulación eficiente.

### **CRUD (Create, Read, Update, Delete)**

Conjunto de operaciones básicas que se realizan sobre una base de datos.

### **Caso de Uso**

Representa una funcionalidad o acción específica que el sistema debe realizar en respuesta a una interacción con un actor.

### **CSV (Comma-Separated Values)**

Formato de archivo utilizado para almacenar datos tabulares, como una hoja de cálculo, donde los valores están separados por comas.

### **Escalabilidad**

Es la capacidad de un software para manejar un mayor número de usuarios o más datos a medida que crece, sin perder eficiencia.

### **Entidad**

En un diagrama entidad-relación (ERD), una entidad representa un objeto o concepto del mundo real con datos almacenados en una base de datos.

### **Extensión:**

Es una relación que indica que un caso de uso puede ser ampliado o modificado dependiendo de una condición especial.

**Escenario:** Situación específica en la que un usuario interactúa con el sistema para lograr un objetivo.

**Evento:** Acción o suceso que desencadena un cambio dentro de un sistema.

### **Microservicios**

Arquitectura donde una aplicación se divide en pequeños servicios independientes que se comunican entre sí.

### **Modelo Relacional**

Forma de organizar datos en bases de datos con tablas y relaciones.

### **MVC**

(Modelo-Vista-Controlador) es un patrón de diseño de software que separa una aplicación en tres componentes interconectados para mejorar la organización y la capacidad de mantenimiento.

### **Objeto**

Elemento del sistema que participa en la comunicación de mensajes dentro del diagrama.

### **Patrón de Diseño**

Solución reutilizable a problemas comunes en el desarrollo de software.

### **Inclusión**

Es una relación que indica que un caso de uso siempre incluirá otro caso de uso como parte de su ejecución.

### **Interfaz de usuario (UI)**

Es la parte del software con la que los usuarios interactúan, como botones, menús, pantallas, etc. Se asegura de que el software sea fácil de usar.

## **División de Tareas**

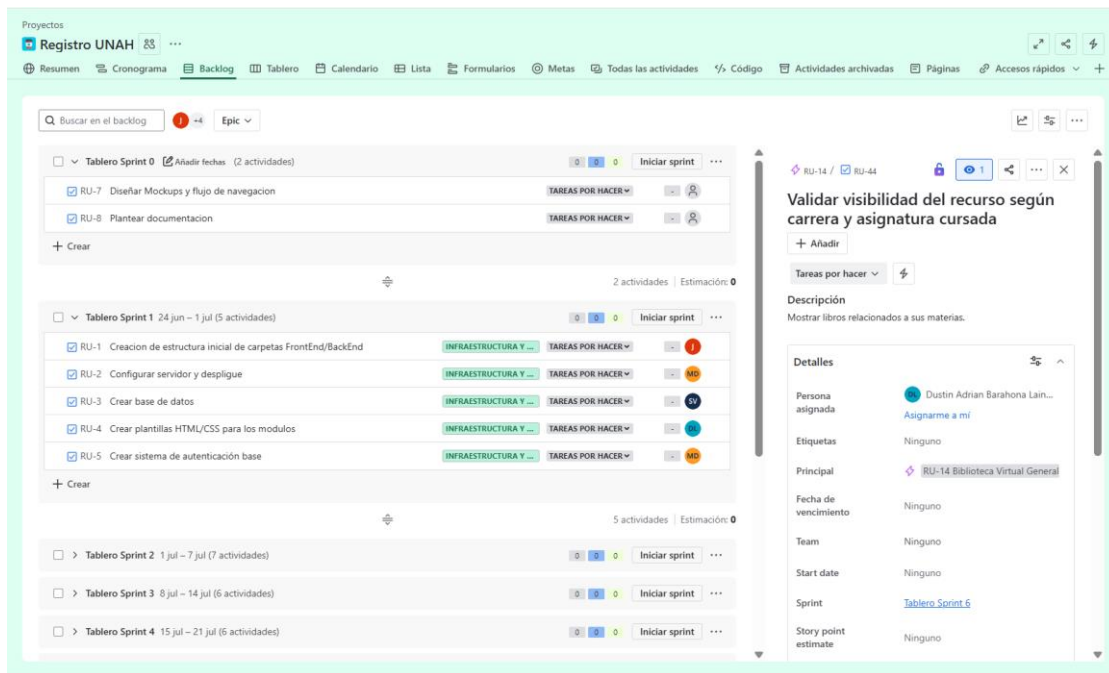
- Dustin Barahona– Frontend Developer
- Mauricio Valesquez - Backend Developer
- Saul - Backend Developer
- Josué Mejía – SCRUM MASTER - Frontend Developer

## Paletas de Colores

### Energética y Accesible

- Azul UNAH (#005F87) → (Cabeceras, botones clave)
- Amarillo vibrante (#FFC107) → (Para destacar acciones importantes)
- Blanco (#FFFFFF) → (Espacios limpios)
- Gris azulado (#E9ECEF) → (Fondos de formularios)
- Verde esmeralda (#28A745) → (Procesos exitosos)
- Rojo coral (#FF6B6B) → (Alertas críticas)

## Backlog



<https://is-project-unah-i-pac-2025.atlassian.net/jira/software/projects/RU/boards/3/backlog?selectedIssue=RU-44>

**Backlog Actualizado hasta 22 Julio**



Proyectos

Registro UNAH

Resumen Cronograma Backlog Tablero Calendario Lista Formularios Metas Todas las actividades Código Actividades archivadas Páginas More

Buscar en el backlog... +4 Epic

☐ Sprint 1 25 jun - 2 jul (7 actividades)  
Terminar todos los entregables para el sprint

☒ RU-1 Creacion de estructura inicial de carpetas FrontEnd/BackEnd INFRAESTRUCTURA ... FINALIZADA - J

☒ RU-2 Configurar servidor y despliegue INFRAESTRUCTURA ... FINALIZADA - MD

☒ RU-3 Crear base de datos INFRAESTRUCTURA ... FINALIZADA - SV

☒ RU-4 Crear plantillas HTML/CSS para los modulos INFRAESTRUCTURA ... FINALIZADA - J

☒ RU-5 Crear sistema de autenticación base INFRAESTRUCTURA ... FINALIZADA - MD

☒ RU-7 Diseñar Mockups y flujo de navegacion INFRAESTRUCTURA ... FINALIZADA - J

☒ RU-8 Plantear documentacion INFRAESTRUCTURA ... FINALIZADA - J

+ Crear

☐ Tablero Sprint 2 1 jul - 8 jul (7 actividades)  
Terminar todas las tareas

☒ RU-16 Crear formulario de inscripción a examen de admisión GESTION DE ADMISI... FINALIZADA - J

☒ RU-20 Crear módulo para revisión de solicitudes por revisor GESTION DE ADMISI... FINALIZADA - MD

☒ RU-17 Validar certificado (tipo, tamaño, resolución de imagen) GESTION DE ADMISI... FINALIZADA - J

☒ RU-18 Crear backend para almacenamiento de solicitudes en la BD GESTION DE ADMISI... FINALIZADA - MD

☒ RU-19 Implementar carga y distribución automática de solicitudes a revisores GESTION DE ADMISI... FINALIZADA - SV

☒ RU-21 Implementar rechazo con retroalimentación y corrección por estudiante GESTION DE ADMISI... FINALIZADA - SV

Proyectos

Registro UNAH

Resumen Cronograma Backlog Tablero Calendario Lista Formularios Metas Todas las actividades Código Actividades archivadas Páginas More

Buscar en el backlog... +4 Epic

☒ RU-22 Enviar resultados por correo al estudiante GESTION DE ADMISI... FINALIZADA - SV

+ Crear

☐ Tablero Sprint 3 8 jul - 14 jul (6 actividades)  
0 0 0 Iniciar sprint

☒ RU-25 Mostrar historial académico y generar certificado en PDF ESTUDIANTES EN CURSO - J

☒ RU-26 Integrar sistema de evaluación docente previo a visualizar notas ESTUDIANTES TAREAS POR HACER - SV

☒ RU-45 Crear cuenta automáticamente desde archivo CSV de admisiones ESTUDIANTES TAREAS POR HACER - MD

☒ RU-46 Generar número de cuenta y correo institucional ESTUDIANTES FINALIZADA - SV

☒ RU-47 Implementar sistema de contactos y solicitudes entre estudiantes ESTUDIANTES FINALIZADA - SV

☒ RU-48 Portal del estudiante ESTUDIANTES FINALIZADA - J

+ Crear

6 actividades | Estimación:

☐ Tablero Sprint 4 15 jul - 21 jul (5 actividades)  
0 0 0 Iniciar sprint

☒ RU-29 Activar matrícula desde administrador MATRICULA TAREAS POR HACER - MD

☒ RU-30 Generar calendario basado en índice académico MATRICULA TAREAS POR HACER - SV

☒ RU-32 Mostrar perfil del docente por sección MATRICULA FINALIZADA - J

☒ RU-33 Permitir cancelación libre de clases durante matrícula MATRICULA FINALIZADA - MD

Link a pagina web

<http://18.221.129.158/>

Modulos en funconamiento

- Admisiones
- Revisores
- Estudiantes



- Biblioteca al 30%

