

TRABAJO PRÁCTICO: CIERRE DEL PRIMER CUATRIMESTRE



INTEGRANTES:

Burdiles Ricardo Adrián

Lagos Lucas

Soto Agustín

Spagnolo Emiliano

ÍNDICE

○ Arquitectura del Sistema.....	3
○ Modelo de datos.....	3
○ Algunas rutas básicas:.....	6
○ Diagramas del Proyecto.....	7
○ Diagramas de casos de uso.....	7
○ Modelo Entidad-Relacion.....	8
○ Diagramas de Clases.....	9
○ Diagramas de Secuencia.....	10 a 12
○ Enlace a repositorio de Github.....	13
○ Bibliografía.....	13
○ Anexo I “links con capturas de rutas”	14

ARQUITECTURA DEL SISTEMA

Tecnología utilizada.

Tecnologías y Herramientas Utilizadas:

Node.js: Utilizado como servidor backend principal.

Express: Framework que facilita la gestión de rutas y solicitudes HTTP.

Joi: Librería utilizada para la validación de datos.

dotenv: Librería empleada para la gestión de variables de entorno.

mysql2: Librería utilizada para la conexión y manipulación de datos en una base de datos MySQL.

nodemon: Herramienta que reinicia automáticamente el servidor al detectar cambios en el código.

XAMPP: Plataforma utilizada para ejecutar una base de datos MySQL (MariaDB) local.

Postman: Herramienta que permite simular un cliente API y probar endpoints sin necesidad de desarrollar un frontend.

Se aplicaron principios MVC y SOLID, separando en tres partes la lógica de la aplicación, modularizado y dando independencia al sistema.

MODELO DE DATOS

Este sistema consta del siguiente modelo de datos:

El sistema descrito abarca la gestión de usuarios, materias, matrículas, tareas y entregas en un entorno educativo. Se identifican tres tipos de usuarios: profesores, alumnos y administradores, cada uno con un ID único y un rol específico. Las materias son asignadas a los profesores y no pueden tener nombres duplicados dentro de la misma asignatura. La matrícula vincula a los alumnos con las materias, prohibiendo inscripciones duplicadas. Además, se permiten la asignación de tareas con fechas de entrega y la recepción de entregas por parte de los alumnos, quienes son evaluados con calificaciones y comentarios.

Este sistema implementa un sistema SEED, que carga los usuarios administradores con sus credenciales genéricas las que luego deben ser cambiadas por los administradores del sistema.

1. Users (Usuarios)

Contiene a todos los usuarios del sistema: profesores, alumnos y administradores.

-
- **id_usuario** (PK): Identificador único.
 - **nombre**
 - **email** (único)
 - **username**
 - **password**
 - **rol**: `ENUM('profesor', 'alumno', 'admin')`
 - **activo**: estado del usuario.
-

2. Materia

Asignaturas dictadas por profesores.

- **id_materia** (PK)
- **nom_materia**
- **descripcion**
- **id_profesor** (FK → Users.id_usuario)
- **activo**

Restricción: Un profesor no puede tener dos materias con el mismo nombre.

3. Matricula

Relación entre **alumnos** y **materias**.

- **id_matricula** (PK)
-

-
- **id_usuario** (FK → Users.id_usuario)
 - **id_materia** (FK → Materia.id_materia)

Restricción: Un alumno no puede estar matriculado dos veces a la misma materia.

4. Tarea

Tareas asignadas en una materia.

- **id_tarea** (PK)
 - **titulo**
 - **descripcion**
 - **fecha_entrega**
 - **id_materia** (FK → Materia.id_materia)
-

5. Entrega

Entregas de tareas por parte de los alumnos.

- **id_entrega** (PK)
 - **fecha_entrega**
 - **id_tarea** (FK → Tarea.id_tarea)
 - **id_usuario** (FK → Users.id_usuario)
 - **calificacion**
 - **comentario**
-

ALGUNAS RUTAS BÁSICAS:

Endpoints de materia: se utilizó query string

- GET/api/materia?id_materia=2 “búsqueda de alumnos por materia”
- GET/api/materia “listado materias”
- GET/api/materia/id “materia por id”
- PATCH/api/materia/id “actualiza materia por id”
- POST/api/materia/ “crea materia”

Endpoints de tarea: se utilizó query string

- GET/api/tarea?id_usuario=2 “búsqueda de tarea por alumno”
- GET/api/tarea “listado de tareas”
- PATCH/api/tarea/id “actualiza tarea por id”
- POST/api/tarea/ “crea materia”

Endpoints de users: se utilizó query string

- GET/api/api/users?rol=profesor “búsqueda de profesores o alumnos en usuarios”
- GET/api/users “listado todos los usuarios”
- GET/api/users/id “usuarios por id”
- PATCH/api/users/id “actualiza usuario por id”
- POST/api/users/ “crear nuevo usuario”

Endpoints de matrícula: se utilizó query string

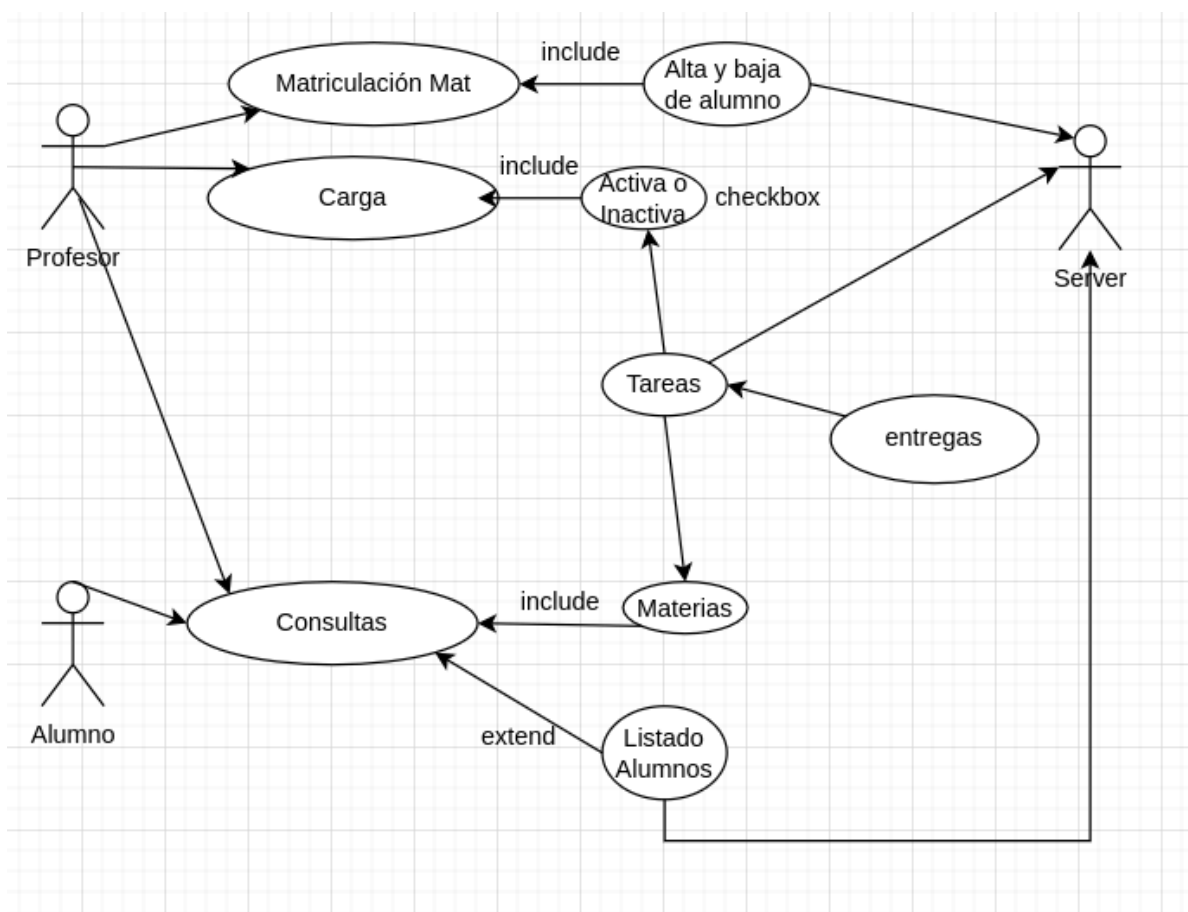
- GET/api/matricula?id_usuario=id “búsqueda de matrículas por alumno”
- GET/api/matricula “listado todos las matrículas”
- GET/api/matricula/id “matrícula por id”
- POST/api/matricula/ “crea nueva matrícula”

Endpoints de entrega: se utilizó query string

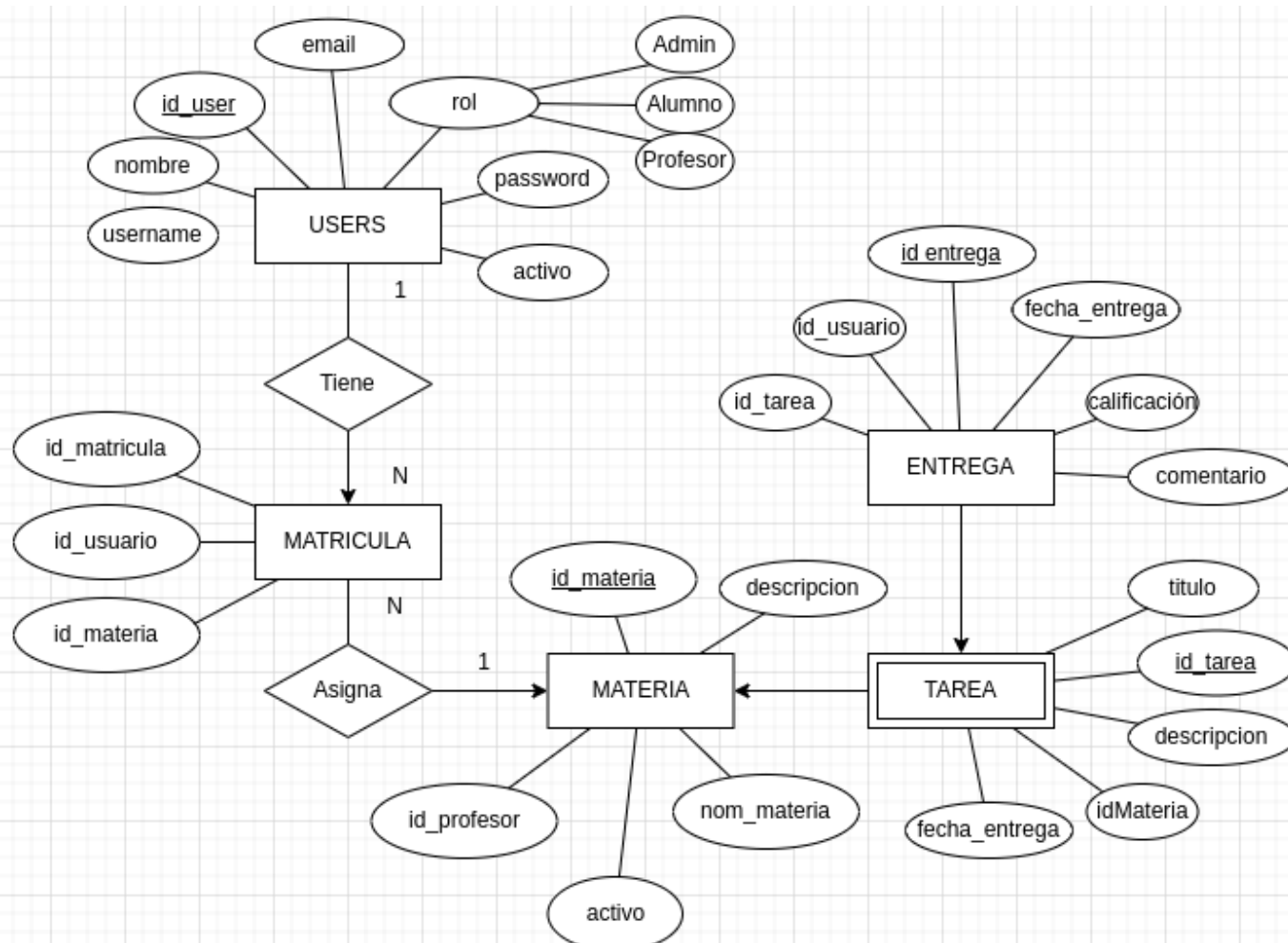
- GET/api/entrega?id_materia=id “búsqueda de entregas por materias”
- GET/api/entrega “listado todos las entrega”
- GET/api/entrega/id “entrega por id”
- GET/api/entrega/alumno/2/entregas “búsqueda de entregas por alumno”
- POST/api/entrega “crea nueva entrega”

DIAGRAMAS DEL PROYECTO

1. Diagrama de Casos de Uso:



2. Modelo Entidad-Relación:



USERS: (id_user, email, rol, nombre, username, password, activo)

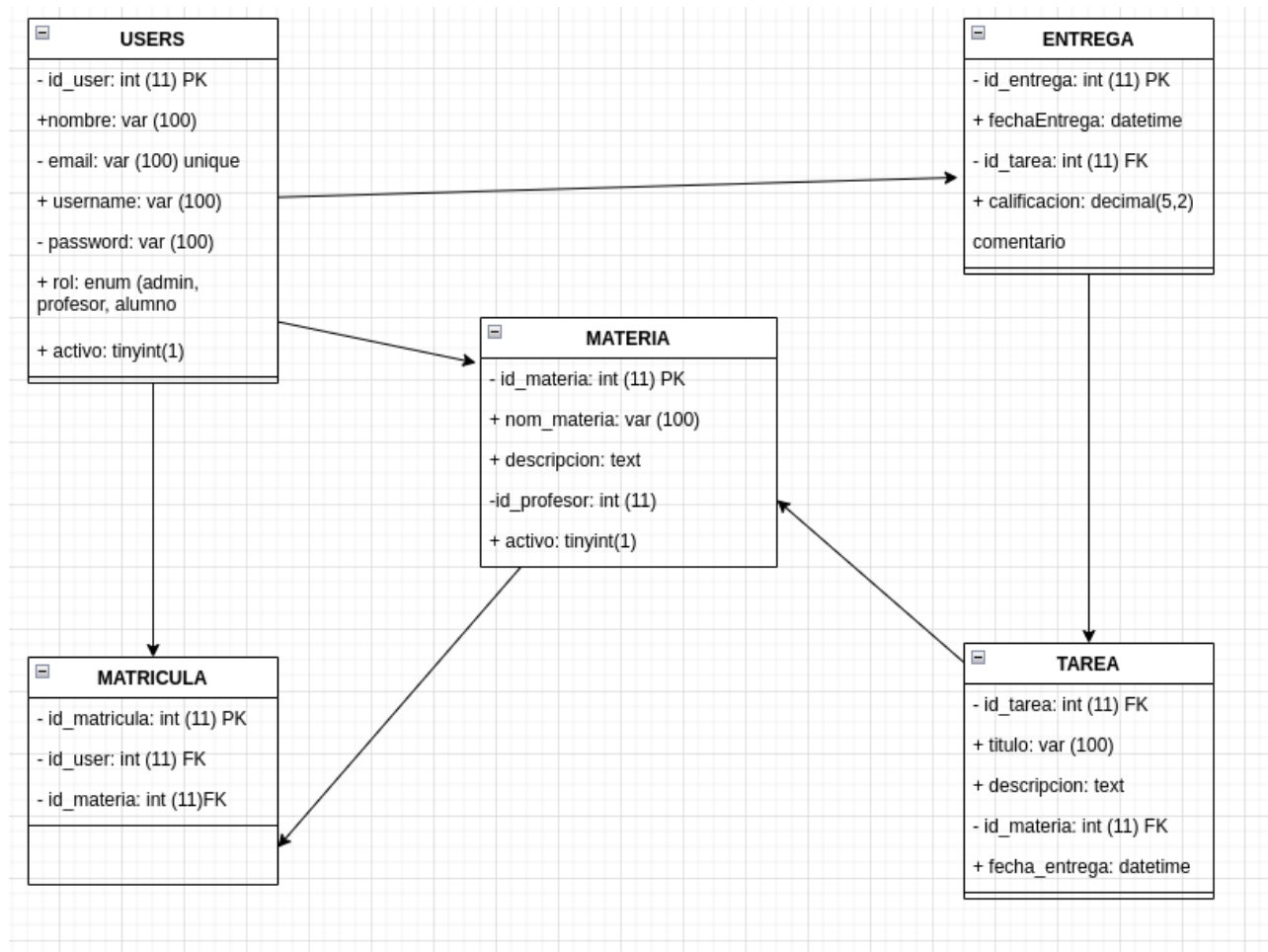
MATRICULA: (id_matricula, id_usuario, id_materia)

MATERIA: (id_materia, nom_materia, descripción, id_profesor(id_usuario), activo)

ENTREGA: (id entrega, fecha_entrega, id_tarea, id_usuario, calificacion, comentario)

TAREA: (id tarea, titulo, descripción, id_materia, fecha_entrega)

3. Diagrama de Clases Simplificado:

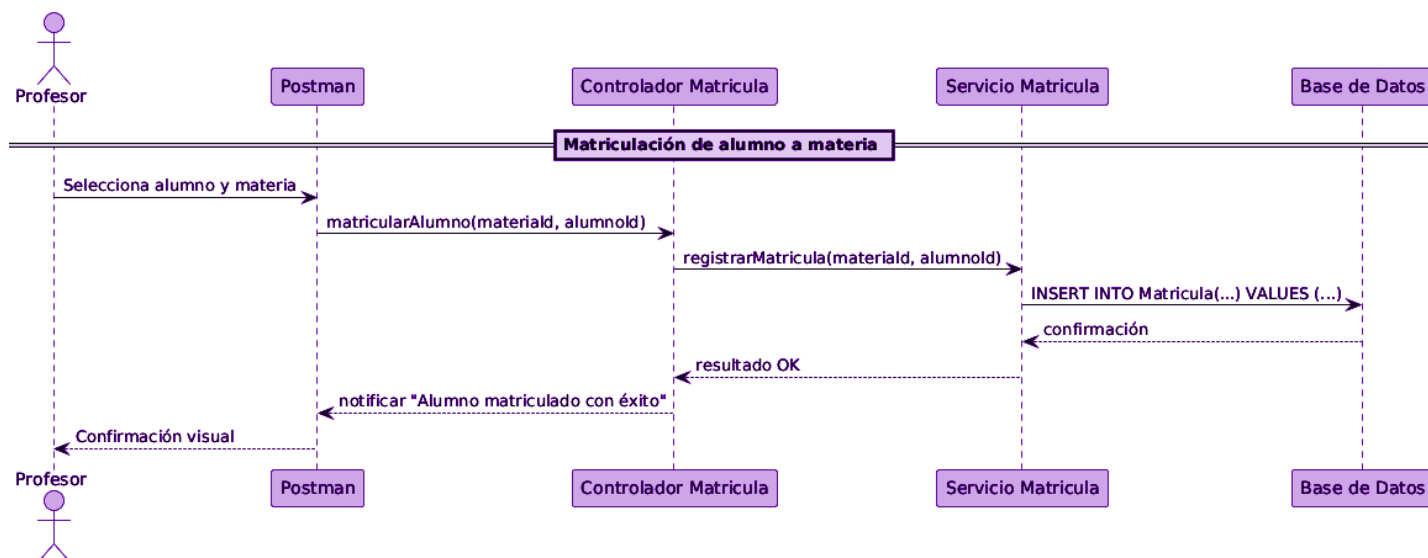


4. Diagramas de Secuencia:

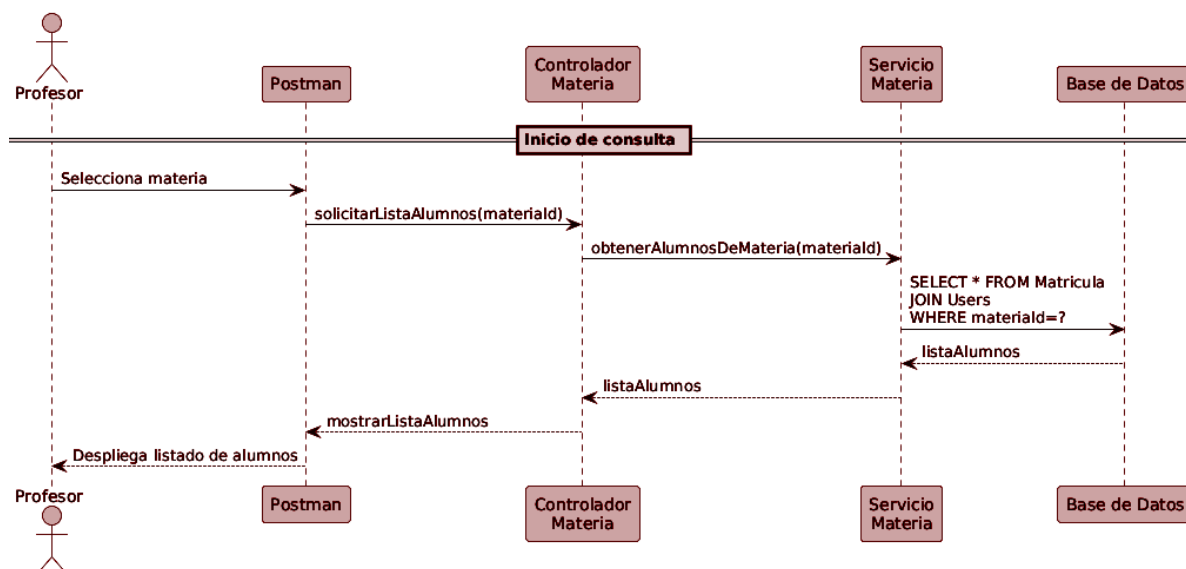
*Aclaración: este proyecto no posee frontend, en un sistema con frontend lo correcto es que cada participante realice una operación a través de una interfaz de usuario. En los siguientes diagramas de secuencia, en lugar de interfaz de usuario para los participantes se graficaron clientes API, como los utilizados para las consultas en este proyecto.

Para el profesor

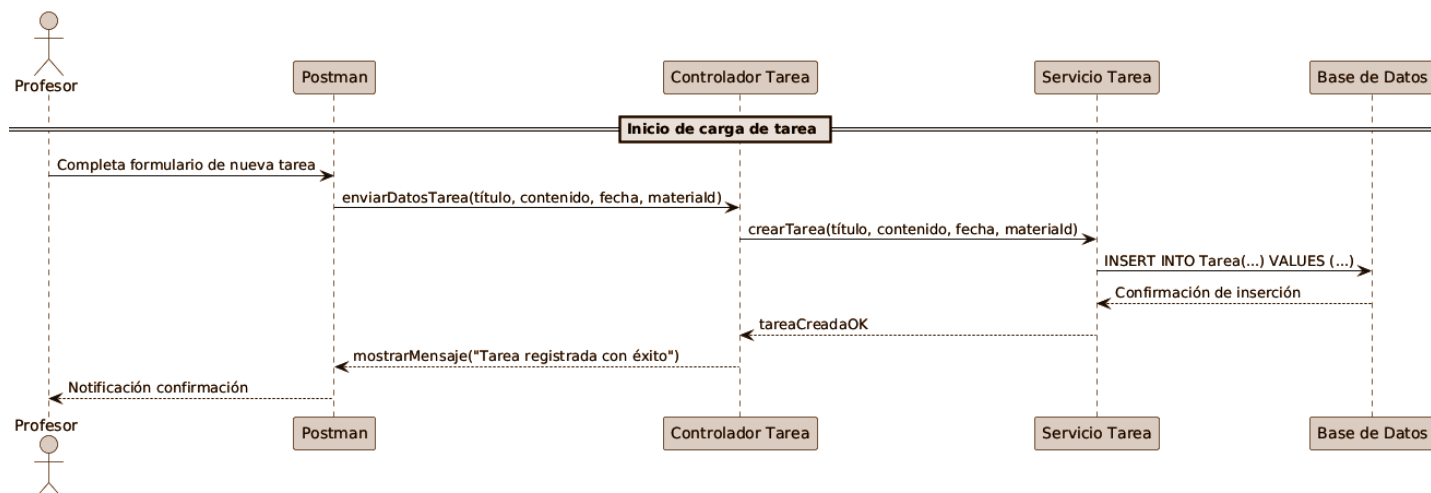
- Matricular alumno en una materia:



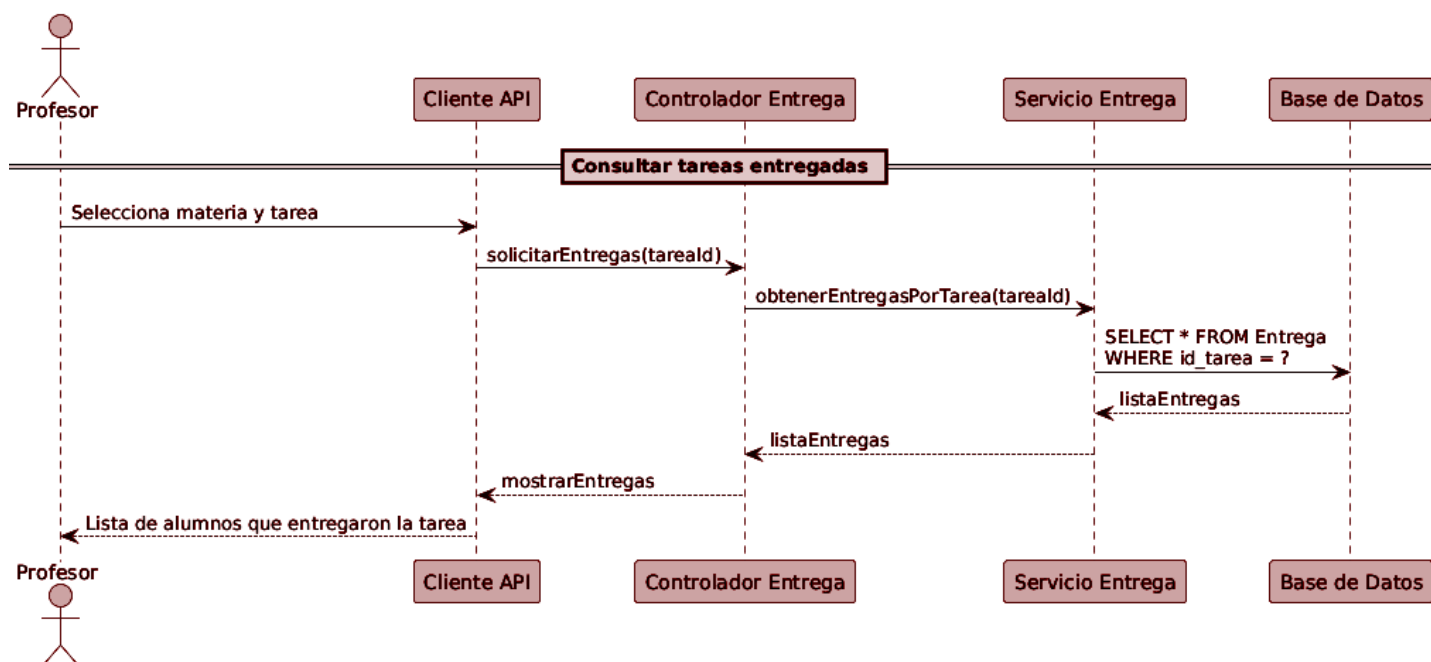
- Consultar qué alumnos están en cada materia:



-Profesor carga una tarea nueva a una materia:

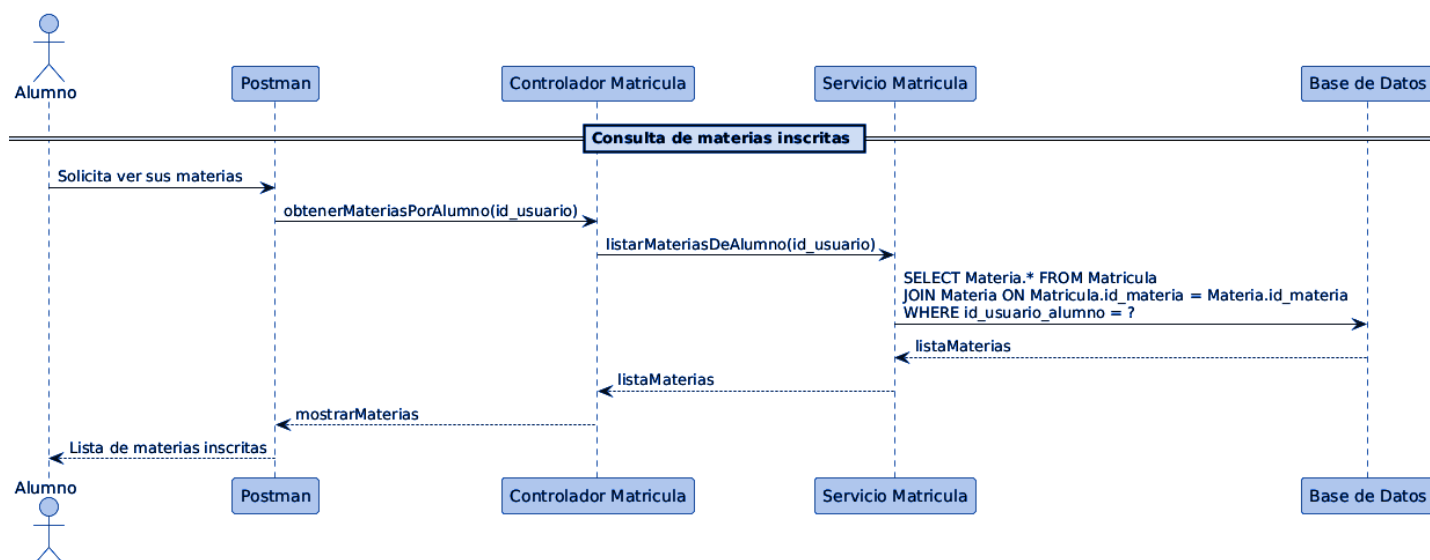


-Profesor consulta las tareas entregadas por los alumnos:

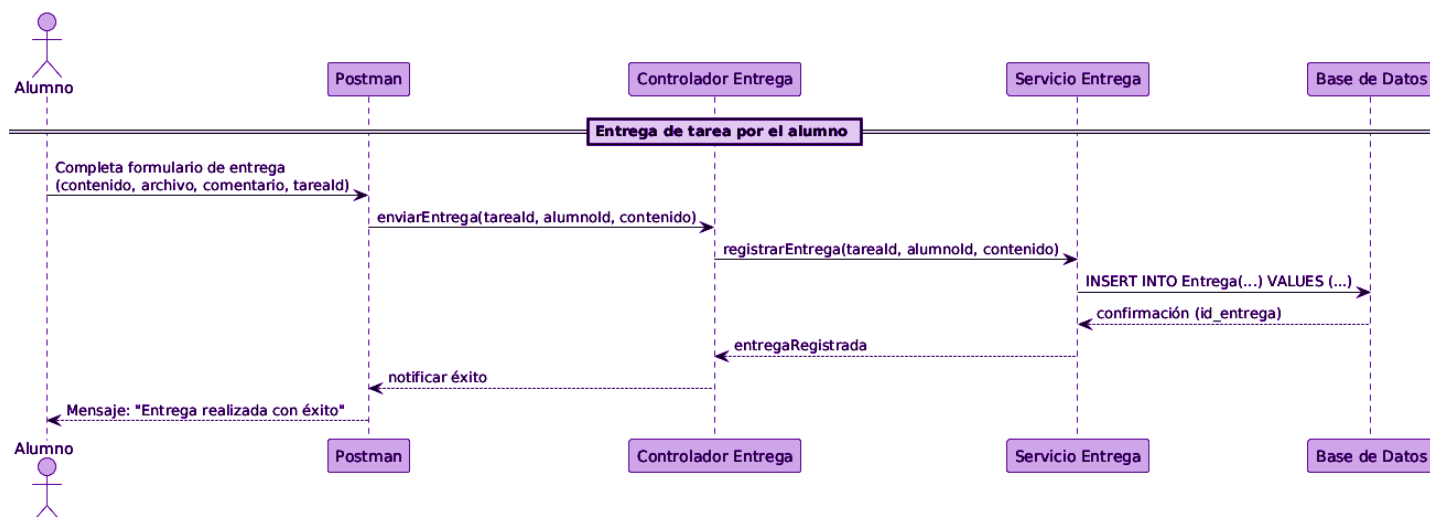


Para el alumno

-Alumno consulta las materias en las que está matriculado:



-Alumno entrega una tarea asignada en una materia:



ENLACE A REPOSITORIO DE GITHUB:

archivo README de repositorio:

Uso

Para iniciar realiza los siguientes pasos:

```
``bash
```

1) Crear la base de datos, vacía solo el nombre en XAMPP o medio ambiente elegido.

2) Crear un archivo .env y agregar el nombre de la DB, Hacer lo mismo en .envs.

3) iniciar el servidor

```
npm run dev
```

-- al iniciar el servidor se ejecutará el archivo SEED2 desde index.js. Este SEED genera el contenido de la DB completa

-- incluyendo el admin con las claves genéricas "que deben ser cambiadas al ingresar".

Repositorio en github:

<https://github.com/Adrian853tp/escuela3.git>

BIBLIOGRAFÍA:

Introducción a la programación Backend. (Clase n° 1)

Material de estudio (Clase n° 1)

Tipos de API's (Clase n° 4)

Variables de entorno (Material de clase virtual n° 4)

Conectándonos a Mysql (Material de clase n°5)

Trabajando con el CRUD (Clase n°6)

Anexo I

Links de capturas de pantallas con algunas rutas seleccionadas:

<https://www.notion.so/Probar-los-endpointst-2213945611b680c29e05fbb059401231>

Se adjunta el script de la Base de datos sql del proyecto.

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Servidor: localhost
-- Tiempo de generación: 30-06-2025 a las 22:42:21
-- Versión del servidor: 10.4.32-MariaDB
-- Versión de PHP: 8.1.25

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `escuela12`
--

--
-- Estructura de tabla para la tabla `entrega`
--

CREATE TABLE `entrega` (
```

```
`id_entrega` int(11) NOT NULL,
`fecha_entrega` datetime NOT NULL,
`id_tarea` int(11) NOT NULL,
`id_usuario` int(11) NOT NULL,
`calificacion` decimal(5,2) DEFAULT NULL,
`comentario` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `entrega`
--

INSERT INTO `entrega` (`id_entrega`, `fecha_entrega`, `id_tarea`, `id_usuario`, `calificacion`, `comentario`)
VALUES
(1, '2025-06-30 16:30:22', 2, 2, 8.50, 'Buen trabajo, Maria'),
(2, '2025-06-30 16:30:22', 1, 2, 9.00, 'Excelente resumen'),
(3, '2025-06-30 16:30:22', 4, 3, 7.00, 'Funciona correctamente'),
(4, '2025-06-30 16:30:22', 3, 5, 9.50, 'Análisis profundo y bien argumentado');

-----

--
-- Estructura de tabla para la tabla `materia`
--

CREATE TABLE `materia` (
  `id_materia` int(11) NOT NULL,
  `nom_materia` varchar(100) NOT NULL,
  `descripcion` text DEFAULT NULL,
  `id_profesor` int(11) NOT NULL,
  `activo` tinyint(1) DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `materia`
--

INSERT INTO `materia` (`id_materia`, `nom_materia`, `descripcion`, `id_profesor`, `activo`) VALUES
(1, 'Matemáticas', 'Curso introductorio a las matemáticas', 1, 1),
(2, 'Historia Universal', 'Un recorrido por la historia del mundo', 1, 1),
(3, 'Programación I', 'Introducción a la programación con Python', 6, 1),
(4, 'Literatura Española', 'Análisis de obras literarias españolas', 6, 1);

-----
```

--

-- Estructura de tabla para la tabla `matricula`

--

```
CREATE TABLE `matricula` (  
  `id_matricula` int(11) NOT NULL,  
  `id_usuario` int(11) NOT NULL,  
  `id_materia` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Volcado de datos para la tabla `matricula`

--

```
INSERT INTO `matricula` (`id_matricula`, `id_usuario`, `id_materia`) VALUES  
(2, 2, 1),  
(1, 2, 2),  
(3, 3, 2),  
(4, 3, 4),  
(5, 5, 3);
```

--

-- Estructura de tabla para la tabla `tarea`

--

```
CREATE TABLE `tarea` (  
  `id_tarea` int(11) NOT NULL,  
  `titulo` varchar(100) NOT NULL,  
  `descripcion` text DEFAULT NULL,  
  `fecha_entrega` datetime DEFAULT (curdate() + interval 15 day),  
  `id_materia` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Volcado de datos para la tabla `tarea`

--

```
INSERT INTO `tarea` (`id_tarea`, `titulo`, `descripcion`, `fecha_entrega`, `id_materia`) VALUES  
(1, 'Entrega 1: Álgebra', 'Ejercicios de álgebra básica', '2025-07-15 00:00:00', 2),  
(2, 'Resumen: Revolución Francesa', 'Resumen detallado de la Revolución Francesa', '2025-07-15 00:00:00',  
1),
```



```
(3, 'Proyecto 1: Hola Mundo', 'Escribir un programa \"Hola Mundo\" en Python', '2025-07-15 00:00:00', 4),
(4, 'Análisis: El Quijote', 'Análisis de personajes y temas en El Quijote', '2025-07-15 00:00:00', 3);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `users`
```

```
--
```

```
CREATE TABLE `users` (
  `id_usuario` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `username` varchar(100) DEFAULT NULL,
  `password` varchar(100) DEFAULT NULL,
  `rol` enum('profesor','alumno','admin') NOT NULL,
  `activo` tinyint(1) DEFAULT 1
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
```

```
-- Volcado de datos para la tabla `users`
```

```
--
```

```
INSERT INTO `users` (`id_usuario`, `nombre`, `email`, `username`, `password`, `rol`, `activo`) VALUES
(1, 'Juan Pérez', 'juan.perez@example.com', NULL, NULL, 'profesor', 1),
(2, 'Maria Gómez', 'maria.gomez@example.com', NULL, NULL, 'alumno', 1),
(3, 'Carlos López', 'carlos.lopez@example.com', NULL, NULL, 'alumno', 1),
(4, 'Ana Rodriguez', 'ana.rodriguez@example.com', NULL, NULL, 'admin', 1),
(5, 'Pedro Sánchez', 'pedro.sanchez@example.com', NULL, NULL, 'alumno', 1),
(6, 'Laura Martínez', 'laura.martinez@example.com', NULL, NULL, 'profesor', 1),
(7, 'Claudio Bidau', 'claudiobidau@universidad.com', 'claudio', 'claudio123', 'admin', 1),
(8, 'Lucas Lagos', 'lucas@universidad.com', 'lucas', 'lucas123', 'profesor', 1),
(9, 'Emiliano Spagnolo', 'emiliano@universidad.com', 'emiliano', 'emiliano123', 'profesor', 1),
(10, 'Adrian Burdiles', 'adrian@universidad.com', 'adrian', 'adrian123', 'profesor', 1),
(11, 'Agustin Soto', 'agustin@universidad.com', 'agustin', 'agustin123', 'profesor', 1);
```

```
--
```

```
-- Índices para tablas volcadas
```

```
--
```

```
--
```

```
-- Indices de la tabla `entrega`
```

```
--
```

```
ALTER TABLE `entrega`
```

```
ADD PRIMARY KEY (`id_entrega`),
ADD UNIQUE KEY `id_tarea` (`id_tarea`,`id_usuario`),
ADD KEY `id_usuario` (`id_usuario`);

--
-- Indices de la tabla `materia`
--
ALTER TABLE `materia`
  ADD PRIMARY KEY (`id_materia`),
  ADD UNIQUE KEY `id_profesor` (`id_profesor`,`nom_materia`);

--
-- Indices de la tabla `matricula`
--
ALTER TABLE `matricula`
  ADD PRIMARY KEY (`id_matricula`),
  ADD UNIQUE KEY `id_usuario` (`id_usuario`,`id_materia`),
  ADD KEY `id_materia` (`id_materia`);

--
-- Indices de la tabla `tarea`
--
ALTER TABLE `tarea`
  ADD PRIMARY KEY (`id_tarea`),
  ADD KEY `id_materia` (`id_materia`);

--
-- Indices de la tabla `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id_usuario`),
  ADD UNIQUE KEY `email` (`email`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `entrega`
--
ALTER TABLE `entrega`
  MODIFY `id_entrega` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
```

```
-- AUTO_INCREMENT de la tabla `materia`
--
ALTER TABLE `materia`
  MODIFY `id_materia` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT de la tabla `matricula`
--
ALTER TABLE `matricula`
  MODIFY `id_matricula` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT de la tabla `tarea`
--
ALTER TABLE `tarea`
  MODIFY `id_tarea` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT de la tabla `users`
--
ALTER TABLE `users`
  MODIFY `id_usuario` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `entrega`
--
ALTER TABLE `entrega`
  ADD CONSTRAINT `entrega_ibfk_1` FOREIGN KEY (`id_tarea`) REFERENCES `tarea` (`id_tarea`),
  ADD CONSTRAINT `entrega_ibfk_2` FOREIGN KEY (`id_usuario`) REFERENCES `users` (`id_usuario`);

--
-- Filtros para la tabla `materia`
--
ALTER TABLE `materia`
  ADD CONSTRAINT `materia_ibfk_1` FOREIGN KEY (`id_profesor`) REFERENCES `users` (`id_usuario`);

--
-- Filtros para la tabla `matricula`
--
ALTER TABLE `matricula`
```

```
ADD CONSTRAINT `matricula_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `users` (`id_usuario`),  
ADD CONSTRAINT `matricula_ibfk_2` FOREIGN KEY (`id_materia`) REFERENCES `materia`  
(`id_materia`);
```

```
--
```

```
-- Filtros para la tabla `tarea`
```

```
--
```

```
ALTER TABLE `tarea`
```

```
  ADD CONSTRAINT `tarea_ibfk_1` FOREIGN KEY (`id_materia`) REFERENCES `materia` (`id_materia`);  
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```