Contenido

[1 Introducción y descripción general 3](#_Toc153782622)

[1.1 Introducción 3](#_Toc153782623)

[1.2 Objetivos a cumplir 3](#_Toc153782624)

[2 Diagramas 4](#_Toc153782625)

[2.1 Infraestructura 4](#_Toc153782626)

[2.2 Entidad – relación 5](#_Toc153782627)

[3 Manual técnico 6](#_Toc153782628)

[3.1 Como empezar 6](#_Toc153782629)

[3.1.1 Clonar los repositorios. 6](#_Toc153782630)

[3.1.2 Instalar las dependencias 6](#_Toc153782631)

[3.1.3 Configurar base de datos 6](#_Toc153782632)

[3.1.4 Configurar las variables de entorno 6](#_Toc153782633)

[3.2 Front-end 6](#_Toc153782634)

[3.2.1 Tecnologías usadas: 6](#_Toc153782635)

[3.2.2 Implementación del código 7](#_Toc153782636)

[3.3 Back-end 20](#_Toc153782637)

[3.3.1 Tecnologías usadas: 20](#_Toc153782638)

[3.3.2 Implementación del código 20](#_Toc153782639)

[3.4 Integración de inteligencia artificial 23](#_Toc153782640)

[3.4.1 Creación de ejercicios 23](#_Toc153782641)

[3.4.2 Registro de asistencia con reconocimiento facial 26](#_Toc153782642)

[4 Manual de usuario 32](#_Toc153782643)

[4.1 Acceso al campus virtual 32](#_Toc153782644)

[4.2 Configurar el campus 36](#_Toc153782645)

[4.3 Cómo inscribir a un estudiante 37](#_Toc153782646)

[4.4 Cargar notas a un estudiante 38](#_Toc153782647)

[4.5 Realizar ejercicios 40](#_Toc153782648)

[4.6 Entregables 42](#_Toc153782649)

[4.6.1 Crear un entregable 42](#_Toc153782650)

[4.6.2 Cómo el estudiante debe enviar su tarea para corregir 43](#_Toc153782651)

[4.6.3 Cómo el profesor debe corregir los entregables 44](#_Toc153782652)

[4.7 Foro de discusión 45](#_Toc153782653)

[4.7.1 Crear discusión 45](#_Toc153782654)

[4.7.2 Añadir comentarios a una discusión 46](#_Toc153782655)

[4.8 Perfil del usuario 47](#_Toc153782656)

[4.8.1 Su progreso 47](#_Toc153782657)

[4.8.2 Configuración de su perfil 48](#_Toc153782658)

[4.8.3 Activar modo accesibilidad 48](#_Toc153782659)

[5 Conclusión 49](#_Toc153782660)

[6 Códigos QR con los repositorios 49](#_Toc153782661)

[6.1 Front-end 49](#_Toc153782662)

[6.2 Back-end 50](#_Toc153782663)

[6.3 Control de asistencia 50](#_Toc153782664)

# Introducción y descripción general

## Introducción

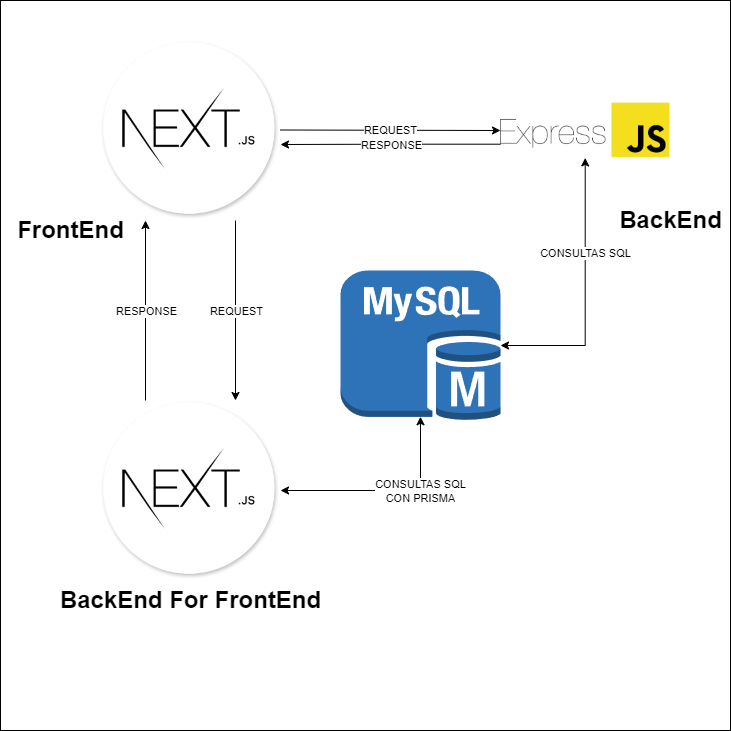
El proyecto consiste en un Campus Virtual con el propósito general de modernizar y optimizar la experiencia de aprendizaje, trasladando el contenido académico a un formato digital/virtual. Este ambicioso proyecto busca cumplir diversos objetivos con el fin de ofrecer una educación más accesible, interactiva y eficiente, tanto para el estudiante como para el profesor.

## Objetivos a cumplir

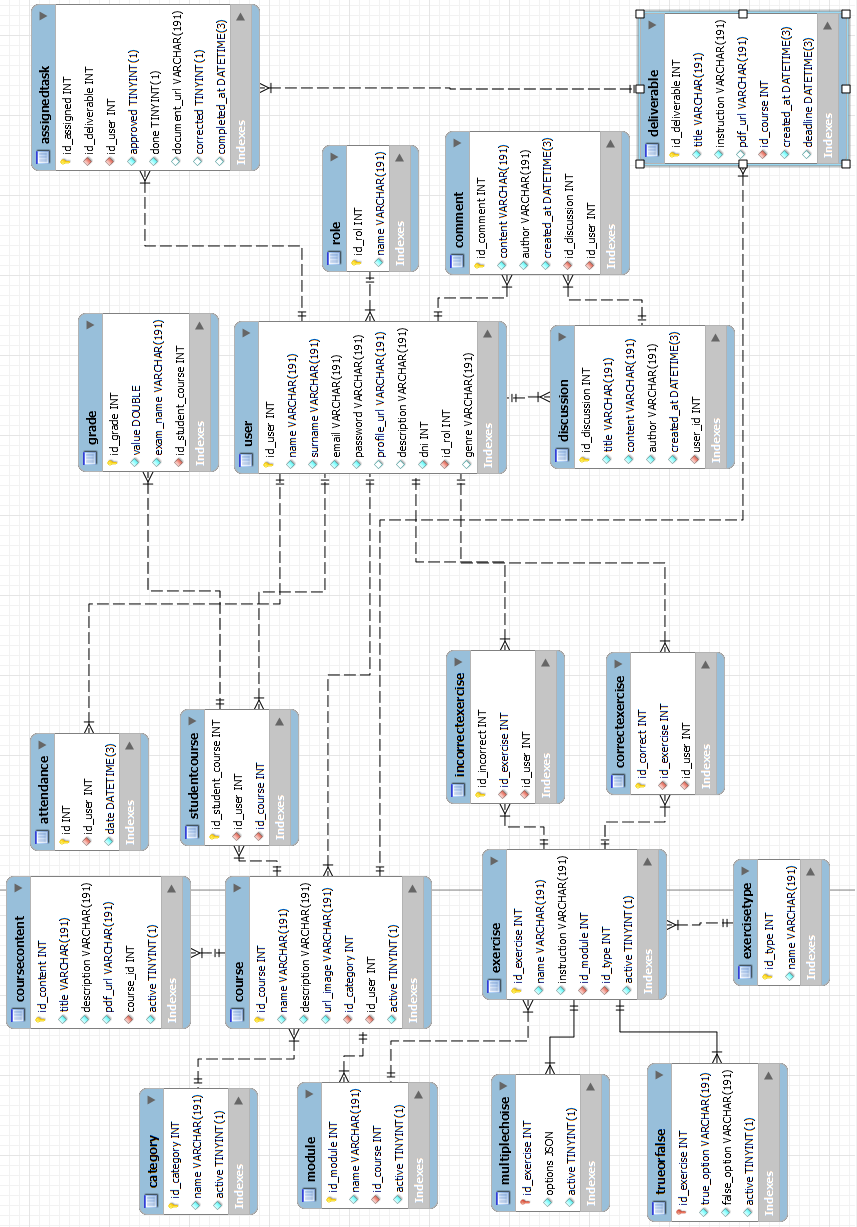
* Transferencia de Contenido Académico: El campus virtual se enfoca en migrar el contenido educativo tradicional a un entorno digital, proporcionando a estudiantes a profesores acceso fácil y rápido a recursos digitales actualizados.
* Mejora la Experiencia de Aprendizaje: La plataforma ofrece recursos interactivos que enriquecen la experiencia de aprendizaje de los estudiantes, promoviendo un enfoque más dinámico y participativo en la adquisición de conocimientos.
* Automatización de Creación y Corrección de Ejercicios: Agilizando el proceso de evaluación, el Campus Virtual incorpora sistemas automáticos de corrección para los ejercicios previamente cargados por los docentes, ya sea cargados a mano o generados a través de Inteligencia Artificial solo introduciendo el tema del ejercicio, permitiendo una retroalimentación más rápida y eficiente.
* Colaboración y Comunicación: Se fomenta la colaboración entre estudiantes y profesores a través de un foro de discusión, facilitando la interacción y el intercambio de ideas para enriquecer el proceso de aprendizaje.
* Seguimiento de Progreso: La plataforma ofrece monitorear el progreso de los ejercicios, proporcionando una visión clara del rendimiento individual.
* Acceso a Recursos Bibliográficos. Abandonando los métodos convencionales, el Campus Virtual garantiza un acceso rápido y sencillo a recursos bibliográficos en formato digital, mejorando la disponibilidad de materiales de estudio.
* Simplificación a procesos administrativos: Se busca facilitar la creación de cursos, módulos, la inscripción, seguimiento de asistencia, la carga de notas y cálculo de promedios.

# Diagramas

## Infraestructura



## Entidad – relación



# Manual técnico

## Como empezar

### Clonar los repositorios.

* Front-end: git clone <https://github.com/TommySosa/campus>
* Back-end: git clone <https://github.com/TommySosa/campus-api>
* Asistencia: git clone <https://github.com/TommySosa/control-asistencia-ia>

### Instalar las dependencias

* En las carpetas de los proyectos tanto del Front-end como del Back-end se tiene que ejecutar el comando: ***npm install.***
* En el proyecto de asistencia se tiene que instalar una por una las dependecias que tiene en las primeras líneas, por ejemplo: ***py install cv2.***

### Configurar base de datos

* Se tiene que ejecutar los siguientes comandos en el proyecto del Front-end: ***npx prisma generate*** y ***npx prisma migrate dev.***
* Una vez creada la base de datos se tiene que ejecutar las siguientes consultas:
* insert into role(name) values ("Estudiante");
* insert into role(name) values ("Profesor");
* insert into category(name) values ('English');
* insert into exercisetype(name) values('Multiple Choise');
* insert into exercisetype(name) values('Verdadero o falso');

### Configurar las variables de entorno

* Chequear y configurar con sus datos las variables de entorno en el archivo .ENV

## Front-end

### Tecnologías usadas:

* NextJs como framework que se basa en ReactJs.
* TailwindCSS como librería de estilos de CSS.
* React-hook-forms para validaciones en los formularios.
* Axios para peticiones HTTP.
* Chart.js para la visualización de los datos del progreso mediante gráficos.
* SweetAlert2 para mostrar mensajes Toast.

### Implementación del código

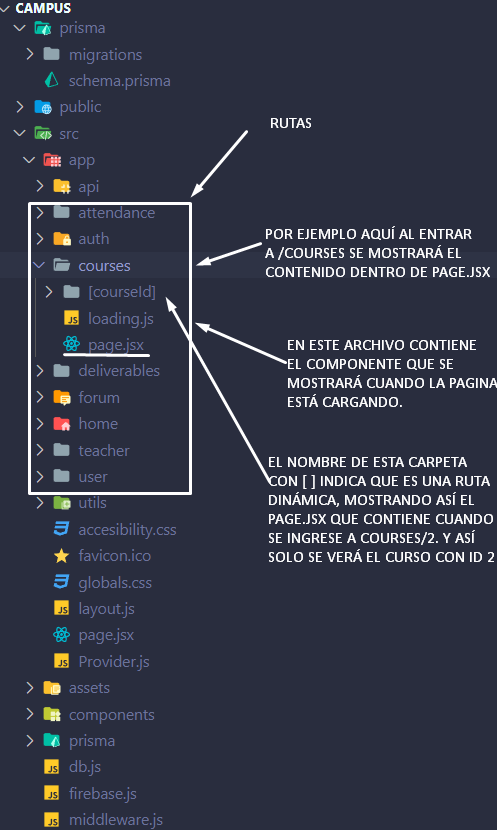
Decidí usar NextJs como framework en el FrontEnd por la capacidad de tener un BFF (Backend for FrontEnd) que me permite crear endpoints de cierta forma más rápida, solamente teniendo una carpeta /api donde dentro de esa carpeta se tiene que crear otro directorio con el nombre de la ruta, por ejemplo /api/courses.

Por ejemplo, aquí se muestra un GET que hace la petición a mi api de Express, brindando así otra capa de seguridad, enmascarando la URL real del endpoint.



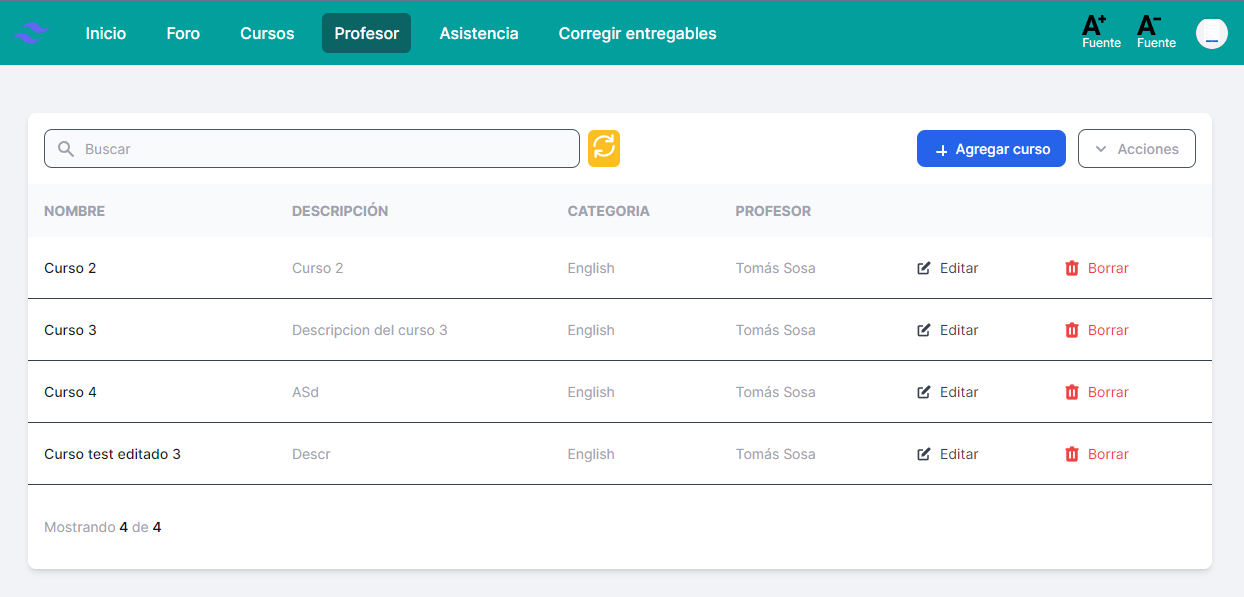
Nextjs también facilita un sistema de rutas muy fácil de usar, ya que solo con crear una carpeta con el nombre de nuestra ruta dentro de la carpeta src/app, y creando un archivo llamado page.jsx ya le “decimos” a nextjs que nuestro sistema web tendrá una página en esa dirección.

Por ejemplo, aquí se muestra el sistema de carpetas del proyecto con sus respectivas rutas:

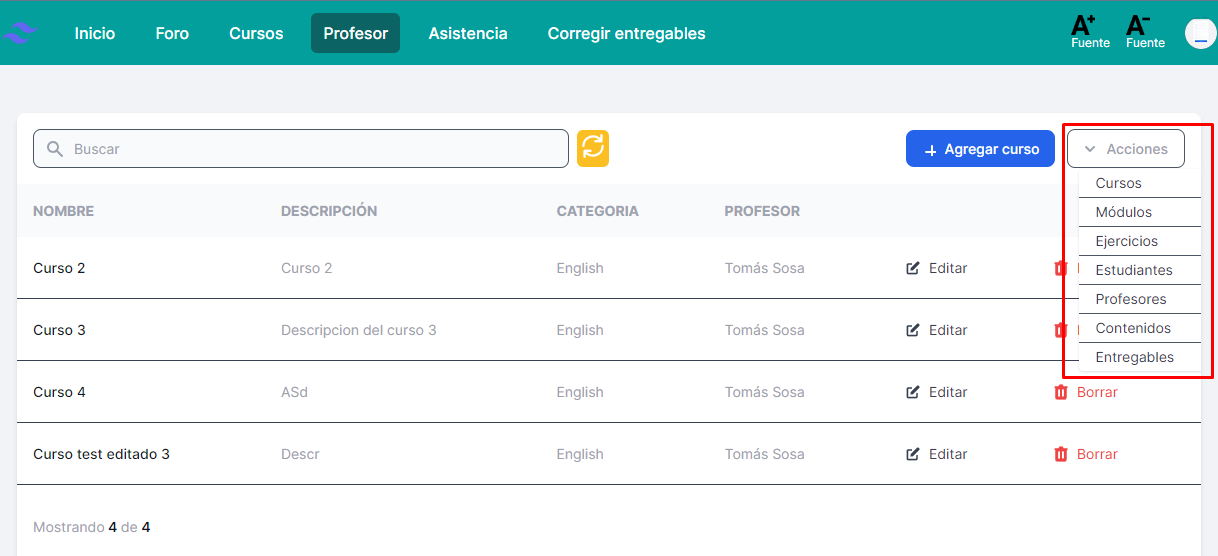


#### Vista rol profesor:

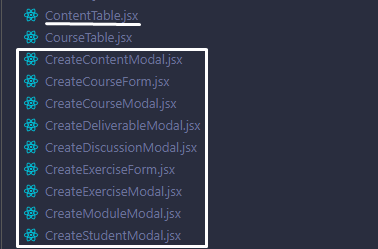
De la siguiente manera se muestran los datos para la administración, independientemente de que tabla sea, siempre con el estilo prácticamente igual. En este caso se muestra la tabla de cursos:



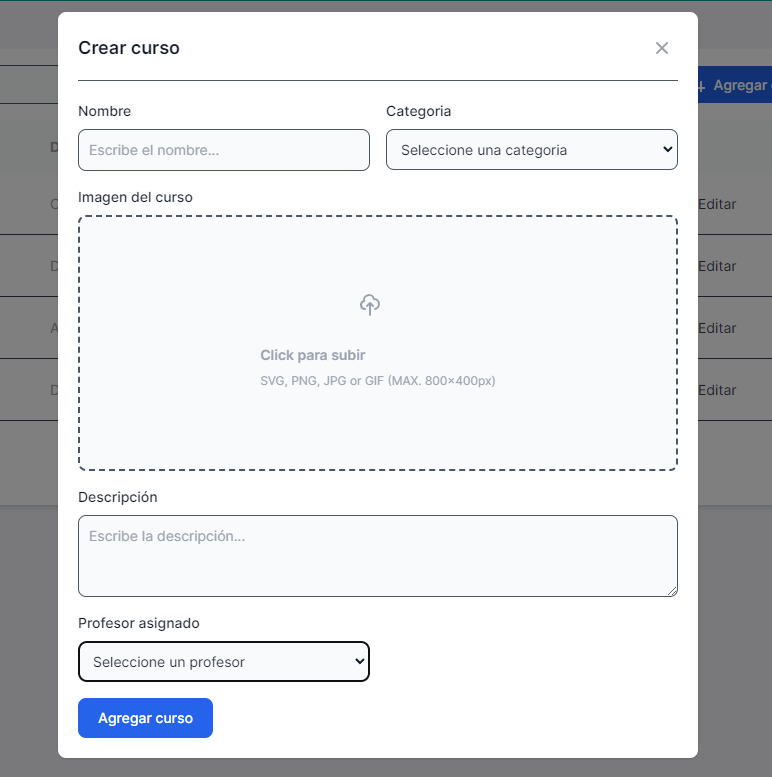
En el apartado de acciones se despliegan las siguientes opciones, que renderizan su tabla correspondiente



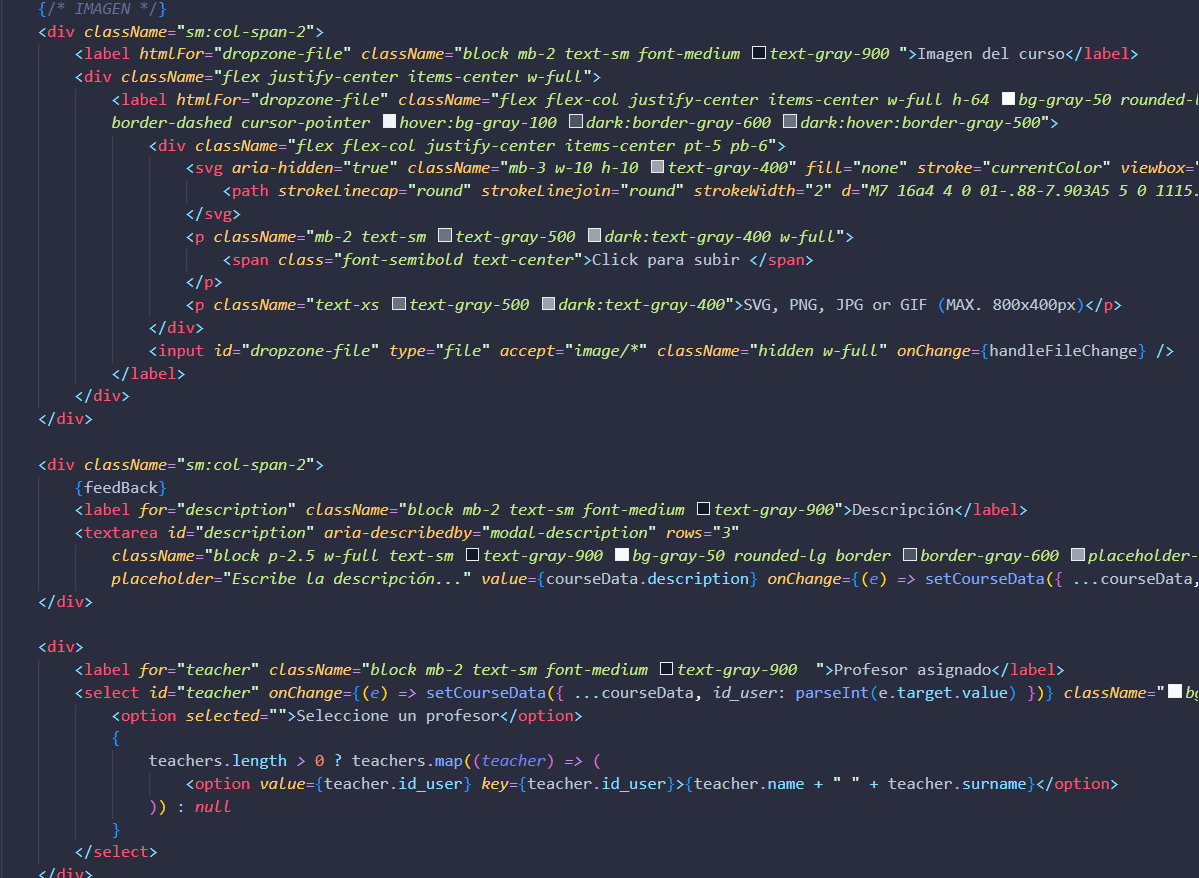
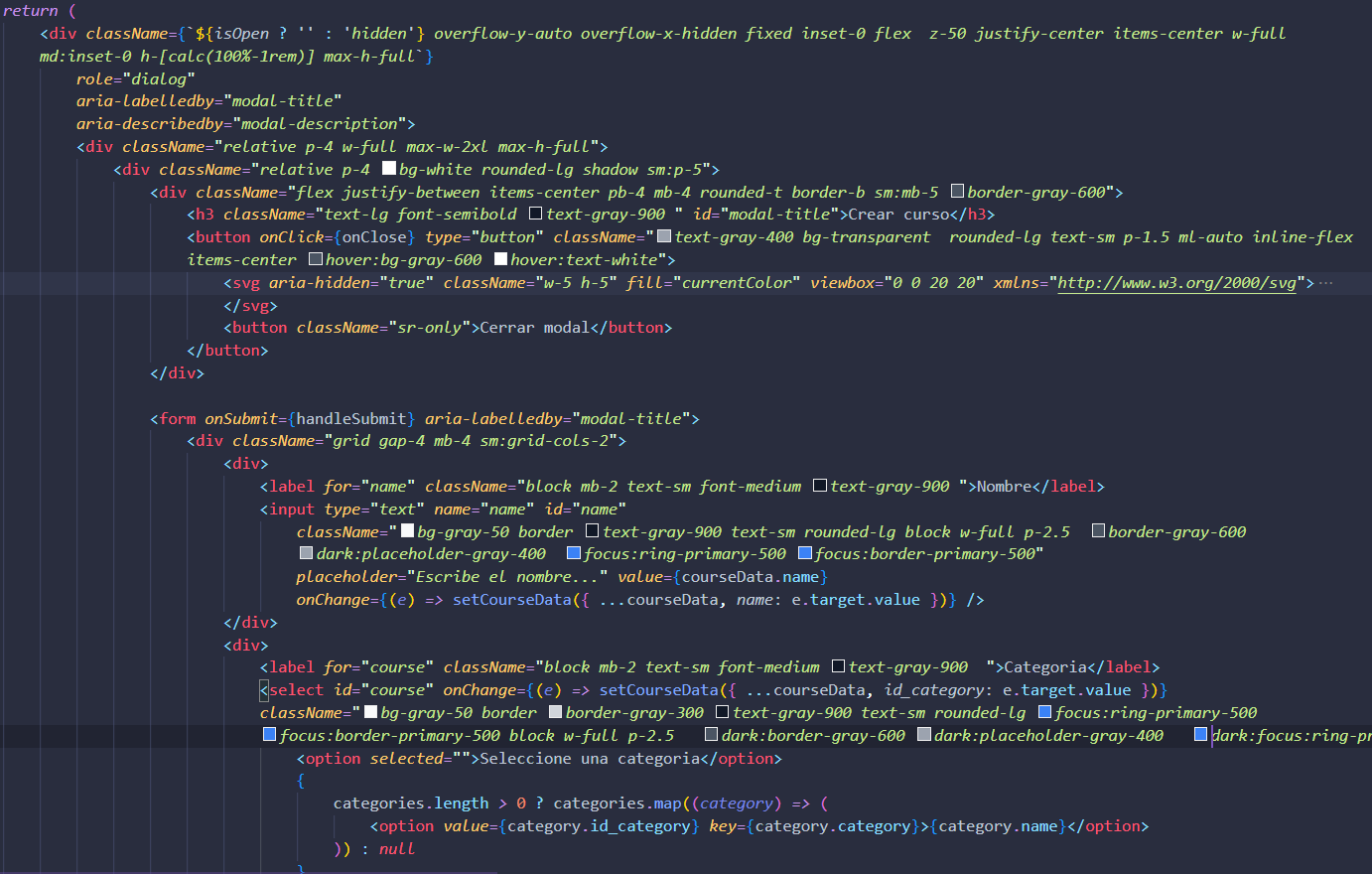
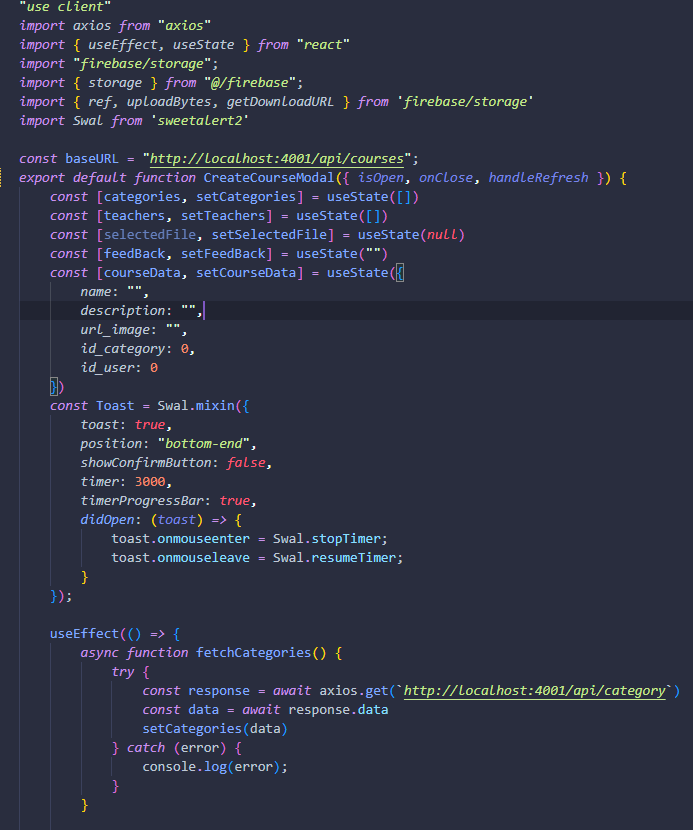
Con respecto al diseño de los ABM decidí crear un componente modal por cada acción del ABM, por ejemplo, los de alta/crear.



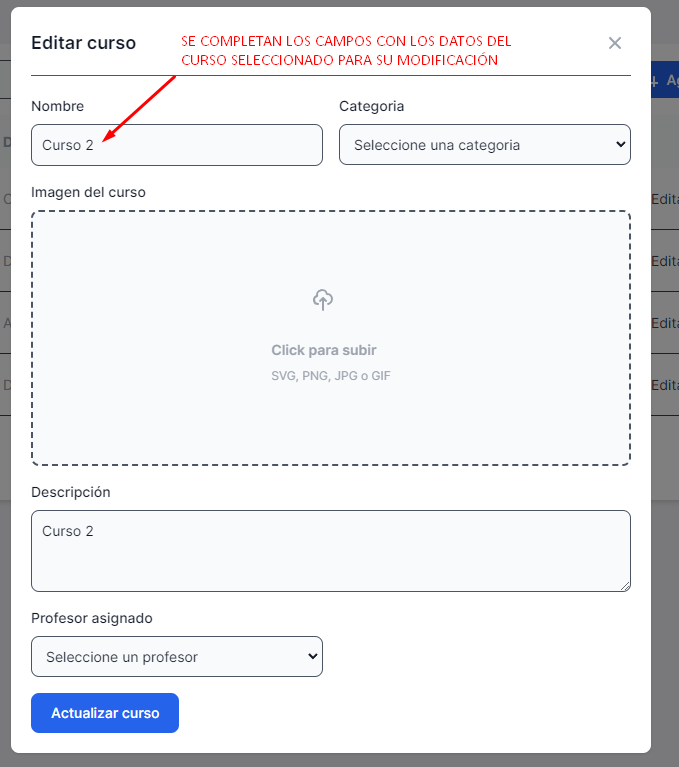
Por ejemplo, para el alta de cursos se muestra con el siguiente diseño:



Es el siguiente código:

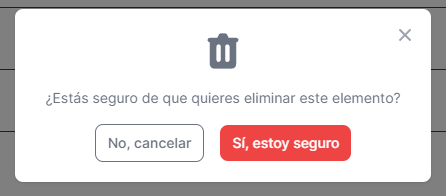


Para la modificación se muestra de la siguiente manera, es prácticamente igual que el alta así que se evita mostrar el código por obvias razones.



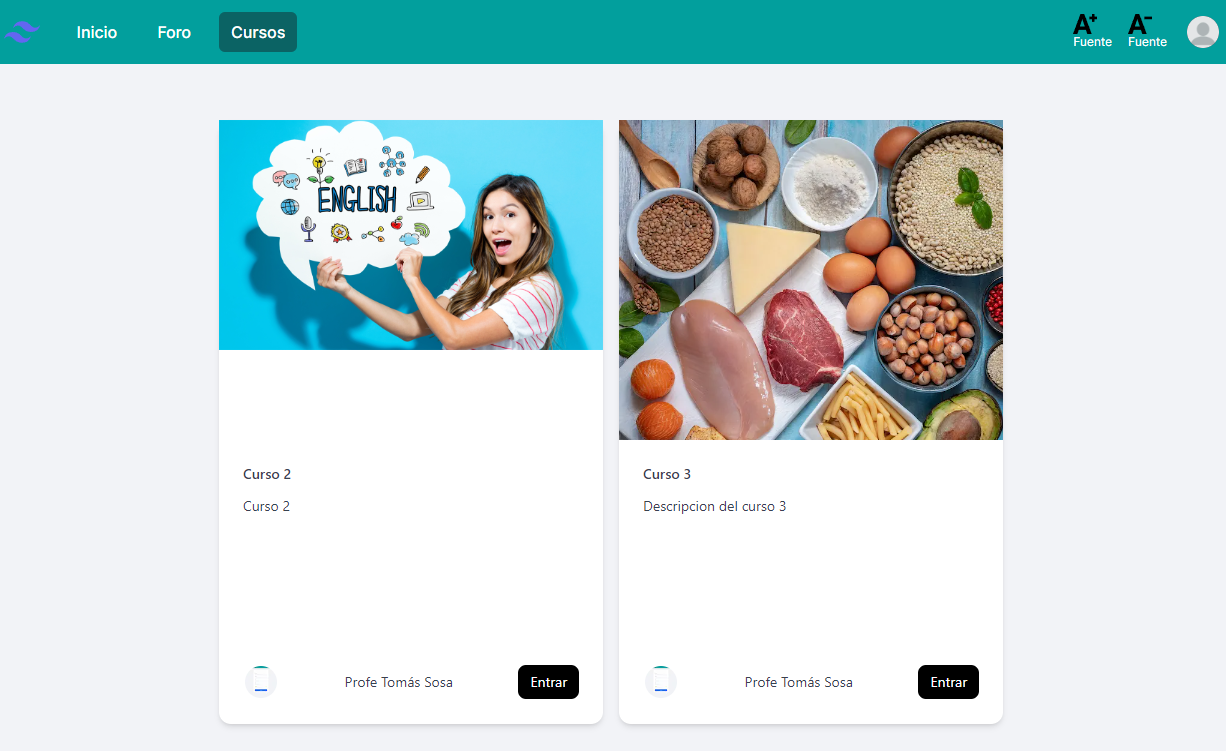
Y para la eliminación de registros, en algunos casos usé soft-delete, actualizando el campo “active” del registro para no mostrarle al cliente, en este caso un curso inactivo. Y para los datos que no están relacionados con otro, si use un hard-delete, eliminando el registro por completo de la bd.

Se ve de la siguiente manera:



#### Vista rol estudiante:

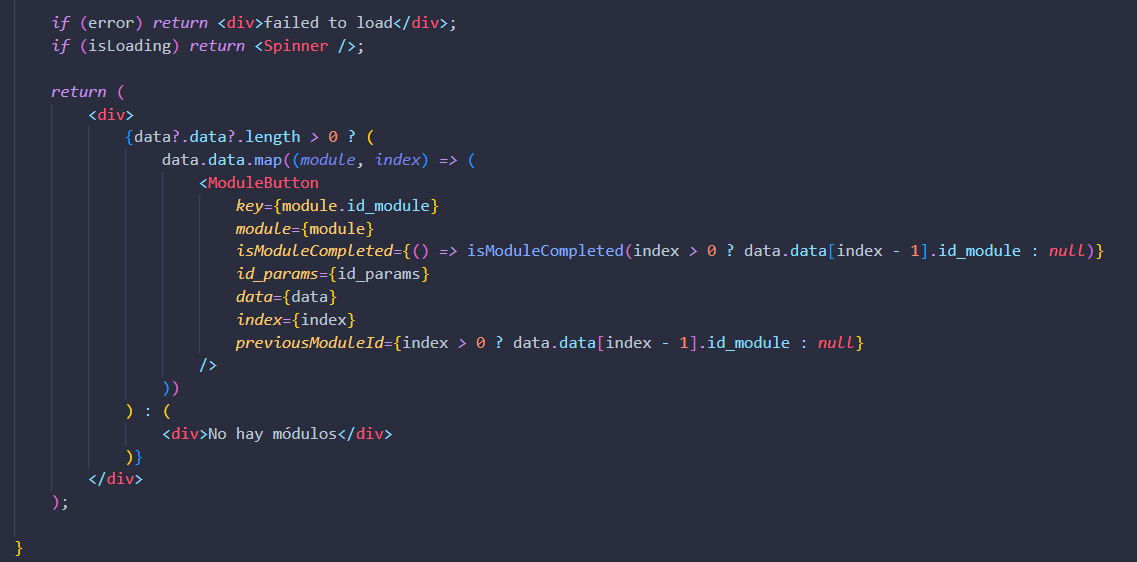
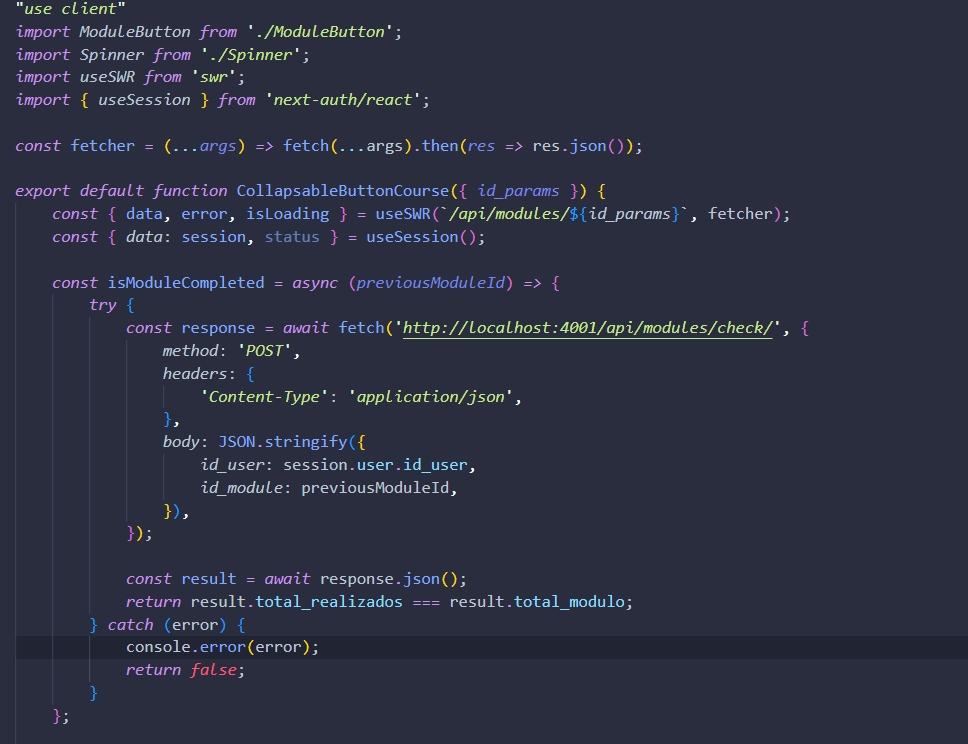
El estudiante solo tiene acceso a los cursos en los que está inscripto, en este caso está inscripto a dos:



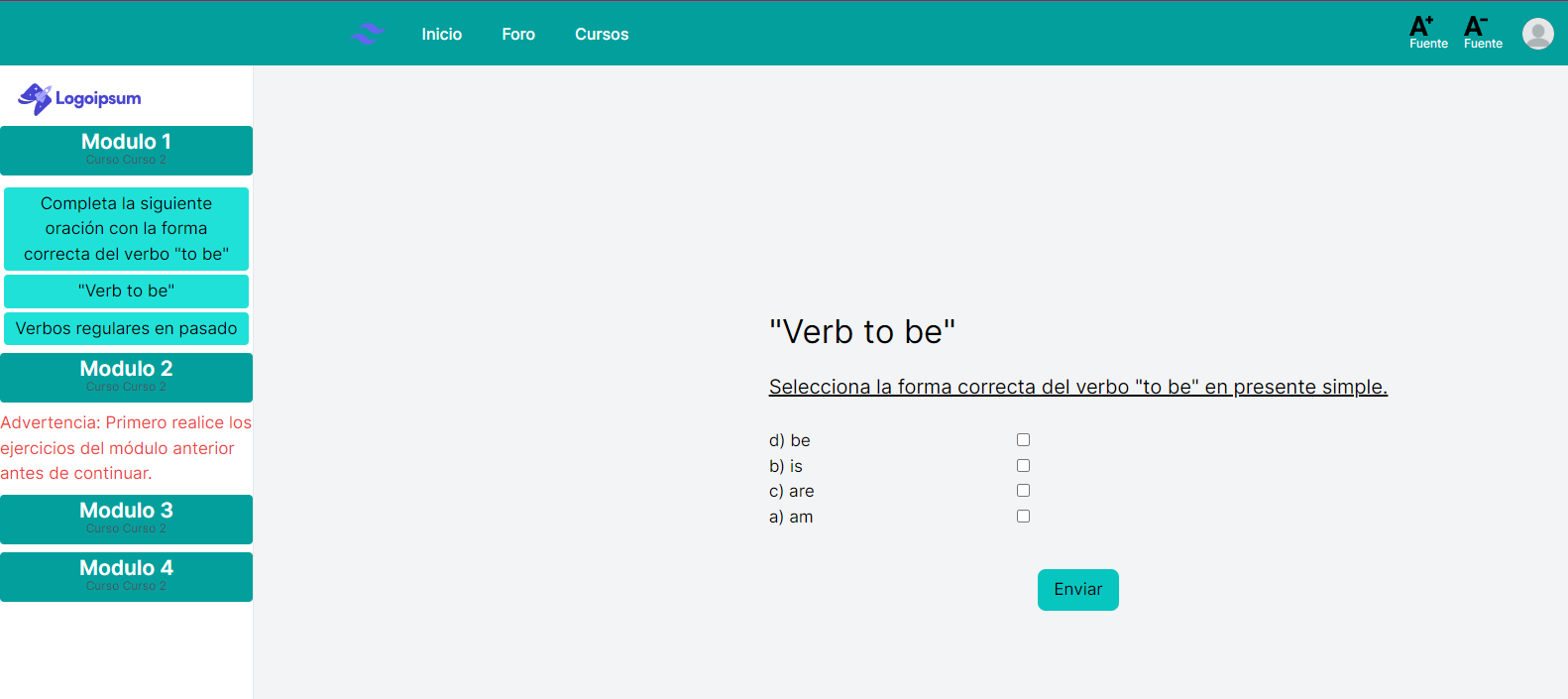
Al entrar a un curso, verá lo siguiente:



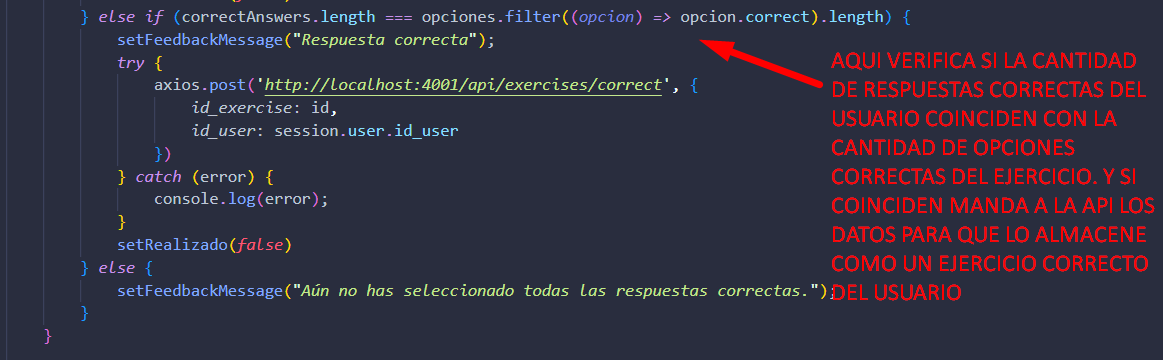
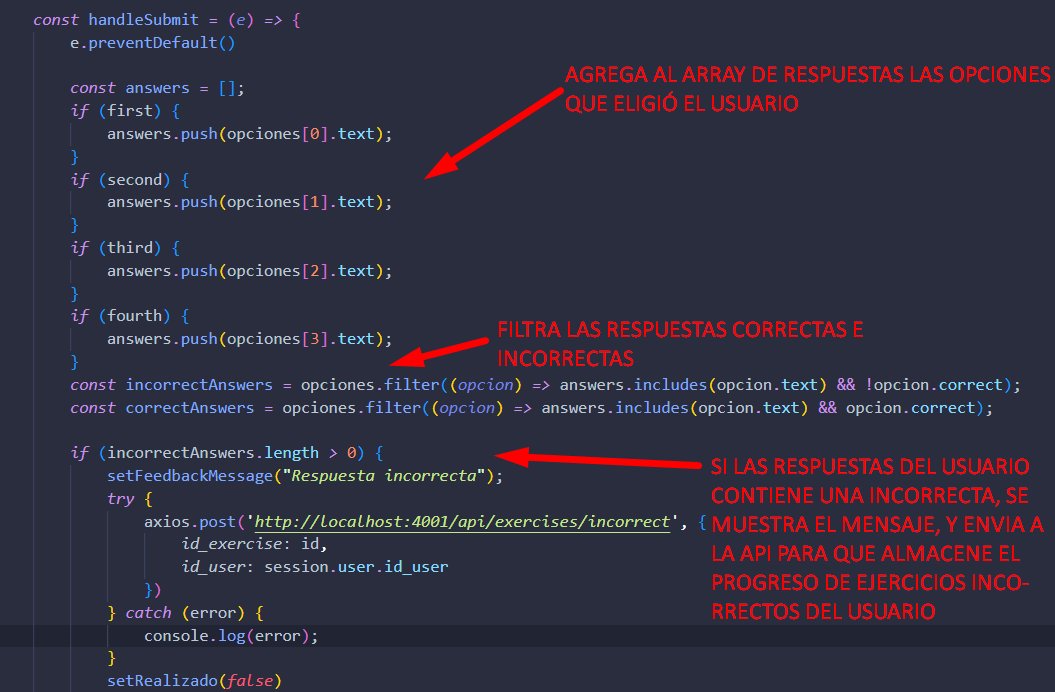
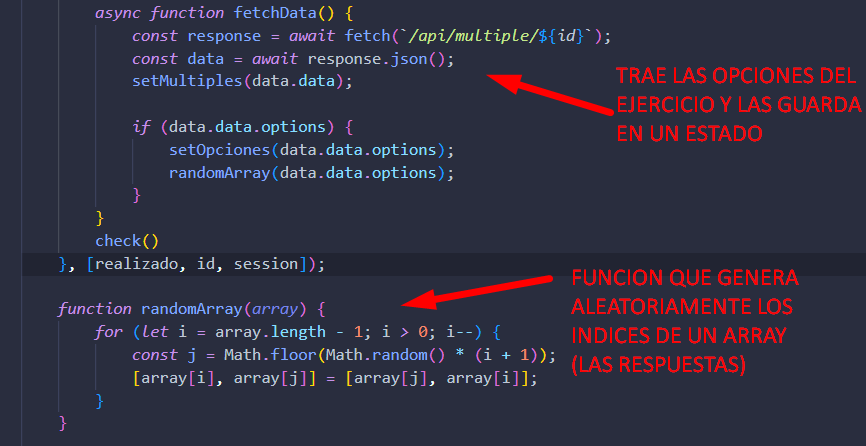
A continuación, se muestra la lógica del componente donde hace la comprobación si el usuario realizó los ejercicios anteriores y la renderizacion de los ejercicios.



Al hacer click en un ejercicio se le muestra de la siguiente forma, en este caso un ejercicio multiple choise.



Aquí se muestra la lógica del componente donde hace el chequeo de que el usuario ya realizó el ejercicio, de “desacomodar” las respuestas aleatoriamente y si la respuesta del usuario corresponde o no con las respuestas correctas del ejercicio.



## Back-end

### Tecnologías usadas:

* NodeJs con Express.
* NextAuth para autenticación con Json Web Token.
* Bcrypt para encriptar contraseñas.
* Firebase para subir archivos a la nube (imágenes y pdf)
* OpenAI para generar ejercicios con Inteligencia Artificial.
* MySql.
* Prisma como ORM.

### Implementación del código

Opté por utilizar Express para mi API debido a su facilidad de uso y flexibilidad en la creación de rutas y endpoints, cuando se trabaja con controladores.

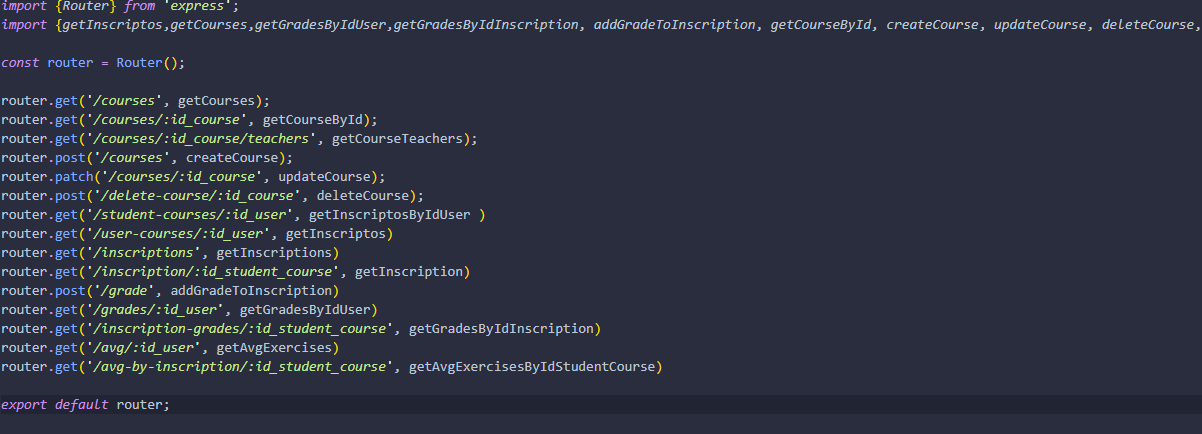
La arquitectura de carpetas del proyecto es la siguiente:



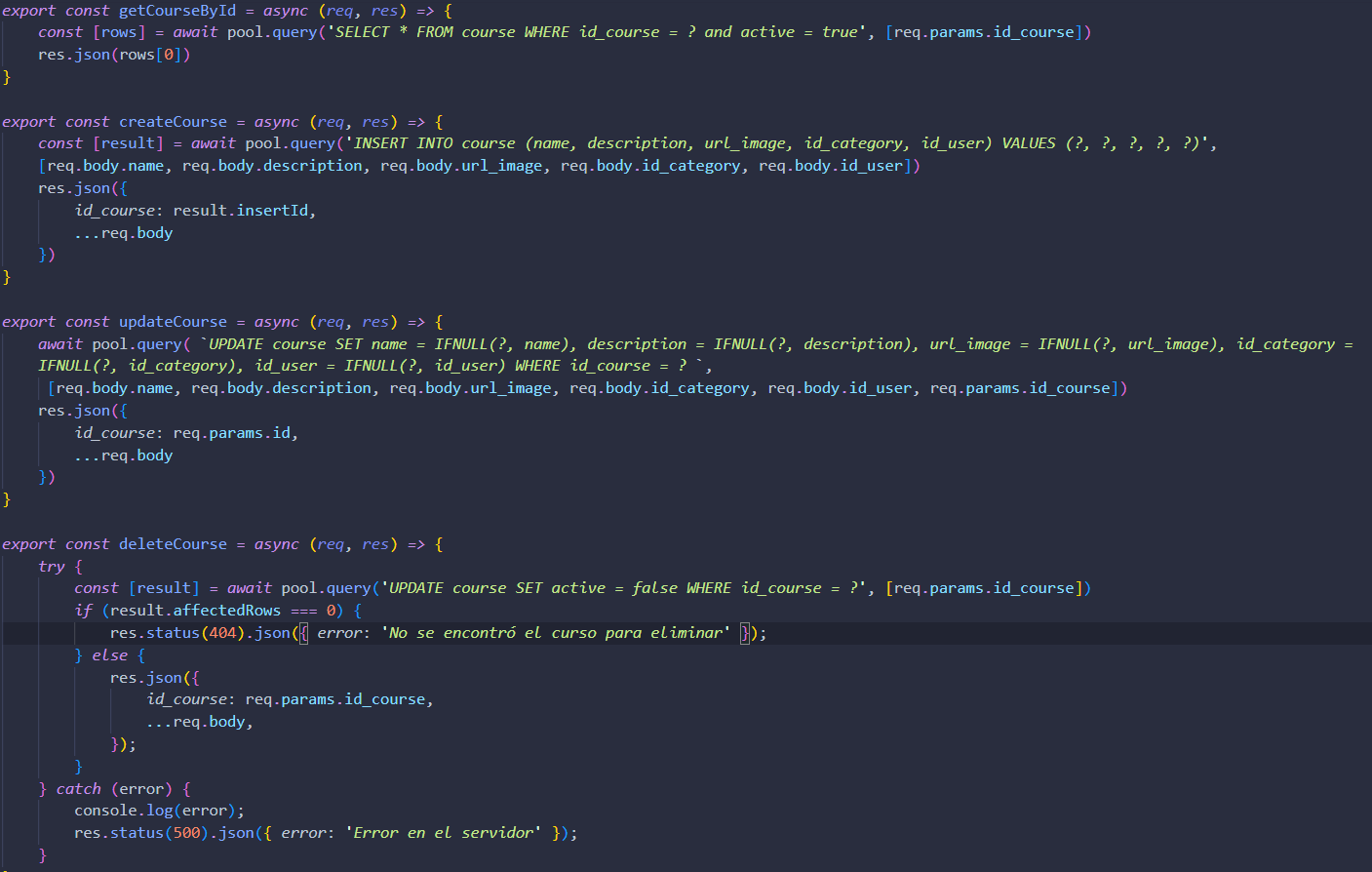
A continuación, se muestra el archivo index.js, que es donde se configura y se pone en marcha el servidor web, define los middlewares para manejar solicitudes y habilita las rutas para diferentes partes de la API con el prefijo “/api”.



Ahora se muestra un archivo de rutas, en este caso de cursos:



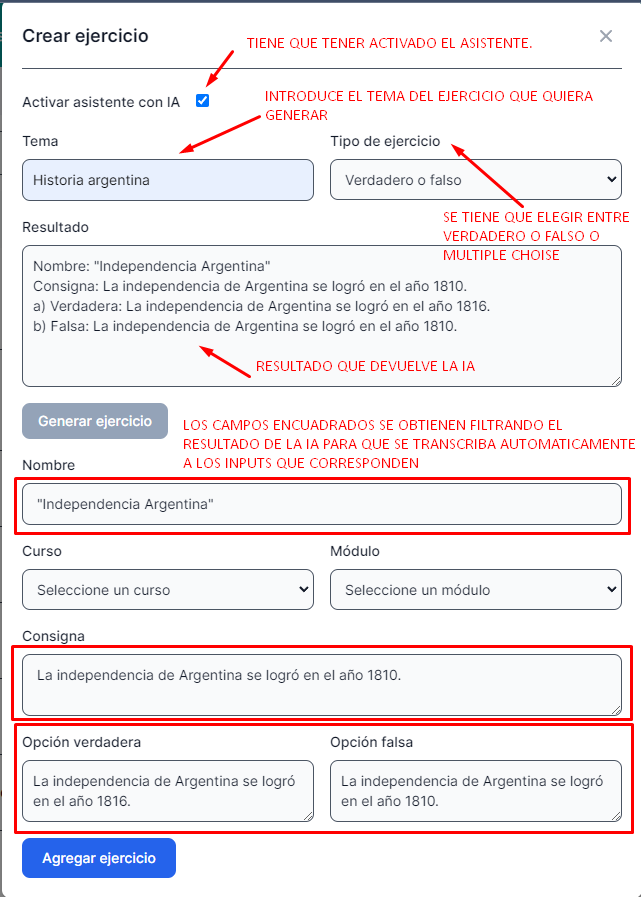
Aquí algunos de sus endpoints:



## Integración de inteligencia artificial

### Creación de ejercicios

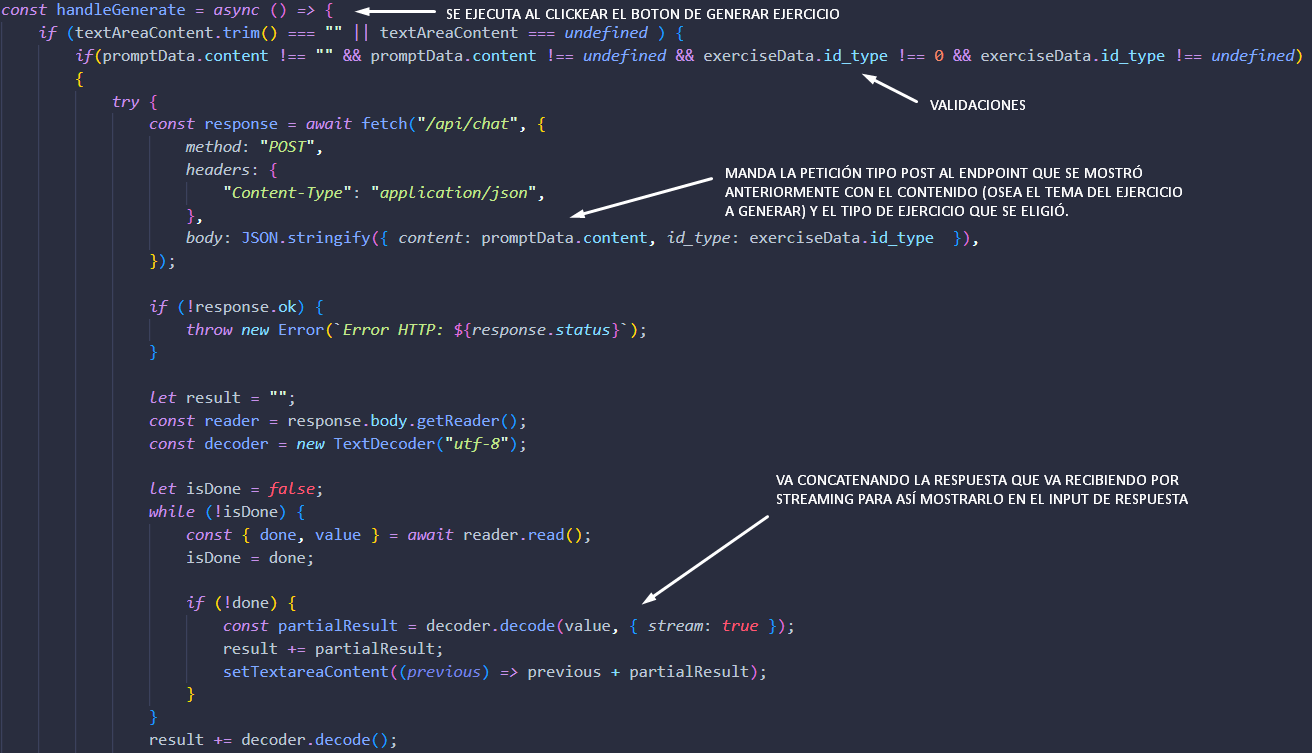
Decidí implementar la API de OpenAI para generar ejercicios con Inteligencia Artificial, para que el profesor pueda agregar ejercicios de forma más rápida.



Y se implementó de la siguiente forma en el backend:

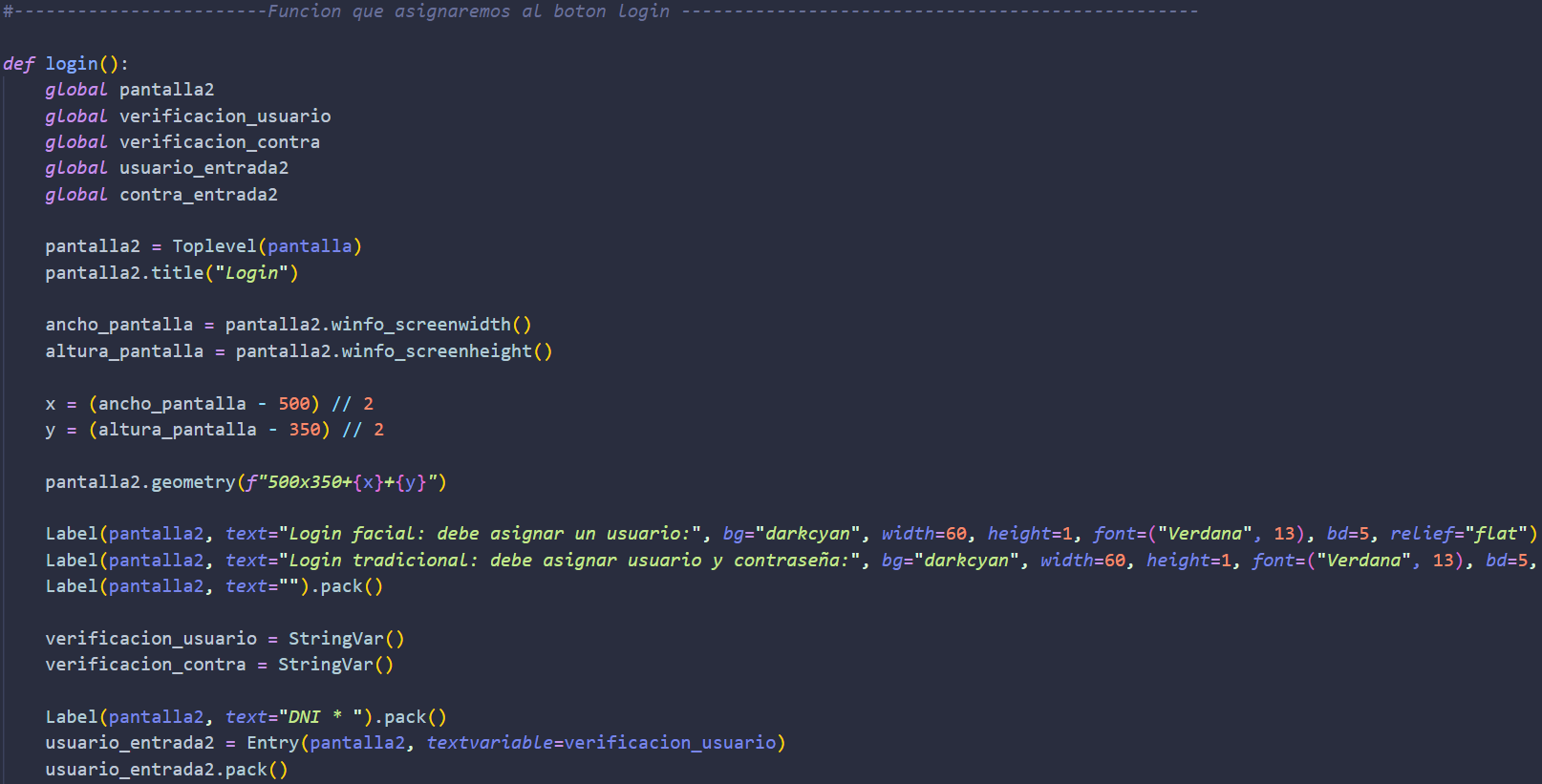
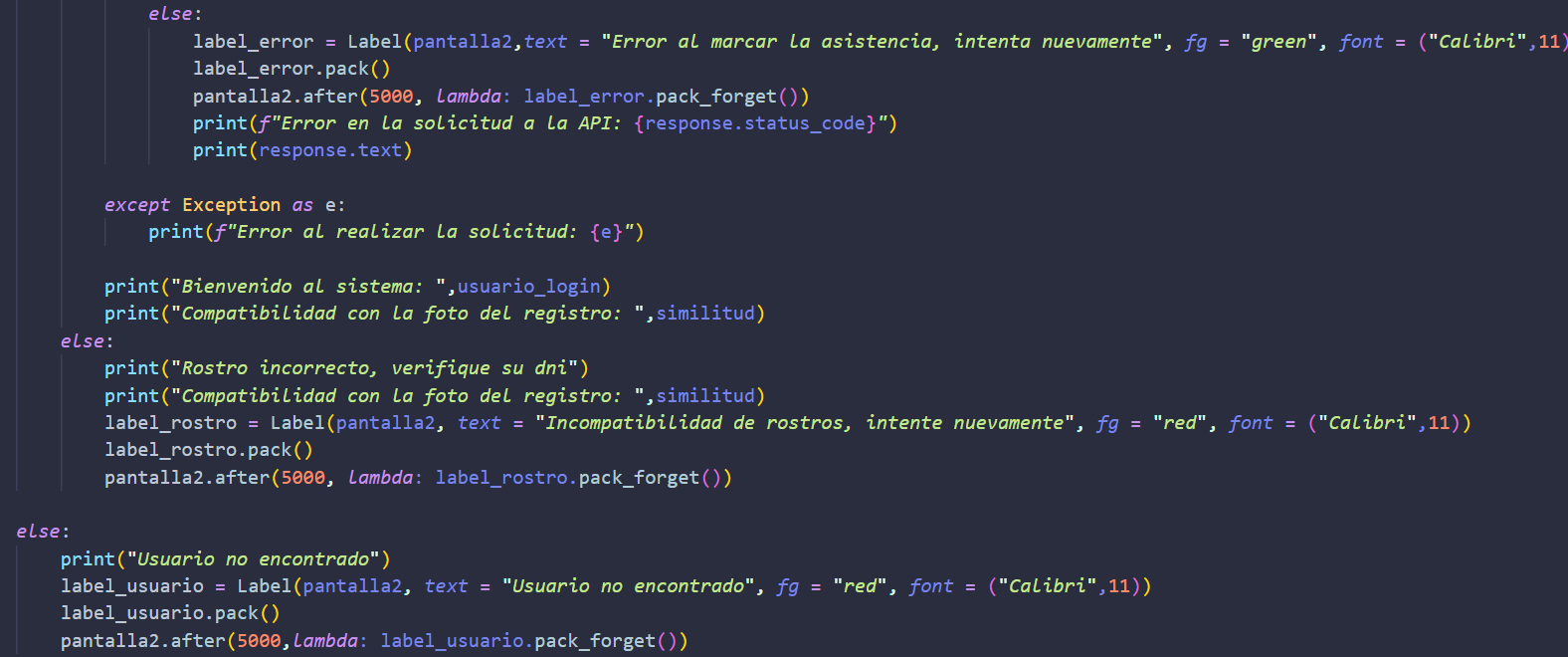
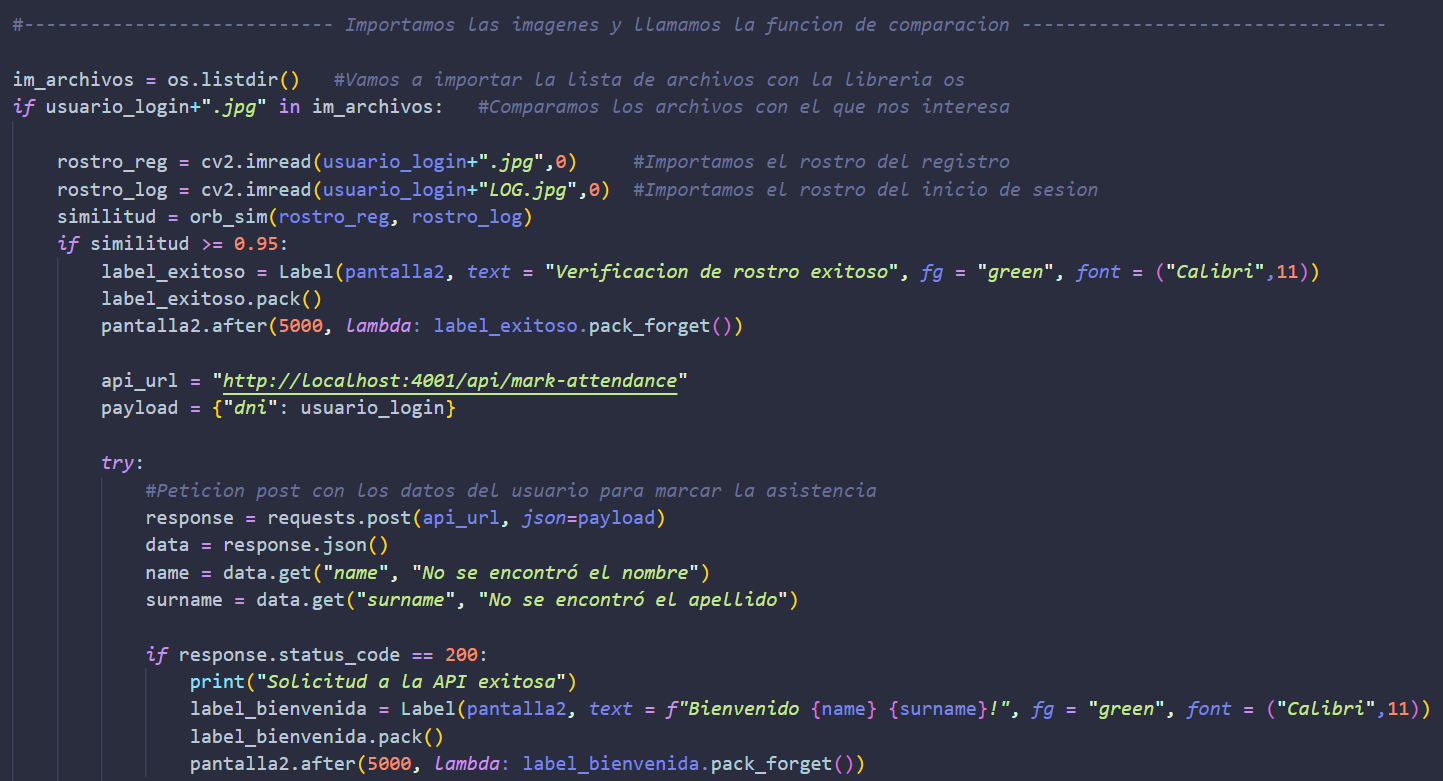
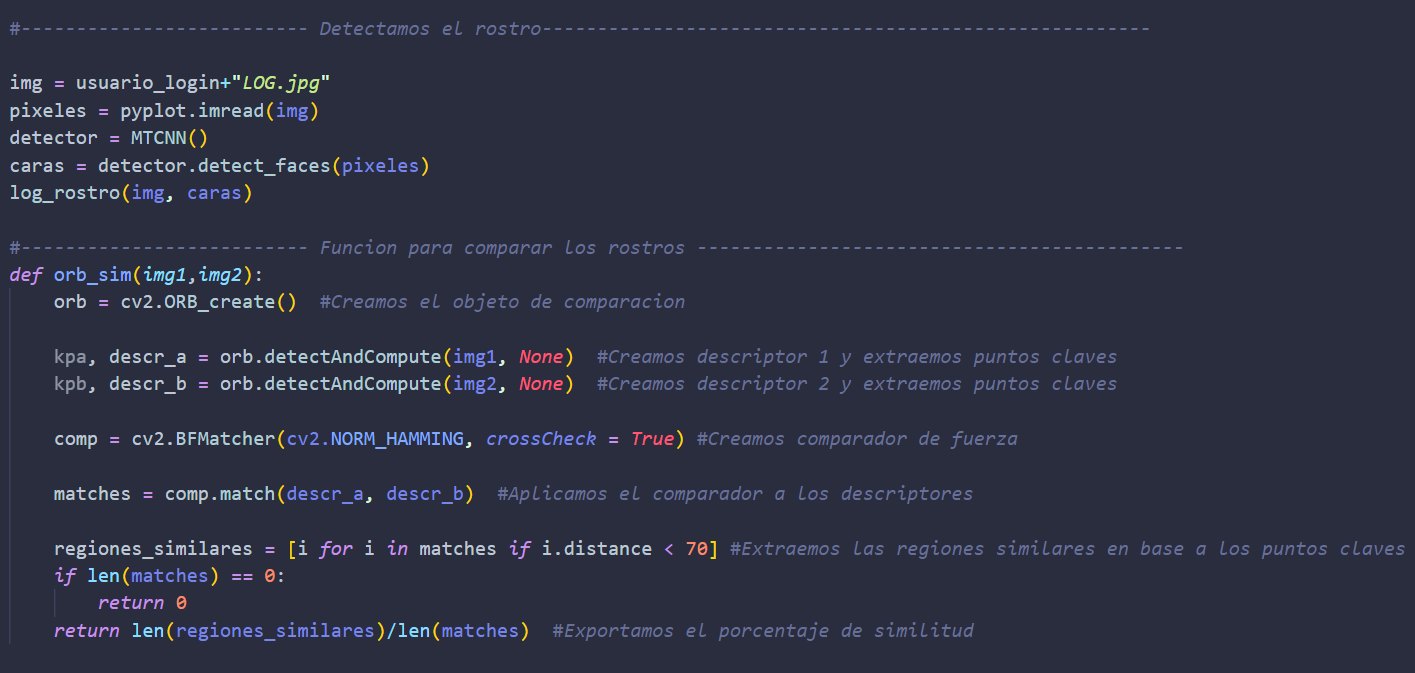
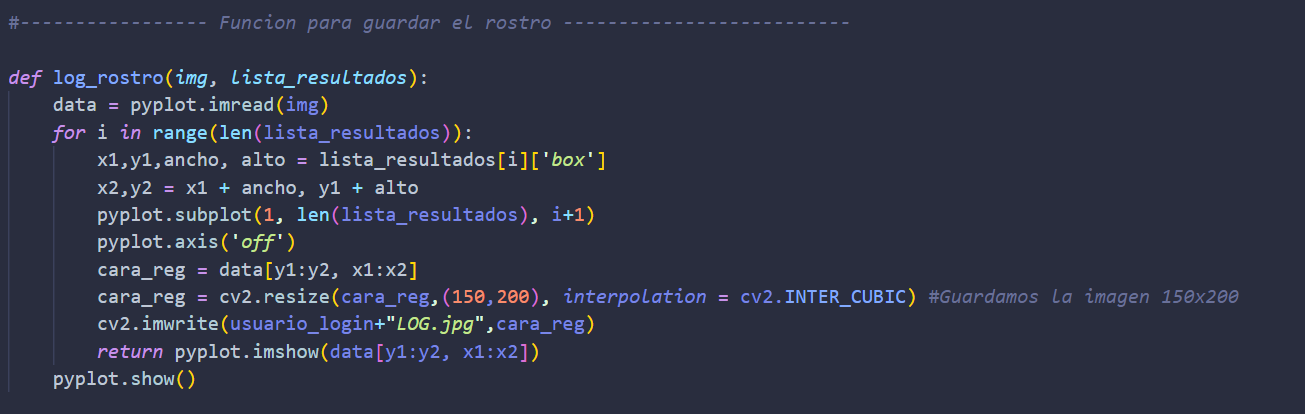
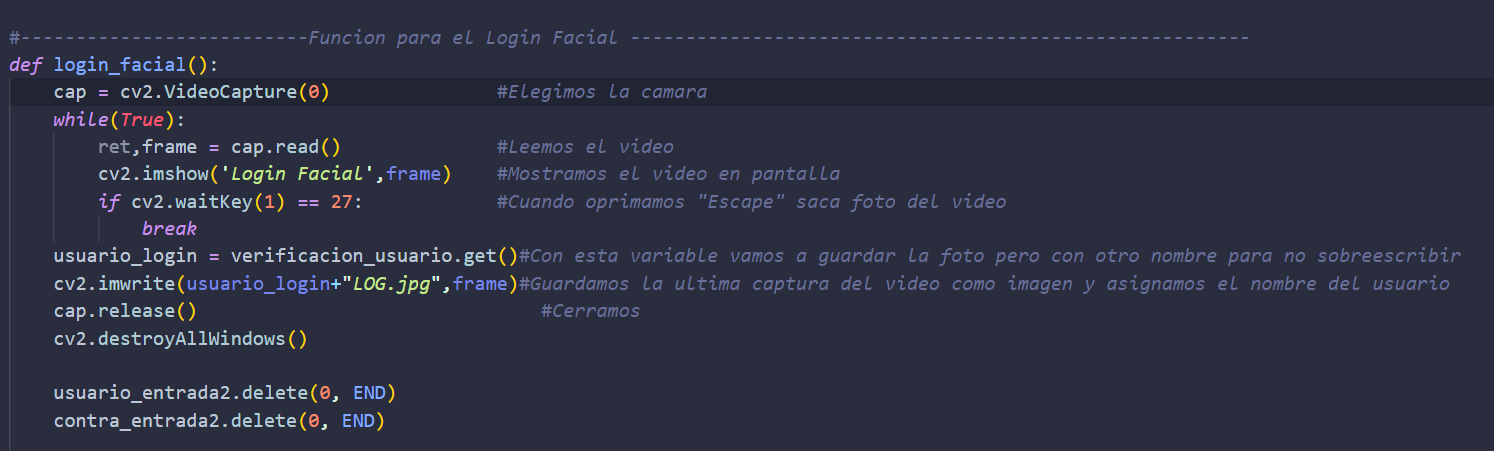
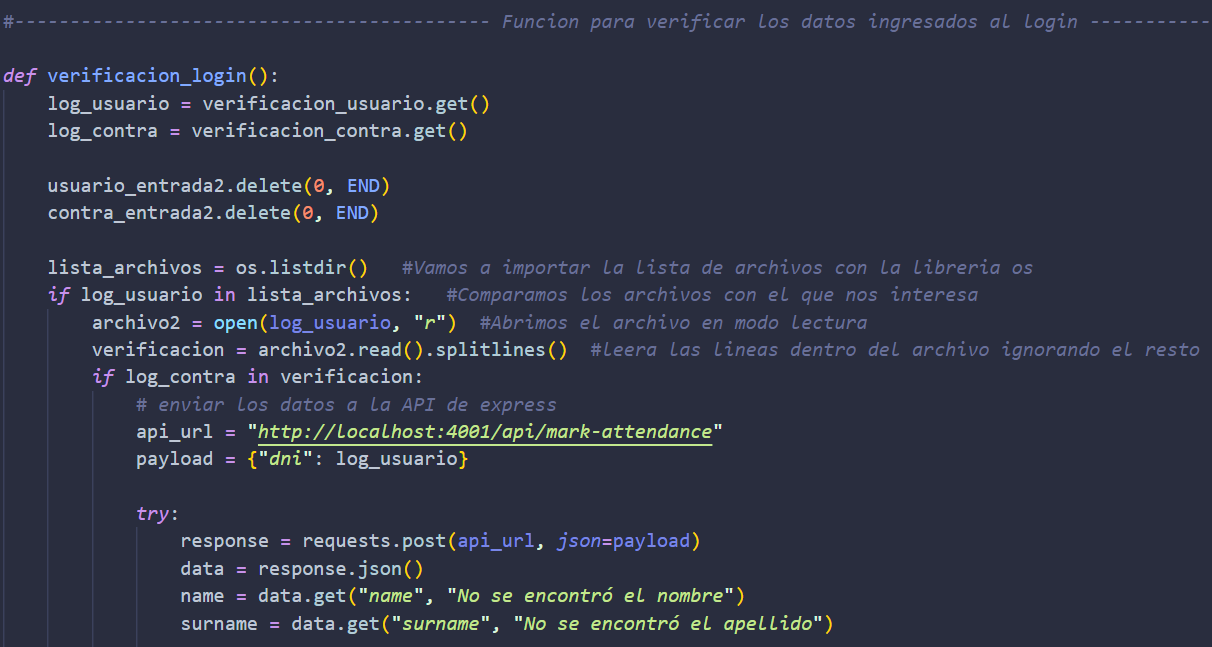
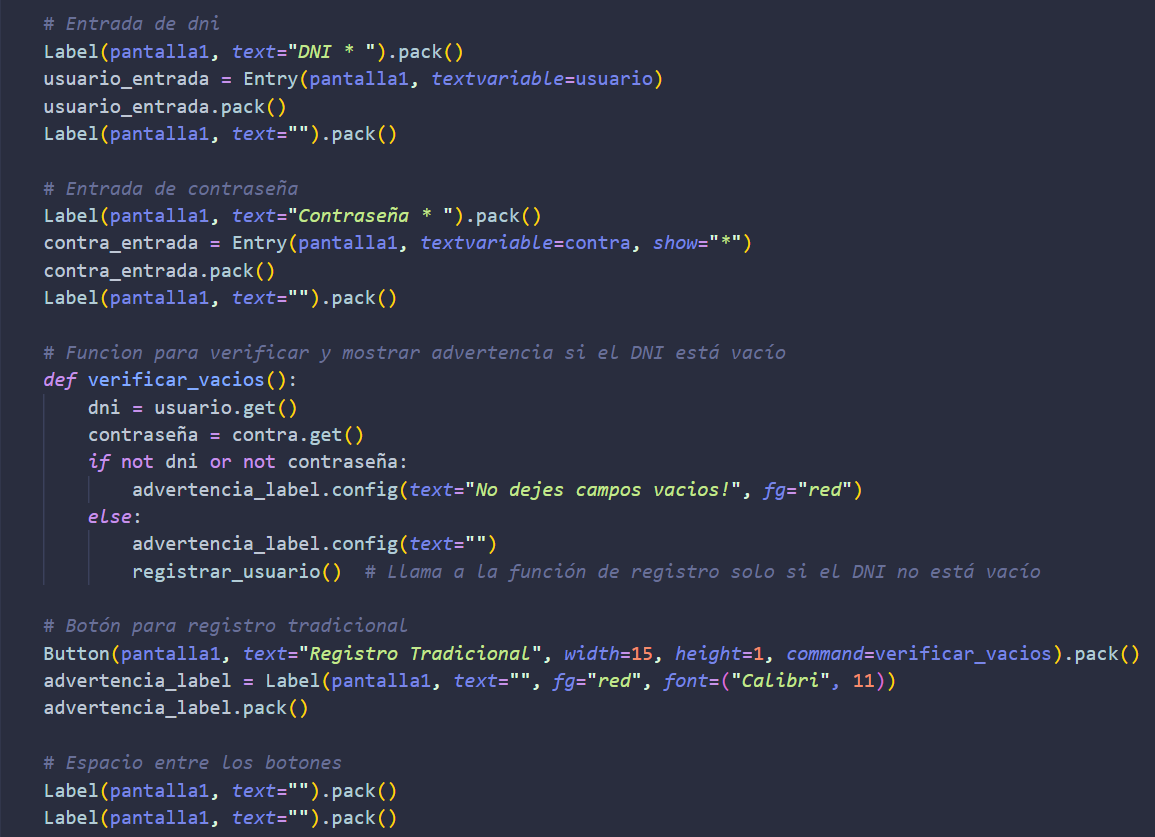
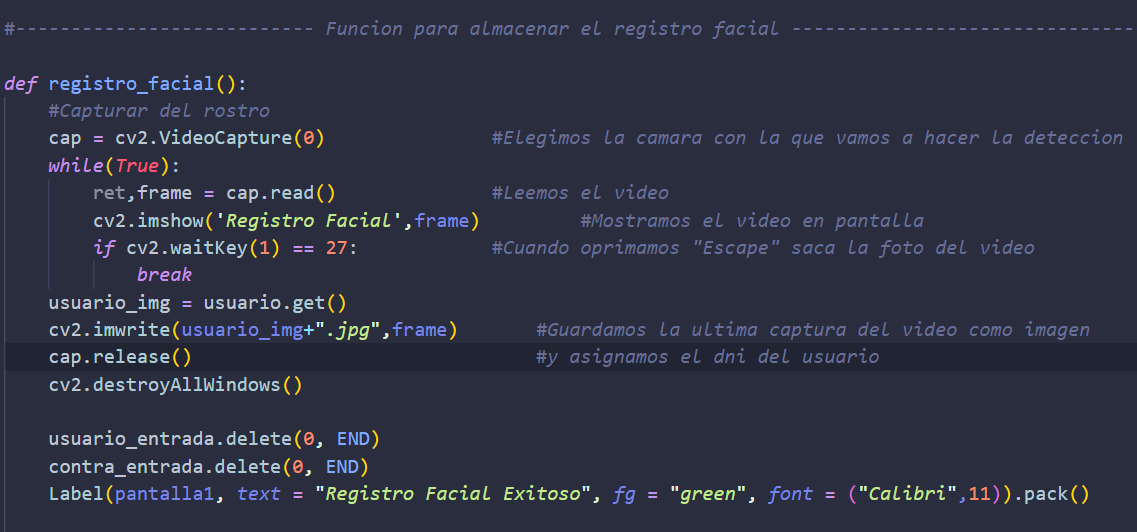


Y en el frontend:



### Registro de asistencia con reconocimiento facial

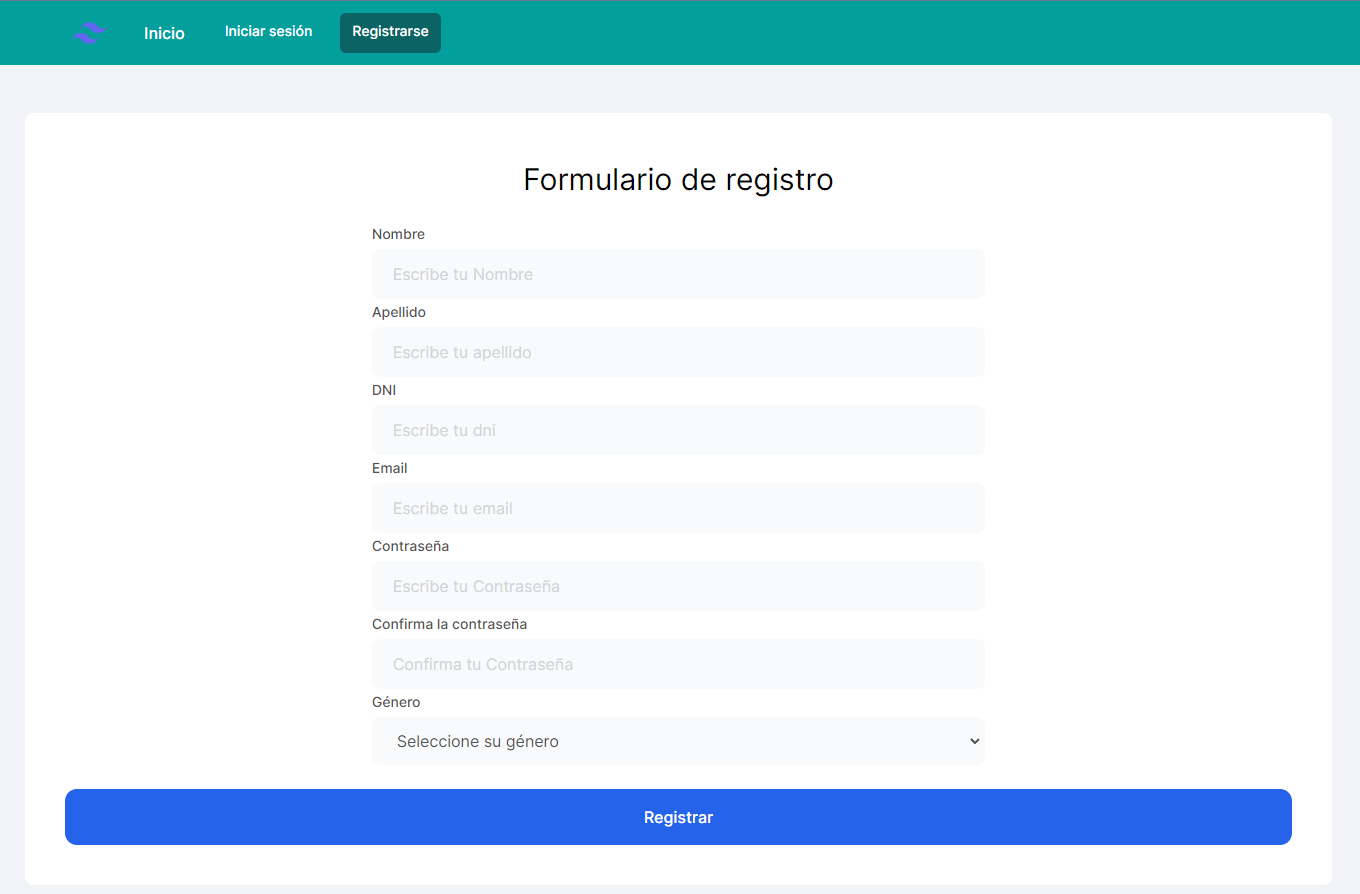
Para el registro de asistencia al ingreso a la institución se desarrolló otro sistema, que se complementa mediante la misma API para mandar la petición para registrar el ingreso del estudiante o profesor. Está desarrollado en Python y usa la librería OpenCV. El código es el siguiente:



# Manual de usuario

## Acceso al campus virtual

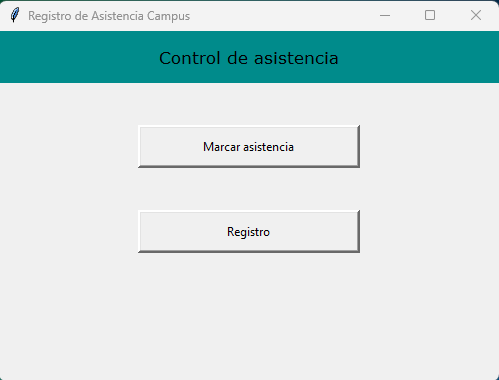
El usuario deberá registrarse con sus datos personales e iniciar sesión como cualquier otra página web, a continuación, se muestra el formulario de registro.



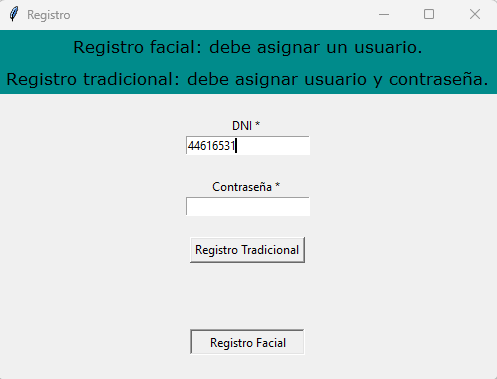
Una vez registrado, para tener el primer acceso como profesor se tiene que asignar el rol manualmente desde la base de datos con la siguiente consulta SQL.

*UPDATE user SET id\_rol = (2) WHERE id\_user = 1;*

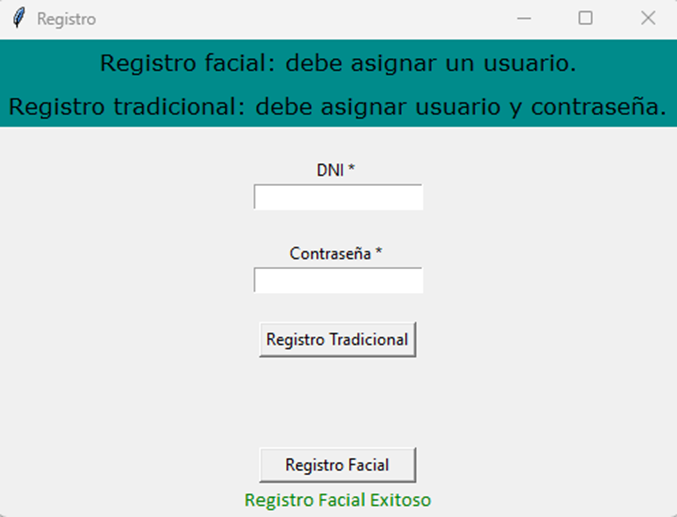
Para configurar el sistema de asistencia tienen que estar previamente registrados en el campus, para registrar el usuario ya sea con reconocimiento facial y/o dni y contraseña, tiene que entrar a la pantalla de inicio.



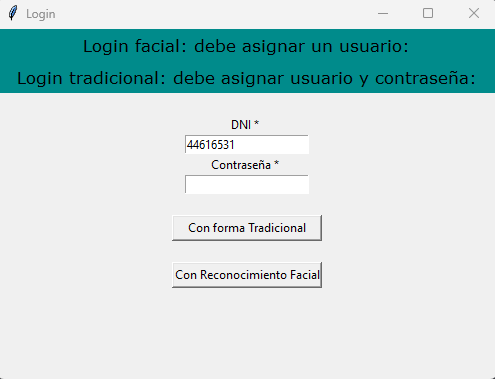
Aquí se muestra el registro con reconocimiento facial, tiene que ingresar el dni y se procede abrir la cámara.



Una vez abierta la cámara y sacada la foto presionando la tecla Escape, se mostrará el siguiente mensaje.



Una vez registrado ya puede proceder a marcar la asistencia de una manera muy similar al registro.

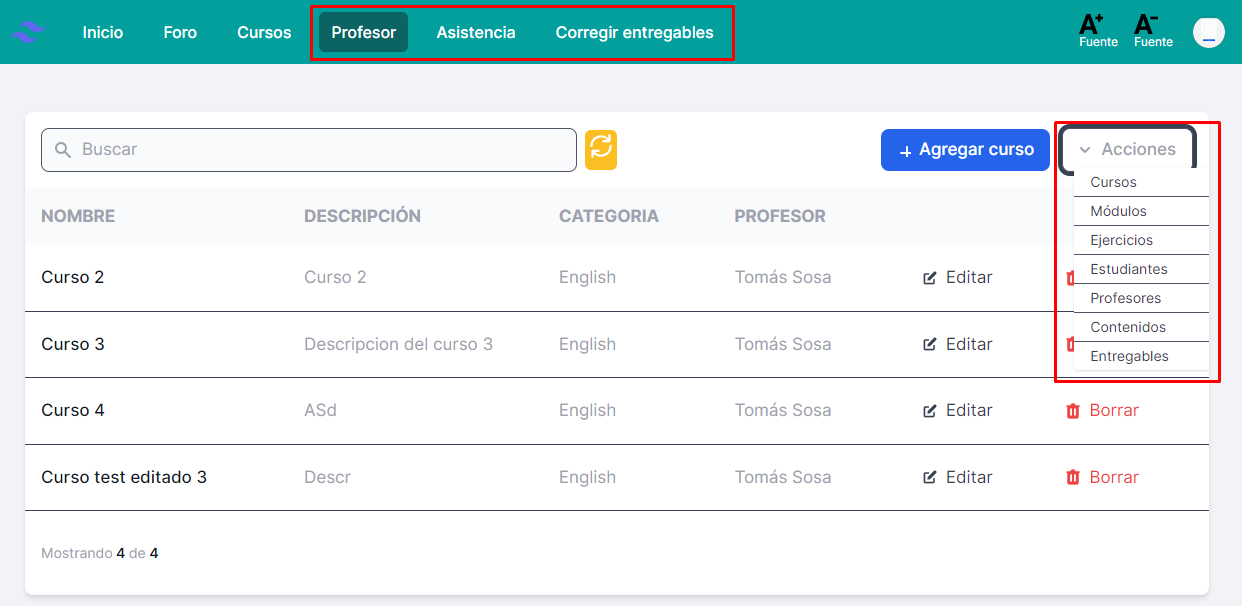


Una vez sacado la foto se verifica si coincide con la foto de registro y manda la petición a la API para guardar la asistencia. Si se hace satisfactoriamente muestra los siguientes mensajes que se eliminan luego de 5 segundos para que no se le muestren a la siguiente persona que quiera marcar la asistencia.

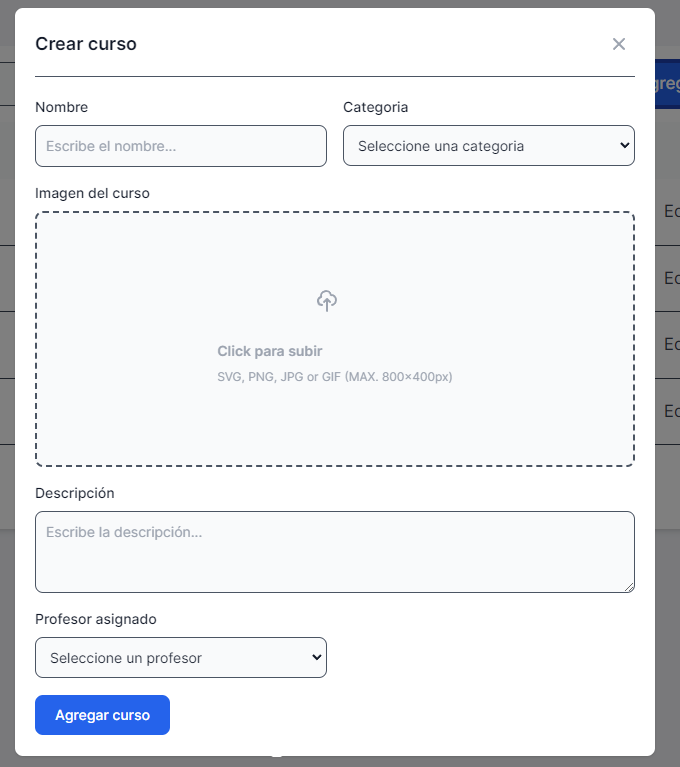


## Configurar el campus

Una vez con el rol profesor se le mostrarán los siguientes botones:

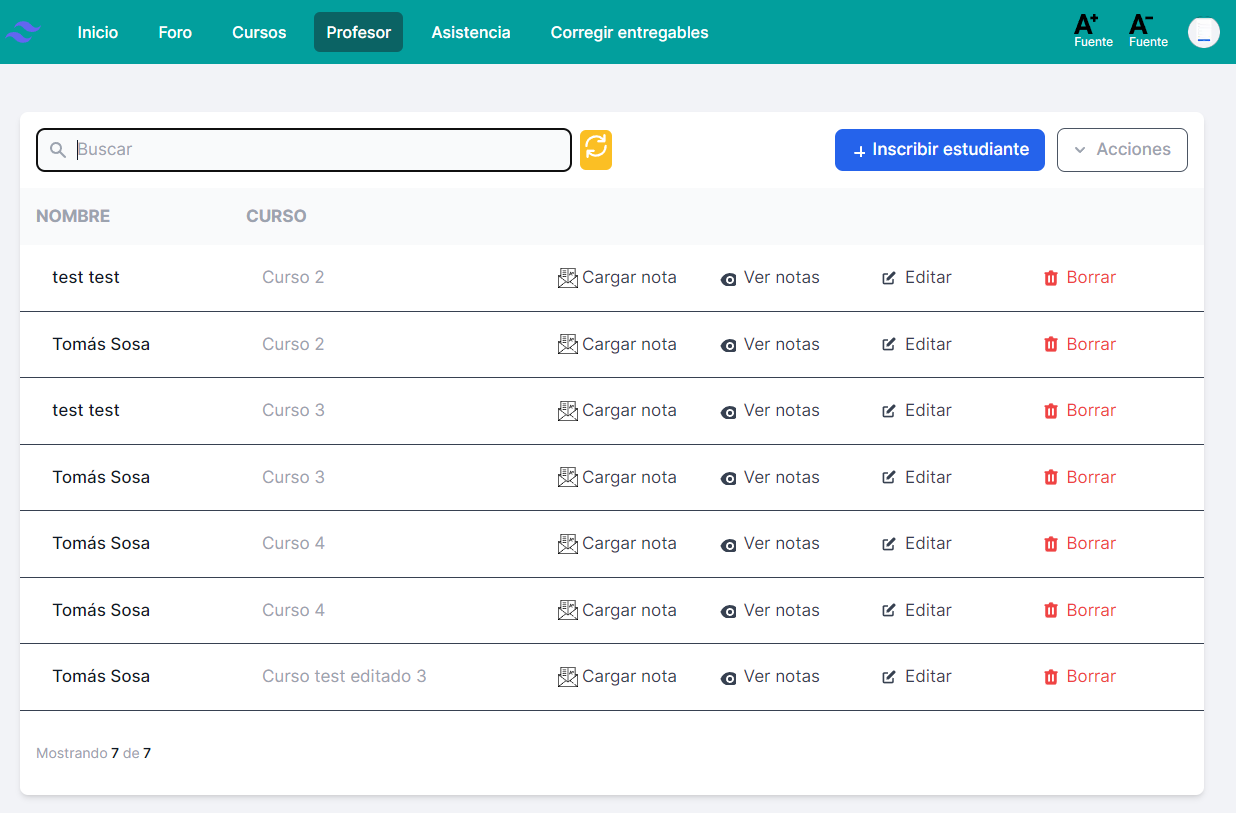


En el botón “Profesor” se pueden realizar todas las acciones de creación, modificación y eliminación que se muestra a la izquierda, para que posteriormente los estudiantes accedan. Este es un ejemplo para crear un curso, donde todos los campos son obligatorios, y todos los demás son prácticamente iguales.

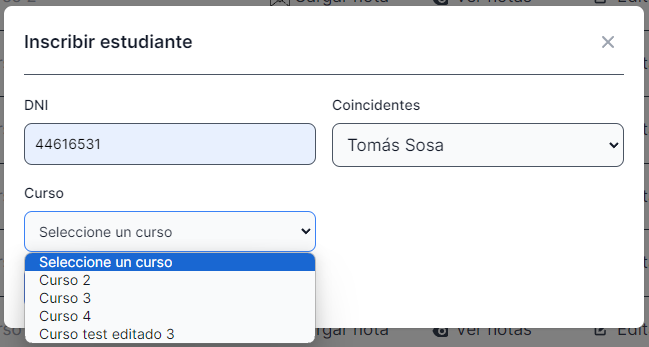


## Cómo inscribir a un estudiante

En la sección de profesor se tiene que ir a la acción “Estudiantes” y se le mostrará lo siguiente.

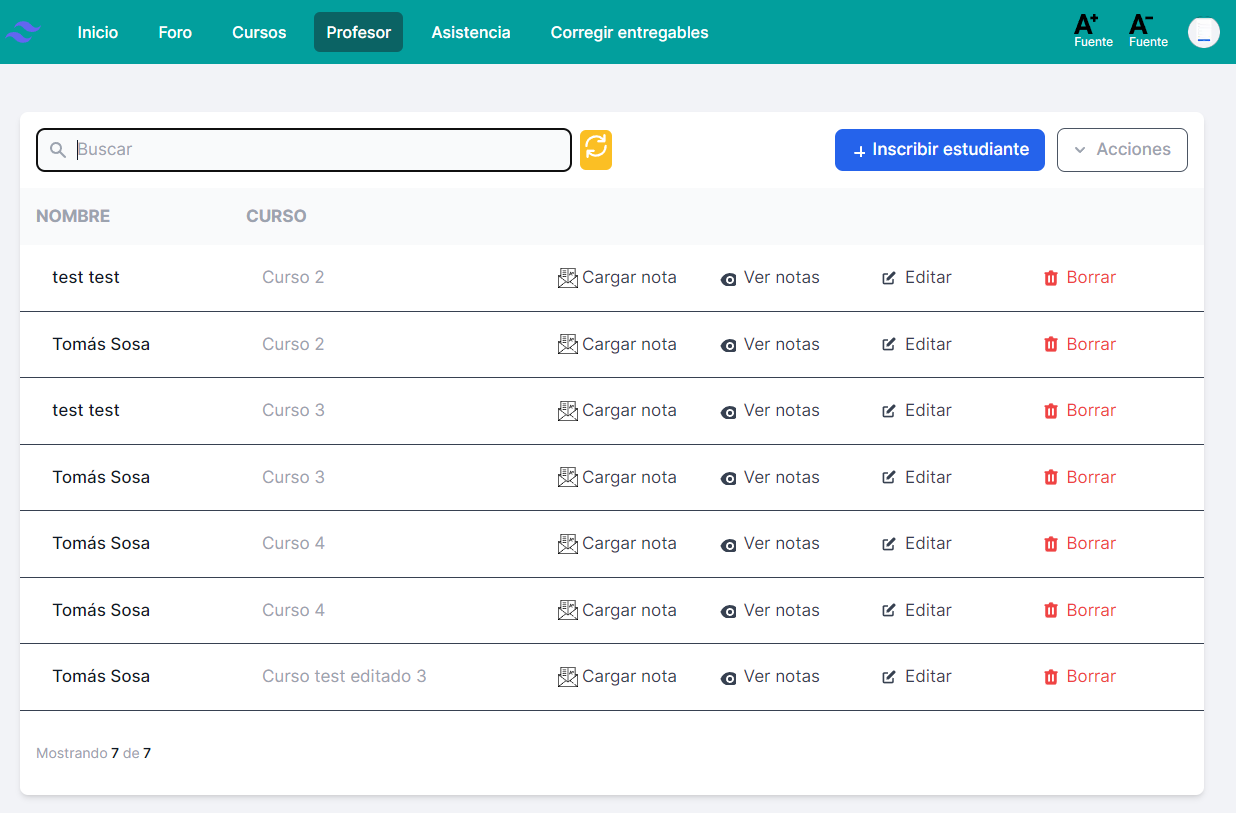


Tiene que presionar en “Inscribir estudiante” y tiene que colocar el DNI del estudiante que ya está registrado en el Campus y seleccionar a que curso inscribirlo.

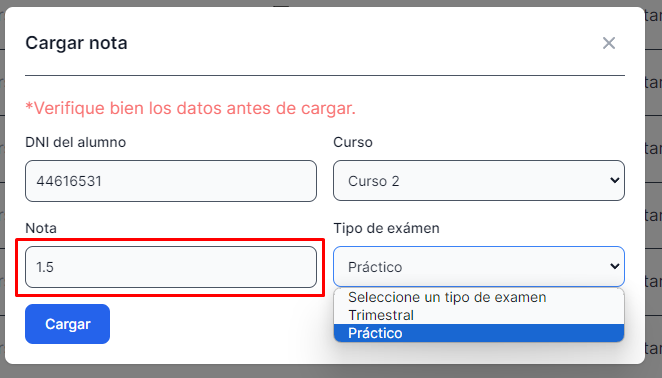


## Cargar notas a un estudiante

Para cargar las notas debe dirigirse a la acción “Estudiantes” y se le mostrará lo siguiente.



Aquí se le muestran todas las inscripciones de los usuarios, y para cargarle la nota debe buscar a la inscripción que pertenece al estudiante que le quiere cargar la nota y hacer click en “Cargar nota” y se le mostrará lo siguiente.

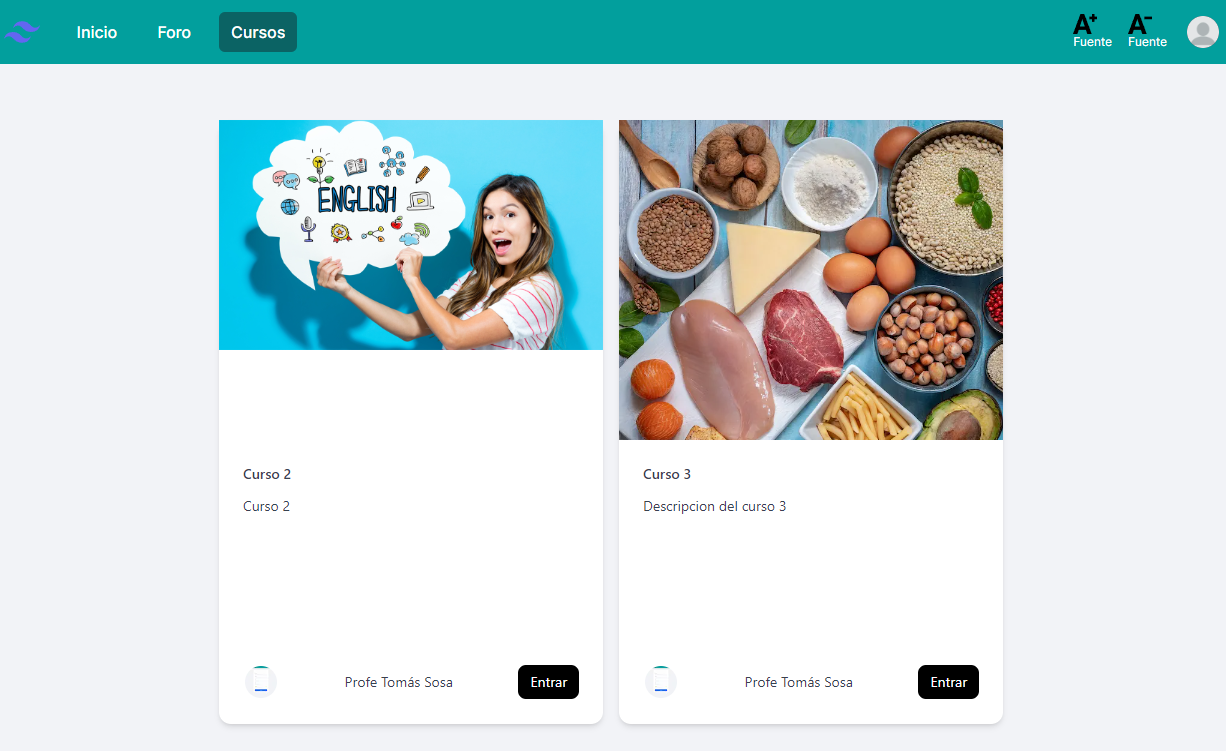


Se puede elegir dos tipos de examen, “Trimestral” que es la nota que el profesor ingresará según el rendimiento del estudiante en un examen fuera del campus, ósea en papel, y “Práctico” que es el promedio que automáticamente se calcula basándose en la cantidad de ejercicios totales del curso y los ejercicios hechos correctamente en ese curso, los ejercicios que no realizó se lo toma como incorrectos y no influyen en el promedio. Luego puede hacer click en “Ver notas” y se le mostrará las notas y el promedio del estudiante en esa inscripción.



## Realizar ejercicios

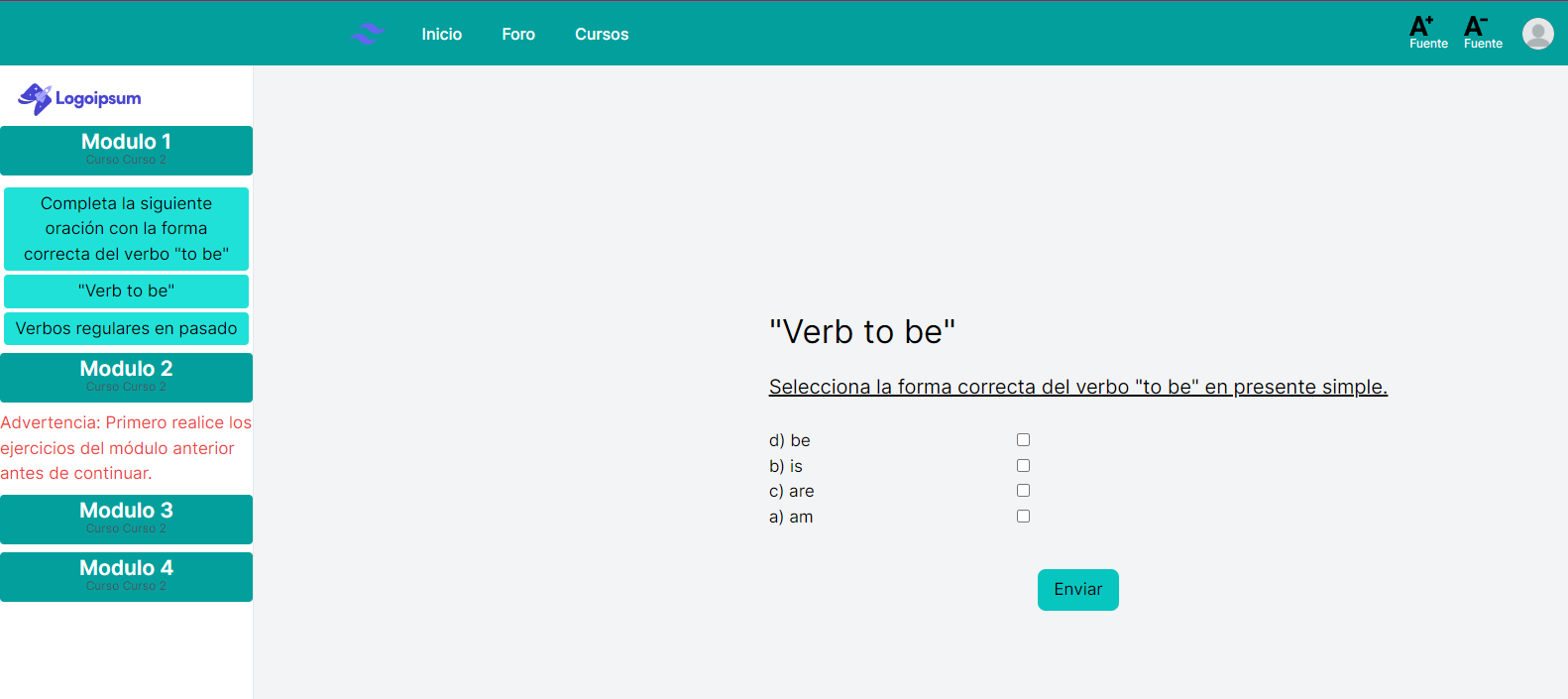
Para realizar los ejercicios el estudiante se tiene que dirigir a “Cursos” y se le mostrarán todos los cursos a los que está inscripto, y solamente le tiene que dar click al botón de “Entrar” en el curso en el que quiera avanzar con sus ejercicios.



Una vez dentro del curso que eligió se le mostrarán los módulos con sus respectivos ejercicios en el lateral izquierdo, y el contenido bibliográfico del curso que el profesor pudo haber subido.



Luego una vez dentro del ejercicio que quiera realizar se le mostrará lo siguiente, en este caso un ejercicio multiple choise.



Luego debe elegir la opción que decida y solamente dar click en “Enviar” y automáticamente se corrige y le muestra el resultado.

## Entregables

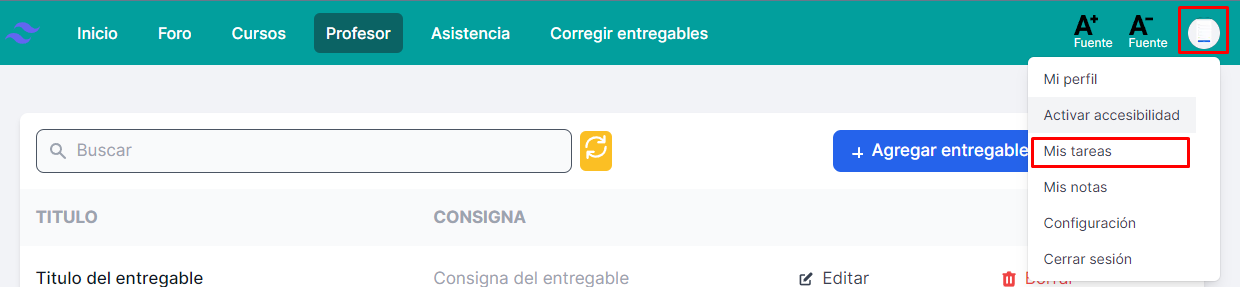
### Crear un entregable

El profesor deberá dirigirse a la sección de “Profesor”, luego a “Acciones” y posteriormente a “Entregables”, donde podrá crear el entregable con su título, consigna, documento pdf que es opcional, fecha límite y a qué curso pertenece. Una vez creado se le asignará esa tarea a cada estudiante inscripto a ese curso

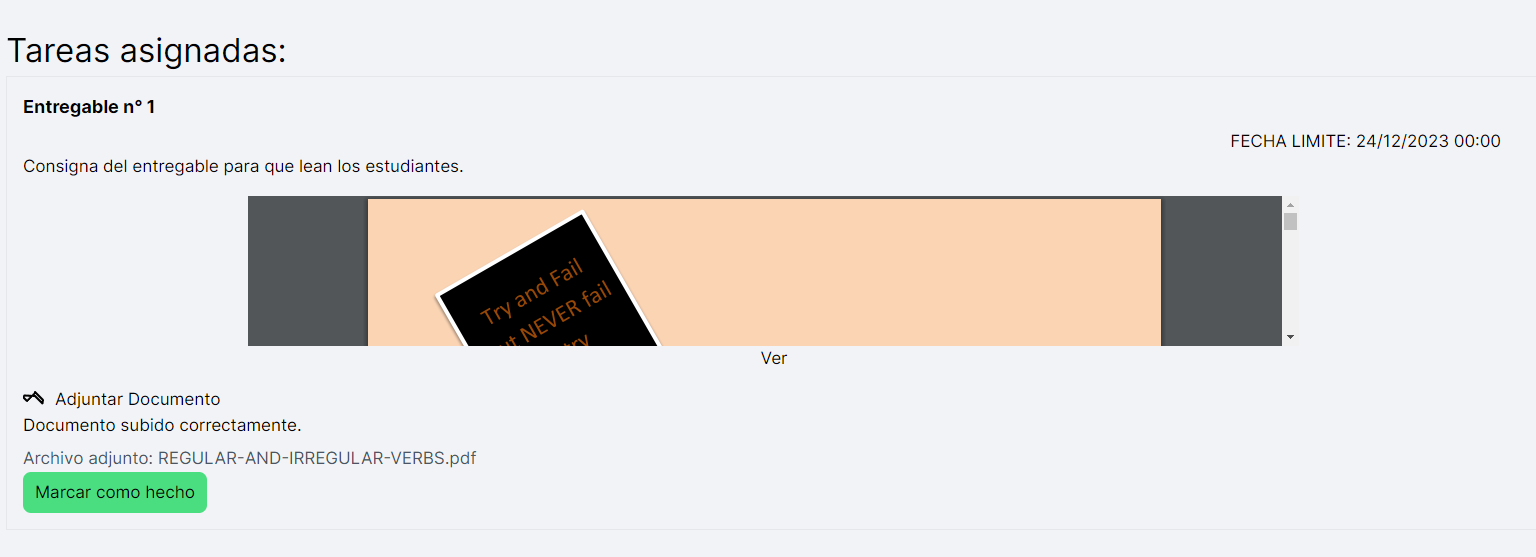


### Cómo el estudiante debe enviar su tarea para corregir

El estudiante deberá clickear en su imagen del perfil dentro de la barra de navegación, y dirigirse a “Mis tareas”.



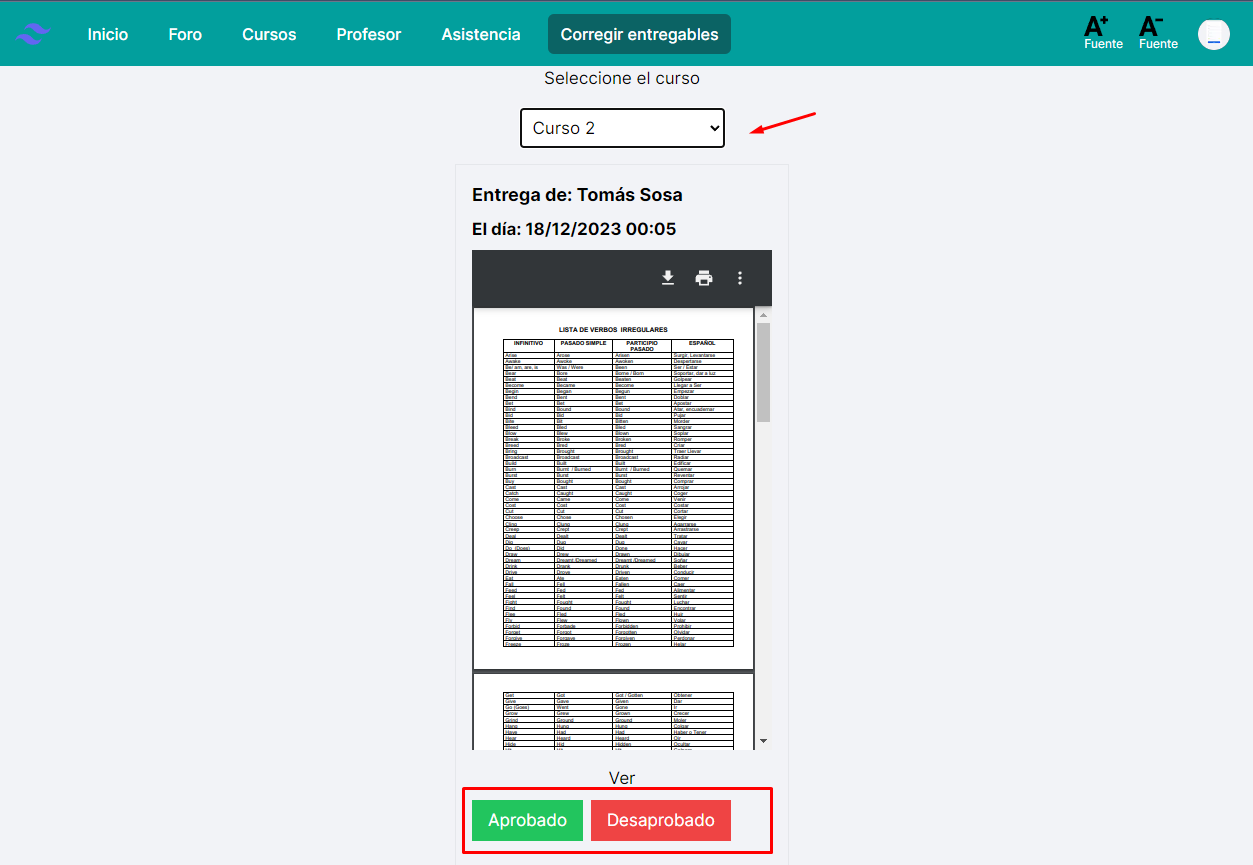
Y le aparecerán todas las tareas asignadas para entregar, de la siguiente manera.



Y para enviarla solamente tiene que adjuntar el documento para corregir y marcar como hecho.

### Cómo el profesor debe corregir los entregables

Tiene que dirigirse al botón de “Corregir entregables” en la barra de navegación. Y tiene que seleccionar el curso del entregable a corregir y se le mostrarán todos los entregables enviados por los estudiantes, con la opción de ver para expandir el documento, y marcar como aprobado o desaprobado, una vez clickeada alguna de estas dos opciones, se le mostrará el siguiente entregable a corregir, y así hasta terminar de corregir todos.

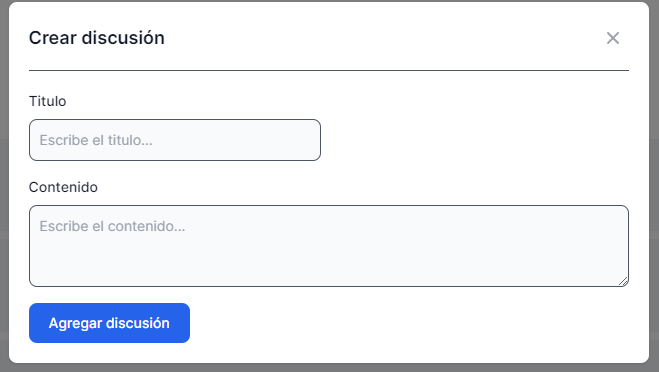


## Foro de discusión

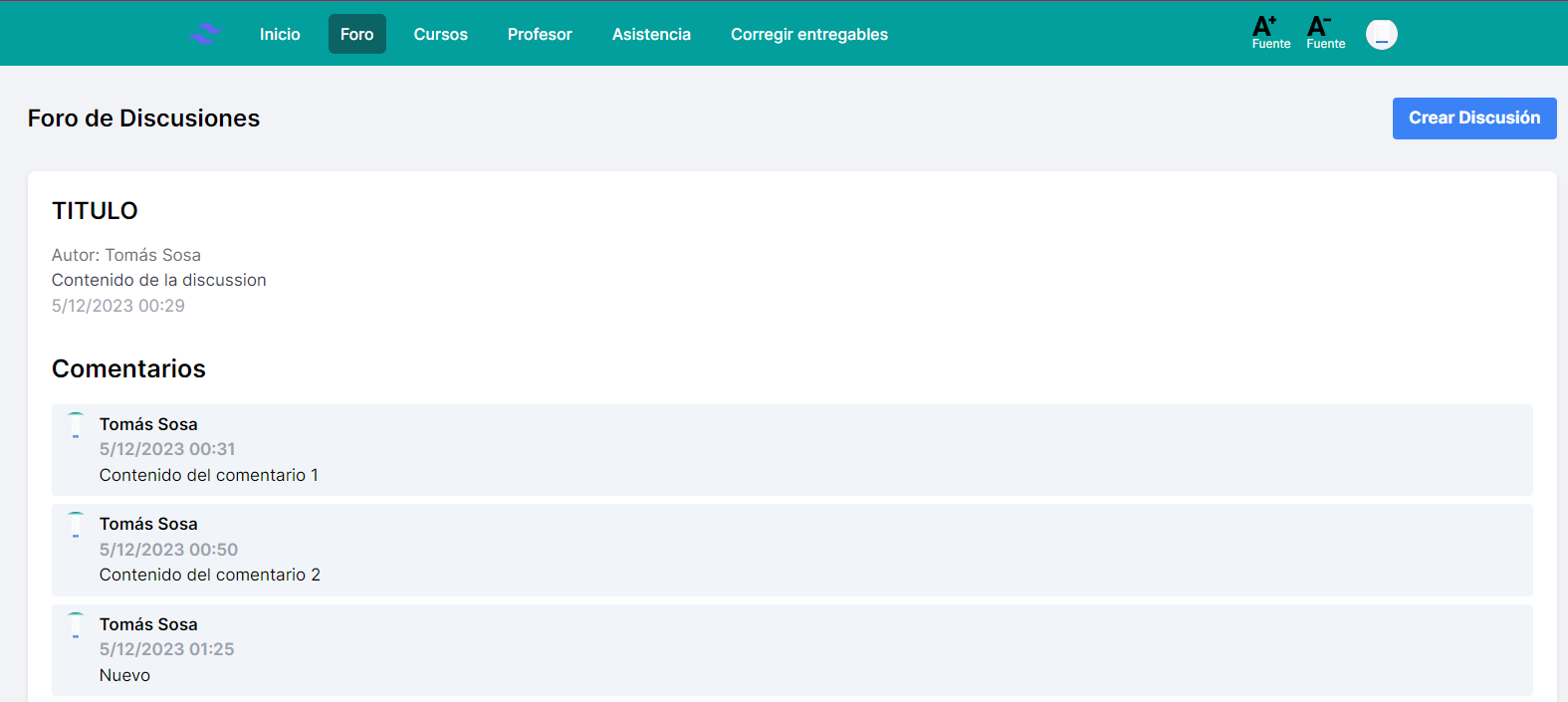
El foro de discusión se puede usar para hacer consultas sobre algo relacionado con la institución, del campus, o para dar avisos de algún evento.

### Crear discusión

Se tiene que dirigir al botón “Foro” dentro de la barra de navegación y darle al botón de “Crear discusión”, y se le mostrará lo siguiente.

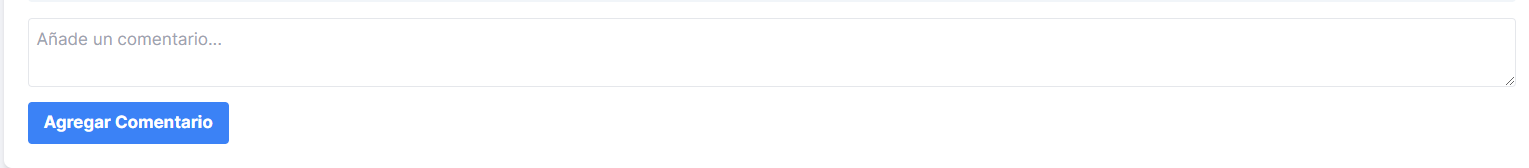


Tendrá que llenar los campos y se le mostrará todas las discusiones con sus respectivos comentarios de la siguiente manera.



### Añadir comentarios a una discusión

Para añadir un comentario solo basta dirigirse a la parte inferior de una discusión y se le mostrará lo siguiente.

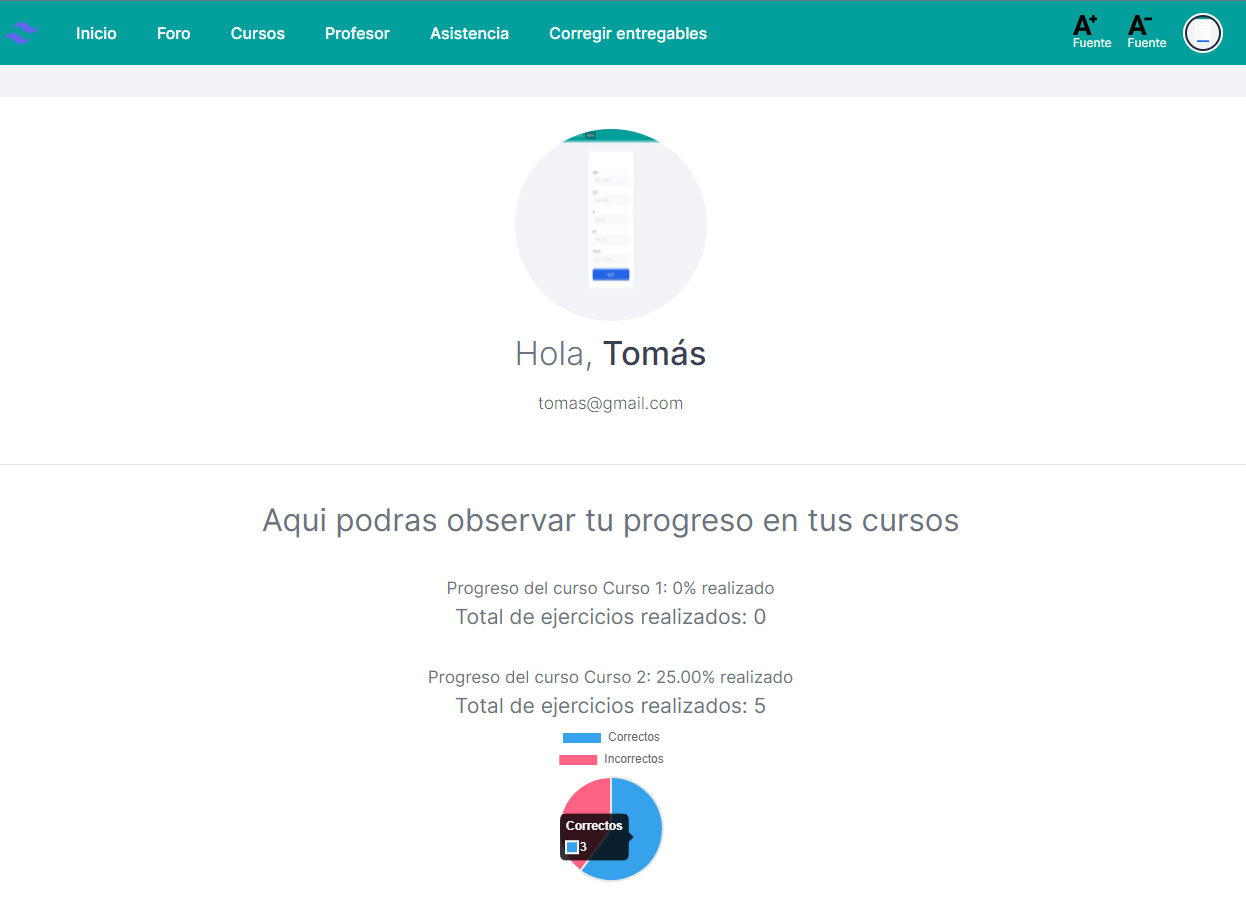


Puede hacer su comentario, por ejemplo, contestando alguna duda. Solo deberá clickear en “Agregar comentario” y ya se posteará para esa discusión.

## Perfil del usuario

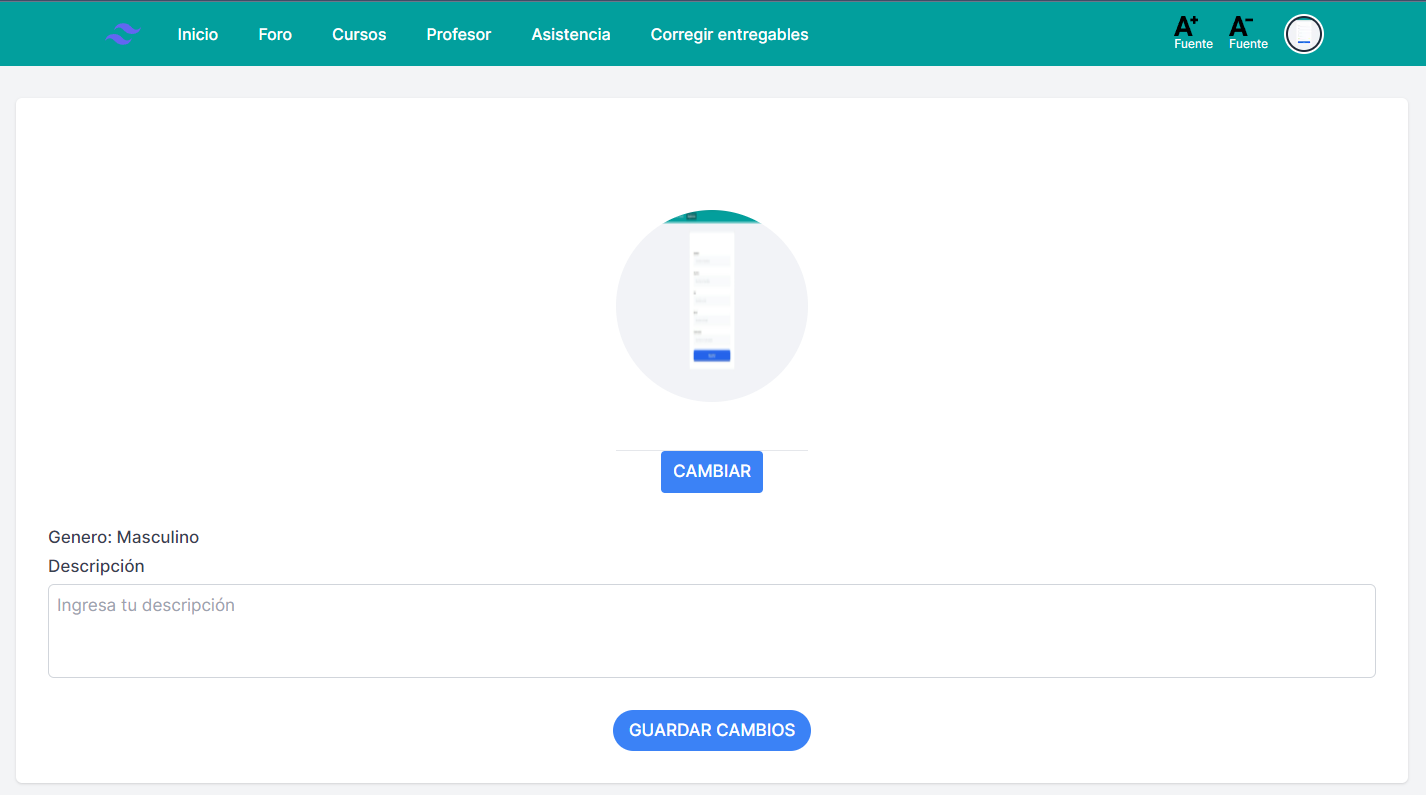
### Su progreso

El estudiante puede ver su perfil haciendo click en su imagen y dirigiéndose a “Mi perfil”, donde podrá ver algunos de sus datos personales, y el progreso que tiene con respecto a los ejercicios que hizo de los cursos en los que está inscripto, como la siguiente imagen.



### Configuración de su perfil

El usuario podrá cambiar su foto de perfil y su descripción en el botón de “Configuración” dentro del desplegable que se muestra al hacer click en su imagen en la barra de navegación.



### Activar modo accesibilidad

Se puede activar el modo de accesibilidad clickeando en “Activar accesibilidad” en el menú desplegable al clickear su imagen en la barra de navegación. Esto permite aumentar o disminuir el tamaño de la fuente de todo el Campus para personas con dificultades visuales.



# Conclusión

Este proyecto ha sido un desafío muy grande porque se propuso trascender los limites convencionales de la educación. En el corazón de esta iniciativa, se busca cubrir diversos aspectos de la vida académica, desde la gestión de notas y promedios hasta la entrega y corrección de tareas. Busqué transformar la experiencia educativa, llevando procesos como la carga de contenido bibliográfico y la corrección de ejercicios automáticamente.

El alcance del proyecto abarca un amplio espectro, reflejando la ambición de proporcionar una solución integral para una escuela moderna. Desde el diseño del backend y la implementación de una base de datos sólida hasta la creación de una interfaz de usuario intuitiva y atractiva, cada fase de desarrollo ha sido una oportunidad de demostrar mi habilidad y creatividad.

# Códigos QR con los repositorios

## Front-end



## Back-end



## Control de asistencia

