Karla Garicoch

Jesica Toloza

Documentación

Sistema de gestión de estudiantes

Contenido

[1. Introducción 2](#_Toc180294932)

[2. Bibliografía 2](#_Toc180294933)

[3. Responsabilidades y roles de los estudiantes 2](#_Toc180294934)

[A) Responsabilidades de Jesica: 2](#_Toc180294935)

[B) Responsabilidades de Karla: 3](#_Toc180294936)

[4. Funcionamiento de la aplicación 3](#_Toc180294937)

[4.0. Roles del sistema: administrador, profesor y alumno 3](#_Toc180294938)

[4.1. Inicio de sesión 3](#_Toc180294939)

[4.2. Registro de estudiantes 4](#_Toc180294940)

[4.3. Registro de notas 4](#_Toc180294941)

[4.4. Vistas y motor de plantillas Pug 4](#_Toc180294942)

[4.5 Uso de JSON para almacenamiento 4](#_Toc180294943)

[4.6. Rutas de la aplicación 4](#_Toc180294944)

[5\_ Manejo de la aplicación 5](#_Toc180294945)

[5.1 Pantalla de inicio de sesión (Login) 5](#_Toc180294946)

[5.2 Panel de control del administrador 5](#_Toc180294947)

[5.3. Registro de un nuevo alumno por parte del administrador 5](#_Toc180294948)

[5.4 Inscripción a materias por parte del administrador 6](#_Toc180294949)

[5.5 Registro de calificaciones por parte del profesor 7](#_Toc180294950)

[5.6 Pantalla de acceso del alumno 8](#_Toc180294951)

[6. Pruebas con Thunder client 10](#_Toc180294952)

[7. Video explicativo 11](#_Toc180294953)

# 1. Introducción

El proyecto que desarrollamos es un sistema de gestión de alumnos que permite a los administradores registrar nuevos estudiantes y a los profesores asignar calificaciones, todo ello mediante una interfaz web sencilla. Tanto administradores como profesores y, luego de su registro, los alumnos pueden acceder a la aplicación mediante un sistema de inicio de sesión.

La aplicación está construida utilizando Node.js, Express, un motor de plantillas Pug y un sistema de almacenamiento basado en archivos JSON para persistir la información.

# 2. Bibliografía

Para desarrollar el sistema de gestión de alumnos, consultamos diversas fuentes de información, incluyendo tutoriales, videos y documentación oficial, que nos ayudaron a profundizar en el manejo de las tecnologías necesarias. Entre las fuentes utilizadas se incluyen:

* Documentación oficial de Express.js: <https://expressjs.com/>
* Documentación de Node.js: <https://nodejs.org/en/docs/>
* Curso de Node.js: [Node Js desde cero](https://youtube.com/playlist?list=PL9prAn3zOWZ403Yz14PqckSq0nHeY_4yj&si=UWgQJof78D3x8VPm)
* Tutorial de Pug: [tutorial](https://youtu.be/K7jysT0T8QM?si=LXOVD7u1bqEMrIb2)
* Material de clases

# 3. Responsabilidades y roles de los estudiantes

Aunque establecimos roles y responsabilidades, la mayor parte del trabajo fue muy colaborativo. Al ser solo dos integrantes, nos pareció lo más adecuado que sea de esa manera y por eso nos reunimos a través de videollamadas para discutir los detalles del proyecto y escribir el código juntas. Esto permitió que ambas pudiéramos tener una visión global del desarrollo y contribuir en cada una de las áreas.

## A) Responsabilidades de Jesica:

* **Diseño y estructura del proyecto**: definió la estructura organizada de carpetas, separando rutas, modelos, vistas y archivos estáticos.
* **Rutas del administrador y registro de estudiantes**: implementó las rutas necesarias para el registro de nuevos estudiantes y el manejo de usuarios administradores.
* **Manejo de archivos JSON**: desarrolló el sistema de almacenamiento utilizando archivos JSON para persistir la información de los estudiantes y administradores.
* **Pruebas en Thunder Client**: realizó las pruebas de las rutas usando Thunder Client para verificar el correcto funcionamiento.
* **Documentación**: redactó parte de la documentación del proyecto.

## B) Responsabilidades de Karla:

* **Rutas del profesor y registro de calificaciones**: desarrolló las rutas que permiten a los profesores asignar calificaciones a los estudiantes.
* **Inicio de sesión para profesores y administradores**: implementó la lógica de autenticación para permitir que los usuarios inicien sesión de manera segura.
* **Configuración del motor de plantillas Pug**: configuró las plantillas para las vistas del sistema, como login, registro de estudiantes y calificaciones.
* **Pruebas funcionales**: realizó pruebas generales para asegurar que las características del sistema funcionaran correctamente.
* **Video explicativo**: propuso la forma en la que debería garbarse el video

# 4. Funcionamiento de la aplicación

## 4.0. Roles del sistema: administrador, profesor y alumno

El sistema de gestión de alumnos está diseñado para manejar tres roles principales:

* **Administrador:** responsable de registrar nuevos estudiantes y gestionar sus datos.
* **Profesor:** encargado de asignar calificaciones
* **Alumno:** tiene acceso al sistema para visualizar sus datos personales, materias inscritas y notas.

Cuando un administrador registra un nuevo alumno, se ingresan varios datos relevantes, como el DNI, correo electrónico, nombre del padre o tuto**r**, un número de contacto, etc. Al completar el registro, el sistema genera automáticamente una contraseña aleatoria para el alumno.

El alumno puede iniciar sesión usando su DNI como nombre de usuario y la contraseña generada en el proceso de registro. Al ingresar al sistema el alumno tiene acceso a una página donde puede visualizar su información personal, las materias en las que está inscrito y las notas asignadas por los profesores.

## 4.1. Inicio de sesión

El sistema de autenticación permite que tanto profesores como administradores inicien sesión. Las credenciales de estos usuarios administrador y profesor ya están almacenadas en archivos JSON, y el sistema las verifica cada vez que se intenta acceder.

Se validan las credenciales y, si son correctas, el usuario es redirigido a su respectivo panel (administrador o profesor). El middleware de autenticación protege las rutas de acceso, permitiendo solo a usuarios autenticados acceder.

## 4.2. Registro de estudiantes

El administrador puede registrar nuevos estudiantes completando una serie de datos como nombre, apellido, DNI, fecha de nacimiento, email y datos del padre o tutor. Los datos ingresados se guardan en un archivo JSON.

Al enviar el formulario, se validan los datos y guarda la información en el archivo JSON. Esto actualiza la lista de estudiantes.

## 4.3. Registro de notas

Los profesores pueden buscar a los estudiantes, ingresando su DNI, para asignarles calificaciones y la nota será almacenada en el archivo JSON.

El profesor ingresa el DNI del estudiante, la aplicación busca en el archivo JSON y carga los datos del estudiante. Se presenta un formulario donde el profesor puede asignar las calificaciones, que luego se guardan en el archivo JSON.

## 4.4. Vistas y motor de plantillas Pug

Las vistas de la aplicación se generan con el motor de plantillas Pug. Esto incluye las pantallas de login, registro de estudiantes, inscripción a materias, asignación de notas, etc.

Las plantillas Pug están organizadas por roles y se encuentran en la carpeta /views. Según el tipo de usuario que haya iniciado sesión, las vistas renderizan el contenido pertinente.

## 4.5 Uso de JSON para almacenamiento

Toda la información relacionada con usuarios, estudiantes y sus calificaciones se almacena en archivos JSON. Estos actúan como una base de datos sencilla para persistir la información.

Cada vez que se necesita leer o escribir datos, la aplicación accede al archivo JSON correspondiente.

## 4.6. Rutas de la aplicación

La aplicación cuenta con varias rutas clave:

* /login: para iniciar sesión.
* /admin: permite al administrador registrar estudiantes.
* /profesor: permite al profesor buscar estudiantes por DNI y registrar calificaciones.
* /alumno: permite al alumno ver su información como materias y notas

# 5\_ Manejo de la aplicación

Vamos a mostrar el flujo de funcionamiento de la aplicación de gestión de alumnos. Cada paso del proceso está documentado visualmente para reflejar cómo interactúan los diferentes roles en la aplicación.

## 5.1 Pantalla de inicio de sesión (Login)

La aplicación comienza con la pantalla de login, donde administradores, profesores y alumnos deben ingresar sus credenciales para acceder. Los administradores y profesores ya tienen su clave asignada, mientras que los alumnos deben usar su DNI como nombre de usuario y la contraseña generada al momento del registro.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 5.2 Panel de control del administrador

Una vez que era administrador inicia sesión con su usuario y contraseña va a ser redirigido a un panel donde va a tener dos opciones: registro de estudiantes o inscripción de materias

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 5.3. Registro de un nuevo alumno por parte del administrador

Si el administrador selecciona la opción registrar estudiantes, debe llenar una serie de datos sobre el alumno como nombre, apellido, DNI, email, etc, para generar el registro del mismo. Al llenar el formulario con los datos del estudiante el sistema genera una contraseña aleatoria para el alumno.

Posteriormente el alumno podrá ingresar al sistema iniciando sesión con su DNI y esta clave que se generó en su registro



Al dar click en registrar alumno el sistema nos va a mostrar un mensaje de usuario registrado y la contraseña generada para el mismo.

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

## 5.4 Inscripción a materias por parte del administrador

En el panel administrativo, el administrador, puede inscribir a materias a los alumnos. Para ello una vez que da click en el botón inscribir materia, se dirige a un buscador de alumnos mediante su DNI. El administrador debe ingresar el DNI del alumno que quiere inscribir a materias y la aplicación va a buscarlo y traer la información correspondiente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El administrados debe seleccionar las materias a inscribir y luego dar click en el botón inscribir alumno. El sistema me va a dar un mensaje del registro exitoso de las materias

Interfaz de usuario gráfica

Descripción generada automáticamente

## 5.5 Registro de calificaciones por parte del profesor

Una vez que el profesor ha iniciado sesión con sus correspondientes credenciales, accede a su panel donde se va a encontrar con un buscador de alumnos mediante DNI.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El profesor debe ingresar el DNI del alumno al que quiere asignarle notas en sus materias y el sistema va a traer la información correspondiente y va a poder asignarle una calificación numérica a cada materia a la que el alumno este inscripto.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez que se hace click en el botón para inscribir las notas, el sistema nos muestra un mensaje de que las notas fueron actualizadas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## 5.6 Pantalla de acceso del alumno

Cuando un alumno inicia sesión con su DNI y la contraseña generada por el sistema, accede a su perfil personal. Aquí puede ver un mensaje de bienvenida junto a sus datos personales, las materias inscritas y las calificaciones obtenidas en cada una.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

El sistema de gestión de estudiantes esta desarrollado para que nos avise cuando se ingresan datos que son incorrectos.

Por ejemplo, si queremos ingresar con usuario o contraseña que no son válidos nos muestra el mensaje de error. De la misma manera si buscamos el DNI de un alumno que no está registrado, etc.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

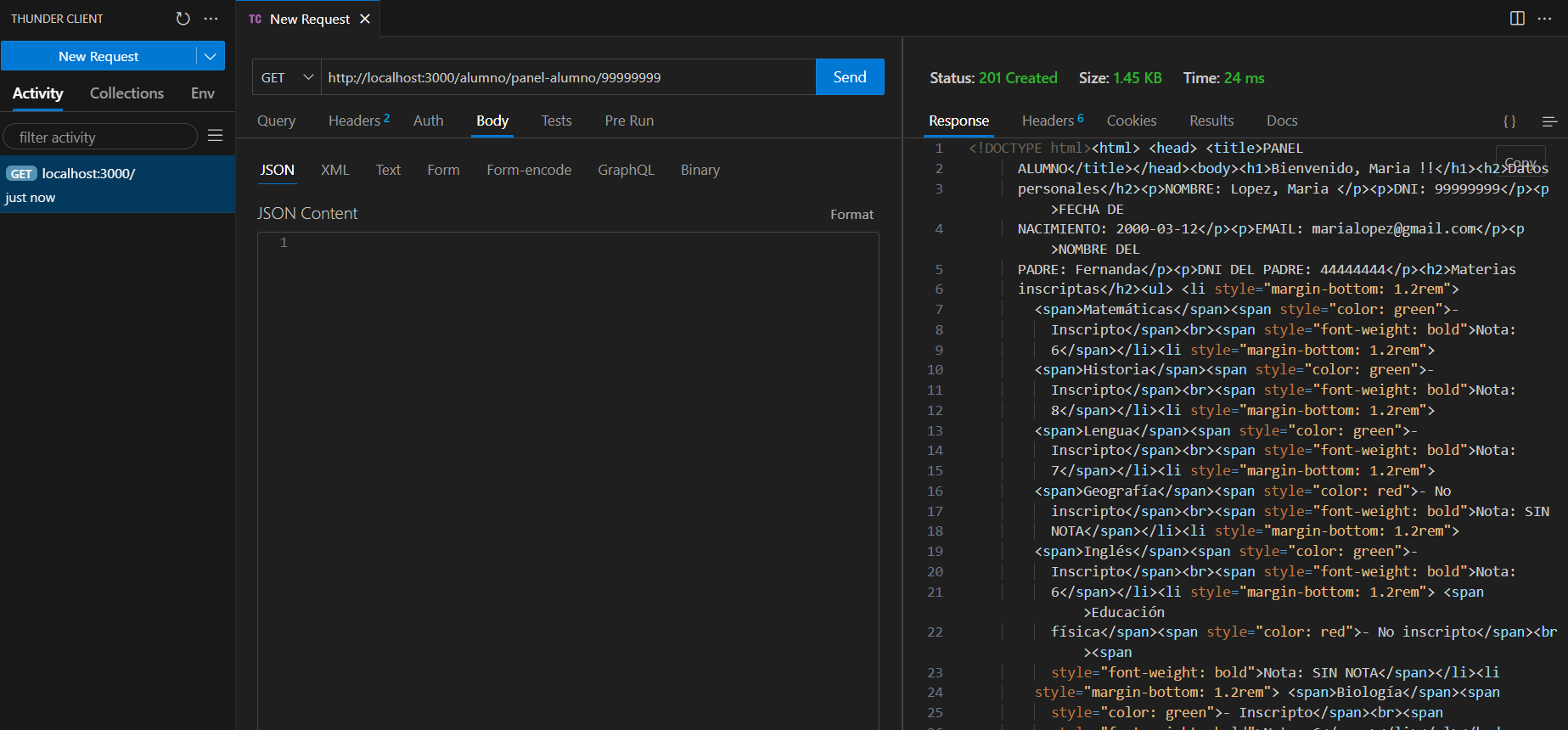
# 6. Pruebas con Thunder client

**6.1 Prueba del método GET para Visualización de datos del estudiante:**

Prueba la visualización de los datos de un estudiante después de iniciar sesión.

Método: GET

Url: http://localhost:3000/alumno/panel-alumno/99999999



**6.2 Prueba del método GET para búsqueda de estudiantes por DNI:**

Realiza una prueba para buscar un estudiante usando su DNI.

Método**:** GET

URL**:** http://localhost:3000/profesor/panel-profesor

Captura de pantalla de computadora

Descripción generada automáticamente

# 7. Video explicativo