Documento de Propuesta de Diseño de Software I, II y II

Módulo actividades interactivas

Autores: Diego Andrés Barón Barrera - Reiner Yair Infante Díaz



Breve reseña

Este módulo tiene como propuesta brindar un espacio esencial para los usuarios en el panorama de la gestión de actividades interactivas dentro de la aplicación CREAVI, esto con el objetivo de fortalecer tanto la herramienta mediadora (creavi) como los procesos académicos de los docentes. Por otra parte, la cantidad de plataformas y aplicaciones digitales que existen hoy día para apoyar a los procesos educativos son una fuerte competencia para cualquier otra aplicación que quiera aparecer en el mundo digital, por eso se debe cumplir con la realización de un trabajo eficiente que permita a los usuarios suplir sus necesidades académicas.

Nivel 1: Diseño de la Aplicación y Análisis de Requisitos

1. Introducción al Diseño de Software
2. Identificación de Necesidades del Usuario
3. Creación de Documentos de Especificación de Requisitos
4. Casos de Uso y Diagramas de Casos de Uso
5. Modelado de Entidad-Relación (E/R)
6. Herramientas para el Diseño de Software (p. ej. Lucidchart, Draw.io)

Contenido

[Breve reseña 1](#_Toc152835331)

[Introducción 2](#_Toc152835332)

[Descripción General: 6](#_Toc152835333)

[CASOS DE USO - DEFINIDOS 7](#_Toc152835334)

[Modelado E/R: 20](#_Toc152835335)

[NIVEL 2: Introducción a la Persistencia de Datos 23](#_Toc152835336)

[INTRODUCCIÓN 23](#_Toc152835337)

[DISEÑO DE LA ARQUITECTURA DEL BACKEND 24](#_Toc152835338)

[ELECCIÓN DE LA BASE DE DATOS 25](#_Toc152835339)

[Diseño de Esquema de Base de Datos: 27](#_Toc152835340)

[IMPLEMENTACIÓN DEL BACKEND 32](#_Toc152835341)

[Desarrollo de Endpoints y APIs 34](#_Toc152835342)

[Pruebas del Backend 37](#_Toc152835343)

[Nivel 3: Consumo de Datos y Desarrollo Frontend 37](#_Toc152835344)

[Etapa 3: Consumo de Datos y Desarrollo Frontend 40](#_Toc152835345)

Introducción

El diseño de software es una fase crucial en el desarrollo de cualquier sistema informático, y la creación de módulos desempeña un papel fundamental en este proceso. El diseño de software se centra en la estructuración y organización de un sistema para satisfacer los requisitos funcionales y no funcionales, mientras que los módulos son unidades independientes y autónomas que componen el sistema de software más grande.

Propósito del Documento:

El propósito de la documentación se define a partir de un formato donde se detallan todos los aspectos fundamentales que se tienen en cuenta para realizar el proyecto.

A continuación, se muestra el documento donde se evidencia distintos aspectos del propósito:

Identificación de la necesidad del usuario

|  |  |
| --- | --- |
| **ANÁLISIS DE LA NECESIDAD EDUCATIVA** | |
| **Tipo:** expresada | La necesidad por tratar es de tipología expresada, debido a que los docentes de las instituciones educativas de Córdoba expresan la necesidad de contar con un apartado independiente dentro de la aplicación “Creavi” que les permita subir contenidos de actividades interactivas desde el marco de trabajo colaborativo de contenidos libre y de fuente abierta basado en Javascript conocido como H5P. |
| **Identificación del aprendizaje ideal** | |
| Los docentes deben conocer e identificar:  ✔ Conocimiento básico de la plataforma Creavi  ✔ Vinculación de actividades a plataformas  ✔ Conocimiento básico en el manejo de h5p. | |
| **Población** | **Rango de edad: Personal mayor de edad** |
| **Escolaridad:** |
| Conocimiento que posee: Los docentes poseen conocimientos sobre algunas aplicaciones que sirven para el desarrollo de actividades interactivas, pero estas aplicaciones no les permite incorporar los contenidos a CREAVI y también los limita en los contenidos que pueden brindarles. |
| Intereses y expectativas: (Población)  Contar con espacios propios para la gestión de contenidos y actividades interactivas de forma abierta y autogestionada. |
| Intereses y expectativas: (Creadores)  Ayudar a los docentes a gestionar espacios para incorporar contenidos de actividades interactivas desde H5P a la aplicación Creavi. |
| **Área de formación** | **Área del saber:** Transversal |
| **Área de contenido:** Gestión de contenidos digitales |
| **Estado actual** | **Diagnóstico:**  Los docentes que utilizan la aplicación CREAVI, no cuentan con un espacio que les permita subir y gestionar los contenidos de actividades interactivas para poder compartirlas con la comunidad estudiantil y que de esta forma el material interactivo de apoyo que ellos desarrollan sea útil y de mucha ayuda para el éxito en el progreso de cada actividad por parte de los estudiantes. |

|  |  |
| --- | --- |
| **Necesidad** | La necesidad educativa parte de la problemática que tienen los docentes para la gestión y carga de contenidos y actividades interactivas creadas en la plataforma H5p. Para esto es necesario que los docentes conozcan las bases de manejo de la plataforma H5P y el manejo de subida de contenidos a diferentes plataformas, en este caso Creavi. |
| **Causas** | * No existe un espacio dentro de la aplicación CREAVI que les permita subir componentes de H5P como material de apoyo a los docentes y esto puede limitar el desarrollo de los procesos académicos. * Una deficiente formación académica por parte de los estudiantes en el aspecto de resolución de problemas, debido a que si no encuentran actividades para resolver, es imposible determinar si poseen los conocimientos esenciales para desarrollar dichas actividades. * Una deserción de la aplicación CREAVI hacia otras aplicaciones o plataformas educativas que sí les de la opción de subir actividades interactivas para compartirlas con sus estudiantes. |
| **Soluciones** | * Mejorar los procesos de gestión de contenidos interactivos para los docentes y facilitar la obtención de diferentes contenidos interactivos en el ámbito educativo. |
| **Conocimientos y habilidades que debe tener el Docente.** | Preconceptos:   * Conocimiento previo de la plataforma H5p. * Conocimiento previo de la plataforma Creavi. |
| Precondiciones:  Conocimientos sobre la creación de actividades interactivas.  Conocimiento y uso de la plataforma H5P  Conocimiento de la plataforma Creavi |
| **Justificación**  Teniendo en cuenta todos los aspectos relacionados a la necesidad encontrada, la justificación radica en brindar un espacio esencial para los usuarios en el panorama de la gestión de contenidos de actividades interactivas dentro de la aplicación CREAVI, esto con el objetivo de fortalecer tanto la herramienta mediadora (creavi) como los procesos académicos de los docentes. Por otra parte, la cantidad de plataformas y aplicaciones digitales que existen hoy día para apoyar a los procesos educativos son una fuerte competencia para cualquier otra aplicación que quiera aparecer en el mundo digital. Por ende, una solución efectiva para abordar la falencia que se determinó en el análisis de la necesidad educativa y para convertir a CREAVI en una herramienta reconocida por sus características funcionales, sería diseñar y desarrollar un apartado dentro de la aplicación CREAVI que les permita a los docentes contar con la función y la opción de subir y gestionar sus actividades interactivas. | |

ALCANCE DEL PROYECTO:

Durante el desarrollo del proyecto se espera alcanzar la meta propuesta, la cual se define como diseñar un apartado independiente dentro de la aplicación “Creavi” que permita subir contenidos de actividades interactivas desde el marco de trabajo colaborativo de contenidos libre y de fuente abierta basado en Javascript conocido como H5P vinculado directamente al estándar de tecnología de aprendizaje llamada Xapi,

DEFINICIONES Y ACRÓNIMOS

**Software**: El software es un conjunto de programas, instrucciones y datos que permiten que una computadora o un dispositivo electrónico realicen tareas específicas, es decir, es la parte interna o lógica de un sistema informático que reparte la capacidad de realizar diversas funciones o tareas.

**Bases de datos:** es una colección organizada y estructurada de datos o información que se almacena de forma sistemática para su posterior acceso, consulta, actualización y gestión. Estos datos suelen estar relacionados de alguna manera y se organizan en tablas, registros y campos, lo que facilita su manipulación y búsqueda. En resumen, una base de datos es un repositorio de datos que se utiliza para almacenar, recuperar y administrar información de manera eficiente.

**Backend**: Es la capa del sistema que es No visible para el usuario final y que se encarga de procesar datos, gestionar la lógica empresarial, interactuar con la base de datos y enviar información al frontend (la parte que se ve y con la que interactúa el usuario).

**H5P**: Es una plataforma de código abierto que permite crear, compartir y reutilizar contenido interactivo en línea, esta se centra en la creación de contenido interactivo enriquecido utilizando tecnologías web estándar como HTML5, JavaScript y CSS. Además, proporciona un conjunto de herramientas y tipos de contenidos predefinidos que permiten a los usuarios crear fácilmente actividades interactivas, como cuestionarios, presentaciones, juegos, simulaciones, videos interactivos y más.

**Xapi:** Es un estándar de tecnología para el aprendizaje electrónico (e-learning) que facilita el seguimiento y la información de diversas actividades de aprendizaje, tanto en entornos en línea como fuera de línea. Se desarrolló como una evolución del estándar SCORM, con el propósito de superar las limitaciones de este último, centrándose especialmente en mejorar la flexibilidad y la capacidad de seguimiento de las interacciones de los usuarios con el contenido educativo. En resumen, xAPI busca proporcionar un marco más robusto y adaptable para rastrear el progreso del aprendizaje en diversos contextos.

Descripción General:

Objetivos del Sistema:

* Crear un entorno dentro de la aplicación que sea fácilmente accesible y utilizable por parte de los usuarios, incluyendo docentes y estudiantes, donde se garantice que el sistema ofrezca una experiencia amigable e intuitiva para fomentar su adopción y uso efectivo. También se busca proporcionar funciones de seguimiento y evaluación como Xapi que permitan a los docentes analizar el rendimiento e interacción de los estudiantes con las actividades y mejorar las estrategias de enseñanza.

Funcionalidad General:

* Integración fluida entre el módulo y la funcionalidad principal de CREAVI.
* Mejora y optimización de la herramienta CREAVI para facilitar la gestión de actividades interactivas.
* Posibilidad de personalizar y adaptar las actividades según las necesidades específicas de cada curso o grupo de usuarios.

Usuarios del Sistema:

* Docentes: Utilizarán el sistema para gestionar y programar actividades interactivas en el entorno académico a través de CREAVI y realizarán un seguimiento del progreso de los estudiantes con Xapi y evaluarán sus desempeños.
* Estudiantes: Son los usuarios que participan en actividades interactivas, reciben retroalimentación y realizan un seguimiento de su propio progreso.
* Administrador: Supervisará y administrará la implementación del módulo en el contexto general de la aplicación CREAVI.

Restricciones:

* Restricciones en el tiempo y los recursos para capacitar a los usuarios, como docentes, estudiantes y administradores, en el uso efectivo del sistema.
* Restricciones en la capacidad del sistema para manejar un aumento significativo en la cantidad de usuarios o datos sin degradar el rendimiento.

Requisitos Funcionales:

* Casos de Uso**:** En el desarrollo de software, los casos de uso son utilizados para capturar, modelar y especificar los diferentes escenarios y situaciones en los que los usuarios interactúan con el sistema.

A continuación, se muestra un diagrama general sobre el caso de uso del módulo de actividades interactivas.

CASOS DE USO - DEFINIDOS

1. Subir archivo h5p
2. Visualizar un archivo h5p
3. Eliminar archivo h5p
4. Modificar un archivo h5p
5. Comentar un recurso h5p (estudiantes)
6. Descargar un archivo h5p
7. Leer xAPI de h5p
8. Realizar seguimientos de interacciones xAPI

* **Descripción detallada de cada caso de uso**

1. **Subir archivo h5p:**

**Actores**:

Usuario Profesor

**Descripción**:

1. Seleccionar Archivo H5P: El usuario navega a la sección correspondiente y selecciona el archivo H5P desde su dispositivo local.
2. Validación del Formato: El sistema verifica la validez del formato del archivo H5P para garantizar la compatibilidad.
3. Ingresar Metadatos: El usuario proporciona información adicional, como título y descripción, para identificar la actividad.
4. Subir Archivo: El usuario confirma la carga, y el sistema almacena el archivo H5P en la base de datos, asociándolo con la información proporcionada.
5. Confirmación: El sistema notifica al usuario sobre el éxito del proceso de carga.

**CASO No. 1 Subir archivo h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-1 | |
| **Nombre** | Subir archivo | |
| **Actores** | Docente | |
| **Objetivo** | Subir un archivo H5P que contenga una actividad interactiva | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Debe haberse autenticado de forma correcta en el sistema. | |
| **Flujo Normal** | Docente | Sistema |
| clic en el botón "Subir archivo" |  |
|  | despliega una ventana donde el usuario busca la ubicación del archivo |
| Selecciona el archivo h5p |  |
|  | valida el formato del archivo |
| Ingresa metadatos (título, descripción). Confirma la carga. |  |
| clic en el botón "Guardar" |  |
|  | Almacena el archivo y notifica el éxito. |
| **Flujo alternativo 1** | Usuario proporciona formato no válido |  |
|  | Sistema muestra mensaje de error. |
|  |  |
| **Flujo alternativo 2** | Usuario cancela la carga. |  |
|  | El sistema cancela la operación y notifica. |
|  |  |
| **Post-condiciones** | Archivo H5P se almacena en la base de datos asociado a los metadatos. |  |
| **Exepciones** |  |  |
|  |  |

**b) Visualizar archivo h5p:**

**Actores**:

Usuario Profesor o Estudiante

**Descripción:**

1. Seleccionar Actividad H5P: El usuario navega a la lista de actividades H5P disponibles y selecciona la actividad de interés.
2. Cargar Interfaz Interactiva: El sistema carga la interfaz interactiva correspondiente al archivo H5P seleccionado.
3. Interacción con la Actividad: El usuario interactúa con la actividad, explorando su contenido y funcionalidades.
4. Finalización: Después de revisar la actividad, el usuario puede cerrar la interfaz o regresar a la lista de actividades.

**CASO No. 2 Visualizar archivo h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-2 | |
| **Nombre** | Visualizar archivo | |
| **Actores** | Docente y/o estudiantes | |
| **Objetivo** | Visualizar una actividad H5P existente | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existen actividades H5P disponibles en el sistema | |
| **Flujo Normal** | Docente | Sistema |
| Usuario navega a lista de actividades H5P. Selecciona actividad de interés. |  |
|  | El Sistema carga la interfaz interactiva correspondiente. |
| Usuario interactúa con la actividad. |  |
| Después de revisar, puede cerrar la interfaz o regresar a la lista. |  |
|  |  |
| clic en el botón "volver" |  |
|  | El sistema redirige al usuario a la lista. |
| **Flujo alternativo 1** | Usuario proporciona formato no válido |  |
|  | Sistema muestra mensaje de error. |
|  |  |
| **Flujo alternativo 2** | Usuario selecciona actividad no existente |  |
|  | Sistema muestra mensaje de error |
| Usuario cierra interfaz sin interactuar. |  |
|  |  | Sistema arroja una ventana de diálogo para confirmar el cierre de la interfaz  “¿Seguro que quieres salir?” |
|  | Usuario confirma y da clic en “Sí, quiero salir de todos modos” |  |
|  |  | El Sistema registra la acción y cierra la interfaz. |
| **Post-condiciones** | Usuario visualiza la actividad H5P seleccionada |  |
| **Exepciones** |  |  |
|  |  |

## 

**c) Eliminar Archivo H5P:**

**Actores:**

Usuario Profesor

**Descripción**:

1. Seleccionar Archivo H5P: El usuario navega a la sección de gestión de actividades y selecciona el archivo H5P que desea eliminar.
2. Confirmación de Eliminación: El sistema solicita una confirmación para evitar eliminaciones accidentales.
3. Eliminar Archivo: Si el usuario confirma, el sistema elimina el archivo H5P y cualquier información asociada de la base de datos.
4. Notificación: El sistema notifica al usuario sobre la eliminación exitosa.

**CASO No. 3 Eliminar archivo h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-3 | |
| **Nombre** | Eliminar archivo h5p | |
| **Actores** | Docente | |
| **Objetivo** | Eliminar un archivo H5P existente | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de archivos H5P en la sección de gestión de actividades. | |
| **Flujo Normal** | Docente | Sistema |
| El Usuario navega a la sección de gestión de actividades. Selecciona archivo H5P a eliminar. |  |
|  | El sistema solicita confirmación. |
| Usuario confirma la eliminación. |  |
|  | El sistema elimina el archivo y notifica el éxito. |
| **Flujo alternativo 1** | Usuario cancela la eliminación. |  |
|  | El sistema cancela la operación y notifica. |
|  |  |
| **Flujo alternativo 2** | Usuario selecciona archivo no existente. |  |
|  | Sistema muestra mensaje de error. |
|  |  |
| **Post-condiciones** | Archivo H5P y su información asociada eliminados de la base de datos. |  |
| **Exepciones** |  |  |
|  |  |

**d) Modificar Archivo H5P:**

**Actores**:

Usuario Profesor

**Descripción:**

1. Seleccionar Archivo H5P: El usuario navega a la sección de gestión de actividades y selecciona el archivo H5P que desea modificar.
2. Realizar Modificaciones: El usuario realiza las modificaciones necesarias en el contenido, descripción o metadatos de la actividad.
3. Guardar Cambios: El sistema actualiza la información en la base de datos, reflejando las modificaciones realizadas.
4. Notificación: El sistema notifica al usuario sobre el éxito de la modificación.

**CASO No. 4 Modificar archivo h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-4 | |
| **Nombre** | Modificar archivo h5p | |
| **Actores** | Docente | |
| **Objetivo** | Modificar un archivo H5P existente | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de archivos H5P en la sección de gestión de actividades. | |
| **Flujo Normal** | Docente | Sistema |
| El Usuario navega a la sección de gestión de actividades. Selecciona archivo H5P a modificar. Realiza las modificaciones necesarias. Guarda los cambios. |  |
|  | El Sistema actualiza la información y notifica el éxito. |
| **Flujo alternativo 1** | Usuario cancela las modificaciones. |  |
|  | El Sistema mantiene la información original. |
|  |  |
| **Flujo alternativo 2** | Usuario selecciona archivo no existente. |  |
|  | Sistema muestra mensaje de error. |
|  |  |
| **Post-condiciones** | Información del archivo H5P se actualiza en la base de datos |  |
| **Exepciones** |  |  |
|  |  |

**e) Comentar un Recurso H5P (Estudiantes):**

**Actores:**

Estudiante

**Descripción:**

1. Interacción con la Actividad: El estudiante interactúa con una actividad H5P asignada.
2. Agregar Comentario: Después de la interacción, el estudiante puede agregar comentarios para expresar dudas, reflexiones o preguntas relacionadas con la actividad.
3. Guardar Comentario: El sistema almacena el comentario en la base de datos, asociándolo con la actividad correspondiente.
4. Visualización de comentarios: Otros estudiantes y profesores pueden ver y responder a los comentarios, fomentando la colaboración y la comunicación.

**CASO No. 5 Comentar un recurso h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-5 | |
| **Nombre** | Comentar un recurso h5p | |
| **Actores** | Estudiantes | |
| **Objetivo** | Agregar comentarios a una actividad h5p asignada. | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de actividades H5P asignadas a estudiantes. | |
| **Flujo Normal** | Estudiante | Sistema |
| Estudiante interactúa con la actividad H5P asignada. |  |
| Agrega comentarios. |  |
|  |  | Guarda el comentario en la base de datos asociado a la actividad. |
|  |  | Sistema permite que otros estudiantes y profesores puedan ver y responder a los comentarios. |
| **Flujo alternativo 1** | Estudiante cancela la adición de comentarios. |  |
|  | Sistema no almacena el comentario. |
|  |  |
| **Flujo alternativo 2** | Comentario no cumple con requisitos. |  |
|  | Sistema muestra mensaje de error. |
| **Post-condiciones** | Comentario se asocia con la actividad H5P y queda disponible para visualización y respuesta. |  |
| **Exepciones** |  |  |
|  |  |

**f) Descargar un Archivo H5P:**

**Actores**:

Usuario Profesor y/o Estudiantes

**Descripción:**

1. Seleccionar Archivo H5P: El usuario navega a la sección de gestión de actividades y selecciona el archivo H5P que desea descargar.
2. Iniciar Descarga: El sistema proporciona una opción para iniciar la descarga del archivo H5P.
3. Descargar Archivo: El usuario confirma la descarga, y el sistema proporciona un archivo H5P descargable en el dispositivo local.
4. Confirmación: El sistema notifica al usuario sobre la descarga exitosa.

**CASO No. 6 Descargar un archivo h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-6 | |
| **Nombre** | Descargar un archivo h5p | |
| **Actores** | Docente y/o Estudiantes | |
| **Objetivo** | Descargar un archivo H5P existente | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de archivos H5P en la sección de gestión de actividades | |
| **Flujo Normal** | Estudiante | Sistema |
| Usuario navega a la sección de gestión de actividades. |  |
| Selecciona archivo H5P da clic en “descargar” |  |
|  |  | Sistema proporciona opción para iniciar la descarga. |
|  | Usuario confirma la descarga. |  |
|  |  | Sistema proporciona archivo H5P descargable. Notifica el éxito. |
| **Flujo alternativo 1** | Usuario cancela la descarga. |  |
|  | El sistema cancela la operación y notifica. |
|  |  |
| **Flujo alternativo 2** | Usuario selecciona archivo no existente. |  |
|  | Sistema muestra mensaje de error. |
| **Post-condiciones** | Usuario descarga el archivo H5P en su dispositivo local. |  |
| **Exepciones** |  |  |
|  |  |

**g) Leer Xapi de H5P:**

**Actores:**

Usuario Profesor

**Descripción:**

1. Seleccionar Actividad H5P: El usuario navega a la lista de actividades H5P y selecciona la actividad de interés.
2. Acceder a Datos xAPI: El sistema proporciona una interfaz que permite al usuario acceder y leer los datos xAPI generados por la actividad H5P.
3. Análisis de Datos: El usuario revisa los datos xAPI para comprender las interacciones del estudiante, incluyendo respuestas, tiempos y eventos específicos.
4. Cierre de la Interfaz: Después de analizar los datos, el usuario puede cerrar la interfaz de lectura xAPI.

**CASO No. 7 Leer Xapi de h5p**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-7 | |
| **Nombre** | Leer Xapi de H5p | |
| **Actores** | Docente | |
| **Objetivo** | Acceder y leer los datos xAPI de interacciones de una actividad H5P | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de actividades H5P con datos xAPI generados. | |
| **Flujo Normal** | Estudiante | Sistema |
| Usuario navega a la lista de actividades H5P. |  |
| Selecciona actividad de interés. |  |
|  |  | Sistema proporciona interfaz para acceder a datos xAPI. |
|  | Usuario revisa datos para analizar interacciones del estudiante. |  |
|  | Cierra la interfaz después del análisis. |  |
|  |  | Sistema lo redirige a la ventana de la listas de actividades |
| **Flujo alternativo 1** | Usuario cierra la interfaz sin analizar los datos. |  |
|  | Sistema registra la acción. |
|  |  |
| **Flujo alternativo 2** | Usuario selecciona actividad sin datos xAPI. |  |
|  | Sistema muestra mensaje informativo. |
| **Post-condiciones** | Usuario revisa los datos xAPI de la actividad seleccionada. |  |
| **Exepciones** |  |  |
|  |  |

**h) Realizar Seguimientos de Interacciones xAPI:**

**Actores**:

Usuario Administrador o Profesor

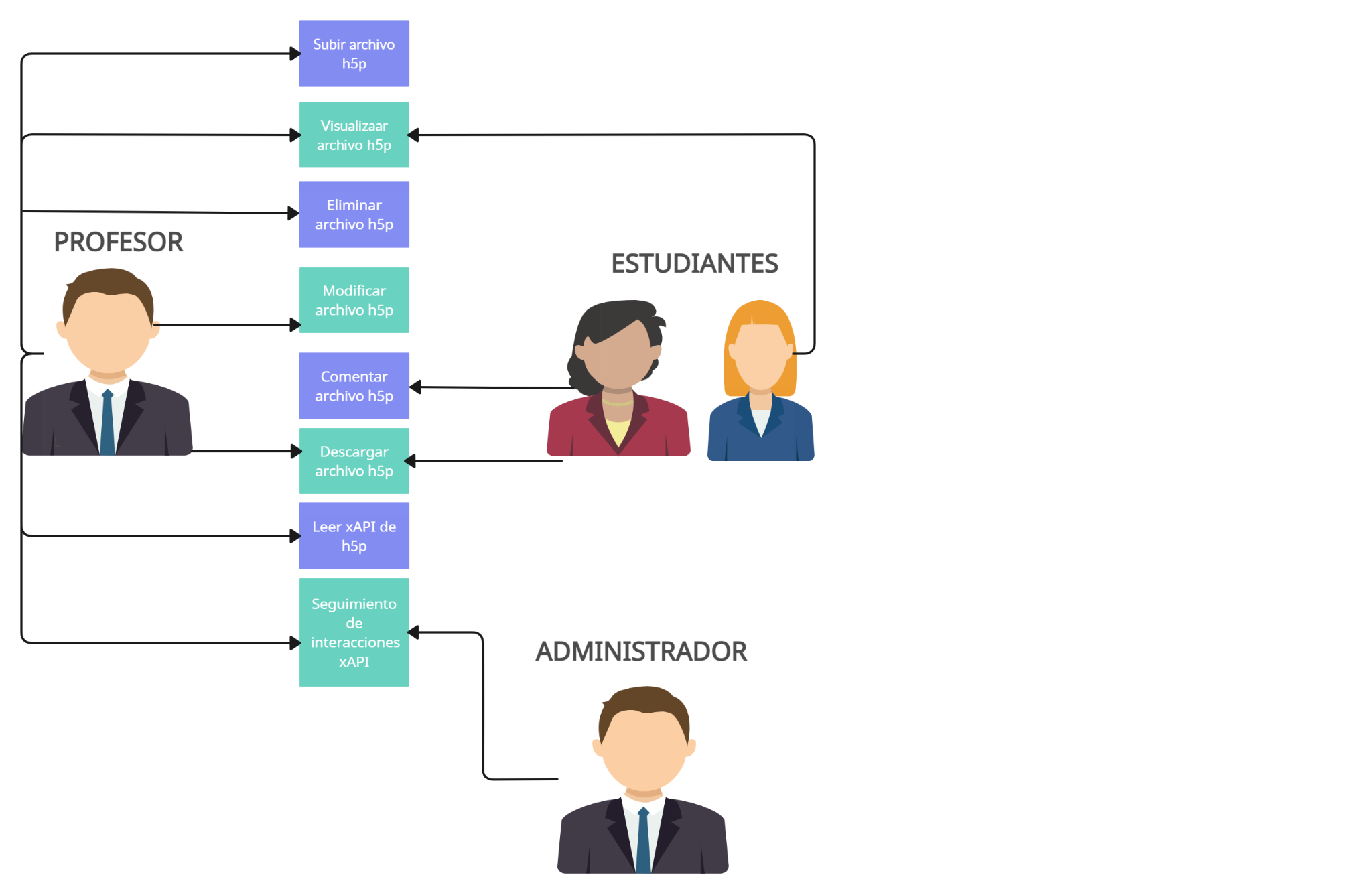
**Descripción:**

1. Acceder a Herramienta de Seguimiento: El sistema proporciona una herramienta o interfaz que permite al usuario realizar un seguimiento continuo de las interacciones xAPI generadas por las actividades H5P.
2. Visualización de Datos: El usuario puede visualizar gráficos, estadísticas y resúmenes basados en los datos xAPI para analizar patrones y tendencias.
3. Tomar Decisiones Informadas: La información recopilada ayuda al usuario a tomar decisiones informadas sobre la efectividad del contenido educativo y ajustar estrategias pedagógicas según sea necesario.

**CASO No. 8 Seguimiento de interacciones con Xapi**

|  |  |  |
| --- | --- | --- |
| **ID:** | CU-8 | |
| **Nombre** | Realizar Seguimientos de Interacciones xAPI | |
| **Actores** | Docente y/o Administrador | |
| **Objetivo** | Realizar seguimientos continuos de interacciones xAPI en actividades H5P | |
| Urgencia | 5 | |
| Esfuerzo | 4 | |
| **Pre-condiciones** | * Existencia de herramienta de seguimiento de interacciones xAPI. | |
| **Flujo Normal** | Estudiante | Sistema |
| Usuario accede a la herramienta de seguimiento. |  |
|  | Sistema muestra visualizaciones de datos xAPI (gráficos, estadísticas). |
|  | Usuario analiza patrones y tendencias. |  |
|  | Toma decisiones informadas basadas en la información recopilada. |  |
|  | Cierra la interfaz después del análisis. |  |
|  |  | Sistema lo redirige a la ventana de la listas de actividades |
| **Flujo alternativo 1** | Usuario no encuentra datos xAPI disponibles. |  |
|  | Sistema muestra mensaje informativo. |
|  |  |
| **Flujo alternativo 2** | Usuario cierra la herramienta sin realizar análisis. |  |
|  | Sistema registra la acción. |
| **Post-condiciones** | Usuario realiza seguimiento y toma decisiones informadas. |  |
| **Exepciones** |  |  |
|  |  |

Diagramas de Flujo de Casos de Uso



Prioridad de Requisitos

**Requisitos No Funcionales:** Los requisitos no funcionales son aspectos del sistema que no están relacionados directamente con las funciones específicas que realiza el sistema, sino con atributos que son cruciales para su operación y éxito general. Estos requisitos a menudo se centran en características como el rendimiento, la confiabilidad, la usabilidad y la seguridad. Aquí hay algunas categorías comunes de requisitos no funcionales:

Requisitos de Desempeño:

* Tiempo de Respuesta: El módulo debe tener un tiempo de respuesta promedio inferior a 2 segundos para todas las transacciones del usuario.
* Capacidad Concurrente: El módulo debe admitir al menos 500 usuarios concurrentes sin una degradación significativa del rendimiento.
* Velocidad de Procesamiento: El módulo debe procesar al menos 500 transacciones por minuto.

Requisitos de Seguridad

* Autenticación de Dos Factores: Todos los usuarios deben autenticarse mediante un segundo factor, como un código de verificación enviado a su teléfono móvil.
* Cifrado de Datos: Todas las comunicaciones entre el cliente y el servidor deben estar cifradas mediante el protocolo HTTPS.
* Auditoría de Acceso: El módulo debe mantener registros de auditoría detallados que registren cada acceso al sistema, incluida la información del usuario, la fecha y la hora.

Requisitos de Usabilidad

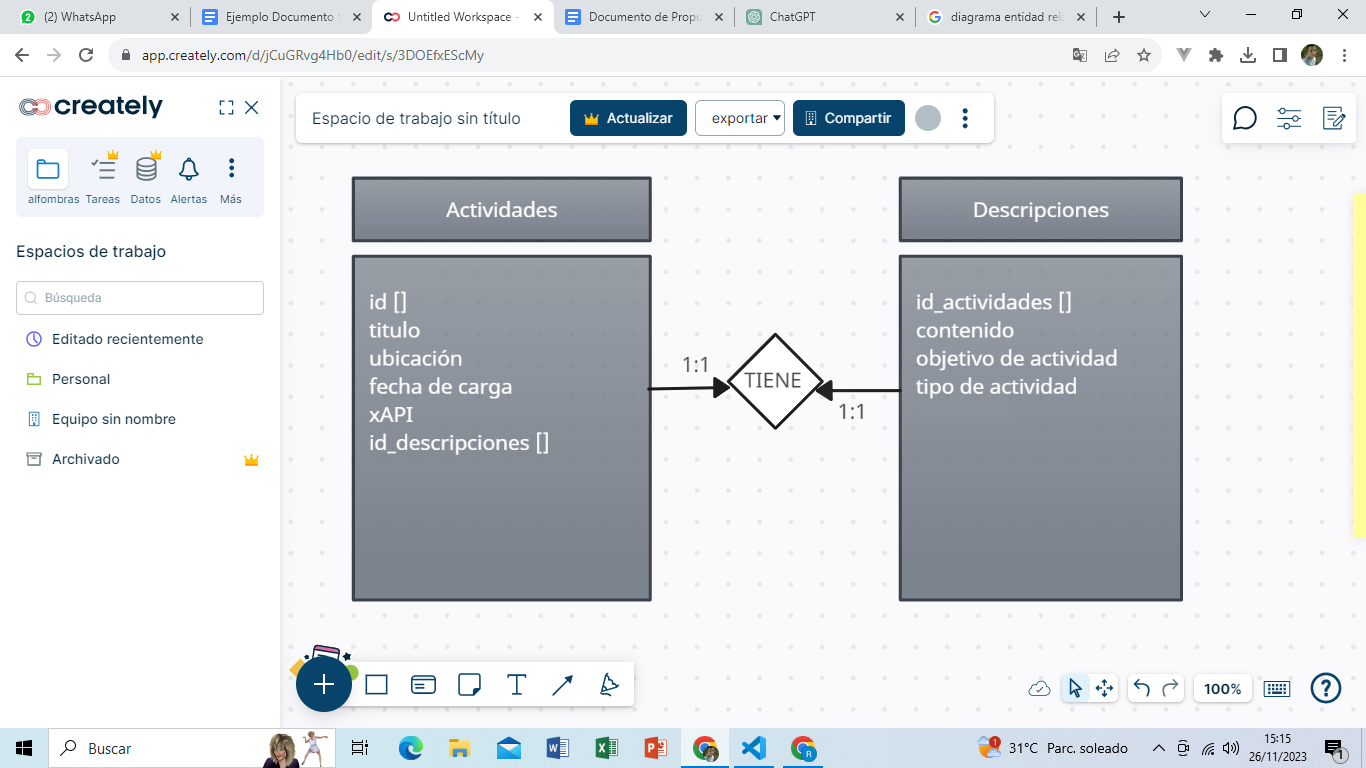
* Navegación Intuitiva: Los usuarios deben poder completar una tarea común en menos de 5 clics.
* Soporte para Accesibilidad: El módulo debe cumplir con las pautas de accesibilidad WCAG 2.0 para garantizar el acceso y uso por parte de personas con discapacidades.
* Feedback del Usuario: El módulo debe proporcionar retroalimentación clara y mensajes de error comprensibles para los usuarios.

Requisitos de Escalabilidad

* Adaptación a Cargas Pico: El módulo debe ser capaz de manejar un aumento repentino del 50% en la carga de trabajo sin afectar el rendimiento.
* Adición de Usuarios: La capacidad de agregar 500 nuevos usuarios al sistema en un período de un mes sin pérdida significativa de rendimiento.
* Escalabilidad Horizontal: El módulo debe admitir la adición de servidores para escalabilidad horizontal cuando la carga aumente.

Modelado E/R:

Diagrama de Entidad-Relación



Descripción de Entidades y Relaciones

***Entidades:***

Actividades (principal): Almacena las actividades subidas por los docentes.

Atributos: ID (identificador único) - título (nombre de la actividad) - ubicación (lugar donde se guarda) - fecha de carga (dia,mes y año en que se sube la actividad) - xAPI (comparte experiencia de aprendizaje) - ID descripciones (información de la descripción de la actividad).

Descripciones: Almacena la información general de las descripciones de las actividades subidas por el usuario profesor.

Atributos: ID actividades (relación directa con las actividades) - contenido (se muestra el contenido completo sobre la actividad) - objetivo de actividad (describe textualmente el objetivo de la actividad) - tipo de actividad (define el tipo de actividad como mapas mentales, sopa de letras, etc.)

***Relaciones:***

“actividades” se relaciona con “descripciones” para guardar y mostrar información sobre las descripciones de las actividades que sube el usuario.

Reglas de Integridad

Clave Primaria y Clave Foránea:

* La clave primaria en la colección "Actividades" (por ejemplo, id\_actividad) debería ser referenciada como clave foránea en la colección "Descripciones" (por ejemplo, id\_actividad en "Descripciones"). Esto establece la relación entre las dos colecciones, asegurando que cada descripción esté vinculada a una actividad existente.
* Restricción de Clave Foránea (Foreign Key Constraint):Debería existir una restricción de clave foránea en la base de datos que garantice que no se pueda agregar una descripción con un id\_actividad que no exista en la colección "Actividades". Esto evita referencias a actividades inexistentes.

Colecciones NoSQL - actividades interactivas

**actividades: {**

id: object

título: string

ubicación: string

fecha de carga: data

xAPI: object

id\_descripciones: object

}

**descripciones: {**

id\_ actividades: object

contenido: string

objetivo de actividad: string

tipo de actividad: string

**}**

NIVEL 2: Introducción a la Persistencia de Datos

1. Diseño de la Arquitectura de Backend
2. Elección de una Base de Datos (SQL o NoSQL)
3. Diseño de Tablas y Modelado de Datos
4. Implementación del Backend (p. ej. usando Node.js, Python, Java)
5. Conexión a la Base de Datos y CRUD Operations
6. Nivel 2: Persistencia de Datos con Backend

Etapa 2: Persistencia de Datos con Backend

INTRODUCCIÓN

Propósito de la Etapa:

La persistencia de datos es un concepto fundamental en la informática que se refiere a la capacidad de un sistema o programa para almacenar información de manera permanente, de modo que los datos se conservan incluso después de que el sistema se apague o se reinicie. Esta funcionalidad es esencial en una amplia variedad de aplicaciones, desde bases de datos empresariales hasta aplicaciones de usuario, ya que garantiza la disponibilidad y la integridad de la información a lo largo del tiempo. La persistencia de datos permite mantener registros, configuraciones, preferencias de usuario y más, lo que facilita la recuperación de datos, el análisis histórico y la personalización de aplicaciones. El propósito de la persistencia de datos es mantener la integridad y disponibilidad de la información a lo largo del tiempo a través de diferentes sesiones.

Alcance de la Etapa:

En esta etapa se espera seleccionar, diseñar e implementar un sistema de bases de datos No relacionales, es decir, una base de datos NoSQL que cumpla con los requisitos de rendimiento, para esto se requiere un alto rendimiento en términos de velocidad y capacidad de respuesta, puede ser necesario realizar ajustes y optimizaciones específicas en la persistencia de datos, lo que aumentará el alcance del proyecto.

Definiciones y acrónimos

**Bases de datos NoSQL**: son sistemas de gestión de bases de datos diseñados para el almacenamiento, recuperación y gestión de datos que no siguen el modelo relacional tradicional utilizado en las bases de datos SQL. A diferencia de las bases de datos SQL, que se basan en tablas y relaciones fijas, las bases de datos NoSQL utilizan estructuras de datos flexibles para manejar información no estructurada o semiestructurada.

**Backend**: Es la capa del sistema que es No visible para el usuario final y que se encarga de procesar datos, gestionar la lógica empresarial, interactuar con la base de datos y enviar información al frontend (la parte que se ve y con la que interactúa el usuario).

**Lenguaje de programación:** Un lenguaje de programación es un conjunto de reglas y comandos que permiten a un programador escribir instrucciones que una computadora puede entender y ejecutar. Estas instrucciones, llamadas código fuente, son escritas en un lenguaje específico que sigue una sintaxis y semántica definida. Los lenguajes de programación se utilizan para crear software y aplicaciones informáticas, controlar hardware, automatizar tareas y resolver problemas computacionales.

**Endpoints**: son puntos de acceso que permiten que los clientes realicen solicitudes a través de una API para realizar diversas operaciones, como recuperar datos, enviar datos, actualizar información, eliminar registros, y más. Cada punto final generalmente está asociado con una operación o función específica de la API.

**API**: (Interfaz de Programación de Aplicaciones) son conjuntos de reglas y protocolos que permiten que diferentes componentes de software se comuniquen entre sí. Las API definen los métodos y estructuras de datos que los desarrolladores pueden utilizar para interactuar con un sistema o servicio en particular. Estas interfaces permiten que las aplicaciones se integren, compartan datos y funcionen de manera conjunta de manera eficiente.

DISEÑO DE LA ARQUITECTURA DEL BACKEND

Descripción de la Arquitectura Propuesta:

Componentes del Backend

Diagramas de Arquitectura

ELECCIÓN DE LA BASE DE DATOS

Evaluación de Opciones (SQL o NoSQL):

Bases de datos NoSQL:

**Modelo de datos flexible**: Las bases de datos NoSQL permiten el almacenamiento de datos en una variedad de modelos, incluyendo documentos, columnas, gráficos y bases de datos clave-valor. Esto permite que las bases de datos NoSQL se adapten a diferentes tipos de datos y necesidades.

**Esquema dinámico:** A diferencia de las bases de datos SQL, las bases de datos NoSQL no requieren un esquema fijo y predefinido. Los datos pueden variar en estructura y no es necesario seguir un esquema estricto.

**Escalabilidad horizontal:** Muchas bases de datos NoSQL están diseñadas para escalar horizontalmente, lo que significa que se pueden distribuir en Múltiples servidores o nodos para manejar grandes volúmenes de datos y tráfico de manera eficiente.

**Alto rendimiento**: Las bases de datos NoSQL suelen ofrecer un alto rendimiento en términos de velocidad de lectura y escritura, lo que las hace adecuadas para aplicaciones de alta demanda, como sitios web y aplicaciones móviles.

**Uso en aplicaciones web y móviles:** Las bases de datos NoSQL son populares en aplicaciones web y móviles debido a su capacidad de manejar datos en tiempo real y escalarse fácilmente para acomodar un crecimiento repentino en el tráfico.

Bases de datos SQL:

**Modelo Relacional:** Las bases de datos SQL utilizan un modelo relacional en el que los datos se organizan en tablas con filas y columnas. Esto permite la representación de relaciones entre diferentes conjuntos de datos.

**Esquema fijo:** Las bases de datos SQL requieren un esquema de datos predefinido en el que se especifica la estructura de las tablas, incluyendo los tipos de datos y las restricciones de integridad.

**Consultas SQL:** Los datos se recuperan y se manipulan utilizando el lenguaje de consulta SQL, que permite realizar consultas complejas para filtrar, ordenar y agrupar datos.

**Transacciones ACID:** Las bases de datos SQL siguen el conjunto de propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) para garantizar la integridad y confiabilidad de los datos.

**Integridad de datos:** Las bases de datos SQL permiten la definición de restricciones de integridad, como claves primarias y extranjeras, para garantizar la coherencia y la calidad de los datos.

**Soporte para múltiples usuarios:** Las bases de datos SQL están diseñadas para admitir múltiples usuarios concurrentes, lo que permite a varios usuarios acceder y actualizar los datos al mismo tiempo.

**Seguridad avanzada:** Ofrecen mecanismos de seguridad, como autenticación y autorización, para proteger los datos y limitar el acceso a usuarios no autorizados.

**Escalabilidad vertical:** La escalabilidad en las bases de datos SQL se logra generalmente aumentando los recursos de hardware, como la CPU o la memoria, en el servidor de la base de datos.

**Aplicaciones empresariales:** Las bases de datos SQL son comunes en aplicaciones empresariales, como sistemas de gestión de clientes (CRM), sistemas de planificación de recursos empresariales (ERP) y sistemas de soporte a la decisión.

**Consistencia y relaciones:** Permiten establecer relaciones entre tablas mediante claves foráneas y garantizan la consistencia de los datos según estas relaciones.

Justificación de la Elección - NoSQL

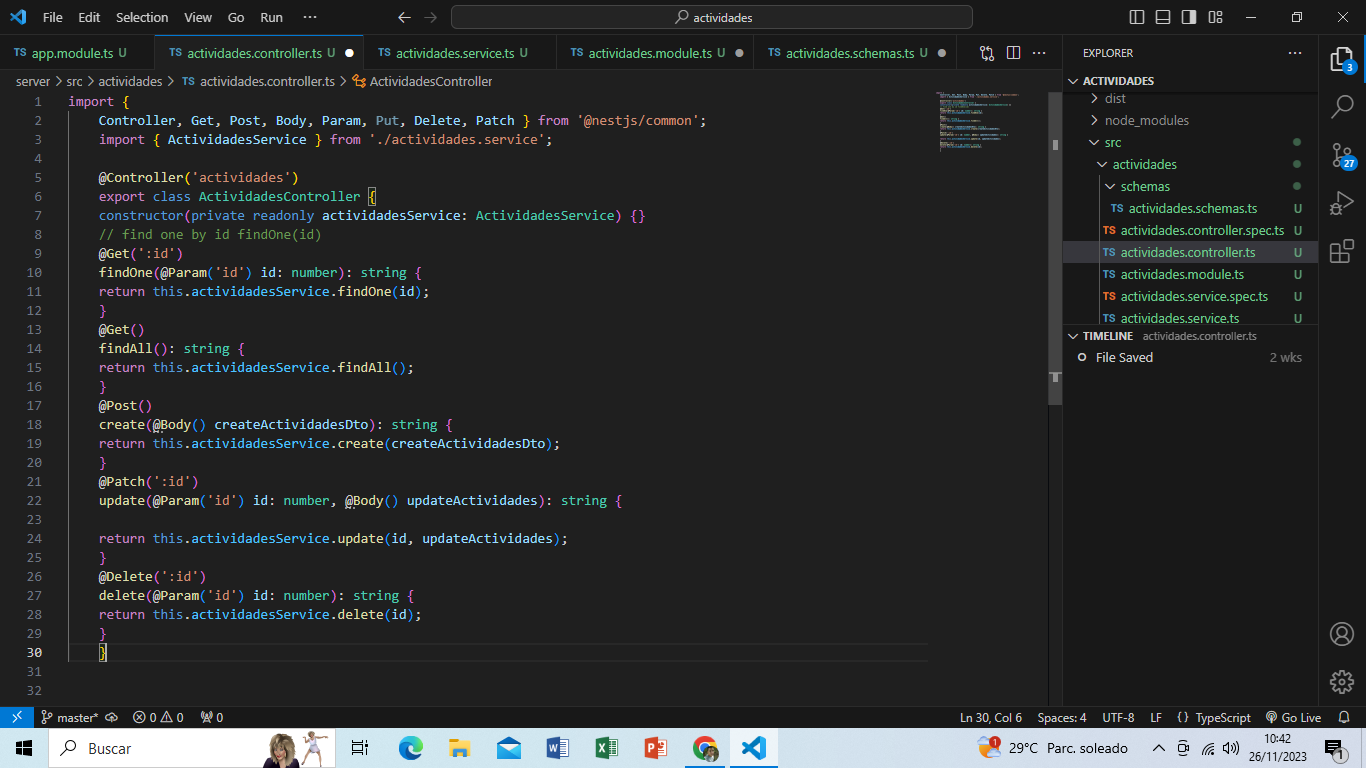
La elección de utilizar una base de datos NoSQL para el módulo "Actividades Interactivas" en una página web educativa “Creavi” se justifica por su flexibilidad en el esquema de datos, permitiendo adaptarse a la diversidad de estructuras y requisitos que requiera Creavi. Además, la escalabilidad horizontal de las bases de datos NoSQL es esencial para manejar el crecimiento potencial en el número de usuarios y actividades. También la capacidad para manejar eficientemente datos no relacionales y adaptarse ágilmente a cambios en los requisitos respalda un desarrollo iterativo y evolutivo. La optimización para operaciones de lectura/escritura intensivas y la posibilidad de escalabilidad independiente para lecturas y escrituras son cruciales en un entorno interactivo. Asimismo, la facilidad de integración con tecnologías web modernas facilita la cohesión con otros módulos y la funcionalidad de una página web educativa eficiente y escalable.

Diseño de Esquema de Base de Datos:

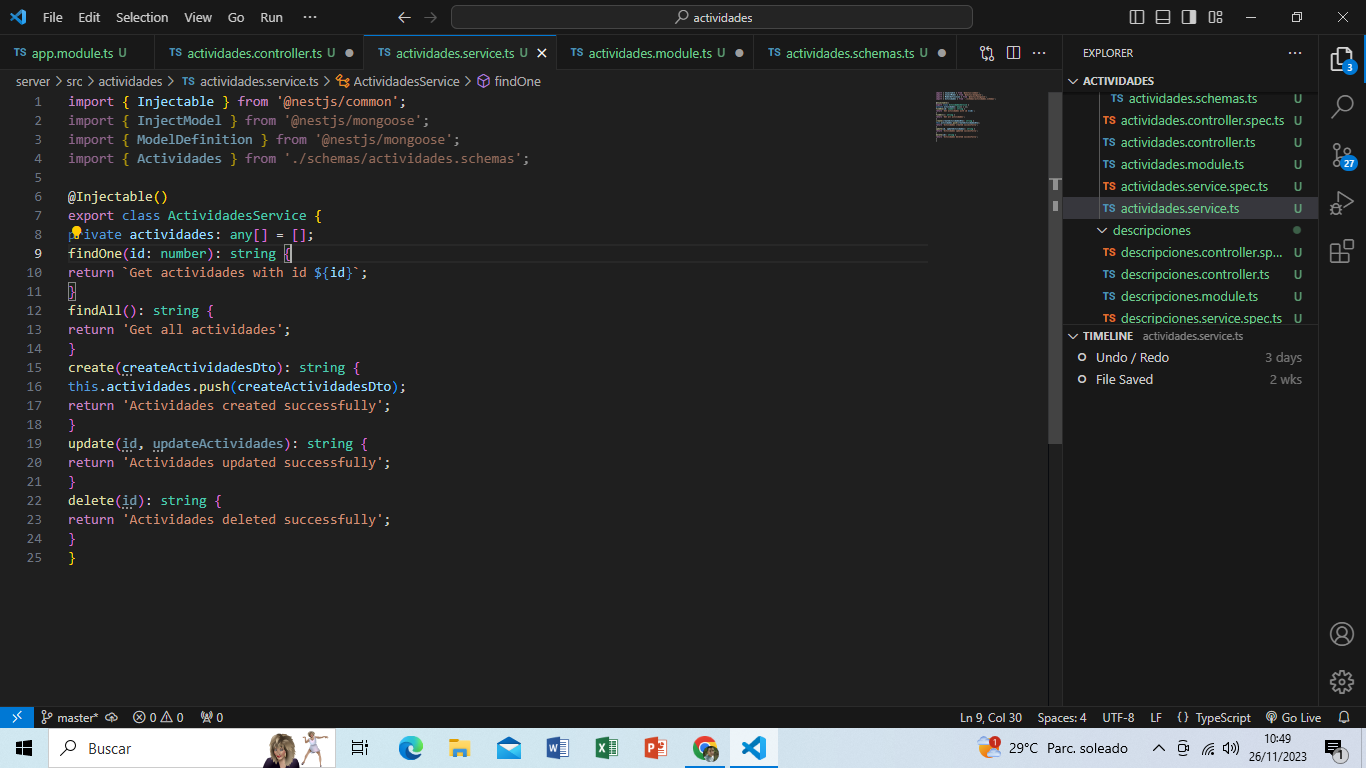
Este módulo tiene 2 esquemas de colecciones

* **esquema de colección actividades**

**actividades.controller.ts**

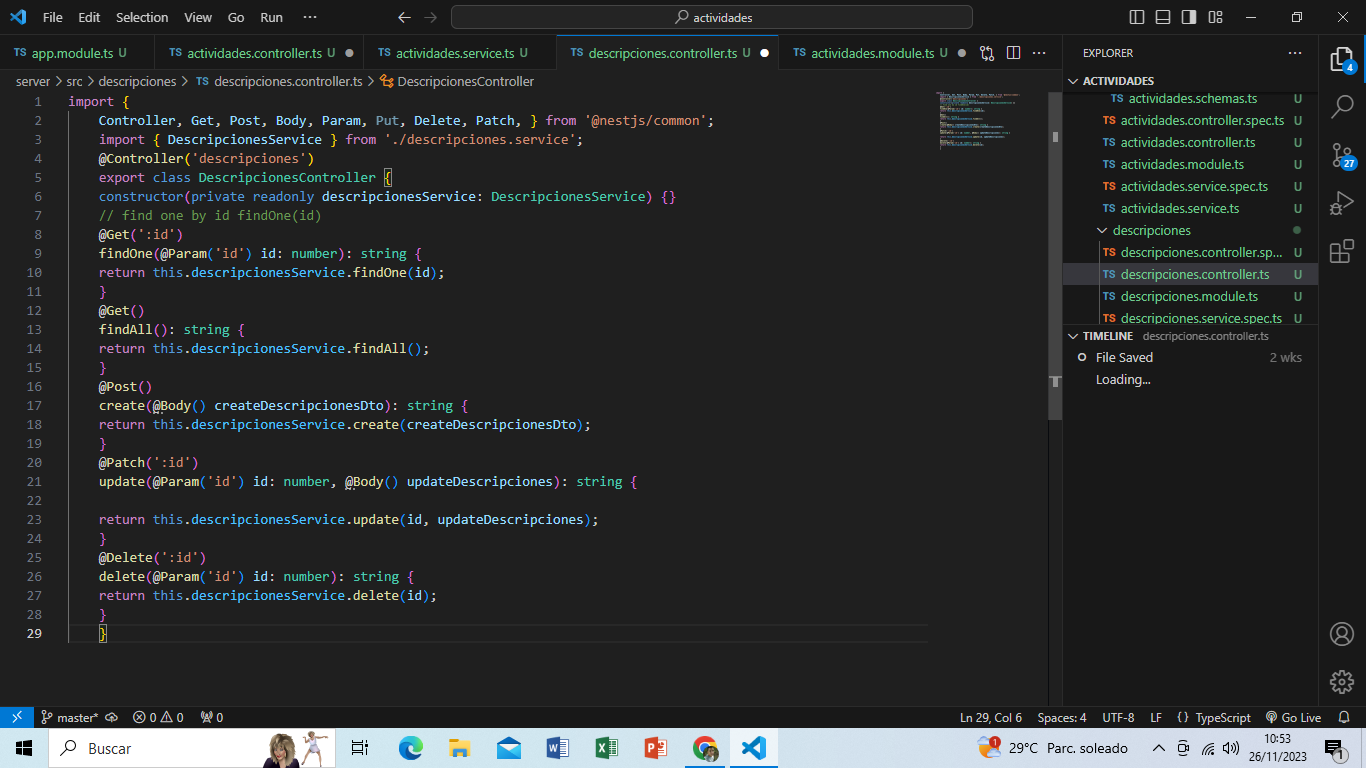
****

**actividades.service.ts**

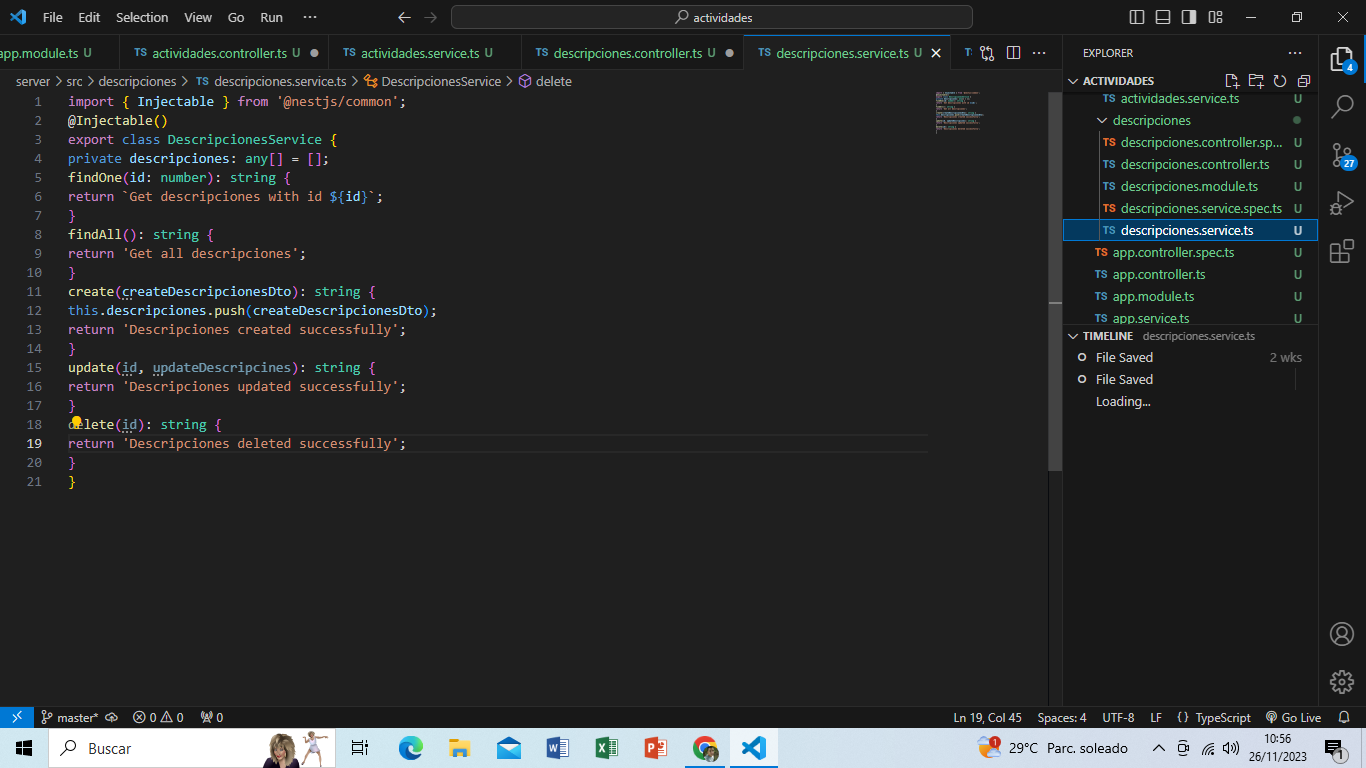
****

**Esquema de colección descripciones**

**descripciones.controller.ts**

****

**descripciones.service.ts**

****

IMPLEMENTACIÓN DEL BACKEND

Elección del Lenguaje de Programación:

Para la implementación del backend relacionado al módulo de actividades interactivas, se desarrollará en los lenguajes de programación como MongoDB, JavaScript y Node.js. Además, se optó por implementar el protocolo HTTP en la construcción de la plataforma mediante el uso de Nest y un servidor MongoDB, ya que presenta diversas ventajas notables. HTTP goza de una amplia aceptación y es un estándar en el ámbito del desarrollo web, lo que asegura la interoperabilidad y una comunicación eficaz entre los clientes y los servidores. Al aprovechar Nest, un framework de Node.js altamente afín al protocolo HTTP, es factible crear con facilidad servicios RESTful o APIs GraphQL, los cuales desempeñan un papel fundamental en la interacción entre el servidor y los clientes. Este enfoque se alinea con las mejores prácticas para establecer una plataforma robusta y eficiente.

A continuación se definen cada uno de estos lenguajes para entrar en contexto con lo que se busca realizar en el módulo de actividades interactivas.

**MongoDB**: Es un sistema de gestión de bases de datos NoSQL (Not Only SQL) de código abierto, altamente escalable y flexible. A diferencia de las bases de datos relacionales tradicionales, que utilizan tablas y esquemas fijos, MongoDB almacena datos en un formato conocido como BSON (Binary JSON) que permite la representación de datos en documentos flexibles y anidados, similares a objetos JSON. Además, MongoDB es una base de datos NoSQL, lo que significa que no utiliza un esquema rígido y permite la flexibilidad en la estructura de los datos. Esto es especialmente útil cuando se manejan datos no estructurados o semiestructurados. También es altamente escalable y está diseñado para funcionar en clústeres de servidores para manejar grandes volúmenes de datos y tráfico. Otro aspecto importante de MongoDB es que puede escalar horizontalmente agregando más servidores al clúster.Documentos y Colecciones: Los datos se organizan en colecciones, que contienen documentos. Cada documento es una unidad de datos individuales que puede variar en estructura y contenido. Esto hace que MongoDB sea adecuado para aplicaciones en las que los datos evolucionan con el tiempo. Por último, MongoDB permite realizar consultas utilizando un lenguaje de consulta similar a JSON, lo que facilita la recuperación y manipulación de datos.

**JavaScript:** JavaScript es un lenguaje de programación de alto nivel que se utiliza principalmente en el desarrollo web para crear interacciones dinámicas y mejorar la funcionalidad de las páginas web. Es un lenguaje de programación versátil y ampliamente utilizado que se ejecuta en el navegador web del usuario, lo que permite a los desarrolladores agregar interactividad, realizar validaciones de formularios, gestionar eventos, manipular el contenido HTML y mucho más.

**Node.js:** es un entorno de tiempo de ejecución de código abierto que permite a los desarrolladores ejecutar JavaScript en el lado del servidor. A diferencia de JavaScript en el navegador, que se utiliza principalmente para la programación del lado del cliente, Node.js se utiliza para la programación del lado del servidor y permite a los desarrolladores crear aplicaciones web y de red altamente escalable y eficiente.

Creación de la Lógica de Negocio:

La lógica de negocio en una base de datos NoSQL (Not Only SQL) se refiere a la parte de un sistema o aplicación que se encarga de definir y gestionar las reglas y procesos específicos relacionados con los datos almacenados en esa base de datos. Esta lógica de negocio se implementa para garantizar que los datos se almacenen, se recuperen y se manipulen de acuerdo con las necesidades y reglas específicas de una aplicación o sistema.

DIAGRAMA DE NEST ES LA LÓGICA DE NEGOCIO

Aspectos que puede incluir la creación de una lógica de negocio:

**Definición de esquema de datos:** Aunque las bases de datos NoSQL permiten flexibilidad en la estructura de los datos, la lógica de negocio puede definir cómo se deben estructurar los datos, qué campos son obligatorios y cómo se relacionan entre sí.

**Validación de datos:** La lógica de negocio puede incluir reglas para validar la integridad y consistencia de los datos, como restricciones de tipos de datos, validaciones de rangos y formatos, y reglas comerciales específicas.

**Lógica de acceso y seguridad:** Se pueden establecer reglas de acceso y seguridad para garantizar que solo usuarios autorizados tengan acceso a ciertos datos y operaciones en la base de datos.

**Reglas de negocio específicas de la aplicación:** La lógica de negocio puede incluir reglas específicas de la aplicación que determinan cómo se deben gestionar los datos en función de la lógica y los procesos comerciales de la aplicación.

**Algoritmos y procesos de análisis de datos:** Si la base de datos NoSQL se utiliza para análisis de datos, la lógica de negocio puede definir algoritmos y procesos para extraer y analizar los datos almacenados.

Desarrollo de Endpoints y APIs

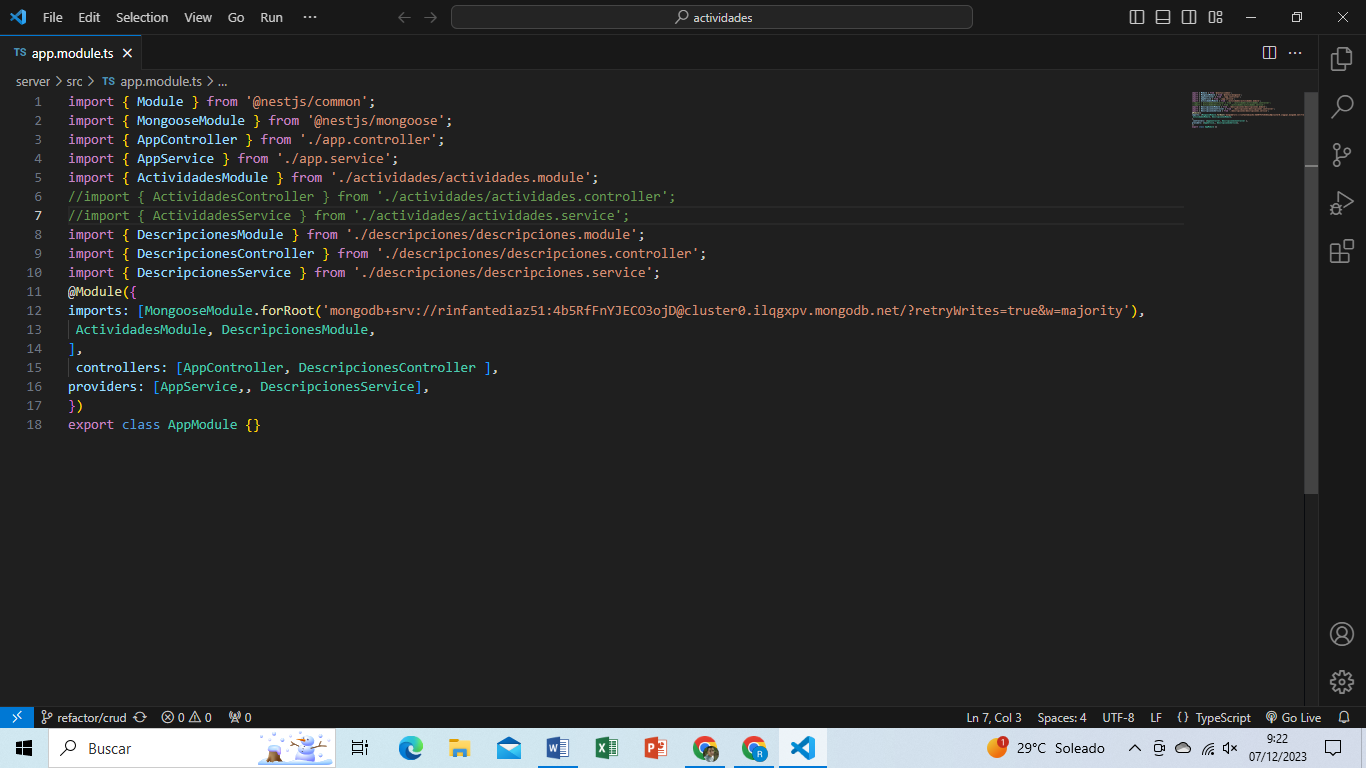
Endpoints son todos los datos verdes y amarrillos que corren en el visual studio code. Y pantallazos de rutas en postman

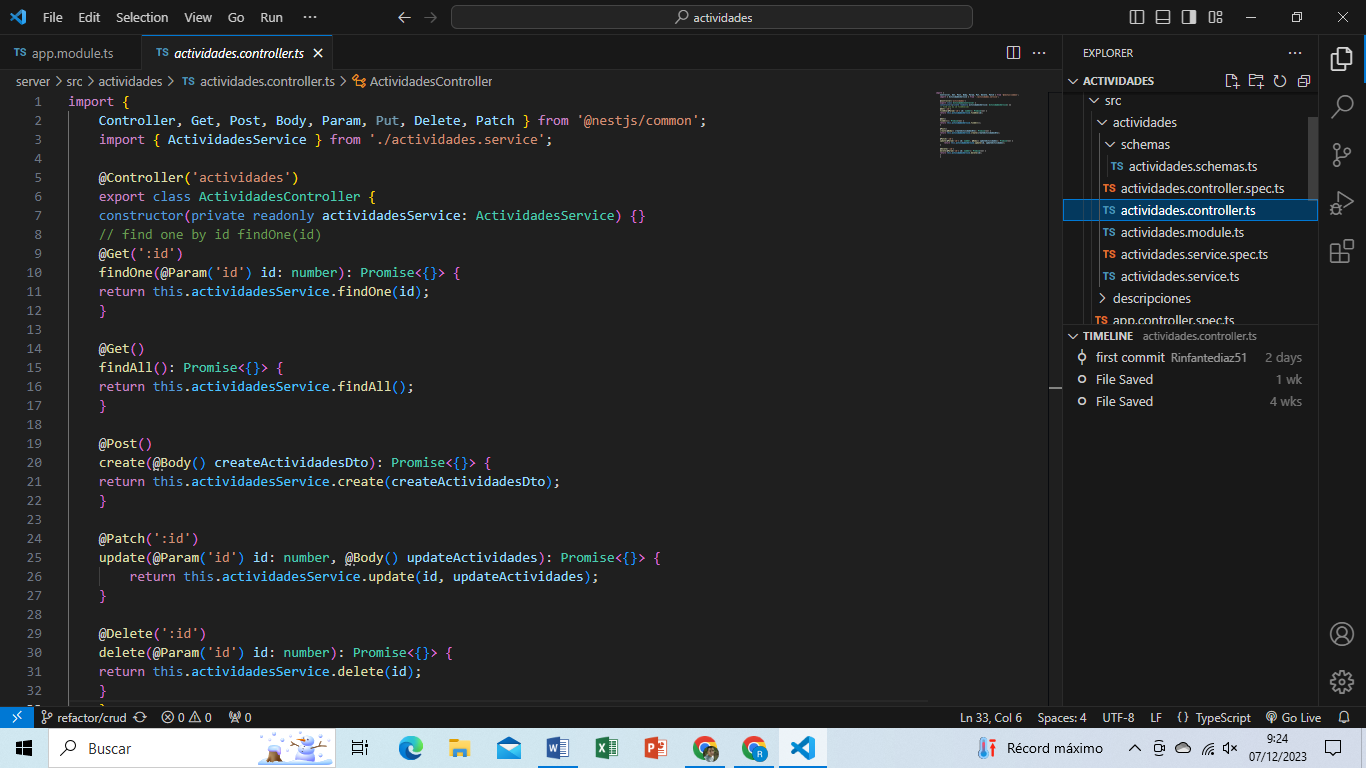
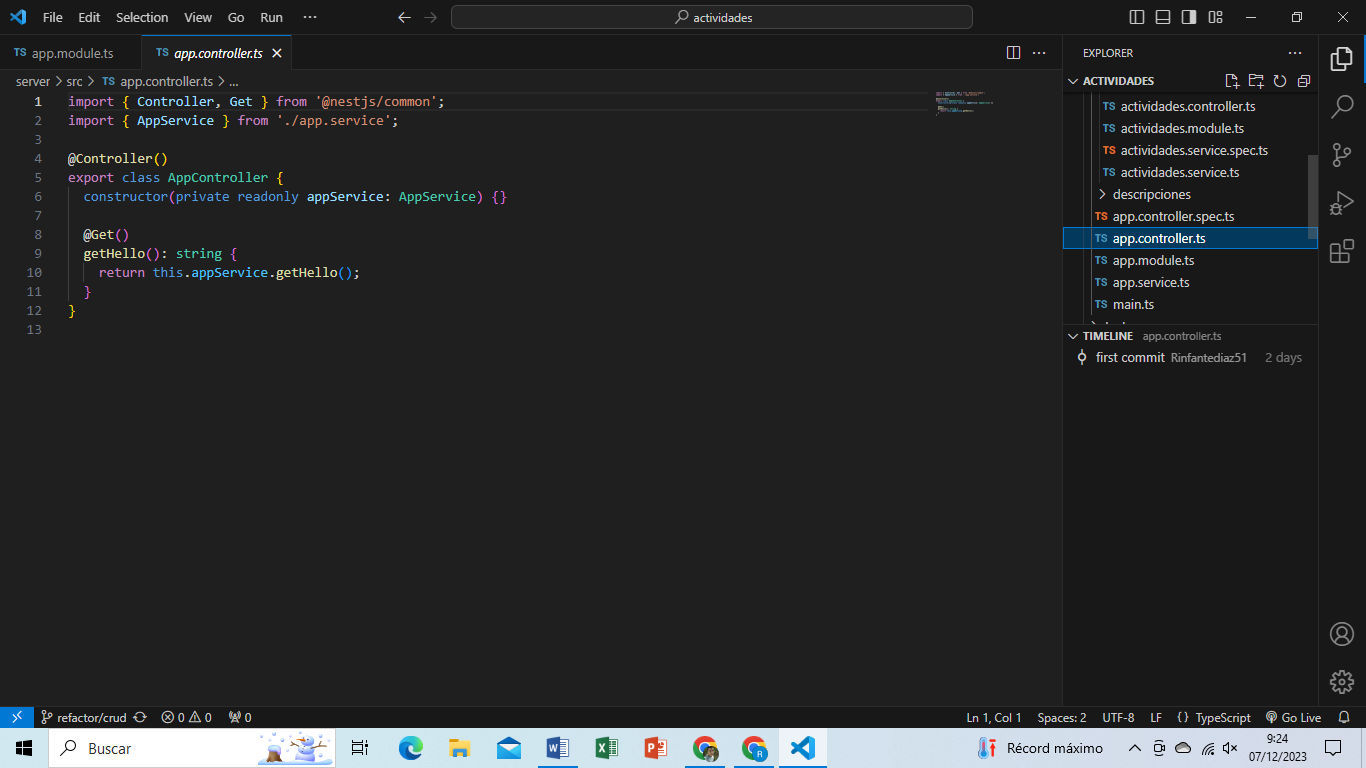
Autenticación y Autorización

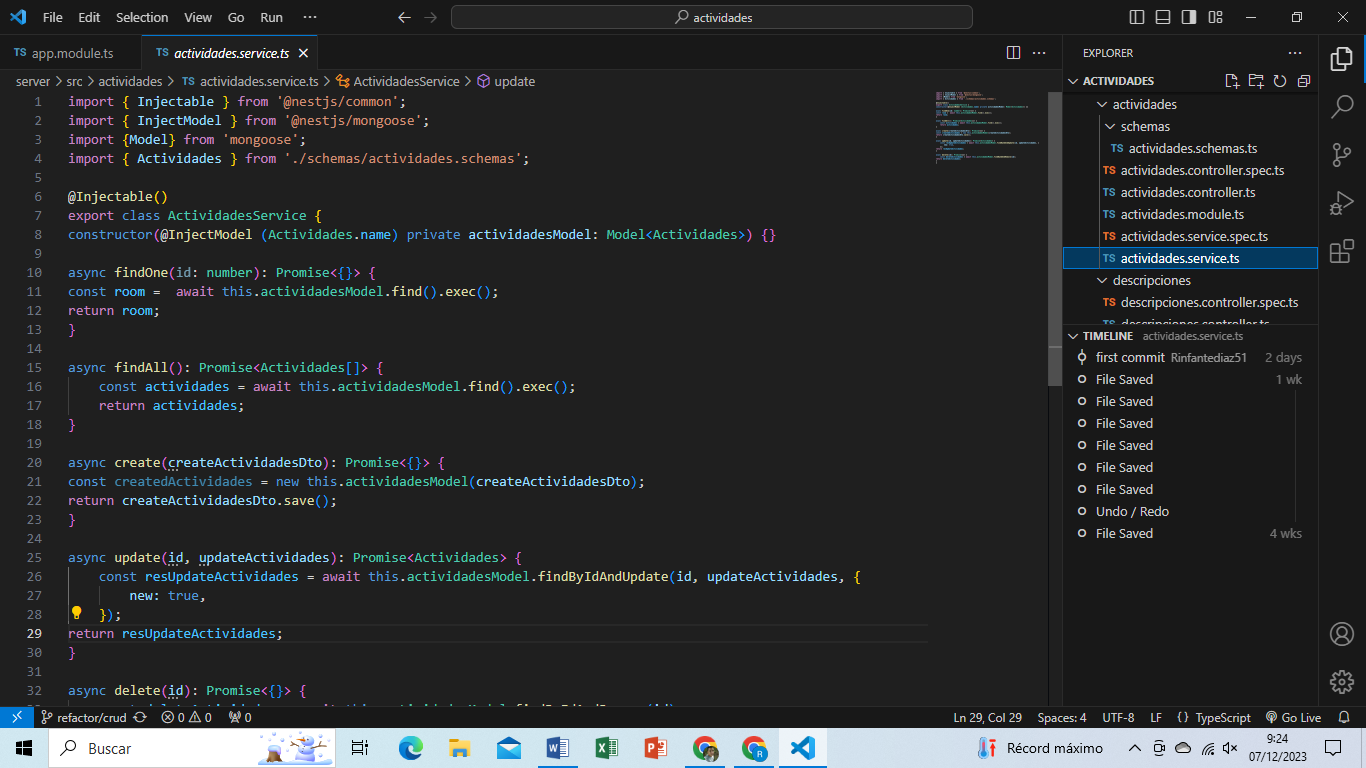
Se gestiona desde el módulo de usuario.

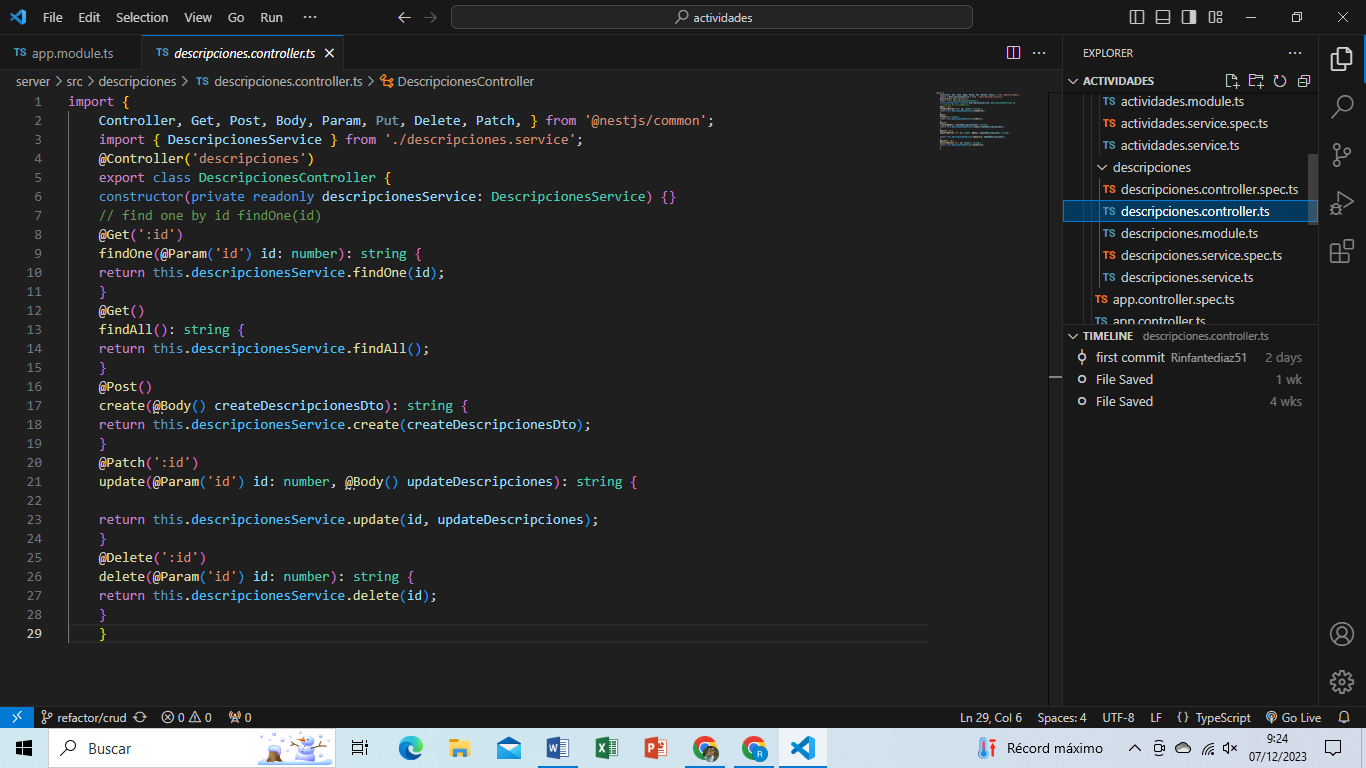
Conexión a la base de datos

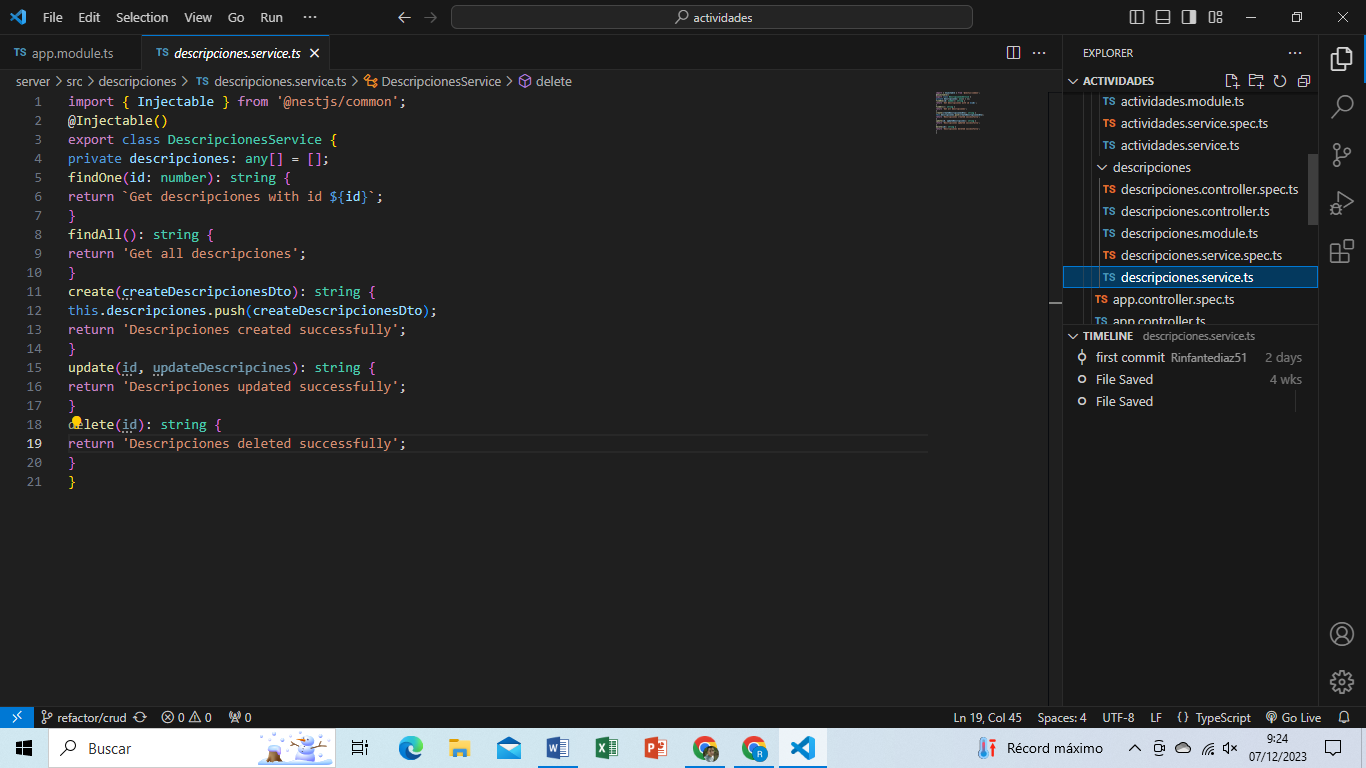
Configuración de la Conexión: archivo app module y se evidencia la conexión en la línea 21



Desarrollo de Operaciones CRUD: ****







Manejo de Transacciones:

Explicar el porqué de los métodos. put, patch, delete, get. Para qué usar cada método en los módulos.

Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Nivel 3: Consumo de Datos y Desarrollo Frontend

Introducción al Desarrollo Frontend

Creación de Interfaz de Usuario (UI) con HTML y CSS

Programación Frontend con JavaScript (JS)

Consumo de Datos desde el Backend

Interacción Usuario-Interfaz (Eventos, Formularios, Validación)

Pruebas y Depuración Frontend

Implementación de la Lógica de Negocio en el Frontend

#### **SPRINTS Nivel 1: Diseño de la Aplicación y Análisis de Requisitos**

| **Ítem** | **Descripción** |
| --- | --- |
| 1 | Investigación sobre el dominio del proyecto |
| 2 | Entrevistas con los stakeholders para identificar necesidades |
| 3 | Creación de un documento de especificación de requisitos |
| 4 | Desarrollo de casos de uso y diagramas de casos de uso |
| 5 | Modelado E/R del sistema utilizando una herramienta adecuada |

#### **SPRINTS Nivel 2: Persistencia de Datos con Backend**

| **Ítem** | **Descripción** |
| --- | --- |
| 1 | Diseño de la arquitectura de backend |
| 2 | Elección de una base de datos y diseño de tablas |
| 3 | Implementación del servidor backend utilizando un lenguaje específico |
| 4 | Conexión a la base de datos y creación de operaciones CRUD |
| 5 | Pruebas del backend y manejo de errores |

#### **SPTINTS Nivel 3: Consumo de Datos y Desarrollo Frontend**

| **Ítem** | **Descripción** |
| --- | --- |
| 1 | Creación de la interfaz de usuario (UI) utilizando HTML y CSS |
| 2 | Desarrollo de la lógica del frontend con JavaScript (JS) |
| 3 | Implementación de la comunicación con el backend |
| 4 | Interacción usuario-interfaz (manejo de eventos, formularios, validación) |
| 5 | Pruebas y depuración del frontend |
| 6 | Implementación de la lógica de negocio en el frontend |

### Etapa 3: Consumo de Datos y Desarrollo Frontend

1. Introducción
   * Propósito de la Etapa
   * Alcance de la Etapa
   * Definiciones y Acrónimos
2. Creación de la Interfaz de Usuario (UI)
   * Diseño de la Interfaz de Usuario (UI) con HTML y CSS
   * Consideraciones de Usabilidad
   * Maquetación Responsiva
3. Programación Frontend con JavaScript (JS)
   * Desarrollo de la Lógica del Frontend
   * Manejo de Eventos y Comportamientos Dinámicos
   * Uso de Bibliotecas y Frameworks (si aplicable)
4. Consumo de Datos desde el Backend
   * Configuración de Conexiones al Backend
   * Obtención y Presentación de Datos
   * Actualización en Tiempo Real (si aplicable)
5. Interacción Usuario-Interfaz
   * Manejo de Formularios y Validación de Datos
   * Implementación de Funcionalidades Interactivas
   * Mejoras en la Experiencia del Usuario
6. Pruebas y Depuración del Frontend
   * Diseño de Casos de Prueba de Frontend
   * Pruebas de Usabilidad
   * Depuración de Errores y Optimización del Código
7. Implementación de la Lógica de Negocio en el Frontend
   * Migración de la Lógica de Negocio desde el Backend (si necesario)
   * Validación de Datos y Reglas de Negocio en el Frontend
8. Integración con el Backend
   * Verificación de la Comunicación Efectiva con el Backend
   * Pruebas de Integración Frontend-Backend