

## Documento de Propuesta de Diseño de Software II

### Creación Módulo de Formulario

#### Integrantes:

Andrés Felipe Díaz Vergara

Jhair Andrés Teherán Florez

María Alma López Mestra

Gustavo Rodríguez Fuentes



## Descripción del software

Se desea crear un componente compatible para la plataforma CREAVI que permita a través de una pizarra digital compartir elementos multimediales entre los usuarios inscritos en un espacio de trabajo (clase, conferencia, taller, etc.) Los elementos para compartir podrán ser videos, diapositivas, imágenes, documentos, audios, segmentos de código, etc. El anfitrión podrá resaltar elementos en pantalla o apuntarlos y los interlocutores podrán interactuar con ellos solo si el anfitrión u otro usuario designado lo permite.

El componente traerá elementos por defecto en su instalación, pero permite la escalabilidad a otros componentes o formatos que sean compatibles con la versión desarrollada.

## **Etapas 2: Persistencia de Datos con Backend**

### **Introducción**

En el proceso de desarrollo de un formulario en la plataforma Creavi, la etapa de "Persistencia de Datos con Backend" desempeña un papel crucial en garantizar que los datos recopilados a través del formulario sean gestionados de manera eficiente y segura. Esta etapa implica la implementación de la lógica del servidor que permite la recepción, validación y almacenamiento de los datos enviados por los usuarios.

El objetivo principal de esta etapa es asegurarse de que los datos recolectados en el formulario se almacenen de manera confiable y estén disponibles para su posterior uso y análisis. Para lograrlo, el backend se encarga de tareas clave, como la recepción de datos, la validación, la comunicación con la plataforma Creavi y la implementación de medidas de seguridad.

En este proceso, es esencial la elección de una base de datos NoSQL adecuada, ya que estas bases de datos son flexibles y eficientes en la gestión de datos no estructurados o semiestructurados, comunes en formularios en línea. Además, la seguridad y la privacidad de los datos son prioritarias, por lo que se deben implementar medidas de protección, como cifrado y autenticación.

Esta etapa también se enfoca en la identificación y registro de eventos y errores, lo que facilita la resolución de problemas y el mantenimiento continuo del sistema. Una implementación cuidadosa de la persistencia de datos con backend garantiza la funcionalidad y la confiabilidad de tu formulario en la plataforma Creavi, al tiempo que proporciona a los usuarios y a la institución un acceso seguro a la información recopilada.

## **1. Propósito de la Etapa**

En esta etapa, se establecen los mecanismos necesarios para gestionar la información de manera efectiva, asegurando que los datos sean consistentes, confiables y estén disponibles para su posterior acceso y análisis. Además, se enfoca en la seguridad de los datos, implementando medidas para proteger la información del usuario y prevenir vulnerabilidades. La persistencia de datos con backend es esencial para el funcionamiento exitoso del formulario y para mantener la integridad de la información recopilada.

## **2. Alcance de la Etapa**

Para la realización de este proyecto se realizarán una serie de actividades y responsabilidades esenciales. Aquí está el alcance típico de esta etapa:

**Recepción de Datos:** En esta etapa, se configura el servidor backend para recibir los datos enviados por los usuarios a través del formulario. Esto implica establecer rutas y endpoints de API para manejar las solicitudes de envío de datos desde la plataforma Creavi.

**Validación de Datos:** Los datos enviados por los usuarios deben ser validados para asegurar su integridad y seguridad. Esto implica verificar que los datos cumplan con los requisitos y restricciones definidos en el formulario, incluyendo la validación de formatos, la prevención de inyecciones de datos maliciosos y, si es necesario, la autenticación del usuario.

**Almacenamiento en Base de Datos NoSQL:** Una vez que los datos son validados, se procede a su almacenamiento en una base de datos NoSQL. Esto implica definir la estructura de la base de datos, crear tablas o colecciones, e insertar los datos en un formato que sea compatible con la base de datos NoSQL elegida, como MongoDB o Cassandra.

**Comunicación con Creavi:** El backend debe estar configurado para comunicarse de manera efectiva con la plataforma Creavi. Esto asegura que los datos almacenados estén disponibles para su posterior visualización y análisis en la plataforma.

**Seguridad y Privacidad de Datos:** La seguridad de los datos es una preocupación central en esta etapa. Se deben implementar medidas de seguridad, como cifrado de datos en tránsito y en reposo, autenticación de usuarios y autorización, para proteger la información del usuario y prevenir posibles amenazas.

### 3. Definiciones y Acrónimos

#### Definiciones:

- **Formulario:** Un formulario es una interfaz en línea que permite a los usuarios introducir y enviar datos a través de campos de entrada, casillas de verificación, botones de opción y otros elementos interactivos. Los formularios se utilizan comúnmente para recopilar información de los usuarios.
- **Backend:** El backend es la parte de una aplicación de software que se encarga de la lógica y el procesamiento del servidor. En el contexto de un formulario en Creavi, el backend es responsable de recibir, validar y almacenar los datos del formulario en una base de datos.
- **Base de Datos NoSQL:** Una base de datos NoSQL (Not Only SQL) es un tipo de base de datos que no utiliza un esquema de tabla relacional tradicional. Estas bases de datos son adecuadas para el almacenamiento de datos no estructurados o semiestructurados, comunes en aplicaciones web modernas.

#### Acrónimos:

- **API:** Siglas de "Application Programming Interface" (Interfaz de Programación de Aplicaciones). Una API define cómo las aplicaciones y los componentes de software deben comunicarse entre sí.
- **SGE:** Siglas de "Sistema de Gestión Educativa". Un SGE es un software que se utiliza en instituciones educativas para gestionar procesos como la matriculación, el seguimiento del progreso del estudiante y la administración de recursos educativos.
- **NoSQL:** Sigla de "Not Only SQL". Hace referencia a bases de datos que no siguen la estructura de bases de datos relacionales tradicionales y son más flexibles en cuanto al almacenamiento y recuperación de datos.
- **HTTP:** Siglas de "Hypertext Transfer Protocol" (Protocolo de Transferencia de Hipertexto). Es el protocolo utilizado para la transferencia de datos en la World Wide Web.
- **HTML:** Siglas de "Hypertext Markup Language" (Lenguaje de Marcado de Hipertexto). Es el lenguaje utilizado para crear y formatear documentos en la web, incluyendo la creación de formularios web.

- CSS: Siglas de "Cascading Style Sheets" (Hojas de Estilo en Cascada). Se utiliza para definir el estilo y la apariencia visual de documentos web, incluyendo la presentación de formularios.
- SQL: Siglas de "Structured Query Language" (Lenguaje de Consulta Estructurado). Es un lenguaje de programación utilizado para gestionar bases de datos relacionales. A diferencia de NoSQL, SQL se utiliza en bases de datos relacionales tradicionales.

## **Diseño de la Arquitectura de Backend.**

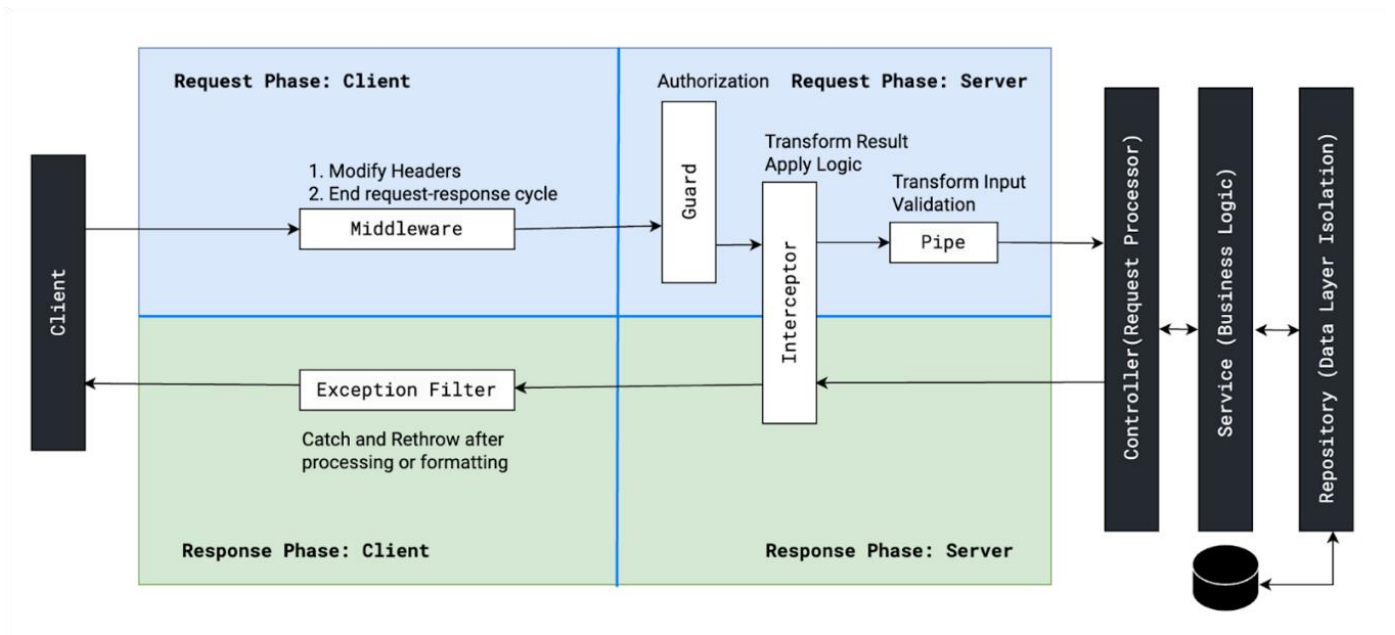
### **4. Descripción de la Arquitectura Propuesta.**

Para el desarrollo de nuestro servidor se tuvo en cuenta la arquitectura Nest.JS, la cual facilita la creación tanto de aplicaciones web como de servidores robustos en un entorno Node.js, centrándose en la capacidad de crecimiento y de mantener el código de manera eficiente, ya que este se basa en una arquitectura modular la cual permite organizar las aplicaciones en módulos, lo que promueve la modularidad y la reutilización del código. Cada módulo puede contener componentes como controladores, servicios y enrutadores, lo que facilita la gestión de la lógica del servidor.

Por otra parte, Nest.JS puede utilizarse creación de una amplia gama de aplicaciones web, incluyendo API REST, aplicaciones en tiempo real, sistemas de gestión de contenido y más.

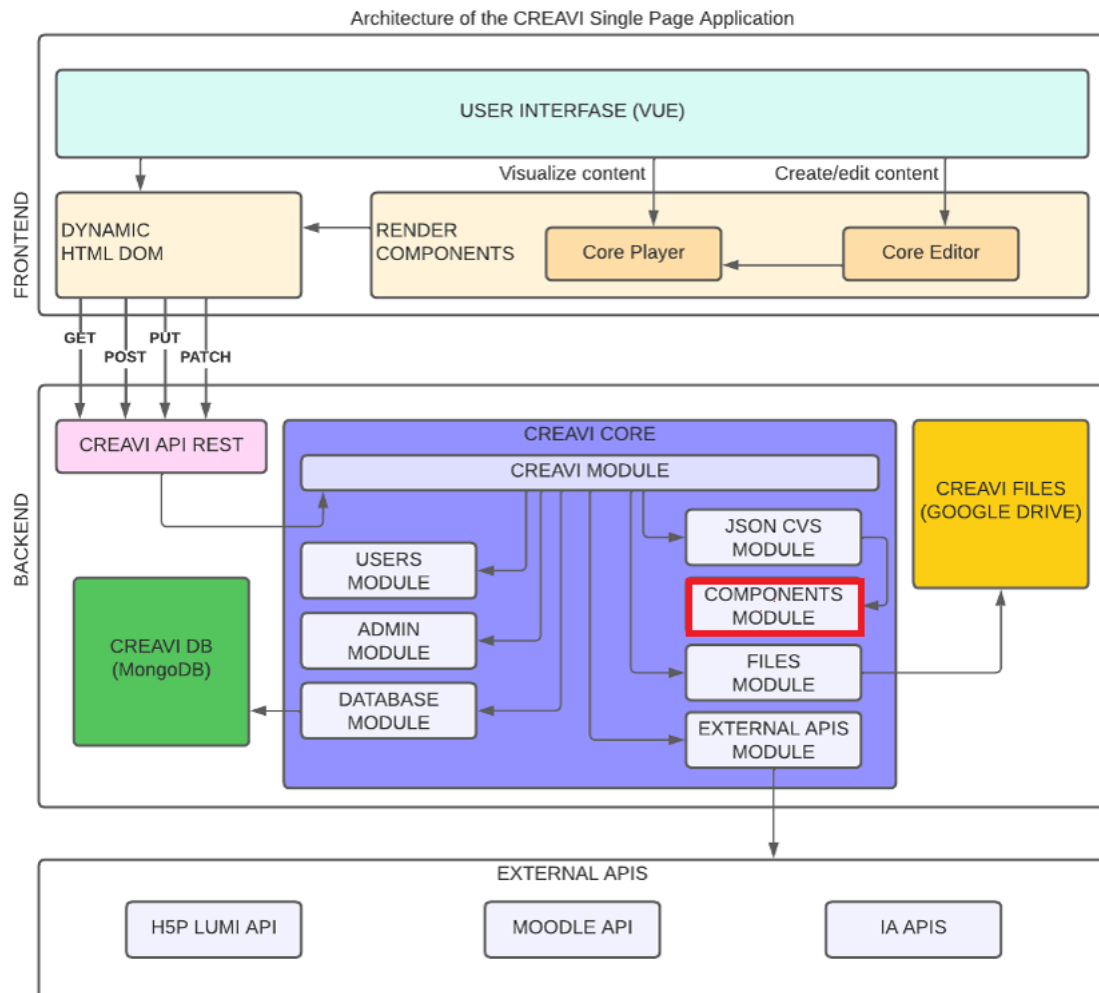
### **5. Componentes del Backend.**

El Backend lo componen una serie de componentes que corresponden a piezas individuales dentro del software y cada una de estas desempeña una función específica dentro del sistema.



## 6. Diagrama de arquitectura.

El diagrama de arquitectura está compuesto por una serie de módulos que en conjunto son la base del software. Cada módulo tiene una función específica que involucra a los demás. Tal es el caso del módulo formulario (Components Module señalado en la figura).



## Elección de la Base de Datos

### 7. Evaluación de Opciones (NoSQL)



El sistema de gestión de bases de datos NoSQL, se toma como elección para el diseño de software educativo, ya que almacena y recupera datos de formas más flexibles y además de ello, no requiere un esquema fijo como lo es SQL. Además, utiliza varios modelos de datos como documentos, o columnas amplias dependiendo del tipo de BD.

## **8. Justificación de la Elección**

Este sistema de base de datos cuenta con las necesidades del proyecto por ello, se trabajará con MongoDB ya que es una base de datos que además de tener un formato JSON que es apropiado para datos no estructurado, es muy adecuada para aplicaciones web y móviles ya que, tienen un crecimiento rápido y eficiente en el manejo de grandes volúmenes de datos.