

Universidad del Valle de Guatemala Ingeniería en Software 1 <b>Estudiante:</b> 22272, Bianca Renata Calderón Caravantes 22473, Madeline Nahomy Castro Morales 22716, Aroldo Xavier López Osoy 22233, Daniel Eduardo Dubon Ortiz 22386, Flavio André Galán Donis	<b>Fecha:</b> Miércoles, 06 de marzo 2024 <b>Sección:</b> 20 <b>Docente:</b> Cristián Muralles
--	--

## Tarea Investigativa 2.

### *Frameworks*

### [Link a Presentación](#)

#### **Backend - Axum**

Axum es un framework para servidores web que se enfoca en modularidad y ergonomía para el programador.

Los principales componentes de Axum son:

- **Router:** Entidad que se utiliza para asociar rutas del servidor a funciones llamadas `Handlers`.
- **Handlers:** Funciones que se encargan de procesar el input enviado hacia la ruta por medio del protocolo HTTP y devolver una respuesta.
- **Extractors:** Funciones que intentan extraer cierta data sobre el payload enviado hacia una ruta por medio de una request. El usuario puede implementar sus propios extractos según vea conveniente.
- **Responses:** Cualquier estructura que implemente la interfaz `IntoResponse` puede ser usada para responder a una request en Axum.
- **Middlewares:** Funciones de lógica de negocio intermedias que se pueden asociar a una cierta cantidad de requests. Estas funciones dependiendo de cuándo se asignen corren antes o después que una request dada.

Un ejemplo de un servidor sencillo creado en Axum se ve de la siguiente manera:

```
use axum::{
    routing::get,
    Router,
};

#[tokio::main]
async fn main() {
    // build our application with a single route
    let app = Router::new().route("/", get(|| async { "Hello, World!" }));

    // run our app with hyper, listening globally on port 3000
    let listener = tokio::net::TcpListener::bind("0.0.0.0:3000").await.unwrap();
    axum::serve(listener, app).await.unwrap();
}
```

Axum es un framework liviano para servidores web, utiliza primariamente el protocolo HTTP aunque también soporta websockets. Debido a su sencillez necesita de otras librerías para realizar tareas más complejas como lo son conectarse con una base de datos, o tener un mejor soporte para el logging. Sin embargo Axum también es consciente sobre su propia sencillez por lo que está diseñado para que integrar este tipo de librerías externas sea sencillo.

El principal patrón de diseño que implementa Axum es el de Chain of Responsibility, pues se tienen los middlewares los cuales intentan procesar la request y opcionalmente pueden decir que el siguiente middleware se encargue de procesar la request.

Se recomienda el uso de Axum cuando se necesite un servidor web rápido y que no consume muchos recursos. Además es positivo si los programadores ya tienen experiencia en Rust debido a que deberán programar en este lenguaje para poder utilizarlo.

## Frontend - HTMX

HTMX es una biblioteca JavaScript que facilita la construcción de aplicaciones web interactivas y dinámicas, centrándose en la simplicidad y la accesibilidad para el desarrollador.

Los principales conceptos de HTMX son:

1. **Atributos de HTMX:** Atributos HTML personalizados que se utilizan para definir el comportamiento dinámico de elementos HTML, como la actualización parcial de la página o la realización de solicitudes AJAX.
2. **Eventos HTMX:** Eventos que se pueden asociar a elementos HTML para desencadenar acciones específicas, como la ejecución de una solicitud HTTP o la actualización de una parte de la página.
3. **Selector de destino:** Selector de CSS que especifica qué parte de la página se actualizará con la respuesta de una solicitud HTMX.
4. **Caché de HTMX:** Mecanismo que almacena en caché las respuestas de las solicitudes HTMX para mejorar el rendimiento y la velocidad de carga de la aplicación.
5. **Transformadores:** Funciones que se utilizan para manipular la respuesta de una solicitud HTMX antes de que se actualice en la página.

Un ejemplo de una aplicación simple creada con HTMX se vería así:

```
<script src="https://unpkg.com/htmx.org@1.9.10"></script>
<!-- have a button POST a click via AJAX -->
<button hx-post="/clicked" hx-swap="outerHTML">
  Click Me
</button>
```

HTMX es una biblioteca frontend ligera y fácil de usar, que se basa en tecnologías web estándar como HTML, CSS y JavaScript. Permite agregar funcionalidades interactivas a una aplicación web sin la necesidad de escribir mucho código JavaScript personalizado. HTMX se integra fácilmente con otras bibliotecas y frameworks, lo que la hace compatible con una amplia gama de herramientas y tecnologías existentes.

El principal patrón de diseño que implementa HTMX es el de Progressive Enhancement, donde se mejora gradualmente la experiencia del usuario agregando funcionalidades interactivas a una aplicación web básica.

Se recomienda el uso de HTMX cuando se necesite agregar interactividad y dinamismo a una aplicación web de forma sencilla y sin complicaciones. Además, es beneficioso si los desarrolladores ya están familiarizados con HTML y CSS, ya que HTMX se basa en estos lenguajes para su funcionamiento.

## **Semejanzas y Diferencias**

### **Semejanzas:**

1. Ambos están diseñados para mejorar la experiencia del usuario en aplicaciones web.
2. Tienen como objetivo principal reducir la cantidad de código JavaScript necesario para implementar interacciones dinámicas en una página web.
3. Permiten realizar actualizaciones de la interfaz de usuario de manera rápida y eficiente, sin necesidad de recargar toda la página.

### **Diferencias:**

1. Axum es un framework de JavaScript mientras que HTMX es una biblioteca.
2. Axum se basa en el patrón de arquitectura MVC (Modelo-Vista-Controlador) mientras que HTMX es más flexible en cuanto a la arquitectura que puedes utilizar junto con él.
3. HTMX se centra principalmente en la manipulación del DOM (Document Object Model) mediante el uso de atributos HTML adicionales, mientras que Axum proporciona una estructura más completa para el desarrollo de aplicaciones web, incluyendo enrutamiento, gestión de estado y manejo de eventos.
4. Axum puede tener una curva de aprendizaje ligeramente más pronunciada debido a su estructura más completa y al uso de un patrón de arquitectura específico.