

The background is a dark navy blue. On the left, there is a large, semi-transparent circular graphic containing a detailed image of a printed circuit board (PCB). Overlaid on the top left of this circle are two overlapping triangles: a blue one in the foreground and a light green one behind it. In the top right corner, there is a faint, light gray pattern of interlocking cubes or a 3D grid.

Frameworks para Backend y Frontend

Flavio Galán
Bianca Calderón
Madeline Nahomy
Daniel Dubón
Xavier Aroldo



Axum





Componentes

01

Routers: Entidad que se utiliza para asociar rutas del servidor a funciones llamadas `Handlers`.

02

Handlers (middlewares): Funciones que se encargan de procesar el input enviado hacia la ruta por medio del protocolo HTTP y devolver una respuesta.

03

Responses: Cualquier estructura que implemente la interfaz `IntoResponse` puede ser usada para responder a una request en Axum.

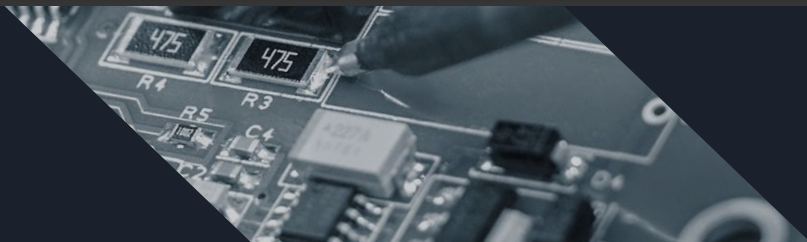


Hello World en Axum

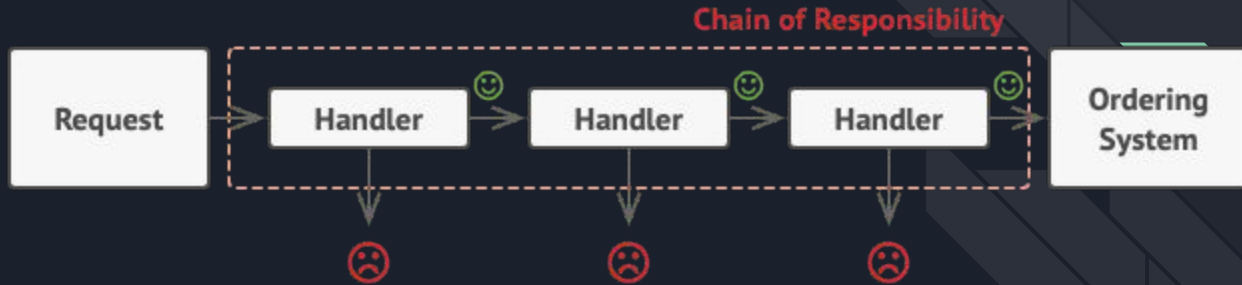
```
use axum::{
    routing::get,
    Router,
};

#[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();
}
```



Patrón de diseño principal "Single Chain of Responsibility"





HTMX

`</> htmx`



¿Qué es?

HTMX es un moderno framework de desarrollo web que permite enriquecer páginas HTML estándar con funcionalidades dinámicas sin necesidad de JavaScript complejo. Fue diseñado para facilitar la creación de interfaces de usuario web interactivas con menos esfuerzo y más claridad, apoyándose fuertemente en HTML para definir la lógica de la interfaz de usuario. Aquí te dejo algunos puntos clave sobre HTMX:

- **Simplicidad y facilidad de uso:** Al centrarse en HTML, HTMX es accesible para desarrolladores con diversos niveles de experiencia.
- **Optimización de la experiencia del usuario:** Permite a los desarrolladores mejorar la interactividad y la velocidad de las aplicaciones web mediante la carga de contenido y la actualización de interfaces de usuario en tiempo real.
- **Bajo acoplamiento y mantenibilidad:** Al mantener la lógica de la interfaz de usuario dentro del HTML, HTMX facilita el mantenimiento y la prueba de las aplicaciones web, ya que reduce la complejidad y el acoplamiento entre el frontend y el backend.



</> HTMX



Motivo de Creación

- **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.
- **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.
- **Selector de destino:** Selector de CSS que especifica qué parte de la página se actualizará con la respuesta de una solicitud HTMX.
- **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.
- **Transformadores:** Funciones que se utilizan para manipular la respuesta de una solicitud HTMX antes de que se actualice en la página.



Componentes

01

Tags: Son atributos que añade HTMX al HTML normal para extender su funcionalidad.

02

hx-{HTTP method}: Tag que indica a HTMX a qué ruta y qué tipo de request debe enviar HTMX.

03

hx-swap: Tag que le indica a HTMX qué debe reemplazar cuando se obtenga una response de parte de hx-{HTTP method}.

HTMX en Acción

```
<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>
```

1. La primera línea carga la biblioteca HTMX desde un CDN (Content Delivery Network). Esto es necesario para poder utilizar las funcionalidades de HTMX en la página web.
2. El elemento `<button>` está equipado con dos atributos personalizados de HTMX:
 - a. `hx-post`: Este atributo indica que al hacer clic en el botón, HTMX realizará una solicitud POST asincrónica a la URL especificada, en este caso, `/clicked`.
 - b. `hx-swap`: Define cómo HTMX actualizará el contenido de la página una vez que se reciba la respuesta del servidor. El valor `"outerHTML"` indica que HTMX reemplazará el elemento completo (en este caso, el botón) con el contenido nuevo que provenga del servidor.

Patrón de diseño principal

HTMX adopta el patrón de diseño Command como su principio central, lo que permite una separación clara entre la iniciación de acciones en la interfaz de usuario y su ejecución en el servidor. Este enfoque facilita la encapsulación de acciones como objetos dentro del HTML, promoviendo un desarrollo web simplificado y mantenible. Gracias a este patrón, HTMX desacopla el frontend del backend, mejora la flexibilidad y extensibilidad de las aplicaciones web, y se alinea con las necesidades de interactividad asíncrona moderna, todo ello sin depender de complejo código JavaScript, manteniendo las interacciones directamente manejables a través de HTML.

Command pattern – Class diagram

