Presentación de proyecto



*Astro\_project*

**Fecha**: 30/10/2024

**Nombre del Autor**: Dmytro Reva

**GitHub**: <https://github.com/DmytroReva2001/Astro_Project>

Índice

[Título del Proyecto 4](#_Toc181194501)

[Objetivo 4](#_Toc181194502)

[Estructura 5](#_Toc181194503)

[1. Tecnología Base 5](#_Toc181194504)

[2. Componentes Principales 5](#_Toc181194505)

[3. Componentes Secundarios 5](#_Toc181194506)

[4. Páginas Principales 6](#_Toc181194507)

[5. Gestión de Datos 6](#_Toc181194508)

[6. Características Técnicas 6](#_Toc181194509)

[7. Internacionalización 7](#_Toc181194510)

[8. Diseño y Experiencia de Usuario 7](#_Toc181194511)

[9. Seguridad y Eficiencia 7](#_Toc181194512)

[10. Documentación 7](#_Toc181194513)

[Análisis Técnico 8](#_Toc181194514)

[Complejidad Temporal 8](#_Toc181194515)

[1. Obtención de Datos: 8](#_Toc181194516)

[2. Renderización de Componentes: 8](#_Toc181194517)

[3. Procesamiento de Datos: 8](#_Toc181194518)

[Manejo de Errores 8](#_Toc181194519)

[1. Estrategias Implementadas: 8](#_Toc181194520)

[2. Mejoras en Accesibilidad: 8](#_Toc181194521)

[Optimización de Rendimiento 9](#_Toc181194522)

[1. Pool de Conexiones: 9](#_Toc181194523)

[2. Consultas Eficientes: 9](#_Toc181194524)

[3. Carga Asíncrona: 9](#_Toc181194525)

[Seguridad 9](#_Toc181194526)

[1. Prevención de Ataques XSS: 9](#_Toc181194527)

[2. Gestión Segura de Conexiones: 9](#_Toc181194528)

[Escalabilidad y Mantenibilidad 9](#_Toc181194529)

[1. Estructura Modular: 9](#_Toc181194530)

[2. Funciones Reutilizables: 9](#_Toc181194531)

[Tecnologías utilizadas 9](#_Toc181194532)

[Fortalezas 10](#_Toc181194533)

[Estructura Clara 10](#_Toc181194534)

[Diseño Moderno 10](#_Toc181194535)

[Accesibilidad 10](#_Toc181194536)

[Evaluación General 11](#_Toc181194537)

[Próximos pasos 11](#_Toc181194538)

El proyecto que presentaré es un portfolio diseñado para la evaluación de aprendizaje. A continuación, se detallan los aspectos clave del proyecto:

# Título del Proyecto

Astro\_project

# Objetivo

Desarrollar un portfolio online interactivo y dinámico que demuestre el proceso de aprendizaje continuo, las habilidades adquiridas y la capacidad de adaptación a nuevas tecnologías.

Este proyecto busca no solo mostrar los trabajos realizados, sino también evidenciar el crecimiento profesional y la habilidad de aprendizaje autodidacta.



# Estructura

## 1. Tecnología Base

AstroJS: Framework moderno (creado en 2021) para sitios web estáticos.



Ventajas: Rendimiento optimizado, facilidad de uso, y capacidad para crear interfaces interactivas.

## 2. Componentes Principales

* **Header**: Navegación principal y branding del sitio.
* **Footer**: Información de contacto y enlaces adicionales.
* **Layout**: Estructura base para todas las páginas, asegurando consistencia.

## 3. Componentes Secundarios

* **Card**: Elemento general con diseño animado de una tarjeta con uso de cadenas e imágenes.
* **ProjectCard**: Elemento para datos relacionados con proyectos con diseño animado de una tarjeta con uso de cadenas, imágenes y uso de datos de tecnologías relacionadas con proyecto.
* **SkillCard**: Elemento para datos relacionados con habilidades con diseño animado de una tarjeta con uso de cadenas e imágenes.
* **Intefaces**: Elemento para almacenar interfaces que podrían ser reutilizados.
* **Urls**: Elemento para almacenar las urls a las páginas del sitio que podrían ser reutilizados.

## 4. Páginas Principales

* Página de **Inicio** (**index**): Introducción al portfolio.
* Página de **Contacto**: Información de contacto y disponibilidad.
* Página de **CV**: Curriculum Vitae con opción de cambio de idioma.
* Página de **Portafolio**: Showcase de proyectos realizados.
* Página de **Habilidades**: Showcase de habilidades/tecnologías dominadas.

## 5. Gestión de Datos

Base de Datos MySQL: Almacenamiento de información dinámica.



Funciones Asíncronas:

* **getDatos**(): Recupera información general.
* **getProjects**(): Obtiene detalles de proyectos y tecnologías asociadas.
* **getHabilidades**(): Recupera información sobre habilidades.

## 6. Características Técnicas

Renderizado Asíncrono: Carga eficiente de datos desde la base de datos.

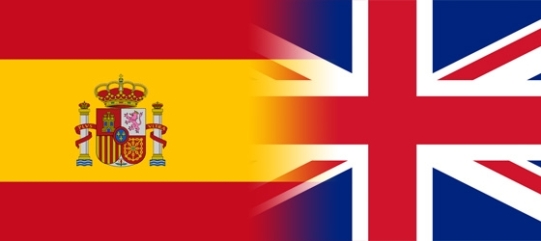
Optimización de Rendimiento: Uso de pool de conexiones para gestionar eficientemente las interacciones con la base de datos.

Manejo de Errores: Implementación de try-catch para gestionar excepciones en las operaciones de base de datos.

## 7. Internacionalización

Documentación Bilingüe: Disponible en español e inglés.

CV en Dos Idiomas: Opción para alternar entre versiones en español e inglés del CV.



## 8. Diseño y Experiencia de Usuario

Diseño Responsivo: Adaptable a diferentes dispositivos.

Interactividad: Elementos interactivos como botones para cambiar idiomas y efectos de hover en iconos sociales.

## 9. Seguridad y Eficiencia

Sanitización de Datos: Prevención de ataques XSS en datos recuperados de la base de datos.

Gestión Eficiente de Conexiones: Uso de un pool de conexiones para optimizar el acceso a la base de datos.

## 10. Documentación

Documentación Técnica: Disponible en español e inglés, detallando la estructura del proyecto, funciones principales y análisis de complejidad temporal.

# Análisis Técnico

## Complejidad Temporal

### Obtención de Datos:

La operación más costosa es la obtención de datos desde la base de datos MySQL.

Complejidad: Varía según la consulta y la estructura de la base de datos.

* Mejor caso: O(1) para consultas simples con índices optimizados.
* Peor caso: O(n) para consultas complejas o tablas no indexadas, donde n es el número de registros.

### Renderización de Componentes:

La renderización de componentes como Header, Footer y Layout tiene una complejidad constante O(1).

El contenido estático (como la información de contacto) también se renderiza en tiempo constante.

### Procesamiento de Datos:

* La función **getDatos**(): O(n), donde n es el número de filas en la tabla 'datos'.
* La función **getProjects**(): O(n \* m), donde n es el número de proyectos y m el número promedio de tecnologías por proyecto.
* La función **getHabilidades**(): O(n), donde n es el número de habilidades en la tabla.

## Manejo de Errores

### Estrategias Implementadas:

Uso de bloques try-catch en funciones asíncronas para capturar y manejar errores de base de datos.

Liberación adecuada de conexiones de base de datos, incluso en caso de error.

### Mejoras en Accesibilidad:

Implementación de atributos ARIA para mejorar la navegación para usuarios con tecnologías de asistencia.

Uso correcto de etiquetas semánticas HTML5 para una mejor estructura del documento.

## Optimización de Rendimiento

### Pool de Conexiones:

Utilización de un pool de conexiones para la base de datos, optimizando la gestión de recursos y mejorando el rendimiento en consultas concurrentes.

### Consultas Eficientes:

Uso de JOINs en la función **getProjects**() para reducir el número de consultas a la base de datos.

### Carga Asíncrona:

Implementación de carga asíncrona de datos, permitiendo que la interfaz de usuario se renderice mientras se esperan los datos.

## Seguridad

### Prevención de Ataques XSS:

Sanitización de datos recuperados de la base de datos antes de renderizarlos en el HTML.

### Gestión Segura de Conexiones:

Uso de variables de entorno para almacenar credenciales de base de datos, evitando su exposición en el código fuente.

## Escalabilidad y Mantenibilidad

### Estructura Modular:

Separación clara de componentes (Header, Footer, Layout) facilitando el mantenimiento y la escalabilidad.

### Funciones Reutilizables:

Implementación de funciones como **getDatos**(), **getProjects**(), y **getHabilidades**() que pueden ser reutilizadas en diferentes partes de la aplicación.

## Tecnologías utilizadas

* JavaScript
* TypeScript
* Astro
* MySQL

# Fortalezas

### Estructura Clara

Reutilización de Componentes: La arquitectura del proyecto está diseñada para facilitar la reutilización de componentes como Header, Footer y Layout. Esto no solo optimiza el código, sino que también mejora la mantenibilidad y escalabilidad del sitio, permitiendo actualizaciones rápidas y consistentes en toda la aplicación.

Organización Efectiva del Contenido: La disposición lógica de las secciones permite una navegación intuitiva. Cada página está estructurada de manera que los usuarios pueden acceder fácilmente a la información relevante, mejorando la experiencia general del usuario.

### Diseño Moderno

Estética Atractiva: El uso de CSS bien implementado garantiza una experiencia visual moderna y atractiva. Se han aplicado principios de diseño contemporáneo para crear una interfaz que no solo es funcional, sino también estéticamente agradable.

Interactividad Mejorada: Se han incorporado efectos visuales y transiciones suaves que enriquecen la interacción del usuario, haciendo que la navegación sea más dinámica y atractiva. Elementos como botones y enlaces responden a las acciones del usuario con retroalimentación visual inmediata.

Diseño Responsivo: La aplicación se adapta a diferentes tamaños de pantalla y dispositivos, asegurando que todos los usuarios tengan una experiencia de navegación fluida, independientemente del dispositivo que utilicen.

### Accesibilidad

Cumplimiento de Normativas: Se han implementado características que cumplen con los estándares de accesibilidad (WCAG), como el uso adecuado de etiquetas ARIA y un contraste de colores que mejora la legibilidad.

Navegación Inclusiva: La estructura del sitio permite una navegación fluida para usuarios con diferentes capacidades, garantizando que todos puedan acceder al contenido sin obstáculos.

# Evaluación General

El código ha sido evaluado con una puntuación de 7/10. (Vía )

Aunque está bien estructurado, se identificaron áreas de mejora en el manejo de errores y accesibilidad. Este portfolio no solo servirá como un recurso para evaluar el aprendizaje, sino también como un blog para hacer seguimiento sobre las habilidades desarrolladas, tecnologías dominadas y proyectos desarrollados reflejados en la presentación y reflexión sobre su trabajo.

# Próximos pasos

Seguir añadiendo contenido a las habilidades y proyectos. Mejorar diseño, implementar mejora manejo de datos.