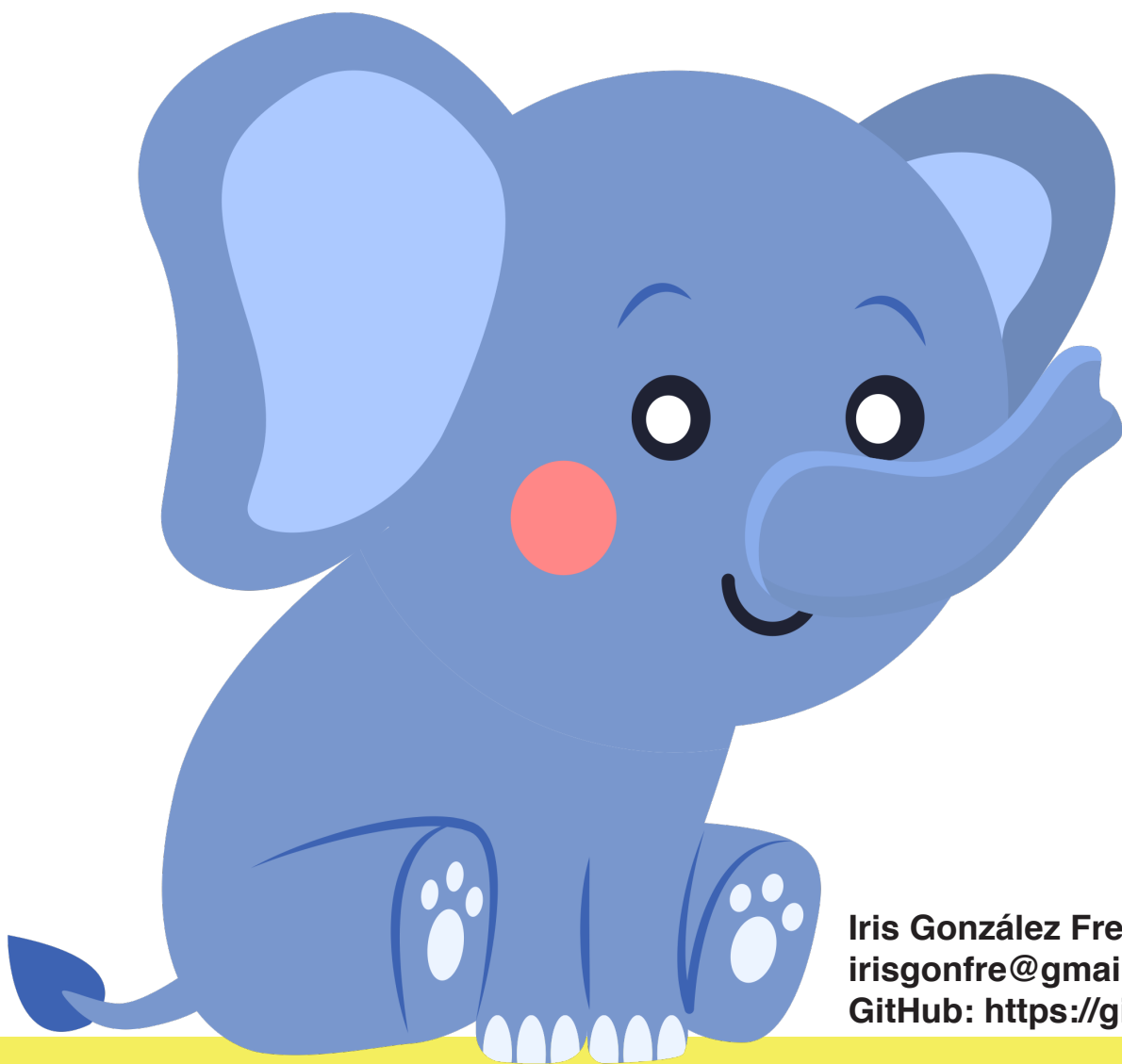


IRIS GONZALEZ FRESNO

JUMP



Iris González Fresno
irisgonfre@gmail.com
GitHub: <https://github.com/Irisgonzf>

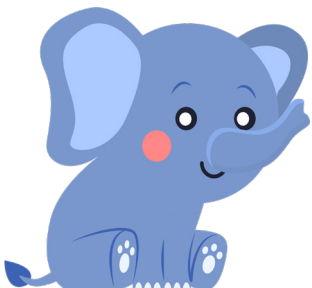
ÍNDICE

1.Breve descripción del proyecto.....	4
2.Tecnología y justificación de la misma	5
3.Guía Breve de Instalación.....	6
4.Guía breve de usuario.....	7
5.Diagrama.....	9
6.Conclusiones finales.....	10

1. Breve descripción del proyecto

El proyecto consiste en una tienda de juguetes, donde los usuarios pueden ver los productos disponibles junto con sus existencias y una imagen del producto. También cuenta con una página de contacto donde pueden enviar un Email. El sistema está diseñado para ofrecer una experiencia sencilla, fácil y atractiva.

El proyecto también incluye una sección de administrador. El administrador puede gestionar los productos y clientes, permitiendo realizar acciones como crear, mostrar, editar, filtrar y eliminar tanto productos como clientes. Además, existe una tabla que permite asignar productos a clientes. También tienen la capacidad de exportar todas las tablas de productos, clientes y asignaciones en archivos CSV, lo que facilita la gestión y análisis de los datos. El sistema también incluye una API REST de productos.



2.Tecnología y justificación de la misma

En este proyecto he utilizado Laravel 10.48.22, un framework de PHP que facilita el desarrollo de aplicaciones web. Elegí Laravel porque es el framework con el que tengo mayor experiencia y me permite realizar este tipo de proyectos de forma sencilla y rápida.

OTRAS TECNOLOGÍAS UTILIZADAS:

HTML, CSS y JavaScript:

HTML: Para la estructuración de las vistas y el contenido web, como la página de productos, formularios de clientes, el administrador entre otros.

CSS: Para darle diseño a las páginas y que fuera atractivo hacia el usuario y el administrador.

JavaScript: Se utilizó para agregar interactividad.

Bootstrap: Me ayudo agilizar el proceso de diseño y asegurarme de que todo fuera responsive.

Maatwebsite Excel (librería para exportación de datos):

Para cumplir con el requisito de exportar las tablas de productos, clientes y asignaciones a archivos CSV, utilicé la librería maatwebsite/excel. Esta librería es muy utilizada en Laravel para exportar e importar datos en diversos formatos, como CSV y Excel.

Mailtrap (para la gestión de correos electrónicos):

Para la funcionalidad de envío de correos elec

trónicos, utilicé Mailtrap.io, un servicio que permite probar el envío de correos electrónicos. Esto resultó útil para garantizar que los correos se enviaran correctamente y que la lógica de la aplicación funcionara.

Postman (para probar la API):

Para verificar que la API de productos estaba funcionando correctamente, utilicé Postman, que me permitió hacer solicitudes HTTP (GET, POST, PUT, DELETE) para interactuar con la API de productos. Esto me ayudó a asegurarme de que las respuestas fueran correctas.

JUSTIFICACIÓN:

Elegí Laravel principalmente por mi experiencia con el framework. Además, Laravel proporciona un conjunto de herramientas que facilita la implementación de características complejas, como la validación de formularios, el manejo de bases de datos y el envío de correos electrónicos.



3. Guía Breve de Instalación

Para instalar este proyecto, sigue estos pasos:

Instalar XAMPP: Asegúrate de tener instalado XAMPP en tu computadora.

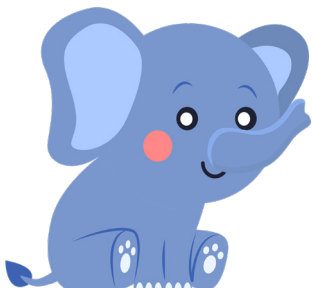
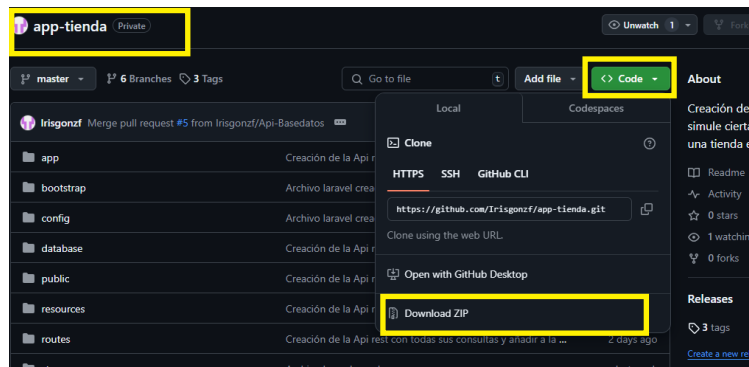
Descargar el Proyecto:

Accede a mi repositorio de GitHub: <https://github.com/Irisgonzf>.

Dentro de mi repositorio, ve a la carpeta app-tienda.

Haz clic en el botón verde Code y selecciona Download ZIP.

Una vez descargado, descomprime la carpeta.



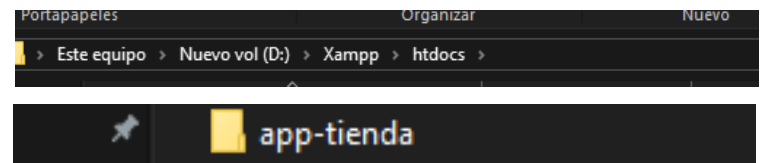
Colocar el Proyecto en XAMPP

Copia la carpeta descomprimida y pégala dentro de la carpeta htdocs de tu instalación de XAMPP

Abrir el Proyecto en Visual Studio Code:

Abre Visual Studio Code y carga la carpeta del proyecto.

Luego en el archivo .env en DB_DATABASE pon bdtienda



Crear la Base de Datos:

Abre el panel de administración de XAMPP y accede a phpMyAdmin.

Crea una nueva base de datos llamada bdtienda.

Importar la Base de Datos:

Dentro del proyecto, encontrarás una carpeta llamada importarbd.

En phpMyAdmin, selecciona la base de datos bdTienda y usa la opción Importar para cargar el archivo SQL dentro de esa carpeta.



Ejecutar las Migraciones:

En Visual Studio Code, abre la terminal y navega hasta el directorio del proyecto.
Ejecuta el siguiente comando para realizar las migraciones de la base de datos:

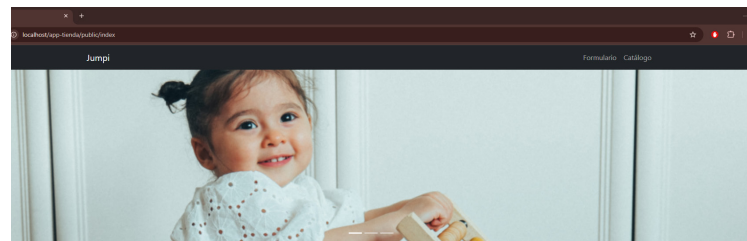
php artisan migrate

Acceder al Proyecto:

Abre tu navegador y visita <http://localhost/app-tienda/public/index> para ver el proyecto en funcionamiento.

Para acceder a la vista de administrador pon: <http://localhost/app-tienda/public/productos>.

Carrusel de Imágenes que muestra fotografías de niños jugando con juguetes.
Un texto de Bienvenida y lo que encontraras



Productos Destacados: Tres productos destacados se muestran en tarjetas. Cada tarjeta incluye el nombre del producto la descripción y un botón de ver más.

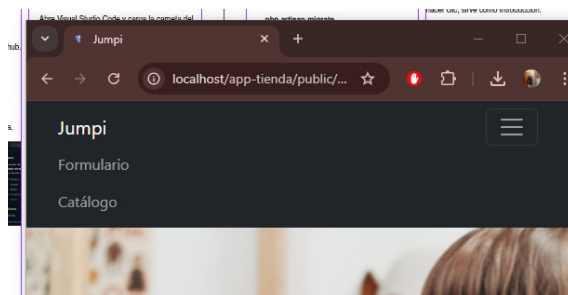


4. Guía breve de usuario

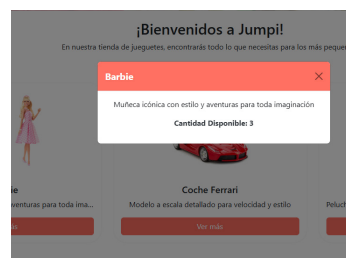
1. Página de Inicio

Al acceder al sitio, el usuario verá una barra de navegación en la parte superior con dos enlaces principales:

Formulario: Lleva a la página de contacto.
Catálogo: Muestra todos los productos.

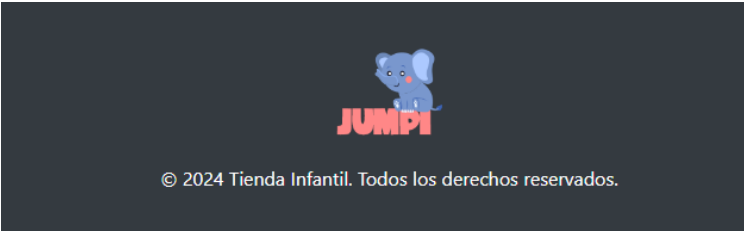


Funcionalidad: Al hacer clic en ver más, se abre un cuadro en la misma ventana que muestra todos los detalles del producto. Para cerrar el cuadro, se puede hacer clic fuera del cuadro o en la X.



Pie de Página

Descripción: Al final de la página, el usuario encontrará el pie de página que incluye el logotipo y la franse © 2024 Tienda Infantil. Todos los derechos reservados.

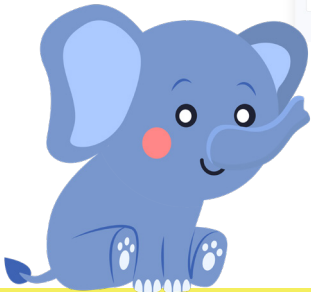


2. Vista de Contacto (Formulario)

Navegador: La barra de navegación igual pero cambiando los enlaces

Inicio: Regresa a la página principal.
Catálogo: Dirige a la página del catálogo.

Formulario de Contacto:El usuario puede rellenar los datos y enviar un mensaje. Al enviar el formulario correctamente, aparece un mensaje de confirmación indicando que se ha enviado correctamente.



¡CONTACTANOS!

Tu mensaje ha sido enviado correctamente

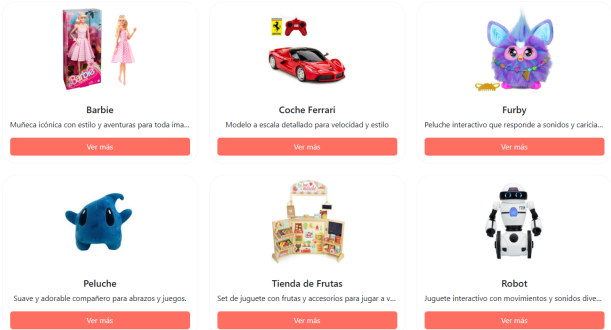
Formulario de Contacto

3. Página de Catálogo

Navegador: La barra de navegación igual pero cambiando los enlaces

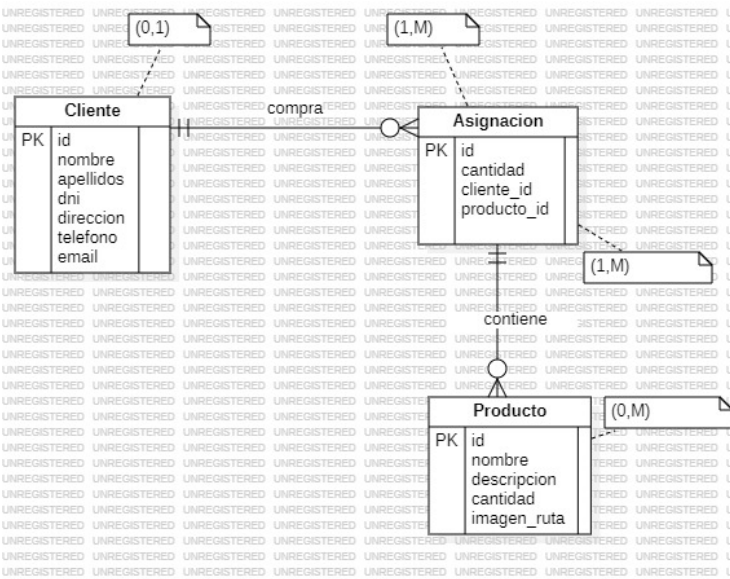
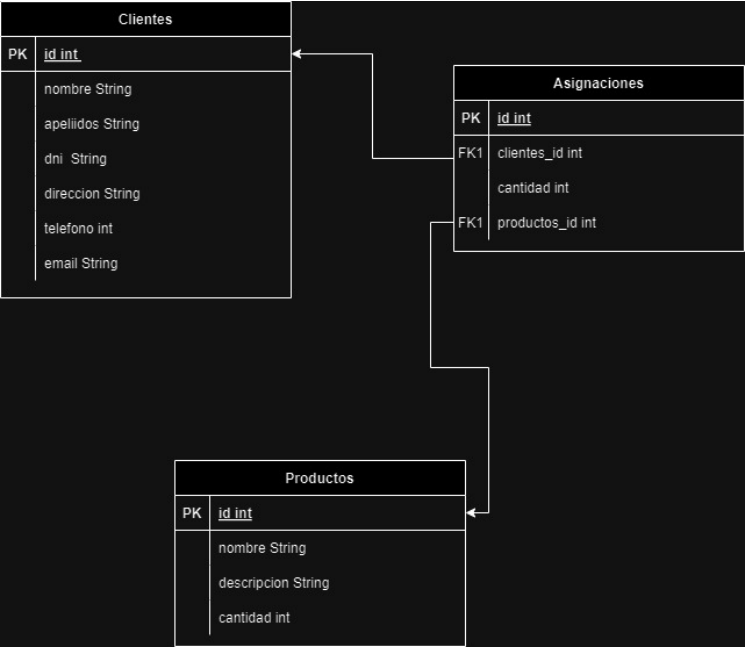
Inicio: Regresa a la página principal.
Formulario: Dirige a la página de contacto

En esta página, se muestran todos los productos de la tienda. Cada producto aparece en una tarjeta, con el botón ver más que abre un cuadro con detalles adicionales.



En todas las vistas tenemos igual el pie de página y en la de catalogo la descripción en cambio la de contacto solo puse contactanos.

5. Diagrama de entidad-relación y un diagrama relacional.



5.Conclusiones finales

Este proyecto me ha permitido crear una página funcional y enfrentar varios desafíos técnicos que, aunque me llevaron más tiempo, me ayudaron a aprender y a mejorar mis habilidades. En particular, uno de los mayores retos fue implementar la API, ya que fue un proceso lioso y que requirió de bastante investigación. Logré resolver algunos problemas iniciales, como enviar correos electrónicos a través de la aplicación y exportar datos en formato CSV, aunque esto último fue complicado porque mi versión inicial de Laravel no era compatible. Tuve que buscar una versión superior e incluso modificar el archivo init.php, lo cual resultó un proceso un tanto confuso y me llevó a ver muchos tutoriales. Sin embargo, el filtro de datos mediante la API se me complicó demasiado, y finalmente no pude implementarlo. En general, la experiencia ha sido muy buena. Si tuviera más tiempo, me enfocaría en ver como hago el filtro de la API y tambien me veria mas tutoriales de GitHub ya que es un mundo.





