

Trabajo Final Entornos Web

Planificación y diseño Página Web

Materia: Entornos Web

Carrera: Ingeniería en Sistemas de Información

Alumna: Bonino Matías

Profesor: Cesar Prato

Año: 2022



Índice

- Planificación y diseño general del Proyecto Web

• Descripción general del proyecto -----	3
• Objetivos del proyecto -----	3
• Alcance -----	3
• Funcionalidades del sitio -----	4
• Diseño responsivo -----	4
• Seguridad -----	4
• Estándares W3C -----	5
• Optimización de Seo -----	6
• Respetto de las imágenes -----	6
• Navegadores -----	6
• Tecnologías y herramientas utilizadas -----	6
• Guía de instalación -----	7
• Imágenes del Sitio -----	9
• Documentación de las tecnologías utilizadas -----	15
• Estimación del tiempo del proyecto -----	15
• Hosting -----	16
• Web Analytics -----	16

Planificación y diseño general del Proyecto Web

Descripción general del proyecto:

El proyecto consiste en un sitio web para la presentación de los productos realizados por la Tienda Maylu, la cual realiza distintos tipos de manualidades en crochet. Estos productos van desde adornos, juguetes, mantas, frazadas, o cualquier otra creatividad que surja del cliente, pueden ser de distintos colores y tamaños según las necesidades y gustos.

Objetivo del proyecto:

Actualmente el medio de darse a conocer es a través de las redes sociales, y con la realización del sitio lo que se busca es personalizar la presentación de los productos y a su vez agregar algunas características adicionales al diseño.

Se agregó la funcionalidad de “Inicio de Sesión” para que clientes logueados puedan acceder a dejar comentarios o “likes” a los productos.

También cuenta con un formulario de contacto para que clientes registrados o no, puedan enviar sus comentarios acerca de la Tienda o también realizar sus consultas o pedidos sobre los productos.

Alcance:

En cuanto a los usuarios a los que está destinado el proyecto, como se mencionó anteriormente, se dirige a todos los usuarios en general, es decir cualquier persona que quiera conocer acerca de los productos realizados por la Tienda.

Toda persona que ingrese al sitio puede contar con toda la información de los productos y sus respectivas imágenes, y acceder al formulario de contacto para realizar cualquier tipo de consulta.

El inicio de sesión está pensado para que sólo usuarios registrados puedan realizar comentarios cada uno de los productos en particular y en un futuro participar de sorteos.

No obstante, al ser un sitio que nace de las redes sociales, se integró la posibilidad de loguearse con cuenta de Facebook, para tratar de mantener los clientes provenientes de dichas redes sociales.

Funcionalidades del sitio:

Al ingresar a la página principal del sitio nos encontramos con un encabezado, el cual está formado por el logo de la Tienda, un menú de navegación y el menú de inicio de sesión.

Como se mencionó, el inicio de sesión puede realizarse a través del ingreso de las credenciales de Facebook o registrar una nueva cuenta en la aplicación haciendo uso del backend propio (la registración de nuevos usuarios aún está pendiente).

Luego una de las opciones del menú de navegación cuenta con un filtro de categorías (este filtro en una próxima versión se quitará de su actual ubicación y se colocará en un costado del cuerpo principal del sitio).

El cuerpo principal cuenta con un “slides” de un grupo de imágenes más representativas del sitio y debajo una sección con cada uno de los productos de la Tienda (los cuales pueden ser filtrados).

Cada uno de estos productos es a su vez un modal donde se muestran mas imágenes e información adicional, desde donde, en una próxima versión, usuarios registrados podrán dejar sus comentarios y “likes”.

En la parte inferior del sitio tenemos información para contactarse con la Tienda y enlaces a las redes sociales.

El sitio consulta además con un formulario de contacto para que los usuarios puedan realizar sus consultas y/o pedidos sobre los productos.

Diseño responsivo:

El sitio fue construido con Ionic, que originalmente es un framework para construir aplicaciones móviles.

A su vez, se instalaron dependencias para construir con dicho framework PWA (progressive web app, aplicaciones web progresivas). Esto nos permite que la misma aplicación pueda ser utilizada desde dispositivos móviles como desde Pc's manteniendo la apariencia correspondiente para cada caso.

Ionic permite con facilidad agregar las configuraciones necesarias para cada tipo de dispositivo (Android, ios, Desktop) y para cada tamaño de plataformas lo que hace más sencillo construir sitios y aplicaciones responsivas.

Seguridad:

El sitio cuenta con dos sistemas de autenticación, uno con credenciales de Facebook y otro a través del backend construido en Laravel.

Para la autenticación con Facebook el sitio fue registrado debidamente mediante el acceso para desarrolladores proveyendo la información requerida por éste.

Con respecto a la autenticación a través de Laravel, el mismo se realiza con la ayuda de la librería Passport instalada en el servidor.

Passport ofrece un sistema de autenticación robusto y de fácil integración con Laravel.

Estándares W3C:

Como se mencionó el sistema está construido en Ionic/Angular (Ionic puede ser utilizado también con React y Vue).

En nuestro caso, al utilizar Ionic/Angular, internamente utiliza la estructura y codificación de Angular. Cada página es un componente independiente, con su código HTML, su lógica y su estilo CSS (si bien el proyecto general también puede incluir un estilo general para toda la aplicación).

Al compilarse el código todos los componentes se “cargan” en un único archivo HTML, el index.html, que es el único que lleva las etiquetas “doctype”, “meta”, “header” y otras. Esto nos lleva a que el único archivo que puede ser probado es el “index.html”. El resto de los archivos HTML de los demás componentes tienen porciones de código. En definitiva, se verificó el “index.html” y efectivamente pasa los estándares, con una única advertencia (de quitar las propiedades de “viewport” del encabezado).

Las propiedades “viewport” en nuestro caso son necesarias para que la aplicación pueda cumplir con su función de Progressive Web App.

Luego, como se mencionó, cada componente tiene su propio archivo de estilos “.CSS”. Se verificaron varios de ellos y no hubo ningún inconveniente en estas pruebas.

Optimización de Seo:

Cabe mencionar que angular al ser un framework que se renderiza del lado del cliente no favorece el tema del SEO, porque de cierta forma el código HTML queda “oculto” (de todas formas es posible agregar las etiquetas para el SEO en el index.html, pero quedaría de manera estáticas y no de todo accesible a los buscadores).

Pero hay una actualización de Angular que se llama "Angular Universal" que convierte a Angular (instalando algunas dependencias adicionales) en un framework que renderiza del lado del servidor (server side rendering).

Actualmente el soporte de Ionic 5 con Angular Universal está en "beta" (según la documentación para la versión 6 de Ionic ya estaría soportado Angular Universal).

No obstante se instalaron las dependencias para convertir la aplicación para que soporte el Server Side Rendering y se verificó (luego de código adicional) que los “meta” de cada componente se modificaban según lo esperado.

Respecto de las imágenes:

Cabe destacar que las imágenes utilizadas son propias y corresponden a los trabajos realizados por la Tienda misma, teniendo los permisos correspondientes para su publicación.

Navegadores:

Se verificó el correcto funcionamiento del sitio en los principales navegadores:

Chrome, Internet Explorer, Firefox, Safari y Opera.

Esto es posible debido a que el código utilizado y sobre todo los estilos aplicados son compatibles con cada una de las versiones de los navegadores mencionados.

Tecnologías y herramientas utilizadas:

Como se mencionó a lo largo de la documentación las principales tecnologías que se utilizaron fueron Ionic/Angular del lado del frontend y Laravel para construir el backend.

La idea de utilizar Ionic/Angular fue para permitir en un futuro una migración rápida a dispositivos móviles.

Sumado al framework de Ionic, se instalaron librerías para convertir el sitio en una PWA (Progressive Web App), lo que permite que el sitio pueda ser accedido desde un browser de una pc o un dispositivo móvil.

Cuando nos referimos a la migración a aplicación móvil nos referimos a agregarle funcionalidades propias de los dispositivos móviles, lo que requiere la instalación de sus respectivos plugin, como ser para acceder a los contactos del dispositivo, a la cámara, etc.

Además de los componentes propios de Ionic para construir las vistas, se instaló la librería Angular Material, ya que provee componentes un poco más vistosos que los de Ionic, y en algunos casos permite simplificar el código (como por ejemplo en la realización del formulario).

Para el envío de los correos del formulario de contactos se utilizó una cuenta en Mailgun y su correspondiente configuración.

En el backend, se decidió por Laravel por su robustez y posibilidad de escalabilidad en caso de ser necesario. Se instaló además la librería Passport para facilitar y gestionar el inicio de sesión.

En cuanto a base de datos, se utilizó MySQL por su sencillez y fácil integración con Laravel.

Para crear el entorno de ejecución de manera local de la base de datos Mysql se utilizó el paquete Xampp.

Con respecto a la escritura del código se utilizó el editor Visual Studio Code.

Guía de instalación:

- 1) Tener instalado el Xampp.
- 2) Instalar el CLI de Ionic (npm install -g @ionic/cli)
- 3) Descargar e instalar composer (<https://getcomposer.org/download/>). Es un instalador según cada equipo, luego es todo "siguiente".
- 4) Clonar el repositorio y colocarlo en la carpeta "xampp\htdocs". (git clone <https://github.com/matias2301/Tienda-Ionic-Laravel.git>).
- 5) Dentro de la carpeta frontend del proyecto correr "npm install".
- 6) Dentro de la carpeta backend del proyecto ejecutar "composer install".
- 6) Dentro de la carpeta backend del proyecto ejecutar "npm install".
- 7) Copiar el archivo .env.example y renombrarlo sólo como .env y dejarlo en la misma ubicación (en el raíz de la carpeta backend).
- 8) En la carpeta backend, en una consola correr "php artisan key:generate".
- 9) Crear una base de datos (por ejemplo Mysql).
- 10) Modificar el archivo .env según la base de datos creada.
Ejemplo de cómo quedaría el archivo .env con la configuración de la base de datos:
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nombreDeLaBD
DB_USERNAME=username
DB_PASSWORD=password

11) En una consola sobre la carpeta backend correr "php artisan migrate" y luego "php artisan db:seed".

Esto último crea un usuario para poder loguearse con email "admin@admin.com" y contraseña "1234". Este usuario tiene role "admin" para poder administrar los productos (aunque esta parte todavía no está implementada).

12) Dentro mismo de la carpeta backend ejecutar "php artisan passport:install".

13) En la carpeta backend ejecutar, en una consola "php artisan serve".

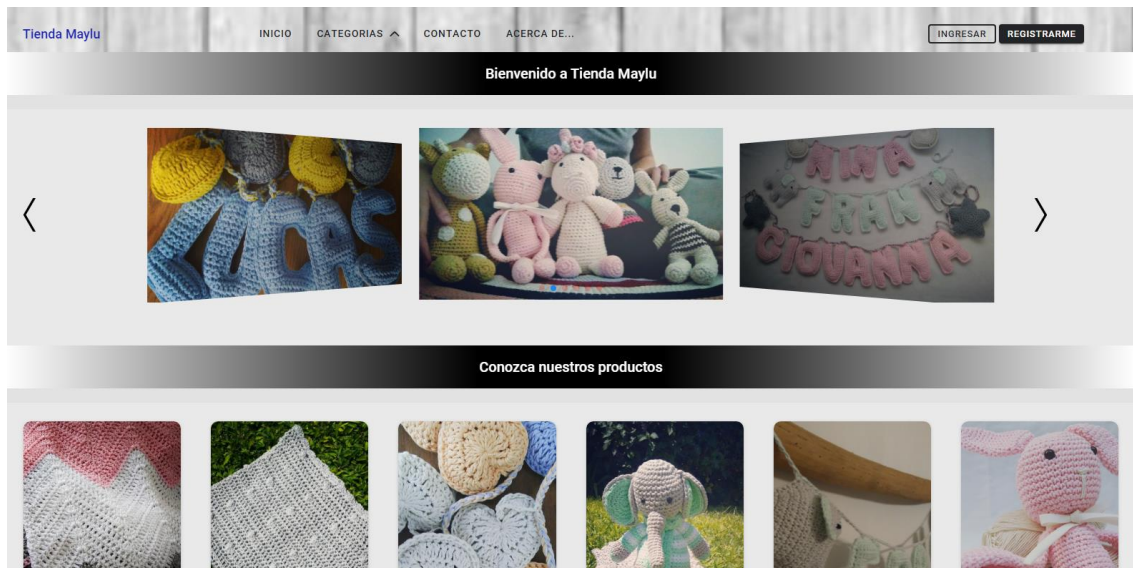
14) En otra consola, en la carpeta frontend, ejecutar "ionic serve -o". Esto abrirá el proyecto en el navegador en localhost:8100 (generalmente).

15) Si se quiere correr el lado del frontend en la funcionalidad de server side rendering utilizando la librería de Angular Universal, en lugar del punto 14) correr el siguiente comando: "npm run dev:ssr".

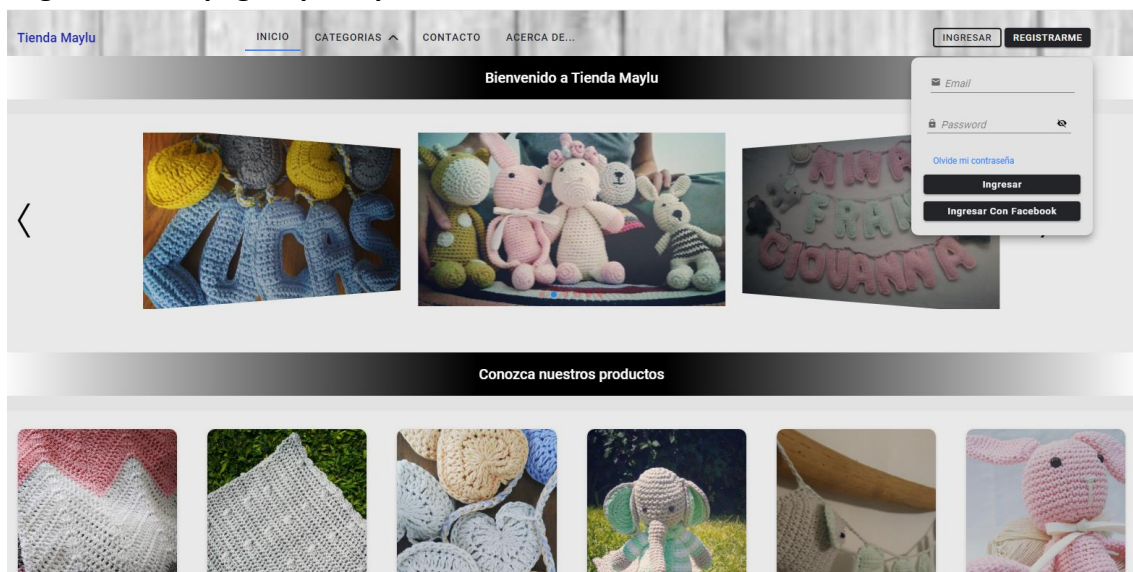
Esta última forma puede generar algún error en la consola que aun requiere ser solucionado.

Imágenes del Sitio:

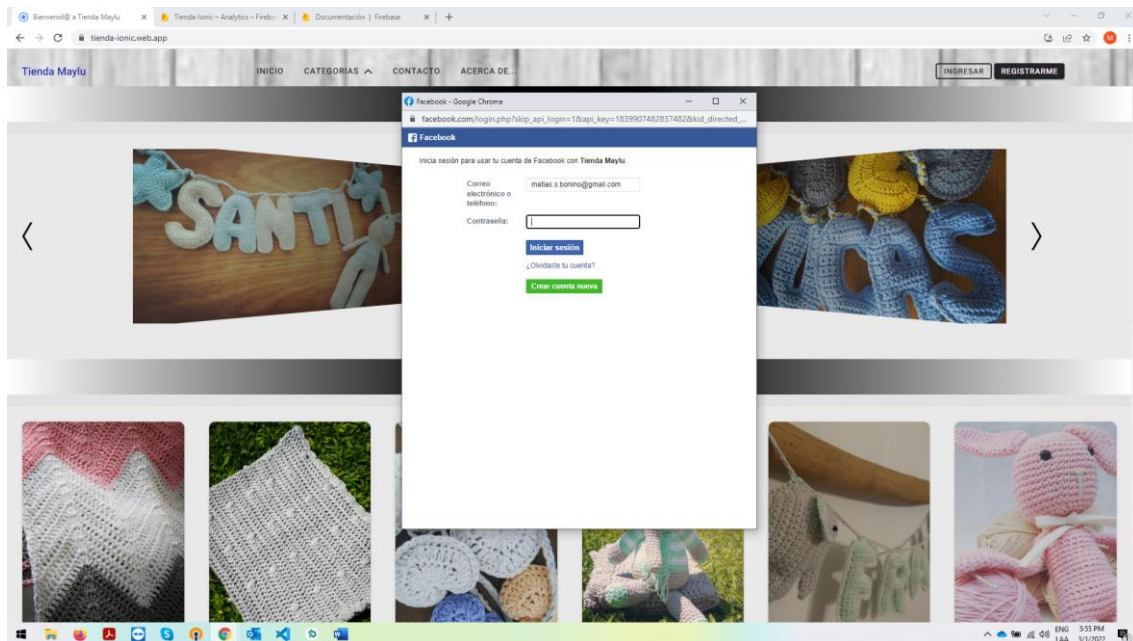
Página principal



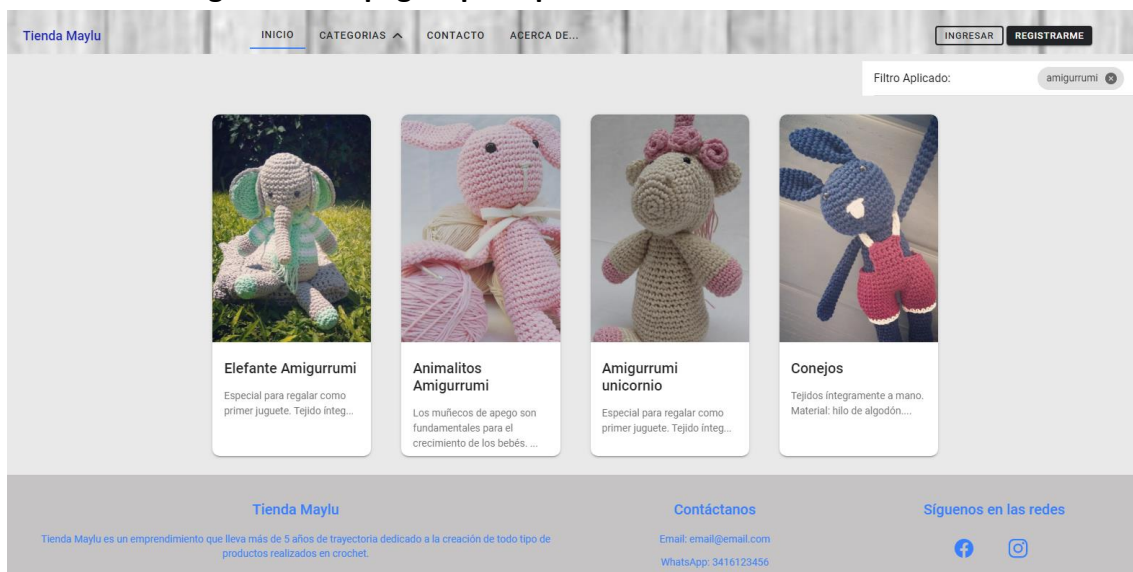
Login desde la página principal



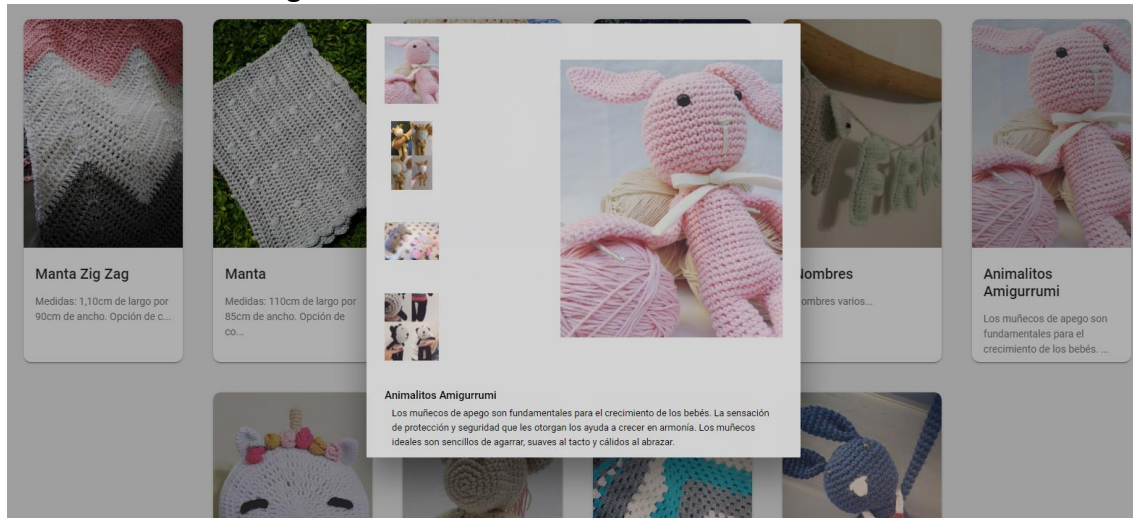
Login utilizando el servicio de Facebook (actualmente deprecado)



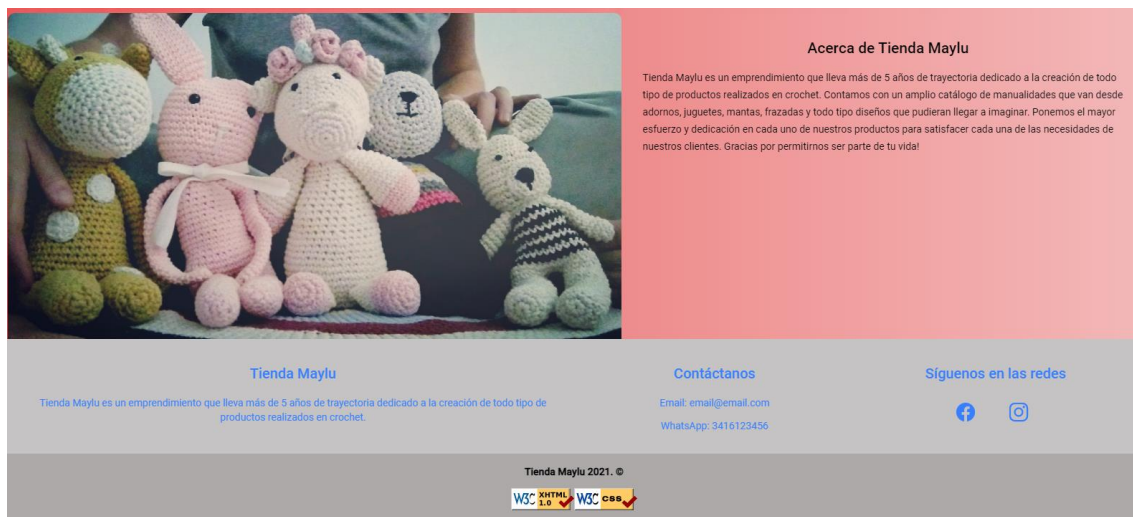
Filtrado de imágenes en la página principal



Selección de una imagen determinada



Página “Acerca de...”



Formulario de contacto

Tienda Maylu

INICIO CATEGORIAS ^ CONTACTO ACERCA DE...

INGRESAR REGISTRARME

Consulta nuestro productos

Nombre

Email (Opcional)

Asunto

Comentario

ENVIAR

Tienda Maylu

Tienda Maylu es un emprendimiento que lleva más de 5 años de trayectoria dedicado a la creación de todo tipo de productos realizados en crochet.

Contáctanos

Email: email@email.com
WhatsApp: 3416123456

Síguenos en las redes

Tienda Maylu 2021. ©

Error en el envío de los datos del formulario

Tienda Maylu

INICIO CATEGORIAS ^ CONTACTO ACERCA DE...

INGRESAR REGISTRARME

Consulta nuestro productos

Nombre

Este campo es requerido

Email (Opcional)

Asunto

Este campo es requerido

Comentario

Este campo es requerido

ENVIAR

Tienda Maylu

Tienda Maylu es un emprendimiento que lleva más de 5 años de trayectoria dedicado a la creación de todo tipo de productos realizados en crochet.

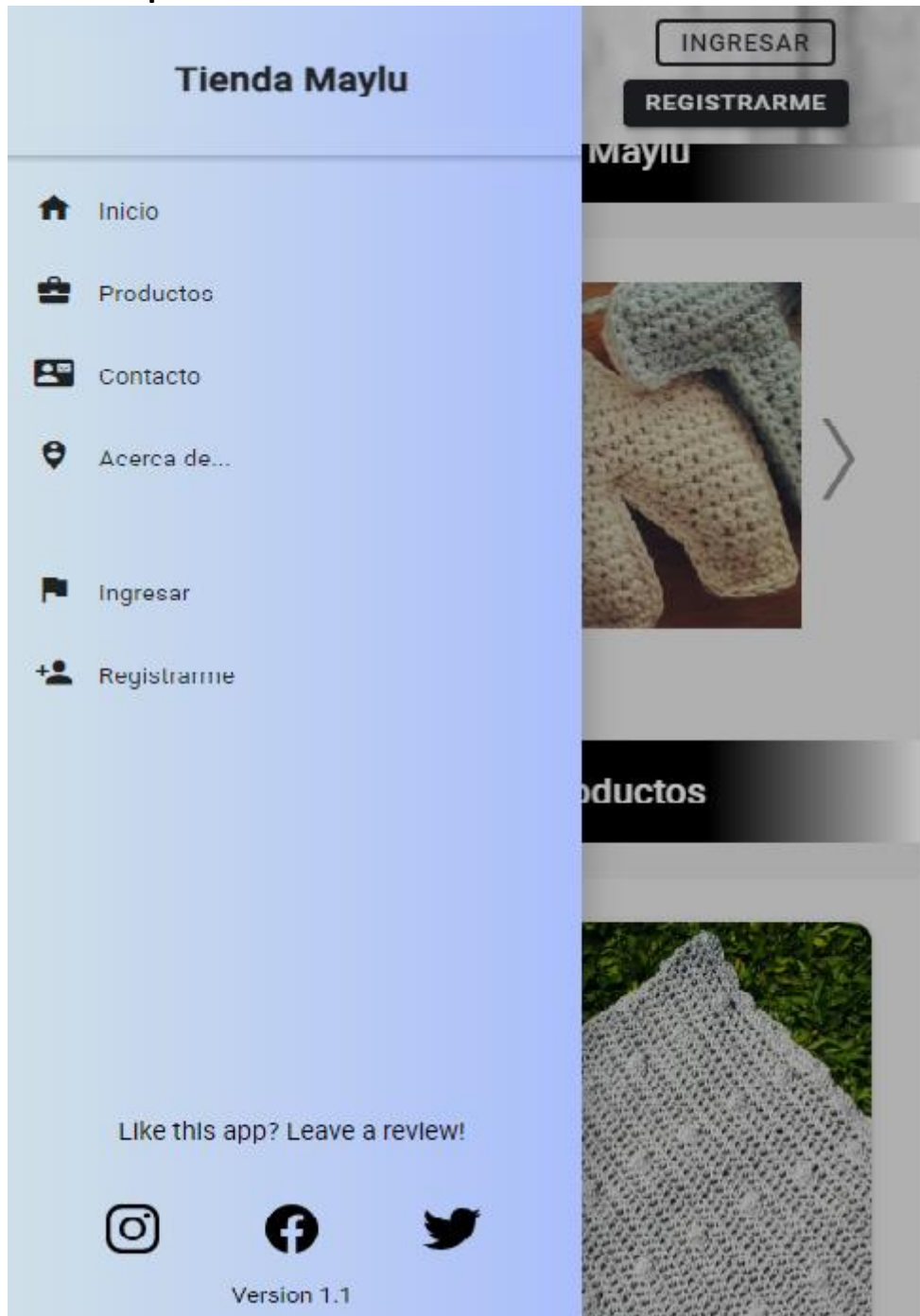
Contáctanos

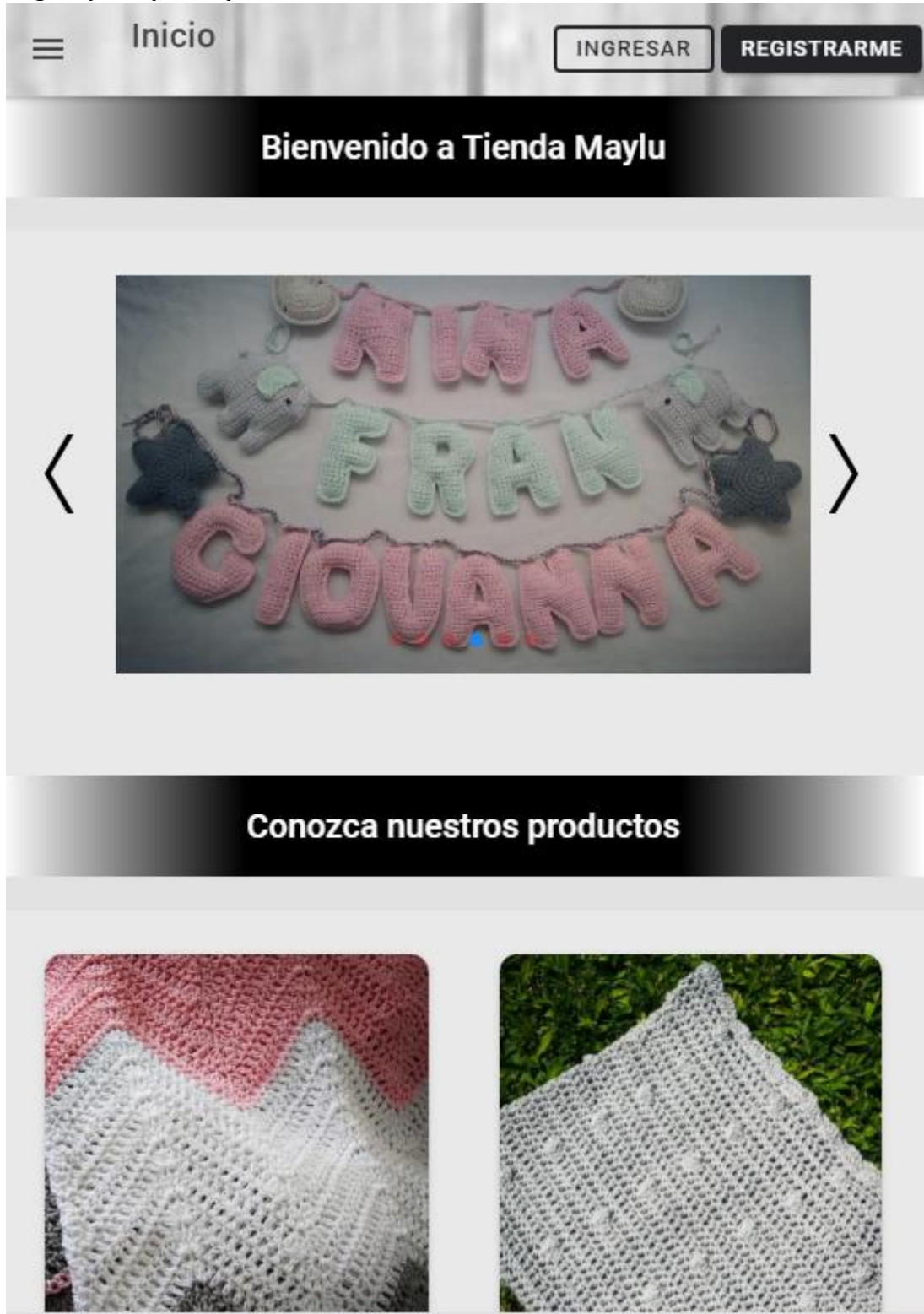
Email: email@email.com
WhatsApp: 3416123456

Síguenos en las redes

Tienda Maylu 2021. ©

Menú responsivo



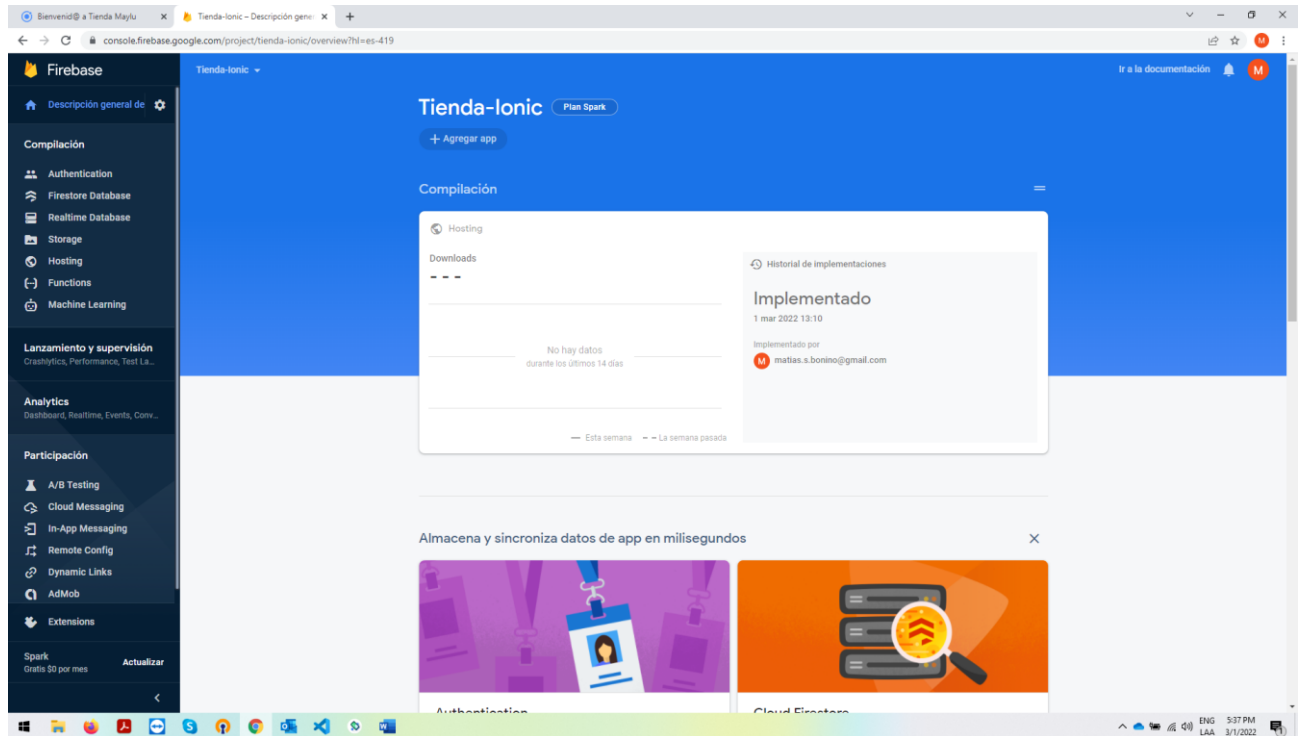


Hosting / Deployment:

Se realizó el “deploy” del sitio con la herramienta de Firebase Hosting. Se creó la aplicación en Firebase y nos asignó una url para acceder al sitio:

<https://tienda-ionic.web.app/>

Este servicio es gratuito, de querer tener una url más personalizada se incurriría en otros costos.



Web Analytics:

A su vez, con la integración de Firebase al proyecto, nos da la posibilidad de integrar otras herramientas de forma gratuita como por ejemplo, Firebase Analytics.

Una breve descripción tomada del sitio:

“Google Analytics es una solución de análisis ilimitada y gratuita que ocupa un lugar central en Firebase. Analytics se integra a distintas funciones de Firebase y te proporciona una capacidad ilimitada de generar informes sobre un total de hasta 500 eventos distintos que puedes definir con el SDK de Firebase. Los informes de Analytics te permiten entender claramente cómo se comportan tus usuarios para que puedas tomar decisiones fundamentadas en relación con el marketing de tu app y las optimizaciones del rendimiento.”

Como ésta, también podrían integrarse herramientas de Testing, Mensajería o Autenticación, entre otras.

Documentación de las tecnologías utilizadas:

Angular: <https://angular.io/docs>

Ionic: <https://ionicframework.com/>

Angular Material: <https://material.angular.io/>

Laravel: <https://laravel.com/docs/8.x>

Consola Firebase: <https://firebase.google.com/docs?authuser=0&hl=es>

Estimación del tiempo del proyecto:

Fases	Inicio	Duración (días)	Fin
Requerimientos	01/04/21	5	05/04/21
Análisis	08/04/21	3	11/04/21
Diseño	12/04/21	25	06/05/21
Implementación	07/05/21	10	17/05/20
Pruebas	18/06/21	2	20/05/21

Total Hs/Días	6
Total Días	45
Total Horas	270