



Nombre de los participantes con su usuario de GitHub

Participantes	Usuario de Github
Valentina	Valentina96-ER
Mateo de los Santos	mateodlso
Sofía Hernández	sofiahernandezas
Oswaldo Castro	osvaldocastroo
Carol Castañeda	Carol-Castaneda

Rol de cada participantes en la entrega

Para esta entrega, trabajamos en conjunto, en diversas instancias, principalmente mediante pantalla compartida. Un/a integrante escribía el código y los aportes del grupo, mientras compartía pantalla. En la medida en la que fueron surgiendo problemas, se fueron buscando soluciones colaborativamente.

Por tanto, todos/as asumimos diversos roles, es decir, diseñamos en conjunto tanto los prototipos como los códigos, y los estilos del login y la lista de productos. Asimismo, nos repartimos los códigos para poder subirlos al GitHub, de forma de que todos/as pudiéramos comenzar a comprender y dominar la aplicación.

Se agrega, por una parte, que el repositorio que usamos fue compartido por Valentina, y por otra parte, quien asumió el rol de la grabación del video fue Oswaldo mientras que quien lo editó fue Mateo.

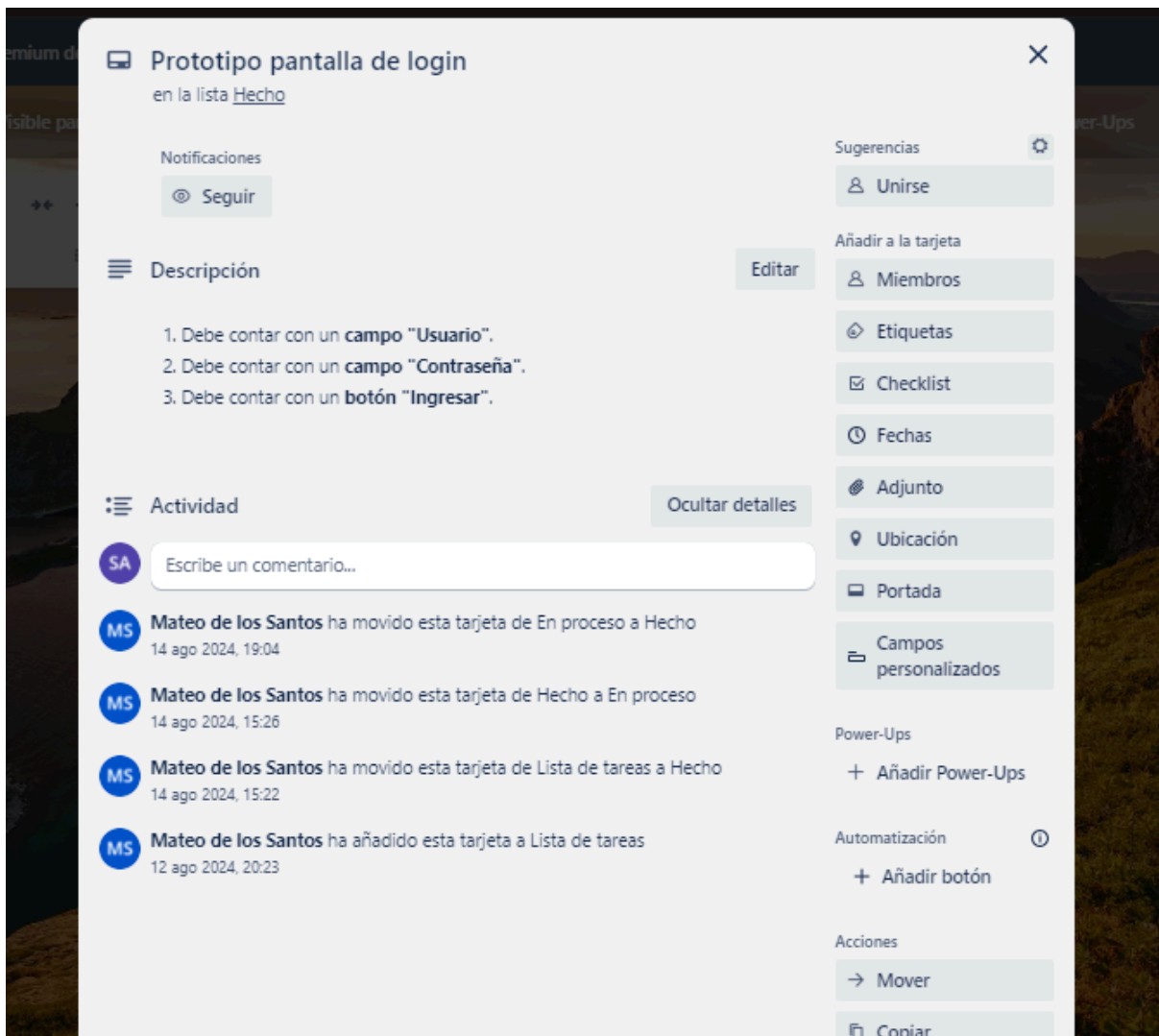
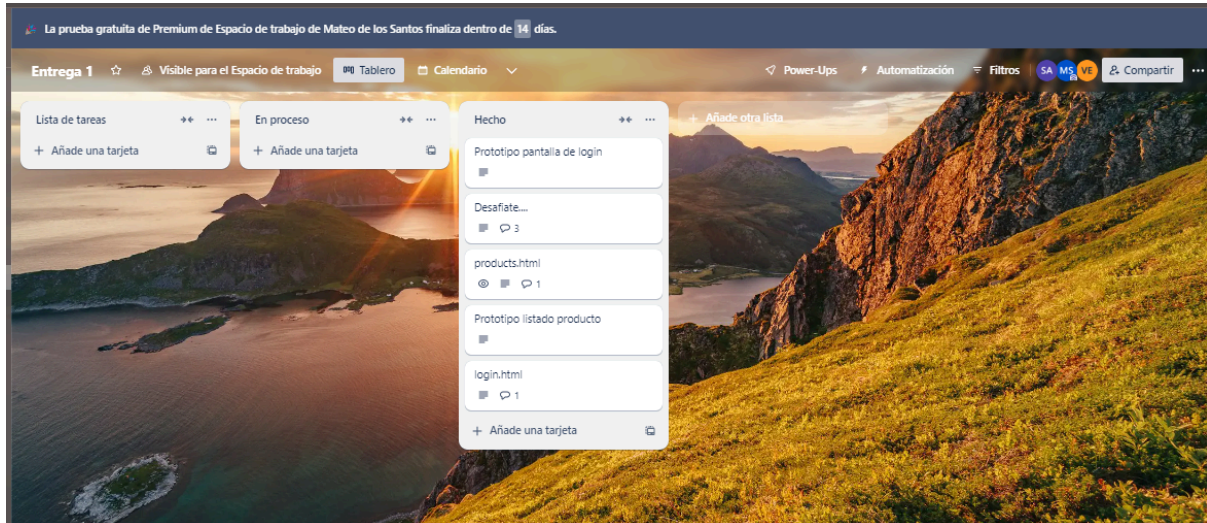
Enlace al repositorio

El enlace del repositorio está localizado en [Valentina96-ER/Entrega-1: Entrega 1 JAP Repositorio \(github.com\)](https://github.com/Valentina96-ER/Entrega-1: Entrega 1 JAP)



Link al tablero de Trello

[Link Trello / entrega-1](#)





Desafiate....

en la lista [Hecho](#)

Notificaciones

Seguir

Descripción

Editar

Investiguen la manera de guardar la sesión al momento de autenticarse e implementenla, y redireccionen a `login.html` **sólo cuando no se ha iniciado sesión.**

Actividad

Ocultar detalles

SA

Escribe un comentario...

VE

Valentina Enciso

ha movido esta tarjeta de En proceso a Hecho

hace 1 hora

VE

Valentina Enciso

ha movido esta tarjeta de Hecho a En proceso

hace 1 hora

VE

Valentina Enciso

ha movido esta tarjeta de En proceso a Hecho

hace 1 hora

VE

Valentina Enciso

ha movido esta tarjeta de Lista de tareas a En proceso

21 ago 2024, 22:38

VE

Valentina Enciso

21 ago 2024, 22:37

Le agregue un id en index.html a este renglon

1

<!-- prueba para agregar login en nav-->

2

Logout

Sugerencias

Unirse

Añadir a la tarjeta

Miembros

Etiquetas

Checklist

Fechas

Adjunto

Ubicación

Portada

Campos personalizados

Power-Ups

Añadir Power-Ups

Automatización

Añadir botón

Acciones

Mover

Copiar



▼ Mercado ▼ Plantillas ▼ Crear

gratuita de Premium d

☆ Visible pa

a tarjeta

products.html

en la lista Hecho 🔒

Notificaciones

🔔 Siguiendo ✓

Descripción

Replicar prototipo

Peticion web a URL con coleccion de productos en formato JSON con la información (precio, nombre, descripción, cantidad vendidos e imagen) respectiva a cada producto.

Actividad

SA Escribe un comentario...

VE Valentina Enciso ha movido esta tarjeta de En proceso a Hecho

21 ago 2024, 22:38

SA Sofía Hernández Astigarraga ha movido esta tarjeta de Lista de tareas a En proceso

20 ago 2024, 9:57

SA Sofía Hernández Astigarraga 20 ago 2024, 9:57

En *products.html* solo modifiqué de la línea 41 a la 43, comentando el código:

```
<main class="pb-5 container">

1 <div _id="container" _="container">
2
3 _<!-- <div class="alert alert-danger text-center" role="al
4
```

Sugerencias

Unirse

Añadir a la tarjeta

Miembros

Etiquetas

Checklist

Fechas

Adjunto

Ubicación

Portada

Campos personalizados

Power-Ups

Añadir Power-Ups

Automatización

Añadir botón

Acciones

Mover

Copiar



login.html

en la lista [Hecho](#)

Notificaciones

Seguir

Descripción

Editar

1 //an products.html

2 <form id="login-form">

3 <h1>Login</h1>

4 <label for="username">Usuario:</label>

5 <input type="text" id="username" name="username">

6 <label for="password">Contraseña:</label>

7 <input type="password" id="password" name="password">

8 <input type="submit" value="Ingresar">

9 <p id="error-message" class="error"></p>

10 </form>

1 //an styles.css

2 /* ESTILOS DEL LOGIN*/

Mostrar más

Actividad

Ocultar detalles

SA

Escribe un comentario...

MS

Mateo de los Santos

14 ago 2024, 19:02

ya lo hice, dejo los codigos si lo quieren probar, lo hice en mi repertorio, no me acorde de abrir una branch por eso lo agrego aqui para que vean si les

Sugerencias

Unirse

Añadir a la tarjeta

Miembros

Etiquetas

Checklist

Fechas

Adjunto

Ubicación

Portada

Campos personalizados

Power-Ups

Añadir Power-Ups

Automatización

Añadir botón

Acciones

Mover

Copiar

Crear plantilla

Archivar

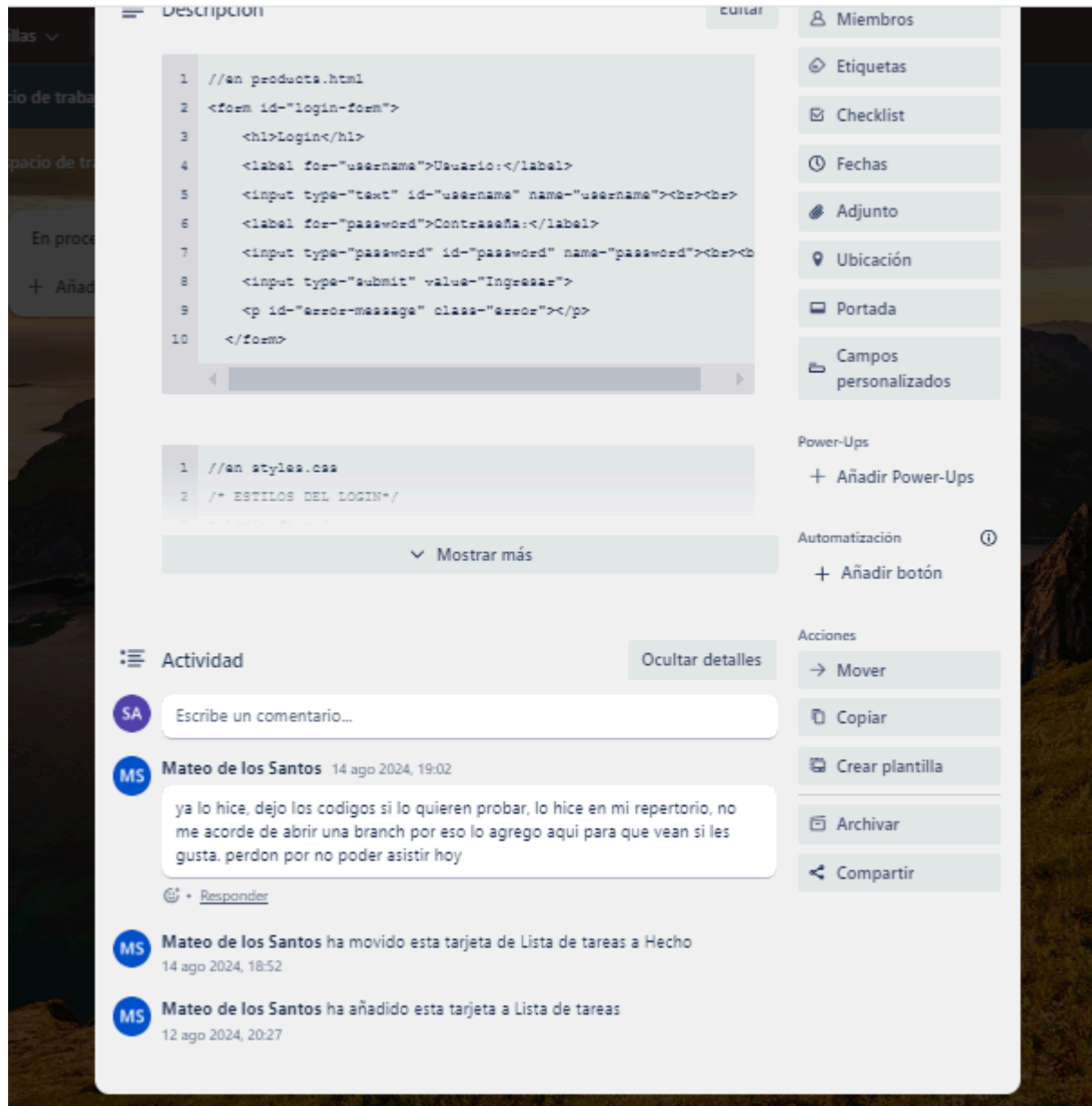


Diagrama y diseños solicitados

Prototipo Login:

<https://www.figma.com/proto/NqQuJpXeC4YFMH63YHjXDh/Login?node-id=0-1&t=t6Ux0sUkbaUhwNwq-1>

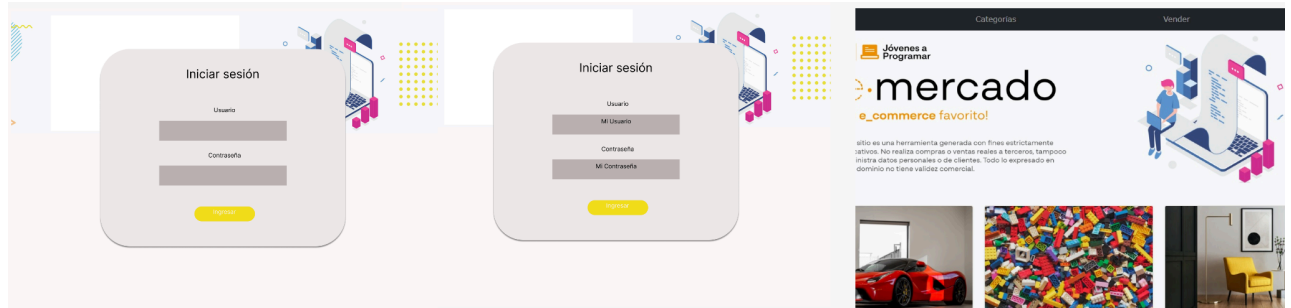
Acceso al Diagrama:

<https://www.figma.com/design/NqQuJpXeC4YFMH63YHjXDh/Login?node-id=0-1&t=t6Ux0sUkbaUhwNwq-1>

En el mismo archivo, pero páginas diferentes, encontrarán los prototipos de *login* y *products* respectivamente.



Login:



Products:

Desktop - 1

Autos

Explora nuestra amplia selección de autos.



Nombre

Cantidad de vendidos: XX

Descripción

Precio: USD XXXX



Nombre

Cantidad de vendidos: XX

Descripción

Precio: USD XXXX



Nombre

Cantidad de vendidos: XX

Descripción

Precio: USD XXXX



Transcripción del video

Introducción:

Para esta entrega trabajamos con Trello para organizar los puntos a trabajar; Figma, para diseñar los prototipos para la pantalla de login (punto 1 de la entrega) y la pantalla donde se visualiza un listado de productos (punto 2). Asimismo, sobre los archivos:

- Init.js
- Index.js
- Login.js
- Index.html
- Products.html
- style.css

Login:

En cuanto al login, creamos el siguiente prototipo en Figma, con las siguientes interacciones: (se muestra prototipo).

Mateo comenzó trabajando sobre style.css y el código, creando un formulario para el inicio de sesión, con dos input type text y sus respectivos labels, y un input del tipo submit para ingresar.

Posteriormente, en login.js, se accede al formulario y se agrega un evento para cuando se haga submit.

`e.preventDefault()`; es un método que previene el comportamiento predeterminado del formulario, que normalmente sería recargar la página al ser enviado.

Las líneas que refieren a las constantes usuario y contraseña, obtienen los valores ingresados por el usuario en los campos de texto con los IDs username y password.

El if verifica si los dos campos no están vacíos, y si se da submit, en ese caso mediante localStorage se guarda como verdadero IsLoggedIn, redirigiéndonos a index.html

Si alguno de los dos campos están vacíos, se ejecuta el bloque de código dentro del else. Aquí se mostrará un mensaje de error.

Después, en conjunto, trabajamos en el estilo del login.

En el style.css se trabajaron principalmente selectores que referían a los id y clases de los elementos, pero también se usaron pseudoclasses como el hover.



Products:

Para mostrar los productos, trabajamos en products.js. Sofia, se ayudó con ChatGPT para tener una estructura sobre la cual seguimos trabajando, primero Carol también con ayuda de ChatGPT, y después el grupo entero mediante pantalla compartida.

Mediante DOMContentLoaded hacemos que el evento se dispare cuando el documento HTML haya sido completamente cargado y procesado.

El fetch hace la solicitud a la API que devuelve un Json.

El primer then, convierte la respuesta de la API en un objeto JSON.

El segundo then llama a la función showData.

El catch, captura y muestra cualquier error que ocurra durante la solicitud o procesamiento de datos.

La función showData:

Selecciona el contenedor donde se mostrarán los productos, usando el ID container, y lo asigna a la constante container.

container.innerHTML = ''; Limpia el contenido anterior para asegurar que no hay duplicados al recargar

Luego se itera sobre cada producto de la lista, se aplica la función de los miles para mejorar la legibilidad, se genera una función que permite agregar un punto cada 3 ceros, y se crea el HTML necesario para mostrar los productos incluyendo: imagen, nombre, descripción, precio y cantidad vendidos.

Posteriormente, agrega el HTML del producto al contenedor.

Desafiate:

En el index.js, con código proporcionado inicialmente por Valentina, se trabajó en conjunto, con pantalla compartida, para unificar códigos y que quede de la siguiente manera:

Se usa de nuevo DOMContentLoaded para que el evento se dispare cuando el contenido HTML haya sido cargado.

El localStorage.getItem('isLoggedIn'): Verifica si en el localStorage del navegador existe un ítem con la clave 'isLoggedIn' y si su valor es 'true'.



El if, verifica si el valor de isLoggedIn es falso, y en ese caso, redirige al login.html

Document.getElementById('logoutButton'): Selecciona el botón de cierre de sesión usando su ID ('logoutButton'). y le añade un evento que se dispara cuando se hace clic en el botón de cierre de sesión.

localStorage.removeItem('isLoggedIn');: debería eliminar del localStorage el ítem 'isLoggedIn'.

Y con el window.location.href = 'login.html'; Después de eliminar el ítem, se redirige al usuario a la página de inicio de sesión (login.html). En el index.html, en el nav, agregamos una etiqueta con “Cerrar sesión” para acceder al login.html.