

PLAN DESARROLLO E-COMMERCE

Gesiel Gimenez
Camper

El liderazgo de tu organización pese a la incertidumbre

Cohorte Front-end 4

PLANTEAMIENTO

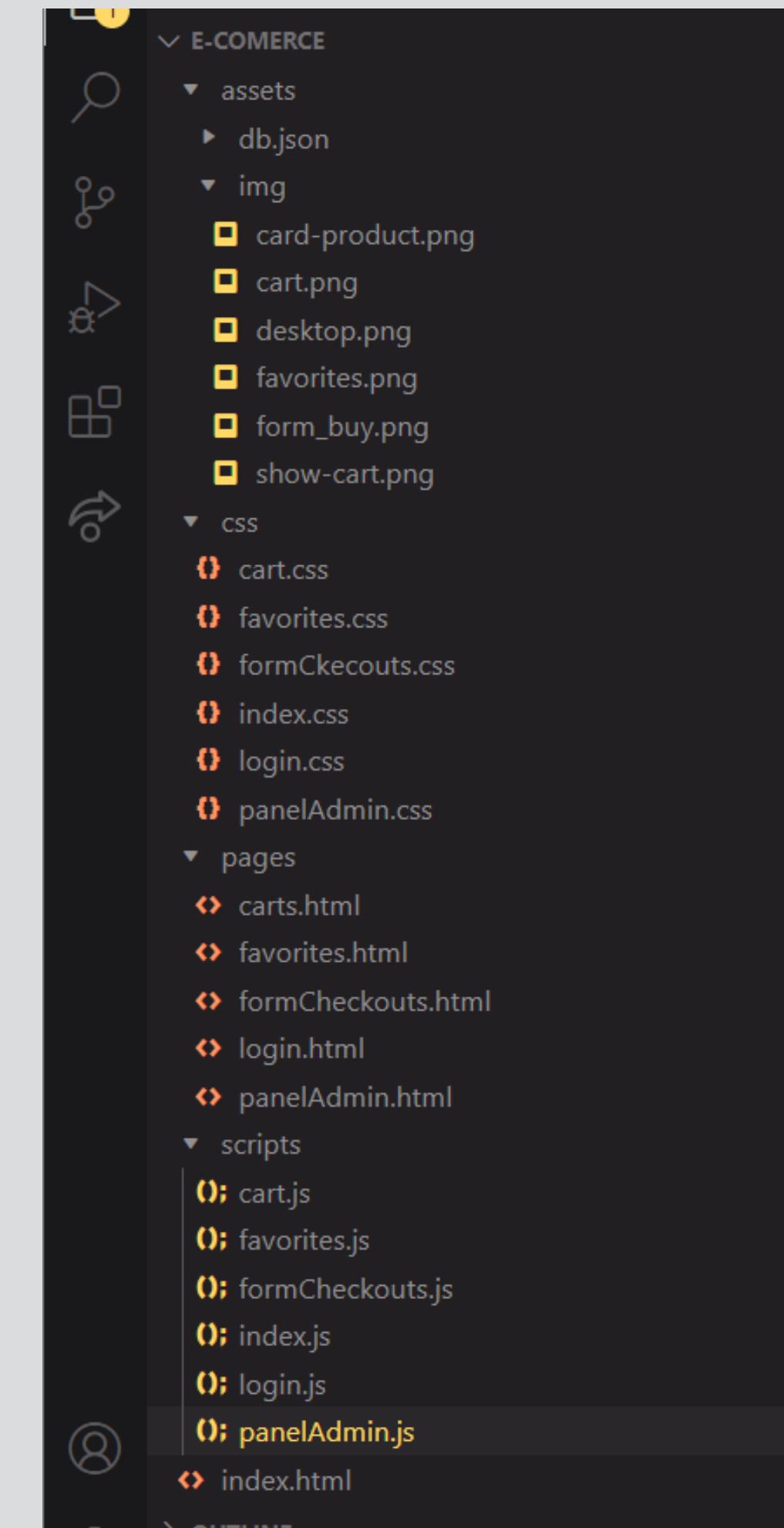
Se presenta un proyecto E-commerce que debe cumplir con los requisitos descritos a continuación

01 Estructura de carpetas y archivos

02 No dudes en escribir tus preguntas en el cuadro de conversación.

03 Las preguntas se responderán en la sesión de preguntas y respuestas al final de la presentación.

1. ESTRUCTURA DE CARPETAS Y ARCHIVOS



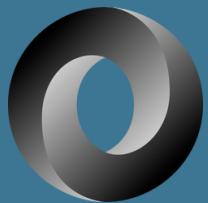
RECURSOS / LIBRERIAS



sistema de Gestión de paquetes
(por defecto del Node.js)



Cuadros emergentes
formulario



Json -Server

Formato de creación de información
(destinada al Storage)



GitHub

Repositorio

Alojamiento principal



Entorno de ejecución
Entrada y salida de datos



Mantener la calma y el optimismo.

Agrega audio o graba tu voz mientras hablas de la presentación y, luego, comparte el video con los demás.



Editor de Código

RESUMEN

● Pto 1

- Creación formato Json de los productos con sus propiedades.
- Implementar funciones que permitan solicitar recursos

● Pto 3

- Crear botones que ejecuten la función de filtrado por búsqueda de categorias.

● Pto 4

- Ejecutamos un evento por medio de Elemnt.getAttribute que sera almacenado en el sessionstorage

● Pto 6

- Ejecutamos la función Fetch, Delete para eliminar un recurso.

● Pto 7

- Ejecutamos la función Fetch, post para crear un recurso

RESUMEN

● Pto 10

-Creamos un formulario que se muestre luego del evento click sobre "process to checkouts.

● Pto 12

-Crear un array con el objeto compra que se guardará en el Json Server (lista de productos)

● Pto 13

-Crear página donde se ejecuten los verbos get (ver productos) , put (modificar) , delete (borrar) , post (crear).

ARROW FUNCTIONS

Funciones flechas: permiten que el código sea más legible

EVENTOS

Notificaciones de características que van a ocurrir

PETICIONES HTTPS

Protocolos de ejecución

Get: ver recursos

Post: crear recursos

Put: modificar recursos

Patch; actualiza una SECCIÓN ESPECÍFICA de un recurso.

Delete: elimina un recurso

CONDICIONALES

Estructuras que basan su resultado en la comparación de una variable con otros valores

CONDICIONALES

Estructuras que basan su resultado en la comparación de una variable con otros valores

ASYNC/ AWAIT THEN O THEN().CATCH

Estructuras que basan su resultado en la comparación de una variable con otros valores

JSON SERVER

Formato de estructura que almacena información, Actúa independiente al lenguaje Javascript

LOCALSTORAGE

Propiedad de sólo lectura que permite que los datos almacenados allí no tengan fecha de expiración