

Final: Trabajo práctico domiciliario

Consignas para la entrega:

- Para la entrega del final, los archivos del desarrollo se deben entregar comprimidos en un único archivo de extensión .ZIP o .RAR, respetando el siguiente formato para asignarle el nombre de archivo:
 - Si la entrega es individual:
 - **p1-final-dw2-[año cursado]-[cuatrimestre cursado]-[apellido]-[nombre].[zip o rar]**
 - TODO EN MINÚSCULAS Y UTILIZANDO EL GUIÓN MEDIO COMO SEPARADOR.
 - Ejemplo:
 - **p1-final-dw2-2023-2-villafane-alejandro.zip**
 - ***En caso de incumplir con el formato del nombre, se restará automáticamente dos puntos, pudiendo desaprobar por este error el final.***
- La entrega será mediante la **INTRANET** de la Escuela, una carpeta de Drive que figurará junto al final en la mesa que se haya anotado para rendir el final.
- Los archivos utilizados para el desarrollo deberán estar ordenados bajo una arquitectura de directorios clara y prolija.
- Además, se deberá adjuntar dentro del archivo comprimido, un archivo llamado alumnos.txt, con los siguientes datos personales: Alumnos/as, cuatrimestre, comisión, turno, materia y profesor.
- **Fecha de entrega:**
 - La fecha y hora de entrega será la que indique la mesa del final en la **Intranet**.
 - Ese día el examen será corregido en vivo junto al/a la alumno/a, que deberá defender su examen, justificando o explicando los puntos débiles del desarrollo.
 - Si debe rendir dos finales el mismo día/hora, acérquese a una de las mesas a dar el presente a fin de informar al docente.

Introducción

- Se debe presentar una aplicación mobile que cumpla la función de un carro de compras que conste de:
 - Navegación de categorías.
 - Catálogo de productos.
 - Agregar y quitar elementos de la compra.
 - Cierre de la compra solicitando datos del pago y del cliente.
 - Contenido 100% navegable offline.
- El trabajo es individual desarrollando una maqueta HTML válida, **implementando las tecnologías y conceptos dictados en el transcurso de la cursada de la materia**. A tal efecto, se aplicará Javascript con manejo de DOM y objetos nativos de ECMAScript.

Sobre el diseño

- Serán desaprobadas las entregas que carezcan de diseño o que el mismo no se visualice de manera apropiada en los navegadores de uso común.
- Debe respetarse una interfaz navegable y comprensible, donde el usuario pueda interpretar el proceso completo de la compra.
- Se debe usar una o más hojas de estilo, a elección de cada estudiante.

Sobre la programación e interfaz del sitio

- Todas las entregas deberán cumplir, como mínimo, con los siguientes requisitos para considerar que han aprobado la evaluación:
 - La interacción de todo el proceso de compra (navegar categorías, ver producto, resumen de compra y checkout) cada uno programado en un documentos HTML distinto.
 - Se deben aplicar las técnicas de manejo de DOM dictadas durante la cursada, haciendo especial implementación en la creación y eliminación de objetos, alteraciones de CSS, etc.
 - Debe poseer un catálogo de productos que pueda filtrarse por categorías.
 - Los productos deben estar divididos, como mínimo, en tres categorías diferentes.
 - En la grilla de productos, mostrar:
 - Nombre.
 - Descripción.
 - Precio.
 - Una imagen en miniatura.
 - Categoría.
 - Debe haber un área del sitio que muestre (mientras el usuario navega por el sitio) la cantidad de productos comprados y el monto a pagar, que al hacerle click debe mostrar TODOS los productos comprados con el subtotal por cada producto y el total de la compra.
 - Los productos deben quitarse, ya sea de uno en uno, así como todos los que se han comprado.
 - Debe haber un botón que indique el final de la selección de productos para dar paso al proceso de ingreso de datos de envío y pago de la compra (también llamado checkout).
 - En el checkout se debe solicitar la información del cliente (nombre, teléfono, email, lugar y fecha de entrega) y del pago (método de pago, cuotas –si corresponde–). Este proceso debe tener la posibilidad de cancelarse para seguir agregando o eliminando productos del carro.
 - Los listados de productos y de cualquier información que se use deben ser almacenados o bien en **localStorage** o en **IndexedDB**.
 - Se deben usar **todos los métodos de localStorage** convenientemente.
 - Para **IndexedDB** también usar **todos los métodos posibles**.
 - Se **debe usar ambos** localStorage e IndexedDB.
 - Se debe implementar **fetch** para solicitar los datos inicialmente, después los datos deben estar todos cacheados.
 - Caché dinámico offline first. Precaching.
 - Manifest completo y bien configurado
 - Service Worker (registro, instalación, activación)

- Para sumar nota, solo en aquellas entregas que hayan cumplido con las condiciones mínimas, las entregas deberán:
 - Ver detalles del producto en una página aparte.
 - Si ofrece más de una foto (a modo de galería de fotos), mejor.
 - Si se selecciona más de una vez el mismo producto debe aparecer una única vez en el detalle de la compra, indicando la cantidad de veces que se compró cada producto.
 - Cada vez que se cambia de categoría debe aparecer una oferta especial que se destaque durante 10 segundos a modo de banner flotante, luego esta oferta DEBE DESAPARECER.
 - Hacer que el banner rote aleatoriamente cada vez que se cambia de categoría y ofrecer alguna interacción (ver o agregar un producto, ir a una categoría, etc.).
 - En los datos del cliente, hacer que algunos datos sean obligatorios. La única validación solicitada es que los datos no estén vacíos.
 - Destacar visualmente los campos que no pasen satisfactoriamente la validación.
 - Desarrollar funciones constructoras para los elementos de los cuales se deban crear instancias.
 - Detección de teclado para los casos donde pueda implementarse (por ejemplo, cerrar la ventana modal al usar la tecla escape, o moverse a la imagen anterior/siguiente con las flechas). Se considerará cumplido este punto si se utiliza en dos ocasiones distintas.
 - El uso de localStorage.
 - Resetear el carrito al finalizar la compra.
 - Mostrar un cartel de "Estas sin conexión" cuando la app se desconecta y "Vuelves a tener conexión" cuando se reconecte.
 - Notificaciones. Preguntar al usuario si acepta recibir notificaciones. Implementar al menos 1 notificación en código. *(Si esta implementación no llegara a funcionar completamente porque se requiere un servidor de HTTPS, pero solo por este motivo, no se descontarán puntos).*
- Por otra parte, sufrirán una pérdida importante de nota las entregas que:
 - Ya tengan TODO el HTML armado y se limiten a cambiar el display (block/none) de los objetos:
 - El final es de DOM, por lo cual la grilla de productos, el detalle de cada producto, la ampliación del detalle, el detalle del carrito de compras y los datos del usuario para el cierre de compra, SE DEBEN CREAR recién cuando sean solicitados.
 - El resto de la interfaz puede crearse desde HTML.
 - Generen los elementos mediante el uso de innerHTML, se deben usar funciones nativas del manejo de DOM para crear, agregar y eliminar los elementos.
 - Sólo se permite el uso de innerHTML para definir el texto interno de las etiquetas (mientras no contenga código HTML).
 - Solo usen 1 o 2 métodos de localStorage y/o IndexedDB
 - Que solamente hagan un console.log / console.error en los eventos de IndexedDB (success, error, complete, etc.)
 - Muestran botones o links que al clickear no realizan ninguna acción porque no está programada o no funcionan por algún error.

- Serán reprobadas automáticamente, sin posibilidad de defensa, las entregas que:
 - No sean descargables y/o instalables.
 - No captureen datos y/o archivos.
 - Hagan uso de ventanas obstrusivas: alert, prompt y confirm. **Todo** debe realizarse desde DOM.
 - Solo sean SPA (Single Page Application) y no varias páginas navegables entre sí.
 - Tengan eventos inline desde alguna etiqueta del HTML.
 - Cualquier implementación de Frameworks que no sea Bootstrap.
 - **Soluciones generadas con inteligencia artificial y/o copias de internet, sean fragmentos o soluciones completas. Solo se aceptan proyectos originales desarrollados por ustedes.**

Evaluación

- Se va a evaluar en primera instancia la funcionalidad del programa y su aspecto más técnico, la codificación.
- El código resultante deberá ser acotado, minimizando el uso de variables y optimizando su modificación.
- En segunda instancia se tendrá en cuenta el proceso de compra del usuario. Y, por último, el diseño y navegabilidad de la interfaz.
- La defensa del final es presencial y el docente puede realizar preguntas sobre el desarrollo del programa, así como pedir que se modifiquen determinadas líneas de código a fin de evaluar el criterio de programación del alumno y su autoría.
- De no poder defender o explicar correctamente su entrega, habrá desaprobado la misma.
- El trabajo debe cumplir con TODAS las pautas solicitadas, la ausencia de alguna es motivo suficiente para considerar la entrega como desaprobada.