

## Consignas de entrega

Para la entrega del presente parcial, se debe crear un repositorio en GitHub respetando ciertas nomenclaturas y pautas:

- El repositorio debe ser personal y tener el nombre: `p1-parcial-1-dw[m|t|n]2[a|b|c]`.
- Dentro del `README.md` deben figurar todos los contribuyentes del trabajo (máximo dos persona).
- Se hará una entrega a la tarea en **Classroom** con el link al repositorio, **no a una carpeta de Drive**. No se aceptará ningún otro tipo de entrega.
- **En caso de incumplir con el formato del nombre, se restarán automáticamente dos puntos, pudiendo desaprobado por este error el parcial.**
- En el repositorio encontraran un `template.md` que tiene lo que debe contener el `README.md` que publiquen en su repositorio personal. A la hora de subirlo, cambien su nombre por `README.md` y completen con los datos que solicita.

Fecha de entrega:

- La fecha de entrega será la que se indique en la actividad del **Classroom**.

## Consigna del trabajo

Se le pide crear un programa usando código Javascript que permita realizar las funciones básicas front-end de un sistema de "carro de compras" que conste de:

- Catalogo de productos.
- Detalle del producto.
- Agregar o quitar elementos de la compra.
- Detalle del carrito

El catálogo de productos, el detalle del producto y el detalle del carrito de compras deben cargarse dinámicamente, no puede estar ya en el HTML. La interacción de todo el proceso de compra (detalle del producto, detalle del carrito) debe programarse en un **único documento HTML**, se recomienda el uso de ventanas modales.

Se deben aplicar las técnicas de **manejo de DOM**, haciendo especial implementación en la creación y eliminación de objetos, alteraciones de CSS, etc. Este parcial es **DOMICILIARIO, INDIVIDUAL o GRUPAL DE A 2 (DOS) PERSONAS**.

De no aprobarse/presentarse la entrega, se pasará a un recuperatorio al final del cuatrimestre.

Este parcial es la base de lo que será el examen final de la cursada.

Cada alumno debe comenzar con el contenido de este repositorio:

- `index.html`: contiene la estructura básica del parcial, incluidas las referencias para el desarrollo del parcial. Se puede modificar libremente.
- `styles.css`: contiene los estilos para darle un mínimo de "diseño" al parcial. Se puede modificar libremente.
- `index.js`: contiene algunas líneas de código para tomar como base para el desarrollo. Puede comenzar con un código desde cero, no es obligatorio partir de la referencia.
- `favicon.ico`: un favicon para mostrar en el navegador. Se puede cambiar libremente.
- `template.md`: es un template de Markdown para el `README.md` que tiene que contener el repositorio que entreguen.

## Desarrollo mínimo y obligatorio (nota 4)

Cualquier faltante de los requisitos que se presentan a continuación, hará que la entrega se considere desaprobada, aunque haya hecho cosas no solicitadas o del punto siguiente. Esta condición es mínima e innegociable.

- Productos:
  - Armar el listado de productos obteniendo la información de cada uno de un Array de objetos. Como mínimo 6 productos.
  - Se deben contemplar como mínimo 3 categorías de productos.
  - Cada producto debe tener como mínimo: Nombre, descripción, precio, una imagen y categoría. Toda esta información se debe mostrar por cada ítem.
  - Debe tener un botón que permita agregar el ítem al carrito.
- Carrito:
  - Debe haber un área del sitio que muestre (mientras el usuario interactúa con el sitio) la cantidad de productos agregados al carrito y el monto total a pagar.
- Clases:
  - Debe implementarse al menos una clase para resolver el trabajo.
- DOM:
  - El manejo de DOM es obligatorio. La creación de elementos de HTML y aplicación de estilos debe ser dinámica a través de los métodos del document (createElement, append, prepend, setAttribute, before, after, parentElement, etc).

## Agregados para llegar a 10

Entre las entregas que cumplan el punto anterior, incrementarán su nota los desarrollos que:

- Productos:
  - Armar la ventana modal del detalle del producto, donde se podrá ver la información en detalle del producto.
  - Esta ventana debe tener un botón que permita agregar el ítem al carrito. Programar dicho botón, para que vaya agregando el ítem al carrito.
  - Debe tener un botón que permita cerrar la ventana modal. Programar dicho botón, para que remueva la ventana modal.
- Carrito:
  - Armar la ventana modal del detalle del carrito, que debe listar los productos agregados.
  - Debe mostrar la cantidad de productos agregados y el monto total.
  - Debe tener un botón que permita cerrar la ventana modal (programarlo).
  - Debe permitir eliminar cada producto del carrito así como la posibilidad de eliminar todos los productos del carrito.
- General:
  - Se deben poder filtrar los productos por categoría.
  - Se valorará la modificación o agregado significativo de HTML / CSS al sitio.

## Consideraciones finales

- Se descontarán puntos si la entrega ya tiene TODO el HTML armado y se limitan a cambiar el display (block/none) de los objetos. El parcial es de DOM, por lo cual la grilla de productos, el detalle de cada producto, el mini carrito y el detalle del carrito de compras, **se deben crear** recién cuando sean solicitados.
- El resto de la interfaz puede crearse desde HTML.
- Esta prohibido generar los elementos mediante el uso de `innerHTML`. Se deben usar funciones nativas del manejo de DOM para crear, agregar y eliminar los elementos. De no ser así, se descontaran puntos.
- Solamente está permitido el uso de `innerHTML` para definir el texto interno de las etiquetas (mientras no contenga código HTML).
- Se descontarán puntos por errores semánticos en el uso de los elementos HTML.
- Se descontarán puntos por el uso incorrecto de las estructuras de programación.
- La lógica del programa también será evaluada.
- Se puede utilizar la base del diseño que se está viendo en otras materias.
- Está permitido el uso de frameworks de CSS, como Bootstrap.