

Condiciones generales:

Se debe presentar un sistema de compra on-line (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.

El trabajo es **individual**, desarrollando una maqueta HTML válida (bajo el DOCTYPE que el alumno desee), 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.

El objetivo principal de este trabajo es lograr la solución interactiva más creativa, que se tendrá en cuenta en la evaluación del trabajo final, respetando las siguientes consignas:

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 (Internet Explorer, Firefox, Chrome)
- Debe respetarse una interfaz navegable y comprensible, donde el usuario pueda interpretar el proceso completo de la compra.
- El código HTML, debe ser válido según la normativa de la W3C.
- Se debe usar una o más hojas de estilo, a elección del alumno. No se pide validar el código CSS.

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) debe programarse en un único documento HTML.
- 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.
- El listado de productos debe obtenerse de un ARRAY multidimensional o de Objetos OBJECT.
- 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.
- En la grilla de productos, mostrar: Nombre, descripción, precio y una imagen en miniatura.
- 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.

Esto es una especie de 'minicarrito' que al hacerle click debe mostrar TODOS los productos comprados con el subtotal por cada producto y el total de la compra.

Esto es un resumen que NO DEBE mostrar cada producto agregado (eso se pide más adelante).

- Los productos deben quitarse, ya sea de a uno como todos los que se han comprado (ambas).
- 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.
- Debe funcionar correctamente en Internet Explorer (el rango a tener en cuenta debería desde la versión 10 u 11). Dejar constancia en un TXT de la versión en que fue probado.

Para sumar nota, solo en aquellas entregas que hayan cumplido con las condiciones mínimas, las entregas deberán:

- Ampliar la imagen de la grilla de productos al hacer click o mouseover (por ej. MercadoLibre). No aplica el cambio de width/height de la imagen, se debe crear una imagen nueva.
- Si el ampliar la imagen ofrece más de una foto (a modo de galería de fotos), mejor.
- Si el ampliar la imagen, ofrece más datos que sólo ver la foto, mucho 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.
- 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 (con que no sea comilla-comilla, alcanza).
- Destacar visualmente los campos que no pasen no pasen satisfactoriamente la validación.
- Desarrollar funciones constructoras para los elementos de los cuales se deban crear instancias.
- Usar temporizadores –Intervalos y TimeOut– para suavizar los efectos y/o generar transiciones (no cuenta el banner, que se debe quitar a los 10 segundos).
- 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.
- Resetear el carrito al finalizar la compra.

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 de la imagen, el mini carrito, 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), aunque se recuerda que debería crearse como nodo de texto.

Serán reprobadas automáticamente, sin posibilidad de defensa, las entregas que:

- Utilicen más de un documento HTML para desarrollar el final.
- Hagan uso de ventanas obstrusivas: alert, prompt y confirm. TODO debe realizarse desde DOM.
- Tengan eventos inline desde alguna etiqueta del HTML.
- Implementen Frameworks, ya sea en Javascript o HTML (prototype, jQuery, sass, bootstrap, etc).
El uso de algún Framework para la manipulación de grillas en CSS (960, bootstrap) es aceptado.

DW2 - Programación I

Fecha de entrega y Defensa:

El trabajo se debe enviar por email dos días hábiles antes de la mesa en que el alumno se haya anotado.

Ese día, el PROFESOR de la materia recibirá en su casilla un zip o rar con los archivos correspondientes al desarrollo. Tanto el asunto como el archivo adjunto deben respetar la siguiente consigna:

p1_final_año_apellido_nombre

Por ejemplo: p1_final_2016_belgrano_manuel

Los horarios destinados a la recepción del email serán dos días antes de la mesa ÚNICAMENTE:

- De 11.00 a 11.30 para mesas del turno mañana.
- De 19.00 a 19.30 para mesas del turno tarde y noche.

La casilla del docente es federico.noto@davinci.edu.ar

FUERA DE ESTE HORARIO, NO SE RECIBIRÁN ENTREGAS DE ALUMNOS, DANDO LUGAR A UN AUSENTE EN LA MESA DE FINAL SIN EXCEPCIONES.

Luego, el día que figura en el DvPanel como mesa de final, el alumno asistirá exclusivamente a **defender** su examen, donde el docente ya tendrá las correcciones pertinentes y el alumno justificará o explicará los puntos débiles del desarrollo.

Esta norma tiene como objetivo agilizar el proceso de corrección (evitando que los alumnos deban esperar un largo rato para recibir su corrección) y ofrecerle al alumno una devolución de errores más detallada y fiel a su entrega.

De esta manera, si una mesa de final se realiza –por ejemplo– el Jueves 5 a las 19:00hs, el alumno deberá:

- Enviar por email el Martes 3 a las 19:00hs con su respectiva entrega.
- Presentarse el Jueves 5 a las 19:00hs para recibir su corrección y defender la entrega.

Se da por notificado y entendido que el alumno que no realice la entrega previa NO GOZARÁ del derecho de final, así como el alumno que no se presente a la DEFENSA no será aprobado aunque haya entregado un final acorde a la presente pauta. Se recuerda que la tolerancia de presentación a la mesa de FINAL es de 45 minutos de iniciada, luego el alumno será considerado AUSENTE. 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.

Formato de entrega:

El trabajo debe presentarse en CD/DVD debidamente rotulado (Alumno, cuatrimestre, comisión, turno, materia y profesor), dentro estarán los archivos utilizados para el desarrollo, ordenados bajo una arquitectura de directorios clara y prolija.

Se recuerda que el archivo principal debe llamarse **index.html**.

Además se deberá adjuntar dentro del disco, un archivo llamado **alumno.txt**, con los mismos datos pedidos para el rótulo del disco.

IMPORTANTE: Los datos del alumno deben estar escritos en el DISCO, siendo esto más importante que la cajita en que se entrega. Si un disco no posee nombre, apellido y final, será rechazado teniendo AUSENTE en la respectiva mesa

DW2 - Programación I



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.

Aquel alumno que no pueda defender o explicar 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.