DWM3A PROGRAMACIÓN II
PROFESORA: PAMELA IGLESIAS

FECHA DE ENTREGA: **48HS ANTES DE LA MESA**

FORMATO DE ENTREGA: RAR - ZIP

Objetivos globales:

El alumno debe desarrollar un sitio web con e-commerce aplicando las tecnologías **HTML, CSS, JS, PHP y MySQL.**

El sitio a realizar es a libre elección. **Se recomienda usar el mismo proyecto de los parciales 1 y/o 2.**

La entrega deberá acompañarse de su respectiva Base de Datos, **contando con información precargada para poder ser evaluada ya funcionando**.

Adicionalmente, este sitio debe contar con un panel de control que permita **manipular la información que se encuentra en dicha base de datos**.

Se recuerda que la entrega del parcial debe hacerse **48hs antes de la fecha de la mesa para poder realizar una mejor devolución del trabajo**.

Este archivo lo van a entregar subiendo el archivo al sistema de entregas propuesto por la escuela desde el panel de Da Vinci, 48hs antes de la mesa hasta las 11hs

El sitio web deberá estar formado por las secciones necesarias para completar el proceso de adquisición de un producto/servicio:

- Listado de productos/servicios
- Detalle del mismo
- Botón para agregarlo al carrito (solo estando logueado)
- Carrito con productos/servicios
- Detalle de la adquisición
- Pago de los productos/servicios

El panel de control debe estar formado por:

- Un listado de los datos.
- Un formulario para dar de alta/editar un nuevo registro.
- Un botón para eliminar un registro.

Todos los sitios deberán contar con por lo menos una función programada en PHP (**HECHA POR USTEDES**).

Deberán tener los errores habilitados desde la programación, tanto para el debug por parte de los alumnos como la correcta corrección por parte del profesor.

Sobre la web:

El sitio tiene que tener un **diseño acorde a la temática seleccionada**, no puede ser un sitio pelado

Sobre la base de datos:

Deberá entregarse la base de datos que usará el sistema para manipular la información. Se pide la entrega del **DER y los códigos SQL de la misma**.

El código SQL debe ser la **réplica exacta del diagrama**, no pudiendo haber entidades o atributos de más ni de menos en el Script de creación.

Sobre el DER:

Deberá tener todas las entidades necesarias para satisfacer el modelo prometido.

El diagrama deberá estar normalizado en su totalidad y los tipos de datos de cada atributo deben estar optimizados para los valores que almacenarán.

Para esta instancia, todos los modelos deberán tener -al menos- **una tabla de relación N-N o una relación de grado 3** que sea coherente con el sistema desarrollado. Además deberá **contar con todas las tablas necesarias para el e-commerce**.

Todas las relaciones deberán tener un nombre que permita entender el objetivo que cumple cada una.

Sobre el código SQL:

Entregar en un archivo con extensión SQL el código necesario para la creación de la base de datos según el diagrama entidad-relación realizado.

Este archivo deberá tener:

- La creación de la base de datos.
- Deberá tener especificado el DROP DATABASE, CREATE DATABASE y USE, de forma tal que solo deba importarse al servidor para ser utilizada.
- Se evaluarán las definiciones de claves primarias y foráneas.
- Deberá tener las consultas necesarias para la inserción de datos en todas las tablas que formen el sistema. Hacer la mayor cantidad de inserciones que sean posibles.

El código entregado deberá ser creado por el alumno tal como se enseñó durante la cursada, no pudiendo ser la exportación del phpMyAdmin, Navicat o software similar.

Sobre el panel de control:

El panel de control debe estar bajo una sub carpeta y debe exigir la **autenticación por medio de usuario y clave**.

Solo los usuarios admin deben poder acceder a este sistema (se quitarán puntos si un usuario común o no logueado puede acceder a alguna sección del panel).

El panel deberá contar con las secciones necesarias como para realizar un alta, baja y modificación:

- Listado.
- Formulario de alta para un nuevo registro.
- Formulario de edición para el registro.

- Botón para eliminar cada registro.

Si desde el panel se administran los usuarios (darlos de alta nuevos, modificarlos, borrarlos); tener presente que **el manejo de usuarios no es considerado "manejo de contenidos"**.

El código PHP deberá ser lo más prolijo posible y los nombres de variables, constantes y funciones semánticas (o sea, el nombre que le pongas, debe ser representativa de lo que hace).

La web debe estar basada en el **modelo de plantilla** que maneje los contenidos a mostrar por medio de datos recibidos por GET y las funciones nativas require e include de PHP.

Todos los datos recibidos por **GET y por POST deberán ser validados desde PHP**. En ningún caso será correcto asumir que "el usuario seguro enviará ese dato".

Tener presente, por ejemplo, que un usuario podría alterar los valores de la URL para acceder a secciones que no le correspondan. Verificar estos datos de forma tal que **PHP nunca muestre un mensaje de erro**r.

El index deberá incluir, al principio, tres archivos que organizarán lo pedido en este punto.

- Un archivo de **configuración global del sitio**, donde se declarará las constantes y los seteos de php (error reporting, display errors, timezone, conexión a la base).
- Un archivo donde se encuentren todas las **funciones desarrolladas por el alumno**.
- Un tercer archivo que tenga todos los **Arrays** que necesite el sitio para funcionar.

Toda variable de tipo Array así como toda función que se encuentren por fuera de los archivos solicitados en este punto, serán motivo de pérdida de puntos.

DWM3A PROGRAMACIÓN II

PROFESORA: PAMELA IGLESIAS

FECHA DE ENTREGA: **48HS ANTES DE LA MESA**

FORMATO DE ENTREGA: RAR - ZIP

ATENCIÓN:

En base a lo que estuve viendo en los tps, tomé la siguiente decisión. Si en el trabajo se detecta aunque sea un archivito, línea de código, etc que fue copypasteado, DESAPRUEBAN AUTOMÁTICAMENTE, no importa si todo lo demás está perfecto.

Requisitos del final:

- El DER de la base de datos y su código SQL de creación e inserción de datos.
- La web debe conectarse a esa base de datos
- Textos e imágenes pertinentes en el sitio.
- Uso de las funciones nativas de php require e include para el manejo de plantillas.
- Verificación de todos los datos recibidos por GET y POST desde php.
- El crud (Create, Read, Update & Delete) de una tabla de la base de datos (en el panel).
- El carrito funcional, es decir, elegir un producto/servicio, agregarlo al carrito, visualizar el total y "pagar". Se recuerda que no hay pasarela de pago, solo la simulación del envío de datos de pago.
- Solo el usuario registrado (no admin) podrá realizar la compra/contratación del producto/servicio.
- Login y verificación de usuario admin al panel.
- Registro de usuarios.
- Tener bien validados los ID que se piden en el ver detalle, de forma tal que si se pide un registro inexistente muestre un mensaje de error apropiado.
- Que el usuario registrado pueda interactuar con los datos del SQL.
- Todo ABM debe mostrar un mensaje indicando si se realizó o no la operación.
- Upload de archivos.
- Definir el tipo de dato más óptimo de cada campo del SQL.
- Tener bien definidos los campos UNIQUE y NOT NULL del SQL

Extra para el 10:

- Paginación de resultados.
- Buscador de contenidos.
- Manejo de filesystem.
- Uso del GDlibrary para el redimensionamiento de imágenes.
- Uso de clases y objetos (el objeto mysqli_result y json_decode no cuenta)
- PHPDocs