

Universidad de Costa Rica
Sede de Occidente
Recinto de Tacares
Informática Empresarial
Profesor: Massiel Rojas Blanco
IF4101 - Lenguajes para Aplicaciones Comerciales
Proyecto 35%

Cree una aplicación web para comercio electrónico de una empresa (**El grupo puede decidir qué productos vender**. Ejemplo: ropa, zapatos, joyería, alimentos, tecnología, etc). El proyecto puede ser estar programado en dos formas diferentes:

1. **Arquitectura MVC:**
 - a. Una aplicación web MVC en ASP.Net o en ASP.Net core.
2. **Arquitectura Cliente / Servidor.**
 - a. Cliente: Aplicación web en React.
 - b. Servidor: API en .Net Framework.

La aplicación de comercio electrónico estará dividida en roles:

1. **Rol administrativo.** Para poder ingresar a la parte administrativa, un usuario debe dirigirse a la url de nuestra página /admin e insertar las credenciales de un usuario con permiso de administrador.
 - a. Manejo de productos.
 - i. Un administrador puede agregar, editar, ver y eliminar productos del sistema.
 1. Un producto puede tener código, nombre, descripción (características), cantidad, categoría, imágenes, etc. Dependerá del tipo de producto que vendan.
 - ii. Un administrador puede buscar productos por código o nombre.
 - b. Manejo de categorías.
 - i. Un administrador puede agregar, editar, ver y eliminar categorías de productos.
 1. Ejemplo: Blusas, zapatos, vestidos, etc.
 - c. Reporte de ventas.
 - i. Un administrador puede ver un reporte de ventas.
 - ii. Se pueden aplicar filtros para ver las ventas en un rango de fechas específico.
 - d. Cerrar sesión.
2. **Rol de ventas:** Para poder ingresar a la parte de ventas, un usuario debe dirigirse a la url de nuestra página /ventas e insertar las credenciales de un usuario con permiso de ventas.

- a. Órdenes de ventas.
 - i. Un usuario puede ver las órdenes de ventas generadas por los clientes y despachar la orden.
 - 1. La página principal consiste en una lista de pedidos ordenados por fecha de la más antigua a la más reciente.
 - 2. Las órdenes de venta se pueden buscar por número de orden, fecha y correo de usuario, etc.
- b. Cerrar sesión.

3. **Rol de cliente:** Para poder ingresar a la página principal /, un usuario no necesita de una cuenta.

- a. Página principal.
 - i. El usuario puede ver todos los productos del catálogo.
 - 1. Puede añadir el producto al carrito de compras.
 - 2. Puede ver más detalles del producto.
 - 3. Ejemplo:
<https://www.free-css.com/free-css-templates/page287/beautiflie>.
 - ii. El usuario puede hacer un ordenamiento de los productos (Menor precio a Mayor, Mayor precio a menor, Más reciente, Más popular)
 - iii. El usuario puede buscar productos por categoría, por nombre, por características (color, tamaño, talla, etc). Debe tener al menos dos características.
- b. Carrito de compras:
 - i. Un usuario puede eliminar un producto del carrito de compras.
 - ii. Puede editar la cantidad de artículos comprados (dependerá del inventario).
 - iii. Puede hacer la compra de los productos.
- c. Compra.
 - i. Cuando un usuario no ha iniciado sesión:
 - 1. Un usuario deberá ingresar sus datos personales, dirección de entrega e información de la tarjeta de crédito/débito y proceder con la compra.
 - ii. Cuando un usuario ha iniciado sesión (**Puntos extra**):
 - 1. Se llenan automáticamente los campos de los datos personales, el usuario puede elegir una de las direcciones de entrega guardados y seleccionar tokens de tarjetas de crédito guardadas.
- d. Órdenes:
 - i. Cuando un usuario no ha iniciado sesión:
 - 1. Un usuario puede buscar una orden por su número.
 - ii. Cuando un usuario ha iniciado sesión (**Puntos extra**):

1. Se le muestra al usuario una lista con todos los pedidos ordenados del más reciente al menos reciente.
- iii. Las órdenes contienen la información de los productos, costos y estado de la orden.
 1. Una orden de compra tiene dos estados.
 - a. En proceso, enviado.
- e. Cuenta (**Puntos extra**).
 - i. El usuario puede crear una cuenta.
 1. Cuando un usuario tiene una cuenta e inicia sesión puede realizar lo siguiente:
 - a. Añadir y editar sus datos personales necesarios para una compra.
 - b. Añadir, editar o eliminar direcciones de entrega.
 - c. Añadir, editar o eliminar tokens de tarjetas de débito / crédito.
 - ii. Lista de deseos.
 1. Puede agregar o eliminar productos de la lista de deseos.
 2. Puede ver todos los productos que ha añadido a la lista de deseos.
 - iii. Órdenes:
 1. Puede ver todas las órdenes de compra.
- f. Cerrar sesión.

Porcentaje:

- Código e implementación 30%. (Nota grupal)
- Defensa del proyecto 5%. (Nota individual)

Notas:

- Se deben hacer todas las validaciones correspondientes.
- La aplicación debe ser responsiva.
- Cada grupo deberá elaborar el diseño de su base de datos.
- Los grupos deberán utilizar GitHub como sistema de control de versiones.
- El diseño de la base de datos deberá ser elaborado por el grupo de proyecto.
- La aplicación y base de datos deben estar alojadas en un servidor (hosting).
- Los grupos puede seleccionar el sistema de gestión de bases de datos (MySQL, SQL server, etc)
- Cualquier cambio al enunciado deberá ser comunicado a tiempo por parte del profesor.

- Se trabajará en los grupos previamente definidos en clases, esto aplica tanto para la entrega de proyecto como la defensa. Si un grupo decide separarse debe primero hablar con el profesor para dialogar sobre la situación. El profesor es quién decide si el grupo se separa e indicar el proceso de separación. Si un miembro de un grupo se separa no puede unirse a otro grupo. No se permite entregar dos proyectos por separado por un grupo que no ha pedido al profesor separarse, en este caso no se recibe ninguna de los dos proyectos realizados individualmente, y la nota es cero.
- Se procederá con lo establecido por el reglamento estudiantil ante cualquier comprobación de plagio.
- No se reciben proyectos ni defensas después de la fecha y hora de entrega límites.
- El proyecto deberá ser subido a mediación virtual.