**Ejercicio 1**

Dado el siguiente modelo que ya esta implementado:



Tomando como base la implementación publicada en aulas, desarrolle el siguiente caso de uso, implementando la vista de la interfaz de usuario en html/css/javascript.

**CU: Alta de nuevo cliente**

**Curso normal:**

1- El sistema muestra la lista de clientes en formato de tabla (Nombre, Cedula, Email)

El sistema deshabilita la posibilidad de ingresar nombre, email y guardar cliente.

El sistema habilita la posibilidad de ingresar la cedula del cliente y de validarla.

2-El usuario ingresa la cedula del cliente e indica que desea validarla.

3-El sistema verifica la cedula (sin puntos ni guiones y que tenga entre 6 y 8 dígitos numéricos) verifica que no exista un cliente ingresado para esa cedula y habilita la posibilidad de ingresar el nombre el email y de guardar el cliente.

4-El usuario ingresa el nombre del cliente, el email e indica que desea guardar el cliente.

5-El sistema ingresa el nuevo cliente, actualiza la tabla de clientes del punto 1, limpia de pantalla los datos ingresados por el usuario y deshabilita la posibilidad de ingresar nombre, email y guardar cliente.

**Cursos alternativos:**

3- El formato de la cedula no es válido. Mensaje “Cedula incorrecta”.

Ya hay un cliente ingresado para la misma cédula. Mensaje “Ya existe el cliente”. Y se muestran en pantalla los datos del cliente correspondientes a la cedula ingresada (nombre e email)

En cualquiera de los dos casos anteriores no se habilita la posibilidad de ingresar el nombre el email y de guardar el cliente.

5-Si el nombre del cliente es vacio o contiene solo espacios o si el email del cliente no contiene “@” ni “.” no se da de alta el cliente y se muestra mensaje “No se pudo agregar el cliente".

Parte 1) Diseñar e implementar la lógica de negocio (modelo)

Parte 2) Diseñar e implementar el controlador con la lógica del caso de uso.

Definir lo eventos del usuario (enpoints)

Definir las respuestas del controlador para cada evento (endpoint)

Parte 3) Implementar la vista en html + javascript

Reglas: Aplicar Experto y división lógica en modelo, controlador y vista.

**Ejercicio 2**

Siguiendo con el modelo del ejercicio anterior implementar el siguiente caso de uso:

**CU: Alta de producto**

**Curso normal:**

1- El sistema muestra la lista de productos en formato de tabla (Nombre, Precio, Unidades, Proveedor, Código)

El sistema muestra los proveedores en una lista (Info: Nombre del proveedor)

El sistema deshabilita la posibilidad de seleccionar un proveedor, de ingresar el precio, las unidades y de guardar el producto.

El sistema habilita ingresar el nombre del producto, y de validarlo.

2-El usuario ingresa el nombre del producto e indica que desea validarlo.

3-El sistema verifica el nombre (que no sea vacio o contenga solo espacios y que el largo sea menor a 50 caracteres), verifica que no exista un producto ingresado con el mismo nombre, y habilita la posibilidad de seleccionar el proveedor, de ingresar el precio, las unidades y de guardar el producto.

4-El usuario selecciona el proveedor, ingresa el precio del producto, la cantidad de unidades, e indica que desea guardar el producto.

5-El sistema ingresa el nuevo producto, actualiza la tabla de productos del punto 1, limpia de pantalla los datos ingresados por el usuario y deshabilita la posibilidad de seleccionar el proveedor, de ingresar el precio, las unidades y de guardar el producto.

**Postcondición:** Si se hace el alta correctamente se le asigna automáticamente un código al producto (auto numerado)

**Cursos alternativos:**

3- El nombre del producto no es válido. Mensaje “Nombre de producto incorrecto”.

Ya hay un producto ingresado con el mismo nombre.

Mensaje: “Ya existe el producto”. Y se muestran en pantalla los datos del producto (proveedor, precio y unidades)

En cualquiera de los casos anteriores no se habilita la posibilidad de seleccionar el proveedor, de ingresar el precio, las unidades y de guardar el producto.

5-Si el precio o las unidades del producto son menores a 1 o no se ha seleccionado un proveedor, no se da de alta el producto y se muestra mensaje “No se pudo agregar el producto".

Parte 1) Diseñar e implementar la lógica de negocio (modelo)

Parte 2) Diseñar e implementar el controlador con la lógica del caso de uso.

Definir lo eventos del usuario (enpoints)

Definir las respuestas del controlador para cada evento (endpoint)

Parte 3) Implementar la vista en html + javascript

Reglas: Aplicar Experto y división lógica en modelo, controlador y vista.

**Ejercicio 3**

Siguiendo con el modelo del ejercicio anterior implementar el siguiente caso de uso:

**CU: Alta de factura**

**Curso normal:**

1) El sistema muestra el total facturado (Suma de todas las facturas del sistema)

2) El usuario ingresa la cédula del cliente que va a comprar e indica que desea iniciar una factura para ese cliente.

3) El sistema inicia una factura para ese cliente, muestra el nombre del cliente, la fecha de la factura (dia/mes/año) y el total de la factura (cero) con un símbolo de “$” delante.

4) El usuario ingresa el código del producto y la cantidad, e indica que desea agregarlo a la factura.

5) El sistema actualiza las líneas de la factura, mostrándolas en una tabla (Cantidad, Nombre Producto, Total Línea).

El sistema actualiza el total de la factura.

6) El usuario confirma la factura.

7) El sistema guarda la factura, actualiza el total facturado y muestra el mensaje “Se guardo correctamente con numero” + el numero de la factura. Fin caso de uso.

8) Opcionalmente el usuario indica descartar la factura (en lugar de confirmarla)

El sistema descarta la factura. Fin caso de uso.

**Postcondición:** Si se guarda la factura el sistema le asigna numero y debe quedar dado de baja el stock (unidades) de todos los productos de la factura. No hay factura iniciada.

**Nota:** El usuario puede agregar N productos a la factura

**Cursos alternativos:**

3) El cliente no existe. Mensaje “No se pudo iniciar la factura, verifique la cedula”.

Ya hay una factura iniciada en proceso. Mensaje “Debe finalizar la factura actual antes de iniciar otra”

5) El código de producto no existe o la cantidad es menor a 1 o no hay stock suficiente. Mensaje “No se pudo agregar el producto”.

Si el producto elegido ya está en una línea previa de la factura se actualiza la cantidad de unidades de la línea existente.

Si no hay factura iniciada. Mensaje “Debe iniciar la factura antes de agregar productos”.

7) Si la factura no contiene productos. Mensaje “No se pudo confirmar la factura, verifique si hay productos”

Si no hay factura iniciada. Mensaje “Debe iniciar la factura antes de confirmarla”.

8) Si no hay factura iniciada. Mensaje “No se ha iniciado la factura”.

Parte 1) Diseñar e implementar la lógica de negocio (modelo)

Parte 2) Diseñar e implementar el controlador con la lógica del caso de uso.

Definir lo eventos del usuario (enpoints)

Definir las respuestas del controlador para cada evento (endpoint)

Parte 3) Implementar la vista en html + javascript

Reglas: Aplicar Experto y división lógica en modelo, controlador y vista.

**Ejercicio 4**

Siguiendo con el modelo del ejercicio anterior agregar el siguiente caso de uso:

**CU: Consulta de clientes que compraron el producto más barato:**

**Curso normal:**

1) El sistema muestra el producto más barato en el siguiente formato de ejemplo:

Codigo: 1

Nombre: Caramelo

Precio: $2

Stock: 3000

2) El sistema muestra la lista de clientes que hayan comprado el producto más barato en una tabla con las siguientes columnas:

Nombre - Cedula - Email – Ultima Compra (formato de fecha dia/mes/año hora:minutos)

**Cursos alternativos:**

1) Si no hay productos ingresados, se despliega mensaje “No hay productos ingresados.” y fin de caso de uso.

2) Si no hay ningún cliente que haya comprado el producto más barato se indica mediante el mensaje “No hay clientes que hayan comprado el producto.” y fin de caso de uso.

Parte 1) Diseñar e implementar la lógica de negocio (modelo)

Parte 2) Diseñar e implementar el controlador con la lógica del caso de uso.

Definir lo eventos del usuario (enpoints)

Definir las respuestas del controlador para cada evento (endpoint)

Parte 3) Implementar la vista en html + javascript

Reglas: Aplicar Experto y división lógica en modelo, controlador y vista.