

Proyecto 01

Modelado y Programación

Objetivo

Usando los patrones de diseño vistos en clase resuelvan la siguiente problemática. **La solución propuesta debe evitar caer en problemas de rigidez, fragilidad, inmovilidad, y viscosidad.**

Requerimientos

La tienda CheemsMart te ha encargado implementar un sistema de prueba de tienda virtual. Si les gusta, usarán tu implementación.

- La tienda debe guardar la información de todos los clientes registrados. La información que les interesa guardar es: nombre de usuario, contraseña, nombre, teléfono, dirección, cuenta bancaria asociada (detallada más abajo), país de origen. También es valioso un identificador único para cada cliente. El país de origen servirá para cambiar la interfaz de la tienda virtual. Por ahora, la tienda funcionará en Estados Unidos, España, y México. Se desea que sea lo más familiar posible para el usuario. Al saludar, despedirse, completar una compra, dar fecha estimada de entrega, y enviar ofertas (detalladas más abajo), se deberá usar el idioma adecuado para el cliente. El catálogo no es requerido que cambie, pues los productos son regionales, por lo que toda su información (detallada abajo) podrá mantenerse en español latino. Se espera que en un futuro cercano, la tienda pueda presentarse en más países.
- Las ofertas cambiarán entre los países. Las ofertas corresponden a un descuento sobre las compras. Estas ofertas se aplicarán de forma azarosa, es decir que se debe incluir una forma de decidir cuando se aplican estos descuentos. Si el cliente alcanza a ver una oferta, sus compras tendrán aplicado ese descuento. Sabemos que el público estadounidense disfruta las ofertas en productos de electrónica. El público español prefiere ofertas en electrodomésticos. Por su parte, el público mexicano aprovecha las ofertas en productos alimenticios. Por ahora, puedes aplicar la cantidad de descuento que tu prefieras. En otras palabras, tú decides si llevan un 20, 30, o 40%, o la cantidad que desees. Estas ofertas se envían a todos los usuarios que les corresponda.
- Naturalmente, la tienda virtual necesita un catálogo de productos. Este catálogo debe guardar código de barras, nombre, departamento (el tipo de producto que es, por ejemplo electrónica, limpieza, lácteos, etc.), y precio. Los productos de la tienda son regionales, por lo que su información puede ser guardada en español latino y mostrarla al usuario de la misma forma sin preocuparse por su nacionalidad. Este catálogo es cargado de forma remota desde el servidor de la tienda a la interfaz virtual del cliente. Por cuestiones de seguridad, es imperiosamente necesario que la tienda virtual no cargue directamente el catálogo real, pues si llega a presentar vulnerabilidades de seguridad, cualquiera podría cambiarlo. Es por eso que se necesita una forma de presentar esta carga remota del catálogo real a su

representación en la tienda virtual con el fin de poder mostrárselo al cliente. Es importante señalar que el catálogo debe mostrarse de la forma más cómoda posible para el cliente, por lo que se espera un buen formato. Para esta simulación, puedes agregar los productos de tu preferencia al catálogo.

- Un cliente puede acceder a su cuenta para realizar compras en línea a través de su usuario y contraseña. Después de identificarse, el sistema mostrará haber cambiado su idioma para comodidad del usuario a partir de su nacionalidad. En caso de que el sistema lo decida, le mostrará las ofertas a todos los usuarios adecuados. Lo siguiente será mostrarle al usuario un pequeño menú con las opciones de ver catálogo, realizar una compra, o salir.
 - Ver catálogo le mostrará al cliente todos los productos disponibles enlistados con un formato agradable.
 - Realizar una compra lo llevará al catálogo igual que el punto anterior, sin embargo, se ofrecerá la opción de agregar un producto a su “carrito de compras”. Un producto a la vez, el usuario podrá agregar un nuevo artículo. Si desea comprar 2 o más del mismo producto, lo hará uno por uno. También debe haber una opción para salir, en caso de que el usuario desee cancelar su compra o salir sin haber comprado nada. Además, necesitará una opción para acabar su compra. Al hacerlo, se mostrará la siguiente pantalla de compra segura (descrita más abajo). Al concretarse la compra, el cliente podrá ver un ticket de su compra realizada, y una fecha estimada del día en que se entregará a su hogar. Para esta implementación, ésta fecha puede elegirse al azar y no se verificará que sea correcta.
 - Salir cerrará su sesión, regresando al menú inicial para ingresar usuario y contraseña.
- La tienda prometió no hacer mal uso de la información del cliente. Una compra segura es una forma de proteger sus datos personales. Desde que un usuario se identifica, deberá realizar cualquier operación a través de un representante. Al realizar una compra y llegar a la pantalla de “compra segura”, el cliente deberá ingresar su número de cuenta bancaria, que se comparará con el atributo del cliente real. En caso de no ser el mismo, la compra se cancelará.

Notas adicionales

Para este proyecto se requiere que hagan diagramas de clases, y de secuencia (para presentar la forma en que un cliente hace una compra).

También será necesario entregar un archivo pdf titulado README como se indica en los lineamientos. **En este archivo incluirán una justificación de los patrones que decidieron usar.** Esta justificación debe ser breve y concisa. Explicarán por qué usaron determinado patrón de diseño, o la ausencia de uno. Por ejemplo “En la parte de ____ decidí/decidimos usar ____ patrón porque...” o “En la parte de ____ no usé/usamos un patrón pues lo resolví/resolvimos de esta manera...”.

Además, deben agregar al menos un ejemplo de entradas para correr el sistema de forma exitosa. Este ejemplo debe incluir un usuario y contraseña existentes, además de su cuenta bancaria para poder ejecutar las funcionalidades.

Sigan los lineamientos para la entrega de prácticas. El documento estará en el classroom. A diferencia de las prácticas, este proyecto puede realizarse en equipos de hasta 3 personas.

Evaluación

50% Patrones e implementación

20% UML

20% Funcionamiento

10% Justificación