

Patrones de Diseño

El proyecto final consiste en un proyecto que se entregará en dos fechas. A continuación se especifica cada una de las entregas de este proyecto final.

Parte Uno **Entrega de esta parte: 11 de mayo de 2022.**

Crear una aplicación Windows que se encargue de la administración y control de pedidos para tiendas de autoservicio.

La aplicación será capaz de levantar pedidos en diferentes tiendas, para ello, una vez que llegue a la tienda y lea un QR para saber su nombre y Id, deberá capturar el producto, y la cantidad. Al momento de guardar el pedido, la aplicación generará una imagen QR, que reemplazará a la leída anteriormente, que llevará almacenado en formato Json la siguiente información:

- Id tienda
- Nombre Tienda
- Lista de productos a surtir en la próxima visita: Id producto, nombre, precio y cantidad

Y este mismo proceso de toma de pedidos se repetirá para cada tienda.

Al día siguiente que se requiera tomar la ruta para surtir los pedidos, el proceso será el siguiente:

- Leer las imágenes de pedidos levantados, en cada tienda.
- Simular la ruta de entrega (punto explicado más adelante)
- Generar la ruta para visitar las tiendas. Esta lógica estará dada por: la tienda que involucre la mayor ganancia, deberá ser visitada primero, y así sucesivamente, dejando al final la tienda que genere menos dinero.
- Por cada tienda que se visite se surtirá el pedido y al mismo tiempo se levantará el pedido del día siguiente, al generarlo, de acuerdo al proceso anterior, se generará una nueva imagen QR que será utilizada para formar la ruta del día siguiente.

Simulación de Ruta de Entrega

Un punto importante es que antes de generar la ruta de entregas, será necesario verificar el número de camiones repartidores necesarios para cumplir con la demanda de los pedidos.

Habrán un máximo de 5 camiones repartidores:

- 3 camiones de refrescos. Cada camión contará con 120 refrescos por surtir.
- 3 camiones de pan fresco. Cada camión contará con 270 piezas de pan por surtir.
- 3 camiones de verdura congelada. Cada camión cuenta con 95 bolsas de 1 Kg de verdura congelada.

La aplicación deberá simular la ruta de entregas, antes de que los repartidores salgan.

Para ello el usuario, elegirá los camiones a salir a ruta (mínimo 1) y ejecutará la simulación. Si el resultado es: No se cumple con la demanda de producto.

El usuario agregará un camión más a la simulación en runtime, y volverá a ejecutar la simulación, y así sucesivamente, hasta dar con el resultado: Se cumple con la demanda y sobran X productos. Donde X representa el número de productos sobrantes, luego de terminar la ruta.

Parte 2 :**Fecha de entrega: 20 de mayo de 2022**

La aplicación deberá contener una bitácora, la cual esté registrando los eventos de:

- · Comienzo de simulación
- · Agregar camión repartidor
- · Fin de simulación
- · Crear pedido
- · Agregar producto
- · Finalizar pedido
- · Creación de imagen.

La bitácora es de tres tipos:

- · En archivo de texto txt
- · A Grid en pantalla
- · A textbox multilinea

El tipo de bitácora podrá ser seleccionado en runtime, en una pantalla o menú de configuración. El tipo puede ser los tres tipos al mismo tiempo. E inclusive varias veces el mismo tipo. Ejemplos:

- · Si el usuario selecciona bitácora de tipo: Archivo de texto. La bitácora solo se mostrará en un archivo de texto.
- · Si el usuario selecciona tres veces el archivo de texto. Al abrir el archivo, la bitácora mostrará tres líneas por cada evento que registró.
- · Si el usuario selecciona: archivo de texto y textbox, la salida de la bitácora deberá ser tanto en archivo como en el textbox.

Puntos a Considerar de Programación

La aplicación deberá consumir un API que codifique y decodifique las imágenes QR y las convierta en la información anteriormente mencionada. API: 2

La aplicación deberá aplicar de manera correcta al menos tres patrones de diseño, uno por cada tipo de patrones que existen.

Deberán entregar un documento con la explicación y/o justificación de los patrones de diseño. Cada patrón aplicado, deberá tener su UML.