



Con esta práctica se pretende simular la *gestión de pedidos* de una web especializada en la venta online. Los usuarios de dicha web pueden ser de dos tipos: clientes registrados (que pagan una cuota anual para la entrega rápida de sus pedidos, en menos de veinticuatro horas (VIP) o en menos de tres días (NVIP)) y clientes no registrados. Para cada pedido se incluirán¹ la descripción del artículo, el nombre del cliente y la dirección de entrega, el tipo de cliente (registrado VIP, registrado NVIP, no registrado NR), la numeración de la tarjeta de crédito y el tiempo necesario para preparar el envío (aleatorio entre 1 y 10 minutos).

Para realizar la simulación, se deberán implementar al menos los siguientes TAD's o clases en C++:

1. Una *pila de pedidos erróneos*, que almacenará los pedidos cuyos datos son incompletos. Los pedidos de clientes no registrados en la web se situarán en el fondo de la pila, debajo de todos los pedidos de clientes registrados (ambos por orden de llegada a la pila, el primer pedido que llegó de un cliente no registrado estará al fondo).
2. Dos *colas de pedidos en espera*, una para clientes registrados (VIP y NVIP) y otra para no registrados (NR), en las que se almacenarán los pedidos por orden de llegada.
3. Una *lista de pedidos para enviar*, en la que se almacenarán los pedidos para enviar ordenados por tipo de cliente (primero todos los pedidos de clientes VIP, después los de clientes NVIP y por último los de clientes no registrados NR) y por orden de llegada a la lista.

El programa permitirá las siguientes opciones:

1. **Generar** (aleatoriamente o directamente en el código) o **leer** (se leerán de teclado según formato en anexo) los pedidos realizados por los clientes. Se almacenarán en las dos *colas de pedidos en espera*, según se trate de clientes registrados o no, por orden de llegada. El número de clientes que se utilizarán para la simulación será entre 10 y 20.
2. **Gestión de pedidos:**
 - 2.1. Para empezar, se insertan en la *lista de pedidos para enviar* tres pedidos de la *cola de espera* de clientes registrados y uno de la *cola de espera* de clientes no registrados. Antes de pasar un pedido a la *lista*, se comprueba si los datos son incorrectos² y en este caso se almacena en la *pila de pedidos erróneos* y se pasa el siguiente.
 - 2.2. A continuación, se **simula el paso del tiempo y la preparación** y envío de los pedidos, comenzando por el primero de la lista. Para ello, en cada minuto se comprueba el primer pedido de la lista:
 - 2.2.1. Si no está preparado todavía (no ha pasado el tiempo de preparación), se pasa al minuto siguiente³.
 - 2.2.2. Si está **listo para enviar** (si ya ha pasado el tiempo de preparación):
 - a) Se escribe en pantalla un mensaje con los datos del pedido, indicando que se ha enviado.
 - b) Se elimina de la lista.

¹ Se pueden incluir otros datos que se consideren necesarios para la implementación de la práctica.

² Se deja a elección del programador las comprobaciones sobre la corrección de los datos.

³ Durante la ejecución, y para comprobar el funcionamiento, se irán escribiendo mensajes que permitan ver las colas, lista y pila.



- c) Se **incluye** en la *lista de pedidos para enviar un nuevo pedido*, de la siguiente forma: primero pasan tres pedidos de la *cola de espera* de clientes registrados, después se pasa un pedido de cliente registrado desde la *pila de pedidos erróneos* (completando los datos) si lo hay, o, en otro caso, uno de la *cola de espera* de clientes no registrados. Es decir, los pedidos en espera de clientes no registrados pasarán¹ a la lista, si no hay pedidos erróneos de clientes registrados, antes que los pedidos erróneos de clientes no registrados¹.

2.2.3. La gestión de pedidos **finalizará** cuando todos los pedidos hayan sido enviados.

3. Salir del sistema.

ANEXO –FORMATO LECTURA DE DATOS

Ordenador portátil//Luís López//Calle Luna 7//VIP//780044545667//3

Libro electrónico//Carlos Pérez//Calle Luna 7//NVIP//78001345667//4

Taladro//C López//Calle Estrella 7//NVIP//780678345667//7

Mesa comedor//J López//Calle Sol 7//NR//780453745667//8

¹ Siempre pasará uno después de pasar tres de la *cola de espera* de clientes registrados.



NORMAS PARA LA REALIZACIÓN DE LA PRÁCTICA

1. Las prácticas se realizarán individualmente o en grupos de dos alumnos que deberán ser los mismos para las dos prácticas de la asignatura.
2. Debe entregarse un fichero comprimido incluyendo todos los ficheros fuente del proyecto C++ y el documento descrito en el punto 6. Se subirá un fichero por grupo a la plataforma **dentro del plazo fijado en la plataforma**. El nombre del fichero será el nombreapellido1apellido2 de uno de los miembros del grupo.
3. La asistencia y realización de la defensa, es **individual y obligatoria** para todos los alumnos del grupo. En el caso de no asistir, la práctica se calificará con un suspenso (0).
4. En la defensa de la práctica se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesaria la calificación de APTO para poder ser evaluado de la práctica. La defensa se realizará a la hora y en el lugar indicado por el profesor.
5. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
6. La documentación que se entregará en formato digital, junto con el código, deberá tener al menos los siguientes apartados:
 - a. Nombre y DNI de los alumnos del grupo.
 - b. Detalles y justificación de la implementación:
 - b.1 Especificación concreta de la interfaz de los TAD's implementados:
 - b.1.1 TAD's creados.
 - b.1.2 Definición de las operaciones del TAD (Nombre, argumentos y retorno).
 - b.2 Solución adoptada: descripción de las dificultades encontradas.
 - b.3 Diseño de la relación entre las clases de los TAD implementados.
 - b.3.1 Diagrama UML.
 - b.3.2 Explicación de los métodos más destacados.
 - b.3.3 Explicación del comportamiento del programa.
 - b.4 Código fuente.
 - b.5 Bibliografía.