

Taller II – Curso 2025 - 2026

Trabajo de Laboratorio

Introducción

Se desea desarrollar un primer prototipo de un sistema a utilizarse en un emprendimiento familiar dedicado a la elaboración y reparto a domicilio de postres artesanales de elaboración casera que se inauguró durante la pandemia de COVID 19.

Los postres son de muy buena calidad y gracias a la recomendación de sus primeros clientes a sus familiares y amistades, junto con algo de difusión en redes sociales, la tiendita de postres ha ido aumentando su clientela desde la finalización de la pandemia, lo que ha generado a su vez un crecimiento sostenido en las ventas realizadas.



Debido a esto, la familia que dio origen a este emprendimiento ha solicitado una aplicación que le permita gestionar más ágilmente las ventas y reparto a domicilio de los postres elaborados en la tiendita. En este taller se va a desarrollar un primer prototipo de dicha aplicación, que servirá para que los vendedores que trabajan en la tiendita (esencialmente, los miembros de la familia) puedan registrar las ventas y repartos a domicilio de los distintos postres que se elaboran. También servirá para que puedan obtener una serie de estadísticas básicas relativas a las ventas realizadas.

Planteo del Problema

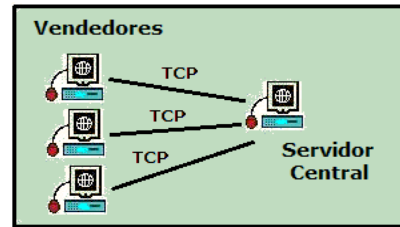
Cada postre ofrecido en la tiendita se identifica por un código alfanumérico y además se registra el nombre del postre y su precio por unidad. Por ejemplo, el postre con código “chocoint001” se llama “*pasión de chocolate intenso*” y cuesta \$250. Algunos postres son light, en cuyo caso se registra además el tipo de endulzante utilizado junto con una descripción adicional. Por ejemplo: el postre light con código “maracuya123” se llama “*explosión de maracuyá*”, cuesta \$320, se elabora con endulzante stevia y su descripción adicional es “*sin gluten, apto para personas celíacas*”.

Se desea llevar un historial de las ventas realizadas, las cuales se identificarán mediante números consecutivos. La primera venta en ser registrada tendrá el número 1, la siguiente el número 2 y así sucesivamente. Además del número, interesa registrar los siguientes datos de cada venta: fecha, dirección de entrega, si es una venta actualmente en proceso o ya finalizada y monto total de la misma. En este primer prototipo, no interesa registrar si la venta es en efectivo o con tarjeta (si es con tarjeta, el comercio usará, por fuera de la aplicación, el dispositivo POS para registrar el pago con tarjeta). Simplemente se quiere tener un registro de las ventas, sin importar la forma de pago.

Por otra parte, interesa llevar un registro de los postres adquiridos en cada venta. Una misma venta puede estar compuesta por hasta 40 postres como máximo. A su vez, un mismo postre puede ser vendido muchas veces. Por ejemplo: el postre “*cheesecake de frutos rojos*” tiene gran popularidad y se vende con mucha frecuencia. Por cada postre adquirido en una venta interesa registrar la cantidad de unidades de dicho postre adquiridas (por ejemplo: 3 unidades de cheesecake de frutos rojos). Debido a la capacidad que tiene la empresa para realizar los repartos a domicilio, ninguna venta podrá superar las 40 unidades en total (no importa qué postres en concreto integren la venta, el total de unidades que la componen siempre será, como mucho, 40).

Arquitectura de la Aplicación

Los usuarios de la aplicación serán los vendedores de la tiendita, quienes la utilizarán desde las computadoras ubicadas en el local. La figura ilustra la distribución de los equipos que integran la aplicación, la cual correrá sobre la red local (L.A.N - Local Area Network) de la tiendita. Las computadoras se conectarán a través del protocolo TCP a otra computadora que oficiará de servidor central.



En los equipos de los vendedores correrá un programa cliente con una interfaz gráfica de ventanas que les permitirá acceder a todas sus funcionalidades y se comunicará con el servidor central, el cual albergará la lógica y la persistencia de toda la aplicación. Por tratarse de una primera versión, los vendedores no necesitarán iniciar sesión para utilizar la aplicación. Simplemente entrarán a la misma y podrán acceder directamente las funcionalidades, sin necesitar ninguna autenticación.

Requerimientos

La aplicación a desarrollar deberá resolver los siguientes requerimientos:

- 1) **Alta de nuevo postre:** Registrar los datos de un nuevo postre que se pone a la venta en la tiendita, verificando que no haya otro postre previamente registrado con el mismo código.
- 2) **Listado general de postres:** Obtener un listado conteniendo código, nombre, precio unitario y tipo (común o light) de todos los postres registrados en el sistema. Este listado se realizará ordenado alfanuméricamente por código de postre.
- 3) **Listado detallado de un postre:** Dado el código que identifica a un postre, listar todos sus datos (código, nombre, precio unitario y tipo). En caso de tratarse de un postre light, se listará también el endulzante empleado junto con su descripción adicional.
- 4) **Comienzo de una nueva venta:** Registrar los datos básicos (fecha y dirección de entrega) de una nueva venta que comienza, la cual se marcará automáticamente como venta en proceso y su monto total será inicialmente cero. Dado que los números de venta son secuenciales, se le asignará en forma automática el número siguiente al de la última venta registrada hasta el momento, verificando además que la fecha de la nueva venta sea igual o posterior a la misma.
- 5) **Agregado de un nuevo postre a una venta:** Dados el código que identifica a un postre, una cantidad de unidades y el número que identifica a una venta en proceso, agregar dicha cantidad de unidades del postre a esa venta. Si el postre ya había sido agregado en forma previa, simplemente se incrementará la cantidad de unidades que se están adquiriendo. En cualquier caso, no se deben superar las 40 unidades de postres en total para la venta.
- 6) **Eliminación de postre de una venta:** Dados el código que identifica a un postre, la cantidad de unidades a eliminar, y el número que identifica a una venta en proceso, eliminar dicha cantidad de unidades de ese postre de la venta. Si se eliminan todas las unidades de ese postre, también se procederá a quitar totalmente ese postre de la venta.
- 7) **Finalización de una venta:** Dados el número que identifica a una venta en proceso y una indicación de si se confirma o se cancela, proceder a su confirmación o cancelación. Si se confirma, se marcará como finalizada, continuará almacenada en el sistema y se emitirá al usuario el monto total a abonar. Si se cancela, será eliminada en ese momento. También será eliminada si no tiene registrado ningún postre, sin importar si se confirma o cancela.
- 8) **Listado de ventas:** Dada una indicación (que puede ser T, P ó F) obtener un listado de las ventas correspondientes a dicha indicación (T = todas las ventas, P = ventas en proceso, F = ventas finalizadas). Por cada venta que integre el listado se desplegará número, fecha, dirección de entrega, monto total de la venta y estado (en proceso o finalizada), ordenado en forma ascendente por número de venta.

- 9) **Listado de postres de una venta:** Dado el número que identifica a una venta, obtener un listado de todos los postres agregados a dicha venta (código, nombre, precio unitario, tipo de postre y cantidad de unidades). Los postres serán listados en el mismo orden en que fueron siendo agregados a la venta.
- 10) **Recaudación por postre y fecha:** Dados el código de un postre y una fecha, calcular el monto total de dinero recaudado por concepto de ventas (finalizadas), junto con la cantidad total de unidades vendidas de dicho postre en esa fecha. Si no se vendió ninguna unidad de dicho postre en esa fecha, ambos resultados serán cero.
- 11) **Respaldo de datos:** Respalidar en disco todos los datos de la aplicación. Este requerimiento podrá ejecutarse toda vez que algún vendedor lo considere oportuno, especialmente luego de haber ejecutado funcionalidades que produzcan cambios en la información del sistema. Todos los datos se respaldarán juntos en un único archivo binario en disco (ubicado en el servidor central), para luego poder ser recuperados a memoria en una próxima ejecución.
- 12) **Recuperación de datos:** Recuperar a memoria todos los datos de la aplicación almacenados en disco. Este requerimiento será ejecutado automáticamente en el servidor central cada vez que el sistema inicie su ejecución (no será disparado por ningún usuario que explícitamente solicite su ejecución).

Lea con atención todos los requerimientos. No olvide hacer todos los chequeos que sean necesarios en forma previa a la resolución de cada requerimiento.

Primera Entrega - *Análisis y Diseño*

Fecha de entrega: **Miércoles 18 de Febrero de 2026** (hasta 23:59 hs). Se debe entregar un documento en formato **PDF** al docente tutor, conteniendo la solución a las partes A y B.

Parte A: Diagrama de clases conceptual de UML correspondiente a la realidad presentada en el planteo del problema propuesto anteriormente.

Parte B: Elección de un diseño para resolver la lógica de la aplicación, desglose de requerimientos para el diseño elegido, especificación completa en UML (diagrama de clases de implementación) y elección de estructuras de datos apropiadas, justificando.

Segunda Entrega - *Implementación*

Fecha de entrega: **Martes 10 de Marzo de 2026** (hasta 23:59 hs). Se debe entregar al docente tutor el código fuente que da solución a las partes C, D y E.

Parte C: Implementación en JAVA de todas las clases y métodos de la capa lógica definidos en las etapas de análisis y diseño. También se deben implementar las clases y métodos necesarios para respaldar y recuperar la información desde disco (capa de persistencia).

Parte D: Implementación en JAVA del programa servidor de toda la aplicación mediante el uso de RMI para la comunicación remota de los programas cliente con el programa servidor.

Parte E: Implementación en JAVA de todas las clases necesarias para el manejo de la interfaz gráfica para los programas cliente de los vendedores. Se utilizarán paquetes con componentes gráficos de Java. Además se deberá realizar integración de la capa gráfica con el programa servidor a efectos de verificar el funcionamiento de los requerimientos.