

# Tienda UPM



**POLITÉCNICA**

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

*Practica de Programación Orientada a Objetos 2025-2026*

*Universidad Politécnica de Madrid*

*E.T.S. de Ingeniería en Sistemas Informáticos*

*Departamento de Sistemas Informáticos*

## Entrega E3:

**El cliente incorpora clientes Empresa** (identificados por **NIF**) y se agregan **Producto-Servicios** (transportes, espectáculos o seguros). Los servicios solo se calculan a posteriori ya que se emitirá una factura y no se puede calcular su precio al agregarlos al ticket, por lo tanto, no tendrán precio, ni nombre, solo **fecha máxima de uso, que limitara su inclusión y cierre en los tickets**. Tendrán un **ID numérico generado de manera secuencial empezando por el 1 y la letra S al final**.

Se extiende el conjunto de tickets a **tickets comunes y tickets de empresa**, que estarán diferenciados por el usuario en el comando de creación del ticket, quedando los tickets comunes para los usuarios normales (los de la entrega 2) y los tickets de empresa para los usuarios de empresa. Aunque ambos tickets tienen el comportamiento similar tienen algunas diferencias. Los tickets de usuario **solo aceptan Productos**. Por otro lado, los tickets de empresa aceptaran **o solo Servicios o Productos y Servicios combinados**. En estos casos, cuando se realice la impresión, **los servicios no aparecerán con precio** y se aplicara un plus de **15% de descuento** en los productos **por cada servicio contratado**. En caso de ser solo un ticket de servicios **no imprimirá precios ni descuentos**, ya que no tiene sentido. Si un ticket es de combinado (productos y servicios) **no podrá cerrarse si no hay al menos y un producto y un servicio**.

Debido a simplificar el sistema será necesario que los tickets hagan uso de la **programación parametrizada** para poder manejar los tickets de manera única por los gestores, asegurar la lógica de los tickets y gestionar los modos de impresión de estos mediante la **inyección de dependencia de clases de manipulación de comportamiento** en el proceso de impresión.

Sera obligatorio, además, por razón de diseño que se usen **al menos un patrón de diseño** al menos.

También **será obligatoria la persistencia** del sistema de tal manera que pueda recuperarse entre ejecuciones. Esta persistencia siempre deberá ser **local** y se adjuntará con el proyecto final por lo que debe estar contenida dentro del proyecto.

Se valorará de **manera positiva** (+1 punto extra sobre esta parte) el uso de librerías de alto nivel (**Tema 5, autónomo**) para agilizar el desarrollo, estabilidad y vistosidad del producto final.

Como producto final no deberán existir errores que queden a la vista del usuario o se impriman por terminal (Excepciones) por lo que deberá **aplicarse algún criterio y mecanismo de control de errores** (Excepciones) justificado.

## PRACTICA - POO 25-26

Los comandos de E2 se extienden con:

- client add "<name>" (<DNI>|<NIF>) <email> <cashId> ( aceptara NIF y decidirá el tipo de usuario en función del mismo )
- prod add <expiration: yyyy-MM-dd> <category>
- ticket new [<id>] <cashId> <userId> -[c/p/s] (default -p option) ( Ya existe pero debe agregarse la inyección de la visualización que distingue clientes y compañías y distinguir sus peculiaridades de ticket para combinado, producto y servicio)
- ticket print <ticketId> <cashId> (mantienene su funcionalidad pero se amplia a las diferentes visualizaciones entre clientes compañías y tipos de ticket)

### Entregables

- Código fuente y empaquetado (jar) comprimidos en un zip subido a moodle. Obligatorio ejecutable con comando “java -jar <fichero.jar>”. No se corregirán prácticas que no ejecuten el proyecto.
- Deberá presentarse un documento PDF de presentación de la persistencia usada, documentación de las librerías usadas (optativo), explicación de la gestión de excepciones de la aplicación e instrucciones de manejo y verificación. Deben adjuntarse imágenes (pantallazos/screenshots) de la ejecución de los comandos (de todos los disponibles)
- Diagrama UML del modelo propuesto en un formato legible ( PNG, JPG, SVG). Se puede pedir justificación de cambios respecto a la segunda entrega.

### Consideraciones

- Para poder defender el proyecto el código entregado no debe tener errores de compilación y debe ejecutar mediante ejecución en consola.
- Se debe respetar el formato de entrada de comandos y salida mostrados en el enunciado de la práctica en entregas anteriores y en la última entrega (que asume la persistencia vacía)
- No se pueden crear más comandos de los pedidos en el enunciado.
- No se mostrarán por terminal errores y debe aplicarse una política razonable de control de excepciones.
- Deben cumplirse todos los hitos finales del producto, persistencia, parametrización y patrones para que sea válida y válida para corrección.
- Es una entrega **final** por lo que debe atenderse a una entrega de un producto a un cliente real como si de una empresa se tratase, respetando la legibilidad, nombre de archivos, formato, comentarios y presentación como si fuera un producto comercializable.
- Debido a su carácter final, puede no ser necesaria la defensa por parte del equipo, o por todo el equipo de manera simultánea, y se realizará en horario de tutorías pactado por los profesores y el grupo no excediendo más de 24h la fecha de revisión del examen teórico o fecha de subida a actas de notas.