

Tarea Final – Entornos de Desarrollo

Una importante empresa de telefonía ha sacado a licitación un proyecto para programar la aplicación de venta online de terminales tipo smartphone. Tus jefes han logrado que vuestra empresa de desarrollo de software sea la elegida para realizarlo.

En la primera reunión, el cliente (Ti Móvil, SL) ha dado una serie de directrices. Tú no asististe a dicha entrevista, pero un compañero que sí estuvo presente te pasa el siguiente apunte:

Se requiere cuenta para poder usar la aplicación	Disponible en smartphone
Habrà un carrito que permita seleccionar el número de artículos de cada modelo a comprar	Compatible con iOS y Android
Los datos de los modelos disponibles están contenidos en una base de datos propia de Ti Móvil SL que actualiza un técnico, y se deben buscar allí para inicializar la información	25 MB como máximo
La selección de artículos reservará los elementos que se desean comprar durante 5 minutos (actualizando la base de datos), y se lanzará una plataforma de pago seguro. El pago será supervisado por un auditor externo	Buen comportamiento con conexión de datos, no sólo wifi
Una vez realizado el pago, se mostrarán los datos de facturación de la compra	Tiempo de respuesta máximo de conexión segura: 10 segundos

- Parte A: entendiendo el proceso

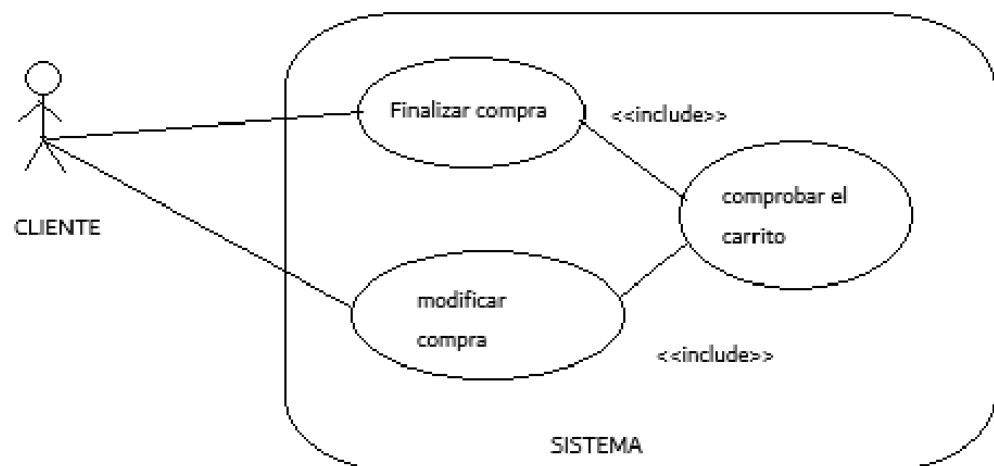
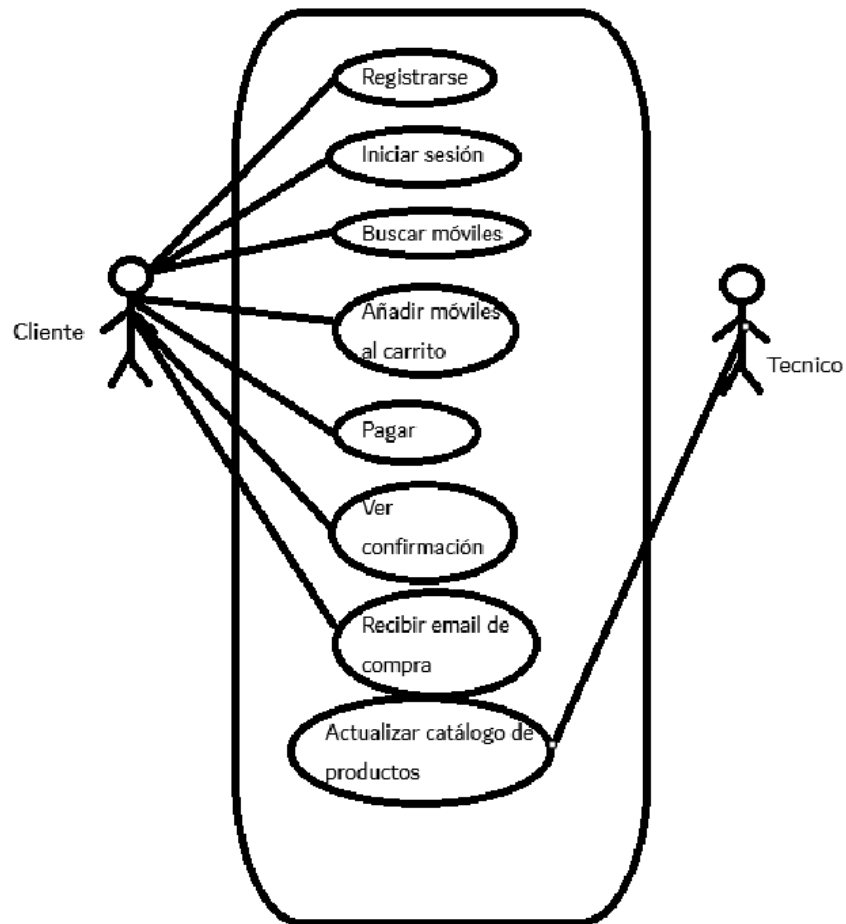
A1) ¿En qué fase del desarrollo estamos?

Estamos en la fase de Desarrollo

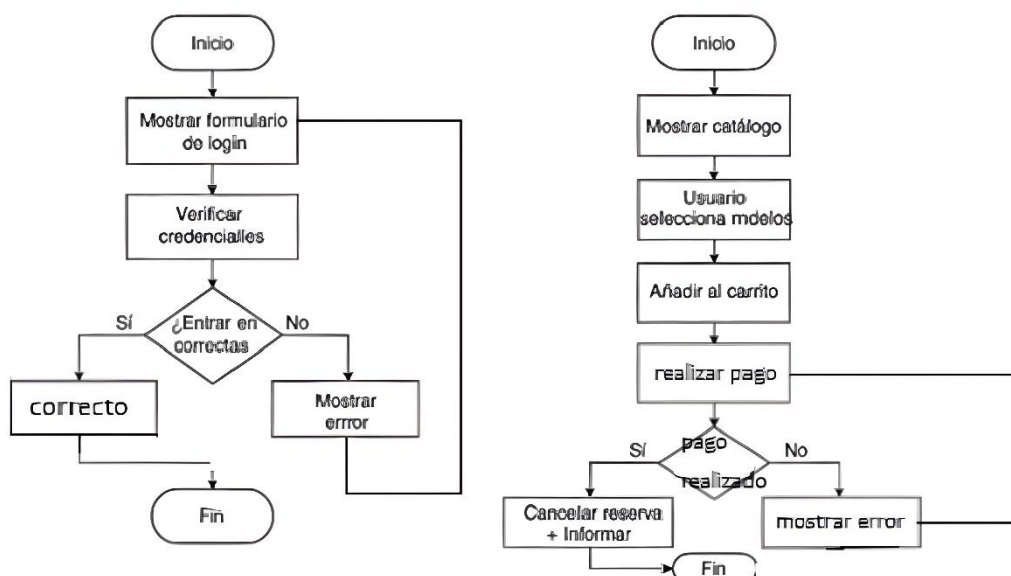
A2) ¿Cómo titularías a cada una de las 2 columnas de requisitos?

Requisitos funcionales y Requisitos no funcionales

A3) Pinta el diagrama de casos de uso para tener claros actores y funcionalidades.



A4) Haz un diagrama de flujo o de cajas del login inicial en la cuenta y de la selección de artículo.



- Parte B: documentando pruebas

B1) El cliente Ti Móvil, SL siempre solicita la entrega de un plan de pruebas inicial (PP) junto al documento de Especificación de Requisitos de Software (ERS). Tu labor es diseñar 5 casos de prueba de cada tipología que te solicitan, plasmándolos en una pestaña distinta del PP.

- Unitarias.

Código Caso	Req. Funcional	Descripción	Instr. Ejecución	Resultado Esperado	Fecha Prueba	OK/NOK
UNIT_01	RF_Carga	Carga &-roid al 30%	1. Crear Android(bat=30); 2. cargar()	Bat=30	01/05/2025	OK
UNIT_02	RF_Llamada	Incremento contador llamada	1. Crear Teléfono; 2. llamar()	llamadas=1	01/05/2025	OK
UNIT_03	RF_PUK	PUK de 8 dígitos	1. Crear Teléfono; 2. getPug()	8 dígitos	01/05/2025	OK
UNIT_04	RF_Linterna	Encender linterna cambia estado	1. Crear Teléfono; 2. encenderLinterna()	isEncendido=true	01/05/2025	OK

UNIT_05	RF_GPS	Actualizar GPS almacena valor	1. Crear PI_iphone; 2. actualizarUbicacion("Madrid")	ubicación="Madrid"	01/05/2025	OK
----------------	--------	-------------------------------	--	--------------------	------------	----

- De integración.

Código Caso	Req. Funcional	Descripción	Instr. Ejecución	Resultado Esperado	Fecha Prueba	OK/NOK
INT_01	RF_LlamadaGlobal	Carga + llamada + totales	1. Crear Android; 2. cargar(); 3. llamar()	cont_ind y cont_tot ++	01/05/2025	OK
INT_02	RF_Carrito	Reserva stock al añadir	1. Mostrar catálogo; 2. añadir carrito	stock provisional-	01/05/2025	OK
INT_03	RF_Pago	Flujo compra + email	1. confirmar compra; 2. procesar pago; 3. enviar email	email recibido	01/05/2025	OK
INT_04	RF_GPS_UI	GPS integra con UI	1. actualizarUbicacion("Sevilla"); 2. mostrar UI	UI muestra "Sevilla"	01/05/2025	OK
INT_05	RF_Pintar	Pintar lanza Paint	1. invocar pintar()	mspaint.exe o msg error	01/05/2025	OK

- De sistema.

Código Caso	Req. Funcional	Descripción	Instr. Ejecución	Resultado Esperado	Fecha Prueba	OK/NOK
SIS_01	RF_FlujoCompletoWeb	Registro→Login→Carrito→Pago	1. registrarse; 2. login; 3. carrito; 4. pagar	flujo sin errores	01/05/2025	OK
SIS_02	RF_ConcReserva	Reservas concurrentes	1. A reserva 2 uds; 2. B reserva simultánea	1 ok; 1 sin stock	01/05/2025	OK
SIS_03	RF_FalloPago	Reversión si pago falla	1. forzar fallo pago; 2. comprobar reversión	reserva cancelada	01/05/2025	OK
SIS_04	RF_StressLogin	1000 logins concurrentes	1. 1000 intentos login	conurrencia sin colapsos	01/05/2025	OK
SIS_05	RF_FlujoMobile	Flujo móvil Android e iOS	1. Android+iOS: login→carrito→pago	ambos completan OK	01/05/2025	OK

B2) Solicitan también 3 pruebas de volumetría/rendimiento. Actualiza el PP con una pestaña adicional que contenga esas 3 pruebas:

Código Caso	Req. Funcional	Descripción	Instr. Ejecución	Resultado Esperado	Fecha Prueba	OK/NOK
-------------	----------------	-------------	------------------	--------------------	--------------	--------

PERF_01	RF_Catálogo	Carga catálogo 50 000 ítems	1. BD con 50 000 regs; 2. solicitar catálogo	Resp. < 2 s	01/05/2025	OK
PERF_02	RF_ReservaConcurrencia	Reservas simultáneas en 500 usuarios	1. 500 sesiones reservan al mismo tiempo	< 5 s por reserva	01/05/2025	OK
PERF_03	RF_PagoConcurrente	Pagos concurrentes (200 hilos)	1. 200 hilos ejecutan flujo de pago	< 10 s por transacción	01/05/2025	OK

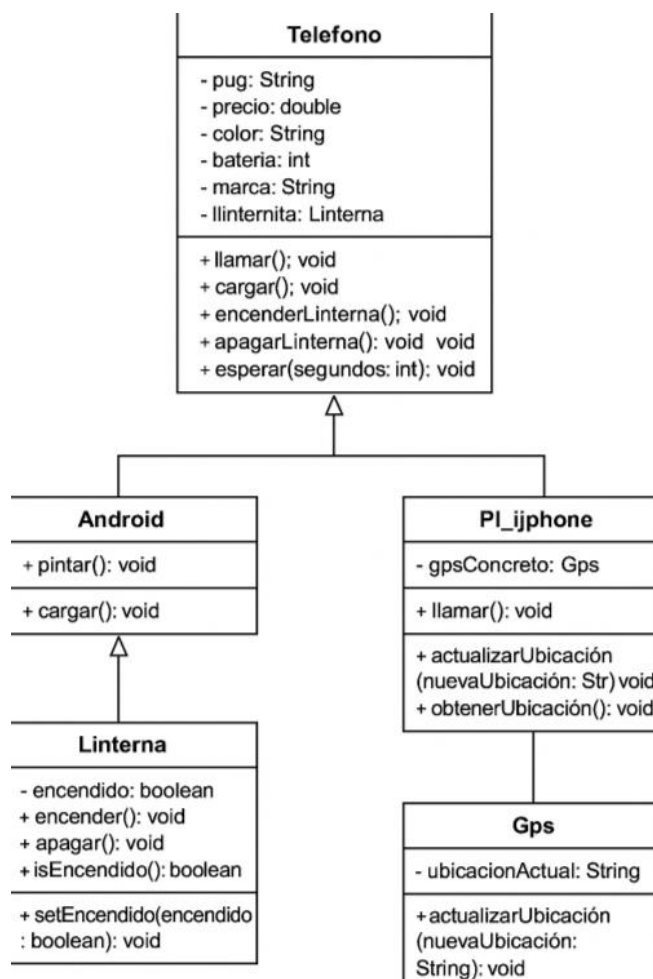
- Parte C: diseñando la solución

A mitad del proyecto recibís una noticia: la Compañía necesita que se programen un par de modelos de smartphones. Como en tu empresa de desarrollo parece que trabajáis bien, os dan también esta parte del proyecto. La información de la que dispones esta vez es:

- Existirán 2 marcas de teléfono: “ π -jiphone” (precio: 700) y “&-roid” (precio: 200);
- Cada teléfono tendrá un PUK asociado (código de seguridad de 8 dígitos). Se generará de forma aleatoria en la creación de cada teléfono y no podrá cambiar.
- El teléfono tendrá un color, elegido por el usuario al comprarlo.
- Todos los teléfonos podrán llamar, en cuyo caso sonará el pitido correspondiente, se iniciará la conversación, y se finalizará (puedes usar el método *esperar* que ya conoces). El número de llamadas realizadas quedará guardado en cada teléfono.
- Además, el “ π -jiphone” dará un mensaje inicial y final extra al llamar, informando de que la llamada se graba en el dispositivo.
- Existirá un control de llamadas totales de la empresa fabricante, por lo que habrá un recuento global de llamadas de todos los teléfonos.
- Todos los teléfonos tendrán un accesorio externo que se podrá conectar llamado “linterna”, que se podrá encender y apagar.
- Los “ π -jiphone” tendrán un dispositivo GPS integrado que dé la ubicación actual. Para ello, tendrá una función de actualizar ubicación, poniendo el dato a disposición hasta una nueva actualización.
- Los “&-roid” tendrán una opción para abrir el Paint y poder dibujar. (Pista: `ProcessBuilder proceso = new ProcessBuilder("mspaint.exe")`).
- No se podrán crear teléfonos sin más, tienen que ser uno de los dos tipos.
- Todos los teléfonos tendrán una capacidad de carga de 0 a 100. El π -jiphone llenará su carga 70 en cada vez que se ponga a cargar, y el &-roid llenará un tercio del nivel previo de carga + 20 extra.

Tú estás liado con la parte de la web de venta de móviles, pero te piden:

C1) Que hagas un diagrama de clases de esta parte del proyecto.



C2) Que codifiques el software básico (mira lo realizado en Programación) y a partir de él hagas un informe de pruebas con 5 pruebas unitarias a partir del software.

Nº	Clase probada	Método / Elemento	Entrada	Resultado Esperado	Resultado Obtenido	OK
1	Android	cargar()	batería actual = 0	batería final = 20 (0/3) + 20 = 20	20	OK
2	PI Iphone	cargar()	batería actual = 10	batería final = 80 (10 + 70)	80	OK
3	Telefono	llamar()	llamadas = 0	llamadas = 1; llamadasTotales++	1	OK
4	PI Iphone	actualizarUbicacion()	"Madrid"	Ubicación actual = "Madrid"	"Madrid"	OK
5	Android	pintar()	—	Se abre mspaint.exe sin error	Paint abierto	OK

- Parte D: interaccionando con el cliente

D1) Han pasado los días, y las fases del proyecto, y el software ya está listo. Te pones a hacer las pruebas, y todo va como la seda. Cuando documentas la última prueba, antes de poder sentir siquiera satisfacción por el trabajo realizado, recibes una llamada. El cliente pide ahora que al final del proceso, una vez mostrada la confirmación de la reserva, se envíe un mail a la dirección de correo del usuario con dicha confirmación. Esto no estaba contemplado al principio. ¿Qué opciones que te ocurren como propuesta para afrontar esta modificación extra?

- Incluir la funcionalidad ahora si todavía hay margen en el plazo de entrega, utilizando una librería para enviar correos sin complicarlo mucho.
- Proponer añadirlo como mejora en una actualización futura, explicando que sería un cambio fuera del alcance inicial.
- Si no se puede hacer desde nuestra aplicación principal, hacer un pequeño programa aparte que envíe el correo automático con los datos de confirmación.

D2) El cliente te pide una reunión para revisar sus pruebas de cliente, que realizaron a partir de las que has documentado en el proyecto. ¿Qué tipo de pruebas han realizado?

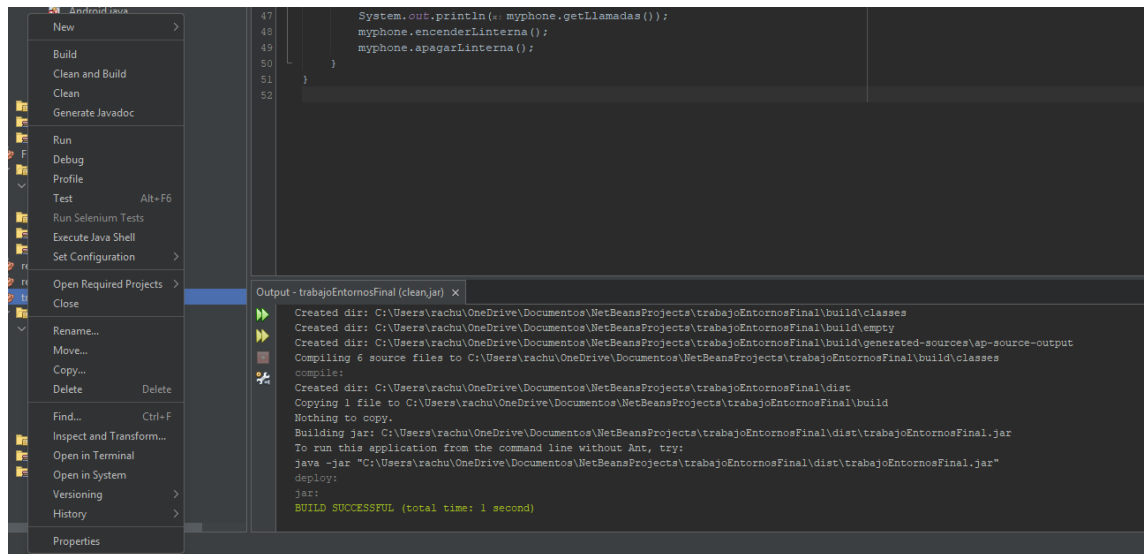
El cliente ha hecho pruebas de aceptación, que son para comprobar que todo funciona como se esperaba. Usan los casos de prueba que les dimos y revisan si la app hace lo que les prometimos. Las hacen ellos mismos, como usuarios reales, y sirven para asegurarse de que pueden usar la aplicación sin problemas antes de dar el visto bueno final.

D3) El cliente te envía una convocatoria para UATs, y te pide que lleves los casos de prueba que utilizaste en las pruebas que documentaste listos para lanzarlos en dicha reunión. ¿Qué importancia tiene que prepares bien esta sesión, con lo que te están pidiendo?

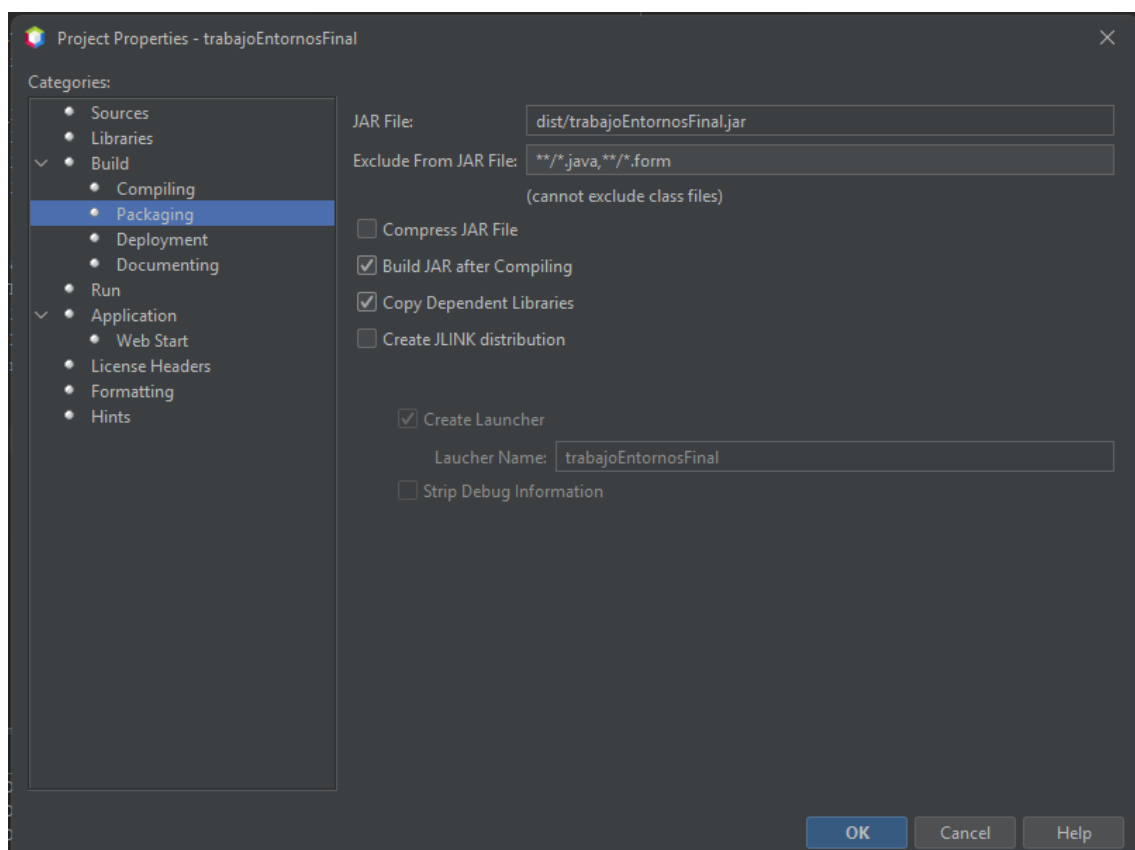
Es una sesión clave porque en ella el cliente va a comprobar si todo funciona como se esperaba. Llevar bien preparados los casos de prueba demuestra que el trabajo está bien hecho y que se han seguido unos criterios claros. También permite detectar cualquier fallo a tiempo, resolverlo y evitar problemas más adelante. En definitiva, es una oportunidad para dejar una buena impresión y asegurar que el cliente aprueba el producto final.

- Parte E: presentación del software

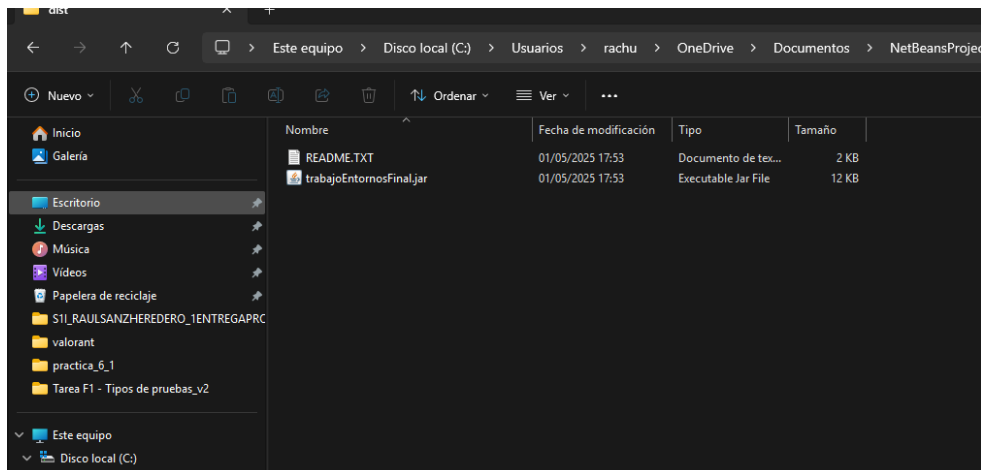
E1) Te solicitan el programa realizado generando su .jar. ¿Sabrías crearlo correctamente? Hazlo.



Haces click en clean and build



Entras en propities y build packaging



E2) Además de la entrega del software se solicita que adjuntes una documentación de la programación de los smartphones (Parte C), que han de tener esta información:

- Todas las clases tienen que tener el autor y la versión.
- La clase principal debe tener además una referencia a los apuntes del tema de Genially.
- La clase Teléfono debe tener documentados todos sus atributos y sus métodos.
- Las clases hijas tendrán una descripción indicando cuál es su madre, y la información sobre los métodos propios (no getters ni setters).

E3) La entrega final se ha de realizar en GitHub. Intenta hacerlo preferiblemente desde Git. Al final darás al cliente (el profesor) una URL donde aparezca toda la información:

- Este Word relleno.
- Los planes de prueba.
- El proyecto en Java.
- Otros documentos que se hayan exigido.