



# SWOP

---

VIABILIDAD, IMPLEMENTACIÓN Y  
COMERCIALIZACIÓN DE UN ECOMMERCE  
BASADO EN UNA DE LAS SAGAS MÁS  
FAMOSAS DEL MUNDO

Sergio Prieto García

Trabajo de fin de ciclo 2ºDAM

Curso 24/25



<b>1. Introducción.....</b>	<b>4</b>
1.1 Descripción general del proyecto.....	4
1.2 Público objetivo.....	5
1.3 Visión, valores y RSC.....	5
<b>2. Contexto y análisis de viabilidad.....</b>	<b>7</b>
2.1 Análisis del entorno económico y social.....	7
2.2 Análisis tecnológico.....	9
2.3 Análisis de riesgos y contingencias.....	9
<b>3. Planificación del proyecto.....</b>	<b>11</b>
3.1 Fases del desarrollo.....	11
3.2 Diagrama de Gantt.....	12
<b>4. Justificación del proyecto.....</b>	<b>12</b>
4.1 Necesidad detectada en el mercado.....	12
4.2 Propuesta de valor.....	14
4.3 Diferenciadores frente a otras tiendas.....	14
4.4 Estrategia de promoción.....	14
<b>5. Arquitectura del sistema.....</b>	<b>16</b>
5.1 Backend: Spring Boot, PostgreSQL, JWT.....	16
5.2 API REST: estructura y seguridad.....	17
5.3 Frontend móvil: Android Studio y Java.....	18
<b>6. Sistema de pujas.....</b>	<b>22</b>
6.1 Funcionamiento general del sistema.....	22
6.2 Reglas de puja y gestión de usuarios.....	23
6.3 Control de stock y finalización de subasta.....	24
<b>7. Base de datos y estructura interna.....</b>	<b>25</b>
7.1 Diagrama entidad-relación (E/R) y UML del Backend.....	25
7.2 UML del Frontend.....	25
<b>8. Recursos utilizados.....</b>	<b>25</b>
8.1 Recursos materiales y humanos.....	25
8.2 Proveedor oficial: OcioStock.....	26
<b>9. Aspectos legales y técnicos.....</b>	<b>26</b>
9.1 RGPD y condiciones de uso.....	26
9.2 Licencias de distribución y propiedad intelectual.....	26
9.3 Coste de publicación en Google Play.....	27
<b>10. Previsión económica.....</b>	<b>27</b>
10.1 Presupuesto inicial y ayudas públicas.....	27
10.2 Estimación de ingresos y gastos (1 año).....	27
10.3 Punto de equilibrio y rentabilidad esperada.....	28
<b>11. Futuras ampliaciones.....</b>	<b>28</b>
11.1 Chat global entre clientes.....	28

11.2 Mejora de diseño UI/UX y experiencia de compra.....	29
11.3 Automatización de stock y logística.....	29
<b>12. Conclusiones.....</b>	<b>30</b>
<b>13. Bibliografía.....</b>	<b>30</b>
<b>14. Repositorios Github públicos y despliegue.....</b>	<b>32</b>

# 1. Introducción

## 1.1 Descripción general del proyecto

**SWOP** es una tienda móvil dedicada en exclusiva al universo Star Wars. Su propuesta combina dos ejes:

1. **Catálogo reducido**, compuesto únicamente por productos oficiales procedentes de OcioStock y otros distribuidores con licencia Disney/Lucasfilm, y
2. **Sistema de pujas** diarias que permite a la comunidad competir por piezas limitadas a partir de la mitad de su precio de venta al público.

### Parte técnica

La plataforma se apoya en un backend desarrollado con **Spring Boot** que expone una API REST segura mediante **JWT**. Los datos (usuarios, productos, pujas y pedidos) se persisten en **PostgreSQL** y ambos se despliegan en un proyecto “Free Trial” de **Render**, lo que garantiza autoscaling y HTTPS sin inversión en hardware (bajo ciertas restricciones y capacidades de esa prueba gratuita).

El front-end es una **app nativa Android** escrita en Java. Utiliza Retrofit para consumir la API, notificaciones push para avisar de pujas mejoradas y creación de pedidos para los clientes. Todo el flujo (navegar, pujar y comprar) está pensado para completarse de la manera más rápida posible.

El sistema de subastas corre una tarea programada que abre un único producto cada 24 h, muestra un contador regresivo y adjudica el artículo al mejor postor. El control de stock reserva la unidad mientras dura la puja, evitando sobreventas.

### Parte económica

El proyecto se financia con **5 000 € de aportación propia** y la ayuda autonómica

**TR880A** (5 500 €), de manera que dispone de **10 500 €** para inversión inicial sin recurrir a préstamos.

En su primer año el objetivo es vender **600 unidades** a un precio medio de 24,20 €, lo que genera una facturación aproximada de **14 500 €**. Restados los costes variables (compra, embalaje, comisión, parte del envío) y los costes fijos mínimos (hosting, marketing básico, cuota de autónomo), el beneficio neto previsto ronda los **2 600 €**. Esta cifra modesta, pero positiva, confirma la viabilidad del modelo y permite reinvertir en stock y publicidad.

A medio plazo la hoja de ruta contempla tres mejoras: chat global, UI renovada y logística automatizada y la posible incorporación de un especialista en marketing cuando las ventas alcancen el umbral de 3 000 unidades anuales. Con gastos fijos aún reducidos y un margen unitario estable, el crecimiento dependerá sobre todo de aumentar la comunidad de usuarios y del éxito de las subastas.

## 1.2 Público objetivo

SWOP se dirige a la comunidad de seguidores de Star Wars en España, de los cuales existen tres perfiles principales:

- **Coleccionista “hard-core”**
  - Compra figuras exclusivas, réplicas numeradas y objetos vintage.
  - Valora la autenticidad y la escasez; es el principal usuario del módulo de pujas.
- **Fan habitual y “self-gifter”**
  - Sigue las series de Disney+, quiere camisetas, Funkos y menaje temático.
  - Agradece la comodidad móvil y los envíos rápidos para auto-regalarse.
- **Comprador ocasional de regalos**
  - No necesariamente fan, pero busca un detalle para amigos o hijos fanáticos.
  - Necesita confianza en la licencia oficial y un proceso de pago sin fricción.

Todos comparten dos rasgos clave: compran mayoritariamente **desde el móvil** y demandan **pruebas de autenticidad** para evitar falsificaciones. El enfoque “app-first” de SWOP y el acuerdo con OcioStock encajan con esas expectativas.

## 1.3 Visión, valores y RSC

### Visión

Convertirnos en la referencia española para el colecciónismo y el regalo Star Wars; un

espacio móvil donde cada fan encuentre piezas auténticas, experiencias exclusivas y una comunidad con la que compartir su afición.

## Valores

<u>Valor</u>	<u>Cómo se materializa</u>
<b>Autenticidad</b>	Solo producto con licencia Disney/Lucasfilm y proveedores verificados.
<b>Comunidad</b>	Chat global (próximamente), decoración de la app acorde al producto y participación activa en redes sociales
<b>Transparencia</b>	Fichas de artículo con fotos reales, histórico de pujas y política de precios clara.
<b>Sostenibilidad</b>	Embalaje de cartón reciclado, cinta de papel y mensajería que compensa CO <sub>2</sub> .

## Responsabilidad Social Corporativa (RSC)

1. **Impacto ambiental reducido:** todo el packaging es recicitable; a medio plazo se contratará el servicio SEUR “Cero CO<sub>2</sub>” para neutralizar emisiones por envío.
2. **Economía local:** la operativa y el almacenaje se realizan en Galicia; la contratación futura de un especialista en marketing priorizará talento local.
3. **Accesibilidad:** la app sigue las pautas WCAG básicas—contraste alto, texto escalable y navegación por lectura de pantalla—para que cualquier fan pueda comprar sin barreras.
4. **Cultura y comunidad:** se destinará un 1 % del beneficio anual a patrocinar eventos fan (proyecciones benéficas, clubs de lectura de novelas SW) y a ONG infantiles que promuevan el acceso a la tecnología.

Con esta combinación de autenticidad, cuidado medioambiental y apoyo a la comunidad fan, SWOP aspira a ser algo más que una tienda: un punto de encuentro responsable para los amantes de la saga.

## 2. Contexto y análisis de viabilidad

*¿Por qué el mercado Star Wars sigue siendo una galaxia rentable?*

### 2.1 Análisis del entorno económico y social

El merchandising de Star Wars ha superado los **32.000 millones de dólares** desde 1977. En un año sin estrenos importantes el mercado sigue rozando los **3.000 millones**, y cuando llega una nueva película o serie la cifra puede saltar a **5–7.000 millones**.

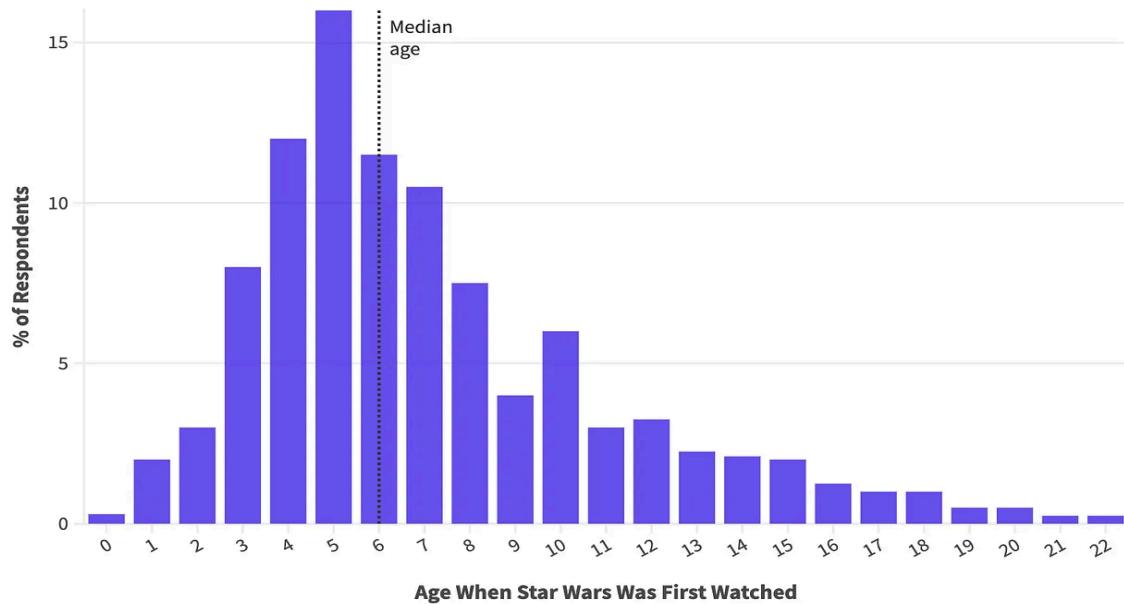
#### Artículo de las cifras

Los fans empiezan muy jóvenes: el 90 % vio su primera película antes de los trece años, y la afición se mantiene estable entre Boomers, Gen X, Millennials y Gen Z. Esta mezcla de generaciones garantiza un público amplio y poder adquisitivo variado.

#### **When Does Star Wars Fandom Begin?**

*Survey: At what age did you first watch Star Wars?*

Source: "Star Wars Fandom Survey: Age and Nostalgia" by Mark H. White, PHD

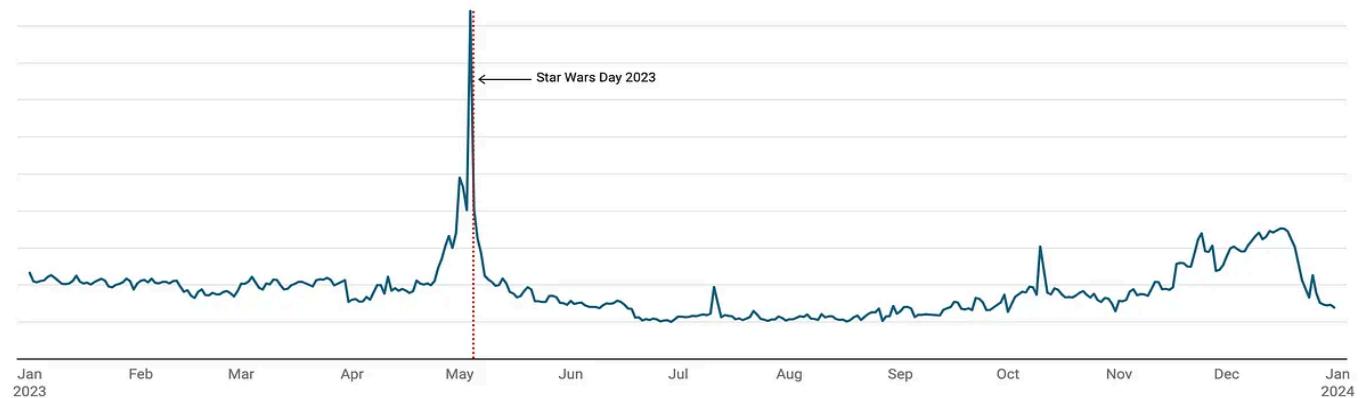


El **4 de mayo** ("Star Wars Day") es un caso aparte: las tiendas online registran aumentos de hasta un **500 %** en ventas. Para SWOP supone la gran fecha del año para lanzar subastas y ofertas especiales.

## Change in Consumer Demand for Star Wars Products in 2023

The change in consumer demand on Amazon for Star Wars products on Star Wars Day compared to every other day of 2023

— Demand for Star Wars Products



*Consumer demand is a measure of the number of consumers on Amazon shopping for a given type of item during a given period.*

Aunque la base sigue siendo enorme, los últimos estrenos han provocado cierta “fatiga Star Wars”: *Solo* perdió 77 millones en taquilla y *The Acolyte* ha recibido críticas duras. Por eso mi estrategia se basa en stock muy curado y en dar sensación de exclusividad a través de pujas.

Además, el **comercio electrónico en España vive su mejor momento**. Los estudios de Statista y Baymard señalan que la facturación online alcanzará los **37.911 millones de euros en 2025**, con un crecimiento anual medio del 8 % y más de **28 millones de compradores habituales** — es decir, tres de cada cuatro internautas españoles ya compra por internet. Este contexto favorece a SWOP porque el proyecto nace directamente en formato móvil: la app Android llega a un público acostumbrado a comprar sin pasar por una web de escritorio y, por tanto, elimina fricciones de pago y carrito

Fuente: [EcommerceNews](#) y [Latevaweb](#)

## **2.2 Análisis tecnológico**

El proyecto SWOP se apoya en servicios “llave en mano” que eliminan la necesidad de comprar hardware y permiten escalar sólo cuando aparezca la demanda.

### **Backend en Render**

Render ofrece un plan gratuito con 500 h mensuales y HTTPS integrado. El autoscaling se activa si el tráfico crece, de modo que sólo se paga cuando realmente se consumen más recursos.

### **Base de datos PostgreSQL en Render**

El servicio gratuito de Render también aplica para la base de datos, que en este caso tiene un límite de 1gb de datos, perfecto para pruebas e incluso para carga pequeña de datos ya en producción.

### **Seguridad estándar**

Las contraseñas se guardan con BCrypt y los usuarios se identifican con JWT. Todo el tráfico viaja en HTTPS, lo que cumple las recomendaciones de Android 9+ y la LSSI.

### **Pagos móviles (para versiones futuras, no implementado por temas legales)**

Stripe y PayPal SDK facilitan el cobro con tarjeta o saldo PayPal desde la app sin almacenar datos sensibles.

## **2.3 Análisis de riesgos y contingencias**

El proyecto afronta cinco riesgos principales que conviene tener presentes desde el primer día.

### **Cansancio de la marca**

Entre 2022 y 2024 las ventas online de productos Star Wars cayeron de forma continua, según la consultora Pattern, tras años de crecimiento. Para evitar que la fatiga del público afecte a SWOP, el catálogo se mantendrá muy selecto (por ejemplo, solo 50

referencias iniciales de las más buscadas) y se reforzará la sensación de exclusividad con subastas semanales de piezas limitadas, como el Funko POP “Grogu” que OcioStock distribuye a 11 € + IVA y cuyo PVP ronda los 18 €

#### Falta puntual de mercancía

OcioStock garantiza entregas en 24h desde Madrid y no exige pedido mínimo, pero si un artículo se agota (p.e. la réplica del sable de Luke), se recurrirá a mayoristas alternativos ya negociados:

- [\*\*Cerdá Group\*\*](#), que sirve mochilas y accesorios licenciados con pedido mínimo de 150 €.
- [\*\*Storline\*\*](#), especializado en menaje; su botella de aluminio Star Wars cuesta 8,99 € al por menor y se ofrece a tiendas con un descuento aproximado del 35 %. El uso de tres proveedores asegura reposición en menos de 48 h para los productos estrella.

#### Caídas de servicio o picos de tráfico

El backend se desplegará en [Render](#). El plan gratuito entra en reposo tras 15 min de inactividad, de modo que para el lanzamiento se contratará el plan “Professional” (19 US \$/mes) que mantiene la instancia activa y ofrece 99.9 % de disponibilidad. La base de datos PostgreSQL en Render dispone de 1GB de espacio gratuito; cuando supere ese consumo se pasará a un plan mayor anteriormente mencionado. Además, en un futuro, [Uptimerobot](#) (plan free) monitorizará la API cada 5 min.

#### Cumplimiento legal

Todos los artículos proceden de distribuidores oficiales, de modo que la tienda evita el riesgo de infracción de marca. La redacción de Política de Privacidad, Términos de Uso y avisos RGPD se contratará en un paquete estándar con una [asesoría TIC local](#). Para pagos, Stripe cobra el 2,9 % + 0,30 € por transacción en España y PayPal 2.90% más una tarifa fija de 0.35€, comisiones que se repercutirán en el margen de venta.

#### Variaciones euro–dólar.

Mucho del stock se importa en USD. En 2024 el cambio ha oscilado entre 0,90 € y 0,94 € por dólar; un pedido de 5 000 US \$ puede encarecerse 200 € en un trimestre. Para proteger el margen se revisarán precios cada tres meses y se pactarán tarifas en euros siempre que el proveedor lo permita.

Con estas medidas específicas —proveedores alternativos cerrados, plan de hosting de pago estable, copias de seguridad diarias y políticas legales profesionales— SWOP reduce los riesgos operativos y financieros a niveles asumibles para un negocio en fase de arranque.

## 3. Planificación del proyecto

### 3.1 Fases del desarrollo

#### Fase 1 – Planificación e idea de negocio

24 mar → 7 abr. 2025 (2 semanas)

Se definen el concepto, el público y el modelo de ingresos. Se compara el merchandising Star Wars con otros nichos y se decide usar pujas para aportar exclusividad. Resultado: estructura funcional de la app, lista corta de proveedores y primeros bocetos de interfaz.

#### Fase 2 – Backend Spring Boot + JWT

7 abr → 5 may. 2025 (4 semanas)

Se crea la API REST con Spring Boot, se modelan las entidades (usuarios, productos, pujas, pedidos) y se incorpora seguridad con JWT y BCrypt.

#### Fase 3 – Frontend Android + despliegues

5 may → 9 jun. 2025 (5 semanas)

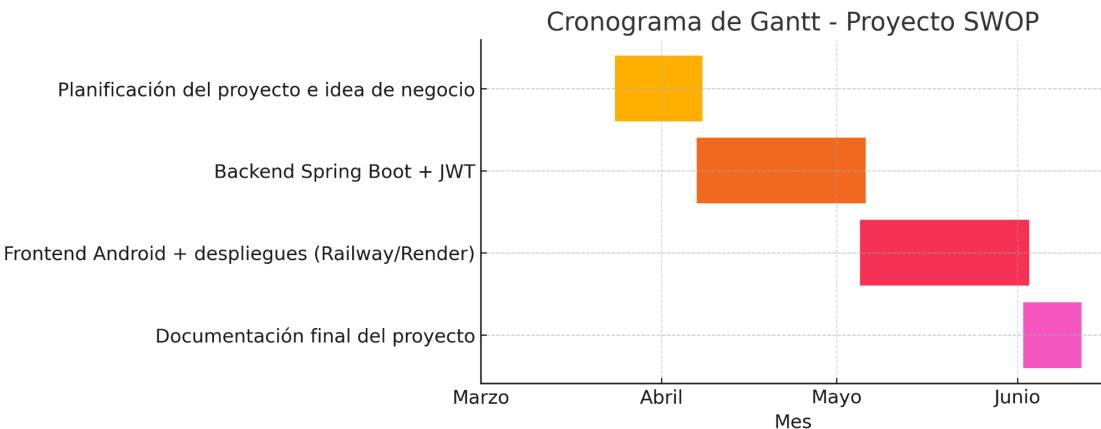
Desarrollo de la app en Android Studio (Java): listado de productos, carrito, pantalla de pujas en tiempo real. Paralelamente se sube la base de datos y el backend a Render, dejando todo accesible por HTTPS.

#### Fase 4 – Documentación final

10 jun → 11 jun. 2025 (Media semana)

Redacción de la memoria técnica y empresarial, capturas de pantalla, diagramas UML, Gantt definitivo y preparación de la presentación para la defensa.

## 3.2 Diagrama de Gantt



## 4. Justificación del proyecto

### 4.1 Necesidad detectada en el mercado

#### Un mercado enorme, pero disperso y poco especializado

El merchandising oficial de Star Wars mueve alrededor de 3 000 millones de dólares cada año y en años de estreno roza los 7.000 millones. Sin embargo, la mayor parte de esa cifra se reparte entre grandes marketplaces (Amazon, eBay) y tiendas generalistas donde el fan debe bucear entre miles de referencias de otras franquicias, soportar largos plazos de entrega o arriesgarse a comprar réplicas no oficiales. La propia Disney Store limita su catálogo a líneas de temporada y apenas ofrece producto de colecciónista. Existe, por tanto, un hueco claro: una tienda vertical, 100 % Star Wars, que garantice licencias oficiales y agrupe en un único lugar las piezas que un seguidor realmente busca.

#### La exclusividad enfermiza

Cuando se siente que una pieza es escasa o “sólo por hoy”, el fan reacciona con una aceleración de compras mucho mayor que en un caso normal. Sin embargo, los portales masivos no ofrecen herramientas para jugar con la escasez controlada. Un

sistema de pujas integrado en la app permite reproducir esa sensación de “edición limitada” cualquier semana del año, al tiempo que incrementa el ticket medio – los coleccionistas compiten y el valor percibido sube.

#### Confianza y autenticidad frente a todas las falsificaciones

En grupos de Facebook y foros especializados proliferan quejas sobre sables de luz o figuras “réplica” que llegan sin sello oficial. Centralizar la oferta en SWOP, con compra directa a mayoristas autorizados como OcioStock, elimina la duda del comprador y pone en valor la procedencia. Para un coleccionista, saber que el Funko o el sable está respaldado por Disney/Lucasfilm es casi tan importante como el precio.

#### Comodidad móvil para las nuevas y actuales generaciones

Más del 75 % de las compras online en España ya se realizan desde el móvil. Aun así, gran parte del merchandising se vende en webs “desktop first” y formularios poco ágiles. SWOP nace como app nativa Android: un par de toques bastan para pujar, pagar con Stripe o PayPal y volver al episodio de The Mandalorian. El proceso refleja la inmediatez con la que el fan consume la saga en *Disney+*.

#### Fatiga de oferta genérica, auge de la “curación”

Después de un lustro de estrenos encadenados, el usuario está cansado de ver siempre el mismo producto licenciado en todas partes. Nuestro catálogo inicial se limita a 50 referencias cuidadosamente elegidas: réplicas de sable con número de serie, Funkos “chase” difíciles de conseguir, réplicas vintage reeditadas y piezas de menaje originales. Menos es más: cada nuevo artículo se anuncia como un evento y la tienda se percibe como un “club de hallazgos” en lugar de un bazar infinito.

#### Conclusión

Existe público, poder adquisitivo y pasión demostrada; lo que falta es un espacio móvil que combine especialización, confianza y juego de exclusividad. SWOP cubre ese vacío: reúne sólo producto oficial, añade un sistema de pujas que refuerza la escasez y empaqueta todo en una experiencia adaptada al teléfono, exactamente donde el fan vive hoy su afición.

## 4.2 Propuesta de valor

**SWOP** es la primera tienda móvil en España dedicada al 100 % a Star Wars que combina catálogo curado + pujas exclusivas.

**Solo Star Wars:** todos los artículos proceden de distribuidores oficialmente licenciados; el usuario compra con total confianza y sin bucear en marketplaces genéricos.

**Catálogo curado:** menos de 50 referencias activas, seleccionadas por rareza, calidad y demanda real (réplicas numeradas, Funkos “chase”, vintage reeditado). Cada producto nuevo se anuncia como un acontecimiento.

**Sistema de pujas in-app:** subastas diarias y campañas especiales (p. ej. 4 de mayo) que devuelven la emoción de la escasez, aumentan el ticket medio y alimentan la comunidad colecciónista.

**Experiencia móvil inmediata:** app nativa Android, pago en dos toques con Stripe/PayPal (futuramente) y notificaciones push para alertas de subasta; perfecta para la generación que ya consume la saga en streaming desde el smartphone.

**Programa de puntos y ranking de ganadores:** cada compra o puja genera créditos que se reinvierten en la propia plataforma, fomentando recurrencia y sentido de pertenencia.

## 4.3 Diferenciadores frente a otras tiendas

**Solo Star Wars:** la única app española con catálogo 100 % dedicado a la saga.

**Catálogo curado:** máximo 50 referencias en rotación; cada alta es un evento.

**Pujas exclusivas:** subastas semanales dentro de la app, inexistentes en la competencia.

**Experiencia móvil rápida:** registro sencillo, pago Tarjeta/PayPal en dos toques.

**Autenticidad garantizada:** todo procede de mayoristas oficiales Disney/Lucasfilm.

**Gamificación:** puntos por compra y ranking mensual de “Maestros Jedi”.

**Logística 24/48 h** y embalaje reciclado.

## 4.4 Estrategia de promoción

Pre-lanzamiento

Dos semanas antes de publicar la aplicación se iniciará una cuenta atrás diaria en Instagram y TikTok. Cada vídeo (15-20 s) mostrará un producto destacado y redirigirá a una landing sencilla con un formulario «Avísame cuando abra». Para acelerar la captación se sorteará un Funko POP de The Mandalorian entre los inscritos: el coste real del premio ronda los 15 €, suficiente para reunir unos 300 correos y generar conversación orgánica.

#### Día de lanzamiento

Durante las primeras 72 horas la app aplicará un cupón de bienvenida del 15 % en la primera compra. Ese mismo día se enviará una nota breve a blogs frikis nacionales (Xataka Basics, La Casa de EL, Zona Negativa) y se activará una campaña “app-install” en Instagram de 35 €, destinada a sumar al menos un centenar de descargas iniciales que den tracción al algoritmo.

#### Promoción continua

Una vez en marcha, la presencia se sostendrá con un reel diario (unboxing, curiosidad o meme) y con micro-influencers cosplay que recibirán producto por valor de 30 € a cambio de un vídeo. Paralelamente se invertirán 30 € mensuales en anuncios segmentados a fans de Star Wars (18-45 años, España), suficiente para mantener un flujo estable de clics y descargas.

#### Campañas de pico

El 4 de mayo (Star Wars Day) se lanzarán subastas “flash” cada hora, anunciadas mediante notificación push. En Black Friday y Navidad se repetirán promociones, la primera con packs “Regalo Jedi” y envío gratis a partir de 40 €, la segunda con un pequeño calendario de Adviento de pujas diarias.

#### Fidelización

Los compradores recibirán una newsletter mensual con lanzamientos y el ranking de ganadores de subastas. El sistema de puntos (1 € gastado = 1 punto; 100 puntos = 5 € para pujar) premiará la recurrencia, y una encuesta enviada 48 h tras la entrega otorgará 50 puntos extra a quienes respondan, generando feedback continuo.

#### Presupuesto anual

El plan completo se cubre con unos 600 € al año: 360 € en publicidad de

mantenimiento, 40 € en refuerzos para el 4 de mayo y Black Friday, y unos 200 € entre sorteos y colaboraciones. Con esa inversión se aspira a 1 500 descargas y 200 ventas en los primeros seis meses, capitalizando el elemento diferencial de la tienda: las pujas de artículos exclusivos.

## 5. Arquitectura del sistema

### 5.1 Backend: Spring Boot, PostgreSQL, JWT

El motor de SWOP está construido con **Spring Boot 3**; arranca con una interacción con el servidor o encendiéndose manualmente. Render despliega con HTTPS automático. La base de datos es **PostgreSQL** en Render y las entidades se generan mediante **Spring Data JPA**, por lo que el script .sql de /resources se conserva sólo como referencia.

#### Modelo de dominio

- **Customer** guarda las credenciales y el rol (USER / ADMIN) y se relaciona con muchas **Order** y muchas **Bid**.
- **Order ↔ Product** es N:M y se resuelve con **OrderDetail**, donde se guarda cantidad y precio final.
- **Bid** funciona también como tabla N:M entre Customer y Product; indica la cantidad ofertada y mantiene siempre *BidStatus.PENDING* hasta que el frontend, en versiones futuras, marque *ACCEPTED* o *REJECTED* para alimentar el ranking.
- **Product ↔ Category** es N:M; un sable láser puede ser simultáneamente “Rélicas” y “Coleccionismo”, por ejemplo.

#### Enums de estado y métodos de pago centralizan reglas estáticas:

- enum BidStatus { PENDING, ACCEPTED, REJECTED }
- enum OrderStatus { PROCESSED, DELIVERED, CANCELED }
- enum PaymentMethod { CREDIT\_CARD, PAYPAL, BANK\_TRANSFER }

#### Arquitectura MVC

- **Repository**: cada entidad principal extiende *JpaRepository*, exponiendo CRUD sin

SQL escrito a mano.

- **Service**: aloja la lógica: *OrderService* agrega o quita *OrderDetail*, *BidService* valida que la puja supere la actual y que el producto siga en subasta.
- **Controller**: publica los endpoints REST y delega en el servicio.
- **DTO**: aísla la capa API del modelo JPA; *ModelMapper* convierte casi todo de forma automática.  
Esta división mantiene el código limpio, facilita pruebas sin arrancar la web y permite insertar reglas (por ejemplo, bloquear pujas si el stock es cero) sin tocar los controladores.

## 5.2 API REST: estructura y seguridad

### Rutas principales ([/api](#))

POST /auth/login	↳ devuelve JWT
GET /products	↳ listado paginado
GET /categories	↳ listado categorías
GET /products/{id}	↳ detalle de producto
POST /bids	↳ crear puja (requiere USER)
GET /bids/my	↳ pujas del usuario
POST /orders	↳ confirmar pedido ganador
GET /customers	↳ listado clientes [ADMIN]
GET /orders	↳ listado pedidos [ADMIN]

*Los métodos sensibles llevan @PreAuthorize("hasRole('ADMIN')").*

### Seguridad

- **JWT** HS256, vigencia 24 h; se envía en *Authorization: Bearer <token>*
- Contraseñas cifradas con **BCrypt PasswordEncoder**.
- Dos roles: **USER** (comprador) y **ADMIN** (gestión interna; hoy, sólo un usuario)

*hardcodeado*, en un futuro se reflejará en la base de datos).

- Filtros de excepción convierten errores en 401, 403, 404 o 422 para que la app pueda reaccionar.

### **Flujo de subasta y pedidos**

- Una puja solo puede lanzarse si el producto está en subasta y el cronómetro de 24 h sigue activado.
- Al expirar el contador, el backend detecta la puja más alta, genera una *Order* con *OrderStatus.PROCESSED* y reserva el stock; la *Bid* ganadora permanece *PENDING* a la espera de funciones de ranking que se implementarán próximamente.
- Cuando el cliente pague (próxima iteración con *CREDIT\_CARD*, *PAYPAL*, *BANK\_TRANSFER*), un webhook actualizará la orden a *DELIVERED*. Si no paga en 48 h pasará a *CANCELED* y el producto volverá al catálogo. (no implementado debido a requisitos legales para procesar pagos)

Con esta estructura la API es segura, comprensible y fácil de extender: basta añadir nuevos DTO y servicios sin romper la lógica principal.

## **5.3 Frontend móvil: Android Studio y Java**

La app de SWOP sigue la arquitectura **MVVM + Repositorios** y se organiza en paquetes que reproducen las secciones funcionales de la tienda y de las pujas. A continuación se describe la finalidad de cada paquete y los flujos clave de sesión y seguridad.

## **Estructura de carpetas (sin layouts, drawables ni iconos, solo lógica)**

```
java/com/example/swop
├── bids
│   ├── BidsActivity.java
│   └── BidsVM.java
├── data
│   ├── local
│   │   └── SecurePrefs.java
│   ├── remote
│   │   ├── api
│   │   │   ├── AuthApi.java
│   │   │   ├── BidApi.java
│   │   │   ├── CategoryApi.java
│   │   │   ├── CustomerApi.java
│   │   │   ├── OrderApi.java
│   │   │   └── ProductApi.java
│   │   ├── models
│   │   │   ├── AuthRequest.java
│   │   │   ├── BidDto.java
│   │   │   ├── BidStatus.java
│   │   │   ├── CategoryDto.java
│   │   │   ├── CustomerDto.java
│   │   │   ├── LoginResponse.java
│   │   │   ├── OrderDetailDto.java
│   │   │   ├── OrderDto.java
│   │   │   ├── OrderStatus.java
│   │   │   ├── PaymentMethod.java
│   │   │   └── ProductDto.java
│   │   ├── AuthInterceptor.java
│   │   └── RetrofitClient.java
│   ├── repository
│   │   ├── BidRepository.java
│   │   ├── CategoryRepository.java
│   │   ├── CustomerRepository.java
│   │   ├── OrderRepository.java
│   │   └── ProductRepository.java
│   ├── sesión
│   │   ├── ReloginCallback.java
│   │   └── SessionManager.java
│   └── útil
│       ├── ApiCallback.java
│       ├── JwtUtils.java
│       └── NotifUtils.java
└── login
    ├── LoginCallback.java
    ├── LoginDialogFragment.java
    ├── LoginResult.java
    └── RegisterDialogFragment.java
├── profile
    ├── auctions
    │   ├── AuctionOrderCardViewHolder.java
    │   └── AuctionOrderListAdapter.java
    ├── orders
    │   ├── OrderCardViewHolder.java
    │   └── OrderListAdapter.java
    ├── EditProfileDialogFragment.java
    ├── ProfileActivity.java
    └── ProfileVM.java
└── shop
    ├── cartwindow
    │   ├── CartActivity.java
    │   ├── CartItem.java
    │   ├── CartPreferences.java
    │   ├── CartProductCardViewHolder.java
    │   ├── CartProductListAdapter.java
    │   └── CartVM.java
    ├── categorywindow
    │   ├── CategoryActivity.java
    │   └── CategoryVM.java
    ├── detailwindow
    │   ├── ProductDetailActivity.java
    │   └── ProductDetailVM.java
    ├── mainwindow
    │   ├── MainWindow.java
    │   └── MainWindowVM.java
    ├── productslist
    │   ├── ProductCardViewHolder.java
    │   └── ProductListAdapter.java
    └── PaymentMethodDialogUtil.java
├── BaseActivity.java
└── WelcomeActivity.java
```

## **Funcionalidad de los paquetes**

- java/com/example/swop

### **/bids**

- **BidsActivity** y **BidsVM** manejan la pantalla de subastas: muestran el producto en puja, comprueban el temporizador de 24 h y, al expirar, envían la puja ganadora al backend para que éste genere la **Order**.
- Si nadie ha pujado, la lógica selecciona un nuevo producto aleatorio para la siguiente ronda.

### **/data/local**

SecurePrefs crea unas **EncryptedSharedPreferences** donde se guarda el token y los datos mínimos del usuario.

### **/data/remote/api**

Interfaces Retrofit (AuthApi, ProductApi, etc.) que reflejan los endpoints del backend.

- **AuthInterceptor** intercepta cada request: si existe token lo añade como header **Bearer**; si no, la llamada sale sin autorización.
- **RetrofitClient** configura Gson para serializar *LocalDateTime*.

### **/data/repository**

ProductRepository, BidRepository... consumen las APIs y exponen *LiveData/Callbacks* a los ViewModels.

### **/data/sesion**

SessionManager guarda el token y verifica fechas de expiración; si caduca, llama a *ReloginCallback* para obtener uno nuevo sin pedir credenciales.

### **/data/útil**

JwtUtils extrae claims, NotifUtils lanza notificaciones push locales y ApiCallback estandariza éxito / error.

### **/login**

Diálogos modales (*LoginDialogFragment*, *RegisterDialogFragment*) que se abren donde sea necesario; devuelven un *LoginResult* al Activity anfitrión mediante *LoginCallback*.

### **/profile**

ProfileActivity y ProfileVM muestran pedidos (*OrderListAdapter*) y pujas ganadas no pagadas (*AuctionOrderListAdapter*). Desde aquí se puede editar la cuenta (*EditProfileDialogFragment*) o pagar pujas pendientes seleccionando un método en *PaymentMethodDialogUtil*.

## **/shop**

El corazón de la tienda:

- **/mainwindow** → *MainWindowVM* obtiene tres productos recomendados (aleatorios) y lanza la búsqueda que funciona por coincidencia léxica.
- **/categorywindow** → filtra por Lego o Merchandising; se invoca desde el toolbar.
- **/detailwindow** → página de detalle con botón “Añadir al carrito”.
- **/cartwindow** → carrito persistente por usuario en *CartPreferences*, con pago en un clic.

## ***BaseActivity***

Clase común con fondo animado, toolbar configurable y control de login: cualquier intento de acceder a secciones protegidas (del toolbar) sin token válido abre el *LoginDialogFragment*.

## ***WelcomeActivity***

Pantalla inicial que redirige al usuario al apartado **Tienda** o a **Pujas**. Sólo la tienda, categorías y detalles se pueden navegar sin sesión; al pulsar “Añadir al carrito” o “Pujar”, se exige login.

## **Flujo de sesión y renovación de token**

1. El usuario inicia sesión → *AuthApi.login()* devuelve JWT de 24 h.
2. *SessionManager* guarda el token cifrado en *SecurePrefs*.
3. *AuthInterceptor* lo inserta en cada petición; si el backend responde 401 y el usuario sigue logeado, *ReloginCallback* renueva el token con las credenciales almacenadas.
4. Sólo si el usuario pulsa “Cerrar sesión” se limpian prefs y se obliga a volver a introducir correo y contraseña.

## **Seguridad en el cliente**

- Token, email y contraseña se almacenan cifrados (AES/GCM) vía **EncryptedSharedPreferences**.

- El interceptor asegura que ningún endpoint protegido se llame sin cabecera **Bearer**; si falla el refresco del token, se lanza de nuevo el diálogo de login.

### Ventajas de la organización

- **Capa de datos desacoplada:** cambiar Retrofit por otra librería o migrar de Gson a Moshi sólo afecta a *RetrofitClient*.
- **Reutilización:** los repositorios se comparten entre distintas pantallas; reduce duplicidad.
- **Seguridad centralizada:** un único interceptor añade o renueva el JWT, evitando errores manuales.
- **UX coherente:**  *BaseActivity* estandariza toolbar, fondo relacionado con Star Wars y lógica de “require login” para esas acciones del toolbar.

## 6. Sistema de pujas

El corazón de SWOP es su **módulo de subastas**, diseñado para transformar la compra en una experiencia dinámica y competitiva. A continuación se detalla su funcionamiento, las reglas actuales y futuras, y cómo se controla el stock y se finaliza cada subasta.

### 6.1 Funcionamiento general del sistema

#### Puja diaria única

- Cada producto en subasta dura **24 horas**. Cuando la subasta se activa, aparece un **contador regresivo** en la ficha del artículo dentro de la app.
- El precio inicial se fija en **la mitad del PVP del producto**. Por ejemplo, si el sable láser oficial cuesta 80 €, la subasta comenzará en 40 €.
- Cada usuario solo puede **hacer una puja por subasta**. Pero siempre se registrará la más alta de dicho usuario.

#### Asignación

- Al cumplirse las 24 horas, el sistema comprueba la **puja más alta**. Ese usuario se

convierte en el “ganador” y se le registrará la compra al precio que ofreció.

- Si **nadie ha pujado**, la subasta queda desierta y el artículo vuelve al catálogo en venta normal o se relanza en otra subasta futura.

#### Ventajas para la tienda

- **Ticket medio elevado**: la puja inicial (50 % del PVP) suele animar a pujadores a superar la competencia, llegando incluso a superar el precio de venta al público.
- **Sensación de escasez**: saber que solo hay 24 horas para conseguir un artículo “a precio reducido” motiva al usuario a entrar diariamente y, en el mejor de los casos, a comprar también otros productos.
- **Retención**: las notificaciones push cuando quedan pocas horas generan revisitadas a la app y mayor probabilidad de venta cruzada.

## 6.2 Reglas de puja y gestión de usuarios

#### Registro y perfil de usuario

- Solo los usuarios registrados pueden pujar. Al registrarse obtienen un **perfil** donde se muestran:
  - **Histórico de pujas ganadas**: cada subasta ganada se almacena con fecha, precio final y nombre del producto.
  - **Puntos acumulados (futuro)**: cada vez que ganan una puja, recibirán puntos canjeables, lo que fomenta volver a la app.

#### Mecanismo de notificaciones

- La app envía una notificación al acabar la puja indicando si has sido el ganador o si no.
- Si un usuario es superado en la puja, puede recibir una alerta si activa esta opción, incentivando **contraofertar** (implementación futura).

#### Futuras implementaciones para gamificación y lealtad

- **Puntos por victoria**: en lugar de pagar en efectivo, el ganador podrá usar puntos acumulados en pujas anteriores para abonar parte o la totalidad del

importe. Esto eleva la sensación de recompensa y hace que algunos compradores ahorren para pujar grandes productos.

- **Ranking de mayores pujadores:** se calcula sumando el importe total gastado en todas las subastas ganadas. Cada mes, el “Jedi Supremo de la Puja” (quien más haya gastado) recibirá un  **premio extra en puntos** para gastar en la tienda. Esto crea **dinámica competitiva comunitaria** y anima a pujar repetidamente.

## 6.3 Control de stock y finalización de subasta

### Control de stock en tiempo real

- Solo se tendrá una **unidad por subasta**. Cuando la subasta se activa, el producto se marca como “reservado para subasta” y se reserva una unidad del mismo.
- Si la subasta queda desierta (nadie puja), la unidad del artículo regresa automáticamente al stock de la **sección “Tienda”** y se podrá comprar al PVP completo o volver a programarse para otra subasta en el futuro.

### Finalización y generación de pedido

- Cuando el temporizador llega a cero, el sistema bloquea el producto y genera un **pedido pendiente de pago** para el ganador.
- El usuario dispone de **48 horas** para confirmar la compra (no implementado). Durante ese tiempo se comprueba que la unidad no haya sido asignada accidentalmente a otro pedido y se retiene la pieza.
- Si el ganador no confirma en 48 horas, la puja se anula y el artículo vuelve a stock disponible.

### Ventajas para la tienda

- **Minimización de roturas de stock:** al reservar la unidad en cuanto finaliza la puja, se evita vender más de lo disponible.
- **Liquidez inmediata:** el pago se procesa tan pronto como el ganador confirma, acortando el ciclo de conversión.
- **Visibilidad de histórico:** cada usuario puede consultar en su perfil todas las pujas ganadas y los precios abonados, reforzando la transparencia y reduciendo reclamaciones.

# 7. Base de datos y estructura interna

A continuación se muestran dos conjuntos de diagramas:

- El **modelo Entidad-Relación (ER)** y el **diagrama de clases del back-end**. Reflejan la estructura de la base de datos y las principales entidades Java en Spring Boot.
- El **diagrama de clases del front-end**. Representa las clases de la app Android (Actividad, ViewModel, adaptadores, etc.) y su relación con los componentes de la interfaz.

Debido al tamaño de las imágenes, colocarlas directamente en el archivo no sería cómodo para verlo, por lo que están subidos a una web (Imgur) en donde se puede ampliar la imagen y que se entienda el texto.

## 7.1 Diagrama entidad-relación (E/R) y UML del Backend

Solamente de las entidades y sus relaciones.

[Diagrama E-R y diagrama UML del Backend](#)

## 7.2 UML del Frontend

Diagramas de todos los paquetes relevantes del frontend.

[Diagrama UML del front](#)

# 8. Recursos utilizados

## 8.1 Recursos materiales y humanos

**Hardware de desarrollo:** portátil i7 / 16 GB RAM y un móvil Android de prueba.

### Software y servicios

- Spring Boot, Android Studio y GitHub (gratuitos).
- **Render “Hobby”** para el backend: gratis, 512MB de RAM, con modo reposo a los 15 minutos de inactividad.
- Base de datos PostgreSQL en Render con el mismo plan que lo anterior.

**Logística y material de embalaje:** cajas de cartón reciclado, cinta de papel y relleno kraft (aprox. 0,35 € por envío).

**Mensajería:** [SEUR](#) 24 h “Península 24h” desde 7,35 € + IVA por paquete para clientes registrados; opción Shop2Shop 48 h desde 5,40 € + IVA para pedidos pequeños.

### Equipo humano

- **Fase inicial:** emprendedora-desarrolladora (autoempleo).
- **Fase de crecimiento (año 2):** incorporación de un/a **especialista en marketing digital** a media jornada ( $\approx$  1 000 €/mes) para gestionar redes, campañas de pago y newsletter.

## 8.2 Proveedor oficial: [OcioStock](#)

Mayorista español con licencia Disney/Lucasfilm. Pedidos sin mínimo y entrega en 24/48 h; dispone de modo dropshipping si se desea enviar directamente al cliente. Esto garantiza autenticidad del producto y reposición rápida del stock.

# 9. Aspectos legales y técnicos

## 9.1 RGPD y condiciones de uso

La app recoge datos de registro (nombre, email, teléfono, localidad, dirección de envío, dirección de facturación) y de pago. Se aplican las obligaciones del RGPD:

- Consentimiento explícito al crear cuenta.
- Política de privacidad con derechos [ARCO](#) y delegado de protección de datos.
- Contraseñas protegidas con BCrypt y transmisión cifrada (HTTPS).
- Acceso, rectificación y borrado en un clic desde la sección “Mi cuenta”.

## 9.2 Licencias de distribución y propiedad intelectual

SWOP no necesita licencia directa de Disney: todos los artículos provienen de distribuidores licenciados (OcioStock, Cerdá Group, Storline). En la web/app se indica

claramente “Producto oficial con licencia Disney/Lucasfilm”. Se evita usar logos o imágenes oficiales fuera de lo suministrado por el proveedor para cumplir la Ley de Propiedad Intelectual española.

## 9.3 Coste de publicación en Google Play

Crear la [cuenta de desarrollador](#) cuesta un **pago único de 25 USD ( $\approx 23$  €)** y permite subir apps de por vida.

No existen cuotas mensuales; sólo se aplican las comisiones estándar de Google Play sobre compras in-app si se implementan en el futuro.

# 10. Previsión económica

## 10.1 Presupuesto inicial y ayudas públicas

El proyecto arranca con **5 000 € de capital propio**, suficiente para cubrir las primeras compras de stock y los gastos de constitución. A esta cantidad se suma la **subvención autonómica TR880A** (55 % sobre una inversión ya realizada de 10 000 € en equipo, software y stock), que aporta **5 500 €** adicionales. En total, SWOP dispone de **10 500 €** para afrontar el primer ejercicio sin recurrir a financiación externa. Esta ayuda es en régimen de mínimis y se concede en concurrencia no competitiva, de modo que basta con cumplir los requisitos formales (alta como autónoma entre 2023 y 2025, inversión ya pagada y domicilio fiscal en Galicia) y presentar la solicitud durante la ventana del 21 de mayo al 30 de junio de 2025.

## 10.2 Estimación de ingresos y gastos (1 año)

Para un escenario prudente se prevén **600 ventas** (50/mes) con un **precio medio de 24 €** por artículo.

### Coste unitario

Cada unidad genera un coste variable de **15,68 €**: 14 € del producto mayorista, 0,35 € de embalaje, 0,63 € del tramo de envío SEUR que asume la tienda y 0,70 € en comisiones de pago. El total de gastos variables asciende así a **9.408 € (15,68 x 600 ventas)**.

### Coste fijo (infraestructura tecnológica, publicidad, gestoría...)

Los costes fijos se reducen a lo esencial: 140 € de hosting en Render, 20 € de dominio y certificados, 400 € en publicidad continua (30 € al mes más refuerzos en picos), 200 € en sorteos e influencers, 960 € de [cuota de autónomos](#) (tarifa plana) y una gestoría online de 300 €. La suma anual es de **2.020 €**.

Con unos ingresos de **14.400 €** y unos gastos totales de **11.428 €** (9.408 € variables + 2.020 € fijos), el beneficio antes de impuestos se sitúa en **3.092 €**. Aplicando el 15 % de IRPF para autónomos, el resultado neto esperado es de **aproximadamente 2.630 €**.

## 10.3 Punto de equilibrio y rentabilidad esperada

El margen real por artículo (diferencia entre precio y coste variable) es de **8,52 €**. Con unos costes fijos de 2.020 €, basta vender **238 unidades al año** (un poco menos de veinte al mes) para cubrir gastos. El objetivo de 600 ventas multiplica ese umbral por 2,5, ofrece un beneficio modesto y confirma que el modelo es sostenible sin necesidad de contratar personal ni asumir préstamos externos.

# 11. Futuras ampliaciones

A lo largo del primer año SWOP se centrará en validar el modelo de catálogo curado y pujas. Una vez asegurada la base de usuarios y un flujo de ventas estable, hay tres mejoras que aportarán valor adicional y prepararán la plataforma para escalar.

## 11.1 Chat global entre clientes

Los fans de Star Wars disfrutan compartiendo hallazgos, discutiendo teorías y, sobre todo, mostrando sus colecciones. Un **chat global** dentro de la app (similar a una sala de Telegram integrada) permitirá a los compradores comentar subastas en directo, intercambiar consejos de conservación o preguntar por la autenticidad de un artículo.

### Tecnología.

Se plantea usar Firebase Realtime Database o un canal MQTT ligero; ambos ofrecen mensajes en tiempo real sin consumir demasiados recursos del servidor.

### Ventajas.

- **Engagement:** más tiempo en la app = más probabilidades de pujar o comprar.

- **Confianza:** las dudas se resuelven entre pares y disminuye la tasa de devoluciones.
- **Comunidad orgánica:** el boca-a-boca interno reduce el gasto en publicidad externa.

## 11.2 Mejora de diseño UI/UX y experiencia de compra

El MVP actual es funcional, pero minimalista. La segunda iteración de la interfaz seguirá dos líneas:

- **Exploración temática:** fichas por era (Trilogía original, Precuela, Secuela, Disney+) para que el usuario navegue por su etapa favorita.
- **Embudo de pago en un gesto:** integración nativa con Google Pay y Apple Pay (cuando salga la versión iOS) para que un usuario registrado finalice la compra con la huella dactilar. También se añadirán micro animaciones para destacar pujas activas.

## 11.3 Automatización de stock y logística

En la versión inicial el inventario se actualiza manualmente al recibir la hoja de reposición del mayorista. El plan a medio plazo es:

- **Webhook con OcioStock:** cada vez que entran nuevas unidades el sistema actualiza el stock y activa una “alerta de reposición” en la app.
- **Etiqueta automática:** conexión vía API con SEUR; al confirmarse el pago se genera la etiqueta y el cliente recibe el número de seguimiento sin intervención humana.
- **Reorden inteligente:** cuando el inventario de un SKU caiga por debajo de 5 unidades, la plataforma enviará un correo de compra automática al proveedor.

**Resultado esperado:** reducir el tiempo de preparación de pedido de 10 min a 3 min, eliminar roturas de stock por descuido y liberar al emprendedor para tareas de marketing y negociación exclusivas.

### Resumen

Estas tres mejoras (comunidad en tiempo real, interfaz pulida y logística sin fricción)

multiplican el valor percibido por el usuario sin suponer un salto desproporcionado en costes, y preparan a SWOP para afrontar un segundo año con un catálogo mayor y, posiblemente, la incorporación de un especialista en marketing a media jornada.

## 12. Conclusiones

Trabajar en SWOP ha sido una experiencia muy gratificante. Desde el primer momento, disfrutar de un proyecto centrado en mi saga favorita (Star Wars) me motivó a fondo. A lo largo de las distintas fases, hubo “baches” que pusieron a prueba mis capacidades: por ejemplo, ajustar el diseño de la app Android para que fuera intuitiva, lidiar con la lógica de las pujas diarias, y validar si realmente una tienda especializada en Star Wars tenía suficientes clientes en España como para ser viable.

A pesar de esas dificultades, el balance es muy positivo. Ha sido emocionante ver cómo la API en Spring Boot respondía a peticiones de pujas y cómo la base de datos se sincronizaba con la app. Además, integrar servicios en la nube (Render) me permitió experimentar con despliegues reales y comprender mejor la gestión de entornos productivos.

Aunque SWOP se queda en un MVP y no se publicará en producción como tal, las lecciones extraídas son innegables. Tener un módulo de pujas, un gráfico de Gantt pequeño pero ajustado a la realidad, diagramas ER y UML, así como una estrategia de marketing, me da una base firme para seguir avanzando como programador. En el futuro, resultaría realmente interesante implementar el chat global, mejorar la interfaz gráfica y automatizar por completo el stock y la logística.

Por último, este proyecto me ha ayudado a consolidar mis habilidades: en Spring Boot reforcé el manejo de las relaciones JPA y aprendí cómo funciona la seguridad JWT; en Android profundicé en la arquitectura MVVM y en la experiencia de usuario móvil; en bases de datos entendí la importancia de un E-R bien diseñado y de una conexión en la nube; y en comercio electrónico vi de primera mano cómo cuadrar números para que un modelo de negocio sea sostenible.

En definitiva, SWOP no solo ha sido una oportunidad para mostrar la saga de películas que más me apasiona, sino también un soporte para seguir perfeccionándome como desarrollador y para conocer en profundidad el mundo del programador “Full-stack”. Estoy convencido de que los conocimientos adquiridos serán muy útiles para futuros proyectos, tal vez no relacionados con Star Wars, pero siempre con ganas e ilusión.

## 13. Bibliografía

Programación:

[Stackoverflow](#)

[Documentación oficial de Android Studio para ciertas librerías](#)

[ChatGPT](#)

[JPA Spring](#)

Empresa:

<https://www.statsignificant.com/p/what-makes-star-wars-fandom-so-unique>

<https://ecommerce-news.es/el-ecommerce-en-espana-preve-alcanzar-los-37-911-millones-de-euros-en-2025>

<https://www.latevaweb.com/ecommerceSpain>

<https://www.cerdagroup.com/>

<https://www.storline.com/esp/76-botellas>

<https://render.com/pricing>

<https://uptimerobot.com/pricing/>

<https://www.esterea.com/servicios/asesoramiento-legal-para-web-y-comercio-electronico/>

<https://www.seur.com/es/particulares/tarifas/tarifas-nacional/>

<https://www.ociostock.com/>

<https://www.obz.es/subvencion/tr880a-xempre-emprendimiento-2025-direccion-general-de-emprendimiento>

<https://www.aepd.es/derechos-y-deberes/ejerce-tus-derechos>

<https://www.stelorder.com/blog/cuota-autonomo-primer-ano/#:~:text=La%20cuota%20de%20aut%C3%B3nomos%20en%20el%20primer%20a%C3%B3%20se%20ha,a%20trabajar%20por%20cuenta%20propia.>

## 14. Repositorios Github públicos y despliegue

- [Backend](#)
- [Front](#)
- Dominio asignado por render al que se le hacen peticiones:  
<https://swopbackend.onrender.com>

Al ser una aplicación móvil, el despliegue de la propia app requiere que esté firmada para subirse a una plataforma legal de APKs o a la Play Store mismamente. Tampoco es posible subirla a Drive para descargar debido a las políticas de Google con este formato de aplicaciones (.apk).