

# **ARMARIO VIRTUAL**

**Patricia Alonso Ayllón**

## Índice

1. Introducción
2. Plan de Empresa
  - 2.1 Justificación
  - 2.2 Nombre y Logo
  - 2.3 Producto
  - 2.4 Modelo de Negocio
  - 2.5 Consumidores y Clientes Potenciales
  - 2.6 Competencia
  - 2.7 DAFO
  - 2.8 Publicidad y Promoción
3. Descripción del Sistema y Historias de Usuario
4. Modelado y Diseño
  - 4.1 Prototipado en Figma
  - 4.2 Diagrama de Clases del Modelo de Dominio
5. Diseño
  - 5.1 Diagrama de Clases de Diseño
6. Implementación

### 1. Introducción

Armario Virtual es una plataforma innovadora diseñada para ayudar a los usuarios a gestionar de manera sencilla y eficiente su guardarropa digital. Su propósito principal es ofrecer un entorno intuitivo donde cada persona pueda almacenar, categorizar y combinar sus prendas, creando conjuntos personalizados que respondan a sus gustos y necesidades.

La visión de Armario Virtual es convertirse en la herramienta de referencia para la planificación de atuendos y la inspiración de moda, facilitando tanto el uso diario de la ropa como la interacción social mediante comentarios, valoraciones y recomendaciones entre usuarios.

Los objetivos generales de este proyecto son:

- **Centralizar la gestión del vestuario:** Permitir al usuario crear un catálogo completo de sus prendas, con imágenes, detalles (color, talla, tipo) y enlaces de compra.
- **Fomentar la creatividad:** Ofrecer funcionalidades para diseñar y guardar conjuntos, ayudando a planificar looks para diferentes ocasiones.
- **Impulsar la interacción social:** Integrar un sistema de comentarios, likes y seguimiento entre usuarios, para compartir inspiración y recibir feedback.
- **Garantizar usabilidad y accesibilidad:** Desarrollar una interfaz clara, responsiva y accesible desde dispositivos móviles y web.

### 2. Plan de Empresa

## 2.1 Justificación

**¿Por qué este negocio?** Armario Virtual nace para cubrir la creciente necesidad de los usuarios de optimizar la gestión de su vestuario en un entorno digital. Con la diversificación de marcas y estilos, llevar un registro manual de prendas y combinaciones se hace tedioso. Nuestra plataforma facilita este proceso al centralizar toda la información en una única app accesible desde cualquier dispositivo.

**Análisis de la competencia** Existen aplicaciones y servicios de planificación de atuendos, pero suelen carecer de características sociales (comentarios, likes, seguimientos) o no integran un catálogo completo de prendas con enlaces de compra. Tampoco ofrecen filtros avanzados por tags, color o tipo.

### **Motivos de viabilidad y proyección de ventas**

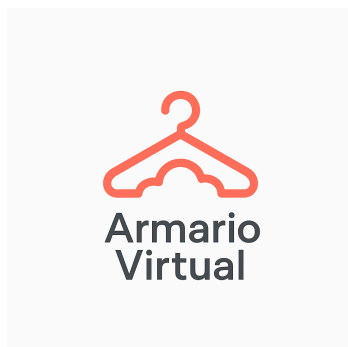
- Crecimiento del mercado de fashion tech y apps de estilo personal.
- Potencial de monetización mediante publicidad segmentada y suscripciones premium.
- Bajo coste de mantenimiento frente a altos márgenes de las ventas digitales.
- Proyección de alcanzar 50.000 usuarios activos en el primer año, con un 5% de conversión a planes de pago.

### **Propuesta de valor: diferencial y novedad del producto**

- **Social e interactivo:** combina gestión de prendas con red social de moda.
- **Machine learning** (futuro): recomendaciones automáticas de conjuntos.
- **Multicanalidad:** sincronización con e-commerce y alertas de ofertas.
- **Flexibilidad:** opciones a medida para usuarios influencers y profesionales de la moda.

## 2.2 Nombre y Logo

- Definición del nombre comercial: "Armario Virtual": nombre claro y descriptivo que comunica la idea de un *armario* digital accesible desde cualquier lugar.
- Muestra del logotipo



- Justificación de la elección
  - El término *Armario* alude al concepto tradicional, mientras que *Virtual* sitúa el servicio en el ámbito digital e innovador.
  - El logotipo integra una percha (símbolo universal de ropa) dentro de una nube sutil, reforzando la idea de almacenamiento en línea.
  - La paleta de colores transmite profesionalidad y frescura, y se alinea con tendencias de diseño actuales.

## 2.3 Producto

- **Descripción de la oferta (servicios/productos)** Armario Virtual ofrece un servicio basado en web y móvil que permite a los usuarios:

- Registrar y almacenar cada una de sus prendas con fotografía, descripción de color, talla y tipo de prenda.
- Organizar su guardarropa mediante etiquetas personalizadas y filtros (color, temporada, uso).
- Crear conjuntos de forma intuitiva arrastrando y combinando prendas.

- Captura de interfaz o mockup

- **Finalidad y necesidades que cubre** La aplicación facilita la planificación de outfits diarios o para eventos especiales, ahorrando tiempo al evitar búsquedas físicas en armarios reales. Cubrimos necesidades como:

- Evitar compras duplicadas al conocer el inventario exacto.
- Inspiración de estilo mediante conjuntos guardados y recomendaciones.
- Gestión eficiente del vestuario al detectar prendas infrautilizadas o en desuso.

## 2.4 Modelo de Negocio

### Fuente de ingresos

- **Publicidad segmentada:** espacios publicitarios para marcas de moda dentro de la plataforma, dirigidos según intereses y perfil de usuario.

### Estructura de costes

- **Desarrollo y mantenimiento:** equipo de ingeniería, diseño UI/UX y pruebas.
- **Infraestructura:** servidores en la nube, bases de datos y ancho de banda.

- **Marketing y adquisición:** campañas en redes sociales, SEO/SEM y partnerships.
- **Soporte y atención al cliente:** personal de help desk y moderación de contenido.
- **Licencias y servicios de terceros:** APIs de machine learning, servicios de email y herramientas de analítica.

### Canales de distribución y venta

- **Web oficial:** plataforma accesible desde navegadores de escritorio y mobile.
- **App Stores:** publicación en Google Play y Apple App Store para dispositivos móviles.
- **Redes sociales y blogs de moda:** promoción directa a través de influencers y contenido patrocinado.
- **Alianzas con tiendas de moda:** integración en e-commerce de marcas y aparición en sus canales de venta.

## 2.5 Consumidores y Clientes Potenciales

### Segmentación de mercado

- **Jóvenes adultos (18–35 años):** usuarios digitales nativos que buscan optimizar su estilo y planificar atuendos.
- **Profesionales y ejecutivos (25–45 años):** necesitan proyectar imagen cuidada en el trabajo y eventos sociales.
- **Aficionados a la moda e influencers:** valoran la interacción social y el feedback de su red.

### Perfil de consumidores

- **Edad:** 18–45 años.
- **Ubicación:** principalmente en zonas urbanas y metropolitanas donde el comercio online y el estilo de vida digital están arraigados.
- **Estilos de vida:** activos, con rutina de trabajo o estudios, interés en tendencias de moda y en aprovechar al máximo su armario.

### Tamaño y características de la zona objetivo

- **España:** más de 15 millones de usuarios de smartphones en el rango de edad objetivo; alto porcentaje de penetración de apps de estilo de vida.
- **Ciudades clave:** Madrid y Barcelona como mercados iniciales, representando aproximadamente el 30% del público objetivo.

## 2.6 Competencia

### Empresas o aplicaciones similares

- **Closet+:** app móvil para registrar prendas, etiquetar, crear conjuntos y planificar outfits con calendario integrado.
- **Stylebook:** disponible en iOS, ofrece gestión de armario, planificación de atuendos, estadísticas de uso y exportación de looks.

- **Pureple:** aplicación gratuita con anuncios, permite catalogar prendas, crear combinaciones y sincronizar datos en la nube.
- **Smart Closet:** plataforma web y móvil con análisis de uso, alertas de prendas infrautilizadas y feed social.

## 2.7 DAFO

- **Debilidades**
  - Ausencia por ahora de recomendaciones automáticas basadas en datos.
  - Dependencia inicial de imágenes y descripciones manuales de prendas.
  - Recursos limitados para marketing y adquisición de usuarios en fases tempranas.
  - Necesidad de incentivar la actividad social para generar contenido (comentarios, likes).
- **Amenazas**
  - Entrada de competidores con mayor financiación o grandes plataformas de moda.
  - Cambios rápidos en tendencias de moda que pueden dejar obsoletas ciertas funcionalidades.
  - Preocupaciones de privacidad y protección de datos personales.
  - Saturación del mercado de aplicaciones de estilo y planificación de atuendos.
- **Fortalezas**
  - Enfoque social y colaborativo que fomenta el engagement entre usuarios.
  - Interfaz intuitiva y accesible tanto en web como en móvil.
  - Propuesta de valor clara: gestión completa de armario y creación de conjuntos.
  - Potencial para integrar en el futuro tecnologías de IA y machine learning.
- **Oportunidades**
  - Alianzas con marcas de moda y plataformas de e-commerce para ampliar catálogo.
  - Crecimiento del sector fashion tech y adopción de herramientas digitales por consumidores.
  - Expansión a mercados internacionales con adaptación cultural.
  - Incorporación de módulos premium o servicios adicionales (personal shoppers, estilistas virtuales).

## 2.8 Publicidad y Promoción

### Canales publicitarios (online/offline)

- **Online:** campañas en redes sociales (Instagram, Facebook, TikTok), publicidad en buscadores (Google Ads), colaboraciones con influencers y blogs de moda, email marketing y anuncios en plataformas de streaming.
- **Offline:** presencia en ferias y eventos de moda, cartelería en tiendas colaboradoras y centros comerciales, flyers en puntos estratégicos y prensa especializada.

### Estrategias de promoción

- **Marketing de influencers:** micro-influencers con audiencia segmentada para generar contenido auténtico.
- **Concursos y retos:** premios para los mejores conjuntos compartidos por la comunidad.
- **Contenido educativo:** webinars y tutoriales de estilo para fidelizar usuarios.
- **SEO y SEM:** optimización de palabras clave y campañas de pago por clic para captar tráfico cualificado.

### Estimación de costes

- Desarrollo de creatividades y producción de vídeos: 2.000 €
- Campañas en redes sociales (3 meses): 1.500 €
- Google Ads (3 meses): 900 €
- Honorarios de influencers (5 micro-influencers): 2.500 €
- Stand en feria de moda local: 1.200 €

### 3. Descripción del Sistema y Historias de Usuario

La plataforma Armario Virtual se basa en una arquitectura **RESTful** desarrollada con **Spring Boot** para el backend y una capa de presentación web (JSP/Thymeleaf) complementada con una API consumida desde aplicaciones móviles o frontends JavaScript.

- **Autenticación y seguridad:** gestión de usuarios mediante JWT para la API y sesiones HTTP para la interfaz web; endpoints protegidos con roles (CLIENTE, ADMIN).
- **Recursos principales:**
  1. /api/prendas: CRUD de prendas, carga y descarga de imágenes, búsqueda y filtrado por nombre, color o tags.
  2. /api/conjuntos: gestión de conjuntos, likes, compartición y carga de imágenes.
  3. /api/usuarios: registro, activación vía email, login, refresh de tokens, perfil, seguimiento entre usuarios.
  4. /api/comentarios: creación y listado de comentarios sobre prendas y conjuntos.
  5. /api/tags: CRUD de tags y asociación con prendas/conjuntos.
- **Tecnologías:**
  1. **Backend:** Java 17, Spring Boot, Spring Security, Spring Data JPA con Hibernate.
  2. **Base de datos:** PostgreSQL (o MySQL) para persistencia relacional.
  3. **Almacenamiento de ficheros:** sistema de ficheros local en **uploads/** o AWS S3 en producción.
  4. **Frontend web:** JSP/Thymeleaf con jQuery o fetch API para llamadas AJAX.
  5. **Movilidad:** consumo de la API desde apps nativas o SPA.
- **Flujo básico:**
  1. El usuario se registra y activa su cuenta vía email.

2. Realiza login y obtiene token JWT para llamadas a la API.
3. Gestiona prendas y conjuntos, sube imágenes y añade tags.
4. Interactúa socialmente: da likes, sigue a otros usuarios y comenta.

1	🕒	HU-01-crear-usuario	#1
2	🕒	HU-08-ver-usuario	#8
3	🕒	HU-06-crear-prenda	#6
4	🕒	HU-14-crear-conjunto	#14
5	🕒	HU-22-dar-me-gusta	#22
6	🕒	HU-02-editar-perfil	#2
7	🕒	HU-03-ver-otros-usuarios	#3
8	🕒	HU-04-desactivar-cuenta	#4
9	🕒	HU-05-seguir-usuario	#5
10	🕒	HU-07-editar-prenda	#7
11	🕒	HU-10-ver-prenda	#10
12	🕒	HU-11-editar-prenda	#11
13	🕒	HU-12-eliminar-prenda	#12
14	🕒	HU-13-ver-prendas-publicas	#13
15	🕒	HU-15-ver-conjunto	#15
16	🕒	HU-19-editar-conjunto	#19



16	⦿	HU-19-editar-conjunto	#19
17	⦿	HU-16-eliminar-conjunto	#16
18	⦿	HU-17-ver-conjuntos-publicos	#17
19	⦿	HU-18-comentar	#18
20	⦿	HU-20-ver-comentarios	#20
21	⦿	HU-21-eliminar-comentario	#21
22	⦿	HU-23-iniciar-sesion	#23
23	⦿	HU-09-ver-todos-los usuarios-admin	#9
24	⦿	HU-24-ver-prendas-admin	#24
25	⦿	HU-28-ver-conjuntos-usuario	#28
26	⦿	HU-25-crear-cuenta-admin	#25
27	⦿	HU-26-eliminar-prenda-admin	#26
28	⦿	HU-27-eliminar-conjunto-admin	#27
29	⦿	HU-28-crear-tipo-prenda	#29
30	⦿	HU-29-borrar-tipo-prenda	#30

## 4. Modelado y Diseño

<https://www.figma.com/design/iS9dc22HknjH2YfXmbUhil/Untitled?node-id=0-1&p=f&m=draw>

### 4.2 Diagrama de Clases del Modelo de Dominio

En el modelo de dominio identificamos las siguientes clases principales:

- **Usuario**
  - Atributos: id: UUID, nombre: String, username: String, email: String, password: String, fechaRegistro: LocalDateTime, enable: Boolean, activo: Boolean, role: Role
  - Relaciones:

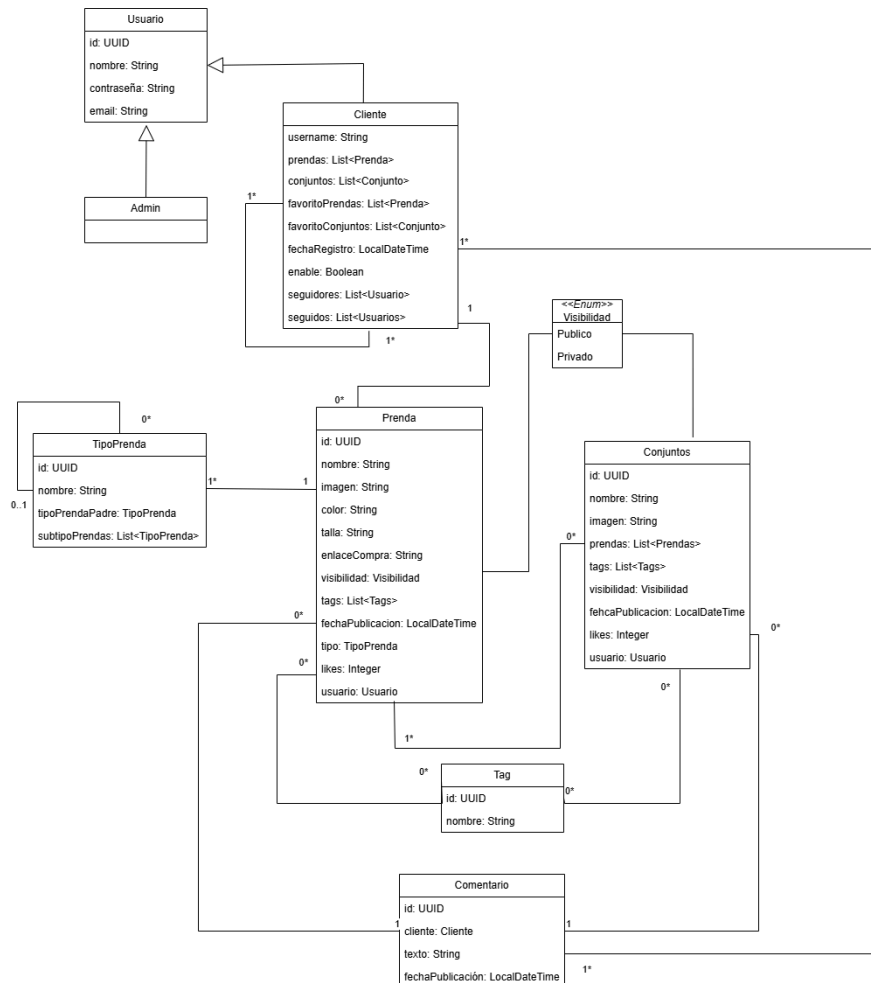
- $1 \rightarrow *$  Prenda (un Usuario puede tener muchas Prendas)
  - $1 \rightarrow *$  Conjunto (un Usuario puede crear muchos Conjuntos)
    - $\leftrightarrow *$  Cliente (seguidores y seguidos)
- **Prenda**
  - Atributos: id: Long, nombre: String, imagen: String, color: String, talla: String, enlaceCompra: String, visibilidad: Visibilidad, fechaPublicacion: LocalDateTime
  - Relaciones:
    - $\leftrightarrow *$  Tag (una Prenda puede tener varios Tags)
    - $1 \rightarrow 1$  TipoPrenda (cada Prenda tiene un TipoPrenda)
    - $1 \rightarrow 1$  Cliente (dueño de la Prenda)
      - $\leftrightarrow *$  Cliente (likes, clientesQueDieronLike)
    - $1 \rightarrow *$  Comentario (comentarios en la prenda)
- **Conjunto**
  - Atributos: id: Long, nombre: String, imagen: String, fechaPublicacion: LocalDateTime, visibilidad: Visibilidad
  - Relaciones:
    - $\leftrightarrow *$  Prenda (prendas que forman el Conjunto)
    - $\leftrightarrow *$  Tag (tags asociados al Conjunto)
    - $1 \rightarrow 1$  Cliente (dueño del Conjunto)
      - $\leftrightarrow *$  Cliente (clientesQueDieronLike)
    - $1 \rightarrow *$  Comentario (comentarios en el conjunto)
- **Tag**
  - Atributos: id: Long, nombre: String
  - Relaciones:
    - $\leftrightarrow *$  Prenda
    - $\leftrightarrow *$  Conjunto
- **Comentario**
  - Atributos: id: Long, contenido: String, fechaPublicacion: LocalDateTime
  - Relaciones:
    - $1 \rightarrow 1$  Cliente (autor del comentario)
    - $0..1 \rightarrow 1$  Prenda
    - $0..1 \rightarrow 1$  Conjunto
- **TipoPrenda**
  - Atributos: id: Long, nombre: String
  - Relaciones:
    - $\leftrightarrow *$  Prenda
    - $1 \rightarrow *$  TipoPrenda (relación jerárquica padre-hijo)

#### **Multiplicidades y cardinalidades:**

- Usuario 1 --- \* Prenda
- Usuario 1 --- \* Conjunto
- Cliente \* --- \* Cliente (seguidores/seguídos)
- Prenda \* --- \* Tag
- Conjunto \* --- \* Tag

- Prenda 1 --- 1 TipoPrenda
- Prenda \* --- \* Cliente (likes)
- Conjunto \* --- \* Cliente (likes)
- Prenda 1 --- \* Comentario
- Conjunto 1 --- \* Comentario

## 5. Diseño



## 6. Implementación

### 6.1 Descripción de los diferentes paquetes y clases de cada aplicación

- **config**
  - **SecurityConfig**: define la cadena de seguridad, filtros JWT y acceso a rutas.
  - **OpenApiConfig**: configura la documentación OpenAPI/Swagger.
  - **WebConfig**: gestiona rutas de recursos estáticos y directorios de subida.

- PasswordEncoderConfig: proporciona el bean de codificación de contraseñas.
- MailConfig: configura el JavaMailSender para envíos de correo.
- **controller**
  - UsuarioController: endpoints para registro, login, perfil, seguimientos y activación.
  - PrendaController: CRUD de prendas, likes, filtrado y gestión de imágenes.
  - ConjuntoController: CRUD de conjuntos, likes, filtrado y gestión de imágenes.
  - ComentarioController: crea, lista y elimina comentarios en prendas y conjuntos.
  - TagController: CRUD de tags y asociación/desasociación con prendas/conjuntos.
  - ViewController: mapea vistas JSP, por ejemplo la página de login.
- **dto**
  - Pares de clases CreateXDto, UpdateXDto, GetXDto para cada entidad (Usuario, Prenda, Conjunto, Comentario, Tag, TipoPrenda).
  - LoginRequest, UserResponse, RefreshTokenRequest para la autenticación.
- **error**
  - Excepciones personalizadas (PrendaNoEncontradaException, UsuarioNoAutenticadoException, etc.).
  - GlobalExceptionHandler y JwtControllerAdvice para formatear respuestas de error.
- **model**
  - Entidades JPA: Usuario (y subclases Cliente, Admin), Prenda, Conjunto, Comentario, Tag, TipoPrenda, RefreshToken, ActivationToken.
  - Enums: Role, Visibilidad.
  - Relaciones y mapeos (@OneToMany, @ManyToMany, etc.).
- **repository**
  - Interfaces JpaRepository y JpaSpecificationExecutor para cada entidad.
  - Consultas personalizadas con @Query para búsquedas con fetch y conteo de likes.
- **security**
  - JwtAuthenticationFilter, JwtAuthenticationEntryPoint, JwtAccessDeniedHandler para el flujo JWT.
  - RefreshTokenService, ActivationTokenService, JwtService, RefreshTokenService para gestión de tokens.
  - CustomUserDetailsService: carga de usuario para Spring Security.
- **service**

- Servicios por entidad (UsuarioService, PrendaService, ConjuntoService, ComentarioService, TagService) con lógica de negocio y transacciones.
- EmailService: envía correos de activación.
- ActivationTokenService y RefreshTokenService: creación y verificación de tokens de activación y refresco.