

Documento de requerimientos de software



SHOWPASS

***Fecha:** 01/10/2025*

Tabla de contenido

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
	1.0	Marcos Beas, Dylan Molina	IES Villablanca	Versión inicial del documento de requerimientos.

Información del Proyecto

Empresa / Organización	ShowPass
Proyecto	Creación de aplicación/Web de venta de entradas online
Fecha de preparación	
Cliente	Todo el público
Patrocinador principal	
Gerente / Líder de Proyecto	
Gerente / Líder de Análisis de negocio y requerimientos	

Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Marcos Beas Sánchez	Desarrollador / Alumno DAM	IES Villablanca		
Dylan Molina Amariles	Desarrollador / Alumno DAM	IES Villablanca		
Miguel Antonio Sutil	Tutor de proyecto	IES Villablanca		

1. Propósito

El presente documento especifica los requerimientos de software de la aplicación **SHOWPASS v1.0 (Venta de entradas online)**.

ShowPass es una plataforma multiplataforma (web y móvil) diseñada para la compra y venta de entradas digitales a eventos, ofreciendo funcionalidades específicas para clientes, vendedores y administradores.

Este documento abarca la **totalidad del sistema ShowPass**, incluyendo:

- **Aplicación web:** desarrollada con frontend en **React (Vite + Tailwind CSS)** y backend en **Java con Spring Boot**.
- **Aplicación móvil (Android):** desarrollada en **Kotlin con Jetpack Compose**, utilizando el mismo backend en **Spring Boot**.
- **Servicios de backend:** encargados de la lógica de negocio, la gestión de usuarios y roles, la administración de eventos, la validación de compras, la generación de entradas en **formato PDF** con código **QR único**, y el envío de si se solicita mediante vía correo electrónico.
- **Paneles de gestión:**
 - **Clientes:** exploración, filtrado, compra de entradas y descarga de tickets.
 - **Vendedores:** creación y gestión de eventos, control de disponibilidad de entradas.
 - **Administradores:** supervisión del sistema, control de usuarios (clientes y vendedores) y gestión de eventos caducados o inválidos.

Versión: versión inicial (**1.0**).

2.Alcance del producto / Software

SHOWPASS tiene como objetivo principal ofrecer una aplicación multiplataforma (web y móvil) para la compra y venta de entradas online de forma segura y sencilla.

Beneficios principales:

- Digitalización del proceso de venta de entradas, reduciendo costes y facilitando el acceso.
- Seguridad en el acceso y las transacciones mediante autenticación con **JWT**.
- Gestión eficiente de eventos para vendedores y administradores.
- Mejor experiencia para los clientes al recibir tickets digitales en PDF con **QR único**.

Objetivos:

- Permitir a clientes, vendedores y administradores interactuar según su rol.
- Facilitar la búsqueda, compra y entrega digital de entradas.
- Ofrecer paneles de gestión para vendedores y administradores.

Este alcance cubre la totalidad del sistema en su primera versión (**1.0**).

3.Referencias

Manual de usuario provisional ShowPass – Documento interno Elaborada por Marcos Beas y Dylan Molina,versión provisional, septiembre 2025, enlace:
https://github.com/MarcosBeasSanchez/TFG/blob/main/docs/ManualUsuarioProvisional_Showpass.pdf

Documentación técnica de Spring Boot – Spring Framework, versión 3.x, disponible en:<https://spring.io/projects/spring-boot>

Documentación de Jetpack Compose – Android Developers, disponible en:
<https://developer.android.com/jetpack/compose>

Documentación Material3: M3 <https://m3.material.io/get-started>

Documentación de Tailwind CSS - Tailwind, versión 4.1.12,
<https://tailwindcss.com/docs/installation/using-vite>

Documentación de Vite - Vite, Ultima versión, <https://es.vite.dev/guide/>

4.Funcionalidades del producto

Registro y gestión de usuarios: permitir que los usuarios se registren, inicien sesión y gestionen su perfil según su rol (cliente, vendedor, administrador).

Exploración y búsqueda de eventos: los clientes pueden buscar, filtrar y consultar detalles de los eventos disponibles.

Carrito de compra y gestión de entradas: añadir entradas al carrito, simular el pago y recibir tickets digitales en PDF con código QR.

Tickets: lista de tickets digitales comprados con opción de descarga y envío al correo electrónico del cliente.

Gestión de eventos para vendedores: creación, edición de eventos, así como control de la disponibilidad de entradas.

Panel de administración: supervisión de usuarios y eventos, eliminación de eventos caducados o inapropiados, bloqueo o suspensión de usuarios.

Seguridad y autenticación: control de acceso basado en roles mediante autenticación segura con JWT.

Compatibilidad multiplataforma: disponibilidad del sistema en web y aplicación móvil para Android.

5. Clases y características de usuarios

La aplicación SHOWPASS está diseñada para diferentes tipos de usuarios, cada uno con características y privilegios específicos de acuerdo con sus necesidades y su rol en el sistema.

Cliente (Comprador)

- **Descripción:** Usuario final que compra entradas para eventos.
- **Frecuencia de uso:** Media-alta (dependiendo de la oferta de eventos).
- **Funcionalidades principales:**
 - Registro e inicio de sesión.
 - Búsqueda y filtrado de eventos.
 - Visualización de detalles de un evento.
 - Compra de entradas (carrito + pago simulado).
 - Recepción de tickets en formato digital (PDF/QR).
 - Gestión de perfil personal.
- **Privilegios de seguridad:** Acceso limitado únicamente a funcionalidades de cliente.
- **Nivel de experiencia requerido:** Bajo, se espera que cualquier usuario general pueda usarlo sin conocimientos técnicos.

Vendedor (Organizador de eventos)

- **Descripción:** Usuario que publica y gestiona eventos, poniendo entradas a la venta.
- **Frecuencia de uso:** Media (dependiendo de la periodicidad de eventos organizados).
- **Funcionalidades principales:**
 - Funcionalidades iguales que el cliente más las de vendedor
 - Creación, edición de eventos.
 - Configuración de disponibilidad y precio de entradas.
 - Control de entradas disponibles.

- **Privilegios de seguridad:** Acceso a funcionalidades avanzadas de gestión de eventos. No puede eliminar usuarios ni acceder a datos administrativos.
- **Nivel de experiencia requerido:** Medio. Se espera que tenga conocimientos básicos de gestión de plataformas digitales.

Administrador

- **Descripción:** Usuario con permisos completos para supervisar el sistema y garantizar su correcto funcionamiento.
- **Frecuencia de uso:** Baja-media, ya que su papel es de control y supervisión.

Funcionalidades principales:

- Funcionalidades iguales que el cliente más las funcionalidades de administrador
- Supervisión y gestión de usuarios (clientes y vendedores).
- Bloqueo/suspensión de usuarios problemáticos.
- Eliminación de eventos.
- Control general de la plataforma y resolución de incidencias.
- **Privilegios de seguridad:** Acceso total a todas las funcionalidades del sistema.
- **Nivel de experiencia requerido:** Medio-alto, requiere conocimientos básicos de administración de sistemas y plataformas online.

6. Entorno operativo

El sistema ShowPass se desarrollará y ejecutará en un entorno multiplataforma, con soporte para aplicaciones web y móviles.

Plataforma de hardware:

- **Servidores backend:** ejecución en entornos Linux (Ubuntu Server recomendado) o Windows Server, con procesadores multinúcleo y mínimo 8 GB de RAM.
- **Estaciones de desarrollo:** equipos con Windows 10/11, macOS o distribuciones Linux, mínimo 8 GB de RAM y procesadores Intel i5/AMD Ryzen 5 o superior.

- **Dispositivos móviles:** smartphones Android con versión mínima 9.0 (Pie), 3 GB de RAM y almacenamiento suficiente para ejecutar la app.

Software y componentes principales:

- **Backend:** desarrollado en Java 17+ con Spring Boot, gestionando la lógica de negocio y API REST.
- **Frontend Web:** React (Vite + Tailwind CSS) para la interfaz web.
- **App móvil:** Kotlin con Jetpack Compose en Android Studio.
- **Base de datos:** MySQL en producción y H2 en entornos de pruebas.
- **Seguridad:** Spring Security con JWT para autenticación y control de roles.
- **Correo electrónico:** servidor SMTP para envío de tickets y notificaciones.

Compatibilidad:

- **Web:** navegadores modernos (Google Chrome, Mozilla Firefox, Microsoft Edge).
- **Móvil:** Android 9.0 o superior.

7. Requerimientos funcionales

1-Registro y gestión de usuarios

Descripción:

Permite a los usuarios registrarse, iniciar sesión y gestionar su perfil. Cada usuario tiene un rol asignado (cliente, vendedor o administrador) que determina las funciones a las que puede acceder.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El usuario selecciona la opción “Registrarse” e introduce sus datos.
- El sistema valida los datos y crea la cuenta para loguearse.
- Los usuarios pueden iniciar sesión y acceder a las funciones de acuerdo a su rol.
- Si hay errores en los datos o el correo ya está registrado, el sistema muestra un mensaje de error.

Requerimientos funcionales:

- **REQ-01:** Registrar nuevos usuarios con correo, contraseña y rol asignado.
- **REQ-02:** Validar que el correo no esté previamente registrado.
- **REQ-03:** Permitir inicio de sesión con credenciales válidas.
- **REQ-04:** Permitir actualizar datos personales y perfil del usuario.

2-Exploración y búsqueda de eventos

Descripción:

Permite a los clientes buscar, filtrar y consultar los detalles de los eventos disponibles.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El cliente accede al listado de eventos.
- Puede aplicar filtros por categoría, ubicación o nombre.
- Selecciona un evento para ver sus detalles completos (descripción, imágenes, fecha, lugar, invitados, precio, etc.).

Requerimientos funcionales:

- **REQ-05:** Mostrar listado de eventos activos.
- **REQ-06:** Permitir filtrar eventos por categoría, ubicación o nombre.
- **REQ-07:** Permitir búsqueda por palabras clave.
- **REQ-08:** Mostrar detalles completos del evento seleccionado.

3-Carrito de compra y gestión de entradas

Descripción:

Permite a los clientes seleccionar entradas (número), simular el pago y recibir tickets digitales.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El cliente añade entradas al carrito de compra.

- Procede a la simulación de pago.
- Si el cliente tiene saldo en su cuenta, se realiza el pago
- El sistema genera entradas en PDF con código QR y las envía al correo del cliente.

Requerimientos funcionales:

- **REQ-9:** Permitir añadir entradas al carrito.
- **REQ-10:** Revisar y modificar el carrito antes de confirmar la compra.
- **REQ-11:** Simular el proceso de pago de entradas.
- **REQ-12:** Generar tickets digitales en PDF con código QR único.
- **REQ-13:** Enviar tickets al correo electrónico del cliente.

4-Tickets

Descripción:

Permite a los clientes consultar sus tickets adquiridos, descargarlos o recibirlos por correo.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El cliente accede a la sección “Tickets”.
- Puede descargar los tickets en PDF o reenviarlos a su correo electrónico.

Requerimientos funcionales:

- **REQ-14:** Mostrar lista de tickets comprados por el cliente.
- **REQ-15:** Permitir descarga de tickets en PDF.
- **REQ-16:** Permitir envío de tickets al correo electrónico del cliente.

5-Gestión de eventos para vendedores

Descripción:

Permite a los vendedores crear, editar y controlar la disponibilidad de sus eventos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El vendedor accede a su panel de gestión.

- Puede crear un evento completando título, descripción, fecha, precio, aforo(número entradas) e imágenes.
- Puede editar o eliminar eventos existentes y consultar la disponibilidad de entradas.

Requerimientos funcionales:

- **REQ-17:** Crear nuevos eventos con toda la información necesaria.
- **REQ-18:** Editar eventos existentes.
- **REQ-19:** Consultar la disponibilidad y ventas de entradas de cada evento.

6-Panel de administración

Descripción:

Proporciona a los administradores herramientas de supervisión de usuarios y eventos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El administrador accede al panel de control.
- Puede revisar usuarios y eventos, eliminar eventos caducados o inapropiados y suspender usuarios que incumplan las normas.

Requerimientos funcionales:

- **REQ-20:** Permitir eliminar eventos caducados o inapropiados.
- **REQ-21:** Permitir suspender temporalmente o definitivamente usuarios
- **REQ-22:** Mostrar listado completo de usuarios y eventos activos.

7-Seguridad y autenticación

Descripción:

Control de acceso seguro mediante autenticación basada en JWT y roles de usuario.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El sistema valida el acceso de cada usuario según su rol y mantiene la sesión mediante tokens JWT.
- Se bloquea el acceso si el token es inválido o expirado.

Requerimientos funcionales:

- **REQ-23:** Autenticar usuarios mediante JWT.
- **REQ-24:** Restringir el acceso a funcionalidades según rol.

8-Compatibilidad multiplataforma

Descripción:

Asegura que SHOWPASS funcione correctamente tanto en la web como en la app móvil Android.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El usuario accede a la plataforma desde web o móvil.
- Todas las funciones disponibles deben comportarse de manera consistente.

Requerimientos funcionales:

- **REQ-25:** Acceso completo desde navegadores web modernos.
- **REQ-26:** Acceso completo desde dispositivos Android 9.0 o superiores.

8.Reglas de negocio

1-Roles de usuario

El sistema distingue tres roles principales: **cliente, vendedor y administrador**.

Cada rol tiene permisos específicos:

- **Clientes:** pueden registrarse, explorar eventos, comprar entradas y gestionar sus tickets.
- **Vendedores:** pueden crear, editar propios, además de consultar las ventas.
- **Administradores:** pueden supervisar el sistema, suspender y eliminar usuarios y también eliminar eventos.

2-Disponibilidad de entradas

- No se permite la venta de más entradas de las disponibles en un evento.
- Si el aforo de un evento se completa, el sistema debe bloquear nuevas compras para dicho evento.

3-Eventos

- Los eventos eliminados no deben estar disponibles en la búsqueda/compra.
- Los administradores pueden eliminarlos para mantener actualizado el catálogo.

4-Integridad de las transacciones

- Una compra de entradas solo será válida si el proceso de pago se completa correctamente.
- En caso de error en el pago, no se deben generar ni enviar tickets.

5-Generación de tickets

- Cada ticket debe ser único e irrepetible, identificado por un **código QR exclusivo**.
- Los tickets deben poder descargarse en PDF y enviarse por correo electrónico al cliente.

6-Seguridad y autenticación

- El acceso al sistema debe estar protegido mediante autenticación basada en tokens JWT.
- Cada usuario solo puede acceder a las funciones correspondientes a su rol.

7-Suspensión de usuarios

- Los administradores pueden suspender cuentas de clientes o vendedores que incumplan las normas (fraude, abuso, etc.).
- Un usuario suspendido no podrá iniciar sesión ni realizar ninguna acción en el sistema.

8-Gestión de datos

- Todos los datos de usuarios y eventos deben almacenarse en la base de datos de manera segura y consistente.
- Los correos electrónicos asociados a las cuentas deben ser únicos.

9.Requerimientos de interfaces externas

1-Interfaces de usuario

Diseño general:

La interfaz de usuario de ShowPass debe ser **intuitiva, responsive y coherente** en todas las plataformas (web y móvil).

Estándares GUI:

- Colores principales definidos en la guía de estilo de la app.
- Tipografía homogénea en todas las pantallas(Sans serif).
- Botones principales con colores diferenciados (ej. “Comprar” en verde, “Cancelar” en rojo).
- Navegación mediante menús claros y persistentes (barra superior en web, barra inferior y superior en móvil).

Pantallas principales:

- **Pantalla de inicio de sesión y registro:**
 - Inicio Sesión: email y contraseña.
 - Registro: nombre,correo,contraseña,fecha nacimiento y rol (cliente o vendedor)
- **Catálogo de eventos:** listado con filtros y búsqueda.
- **Detalle de evento:** nombre,descripción, fecha, ubicación,fotos,invitados, aforo máximo, precio y botón de “Añadir al carrito”.
- **Carrito de compras:** listado de entradas seleccionadas y confirmación de compra.
- **Mis tickets:** historial de tickets con opción de descarga o envío.
- **Panel de vendedor:** formulario para creación y edición de eventos.
- **Panel de administración:** control de usuarios y eventos con acciones de suspensión/eliminación.

2-Interfaces de hardware

Dispositivos soportados:

- Computadores con navegadores web modernos (Google Chrome, Mozilla Firefox, Microsoft Edge).
- Dispositivos móviles Android (versión 9.0 o superior).

Otros dispositivos:

- Impresoras, opcionalmente, para imprimir tickets en formato PDF.

Protocolos de comunicación hardware-software:

- Uso de red (WiFi/4G/5G) para sincronización de datos en tiempo real.
- Soporte a dispositivos de almacenamiento local para descargas (PDF de tickets).

3-Interfaces de software

Base de datos:

- Motor: **MySQL** (producción) o **H2** (desarrollo y pruebas).

Backend:

- Desarrollado en **Java con Spring Boot**, con arquitectura basada en **API REST**.

Frontend Web:

- Desarrollado con **React (Vite + Tailwind CSS)**.

Frontend Móvil:

- Desarrollado en **Kotlin con Jetpack Compose** en Android Studio.
- Librerías externas: **Material3, Retrofit** y otras

Bibliotecas externas:

- **Spring Security + JWT** para autenticación.
- **Librerías de generación de PDF y QR** para emisión de tickets digitales.
- Otras librerías varias

Sistemas operativos soportados:

- Windows, macOS y Linux para acceso web.
- Android para la app móvil.

4-Interfaces de comunicación

Protocolos de red soportados:

- **HTTP/HTTPS**: comunicación entre cliente y servidor.
- **SMTP**: para el envío de correos electrónicos con confirmaciones y tickets adjuntos.

Formatos de mensajería:

- **JSON**: intercambio de datos entre frontend y backend.
- **PDF**: formato estándar para la entrega de tickets.

Navegadores soportados:

- Google Chrome, Mozilla Firefox, Microsoft Edge.

Seguridad en las comunicaciones:

- Todas las comunicaciones deben realizarse mediante **HTTP/HTTPS**
- Tokens **JWT** para autenticación segura y gestión de sesiones.

10.Requerimientos no funcionales

Rendimiento

- **RNF-01**: El sistema debe poder soportar al menos **50 usuarios concurrentes** sin pérdida significativa de rendimiento.
- **RNF-02**: El tiempo de respuesta en operaciones críticas (login, búsqueda de eventos, compra de entradas) no debe superar los **3 segundos** bajo carga normal.
- **RNF-03**: Las consultas de eventos deben ejecutarse en un máximo de **2 segundos** en promedio.

Seguridad

- **RNF-04**: Todas las comunicaciones entre cliente y servidor deben realizarse mediante **HTTPS con cifrado SSL/TLS**.
- **RNF-05**: El sistema debe implementar **autenticación basada en JWT** para acceso seguro.
- **RNF-06**: Las contraseñas deben almacenarse de forma encriptada siguiendo estándares de la industria (ej. algoritmo **bcrypt**).

Usabilidad

- **RNF-07:** La interfaz debe ser **intuitiva y fácil de usar**, con un tiempo de aprendizaje estimado menor a **10 minutos** para un usuario nuevo.
- **RNF-08:** La aplicación debe ofrecer **modo claro y modo oscuro** personalizables.
- **RNF-09:** La navegación debe requerir un máximo de **3 clics** para acceder a las funciones principales (buscar eventos, comprar entrada, ver tickets).

Confiabilidad y disponibilidad

- **RNF-10:** El sistema debe estar disponible el **99% del tiempo** en condiciones normales de operación.
- **RNF-11:** En caso de fallo del sistema, la recuperación no debe superar los **5 minutos**.

Compatibilidad

- **RNF-13:** La aplicación web debe ser compatible con los navegadores más utilizados (Google Chrome, Mozilla Firefox, Microsoft Edge).
- **RNF-14:** La aplicación móvil debe ser compatible con dispositivos **Android 9.0 o superior**.
- **RNF-15:** La interfaz debe ser **responsive**, adaptándose automáticamente a pantallas de distintos tamaños (PC, tablet, móvil).

Mantenibilidad y escalabilidad

- **RNF-16:** El código debe estar documentado siguiendo buenas prácticas de desarrollo (JavaDoc, comentarios en frontend).
- **RNF-17:** El sistema debe ser modular, permitiendo la **incorporación de nuevas funcionalidades** sin afectar la estabilidad general..

Portabilidad

- **RNF-18:** El backend debe poder ejecutarse sin problemas en Windows y Linux
- **RNF-19:** El frontend web debe funcionar en resoluciones desde **1920x1080px** hasta **4K**.

11.Otros requerimientos

Internacionalización:

ShowPass debe estar preparado para soportar al menos dos idiomas: español (por defecto) e inglés (versión futura).

Legales:

Cumplimiento del Reglamento General de Protección de Datos (RGPD) para garantizar la seguridad y privacidad de la información personal de los usuarios. Los datos sensibles deberán almacenarse cifrados y no compartirse con terceros sin consentimiento.

Base de datos:

Cada usuario debe tener un correo único como identificador.

Los eventos eliminados deben marcarse como “inactivos” para mantener trazabilidad.

Reutilización y escalabilidad:

Los módulos de autenticación, gestión de usuarios y generación de tickets (PDF/QR) deberán diseñarse de manera modular para su reutilización en futuras versiones o proyectos relacionados.

El sistema debe estar preparado para integrar pasarelas de pago reales (Stripe, PayPal, etc.) en futuras fases.

12.Glosario

JWT (JSON Web Token): Mecanismo de autenticación y autorización basado en tokens seguros firmados digitalmente.

QR (Quick Response Code): Código bidimensional utilizado para validar y verificar entradas digitales.

PDF (Portable Document Format): Formato estándar utilizado para la entrega de tickets digitales descargables.

Spring Boot: Framework de desarrollo backend en Java que facilita la creación de aplicaciones con arquitectura REST.

REST API (Representational State Transfer): Estilo arquitectónico que define cómo los sistemas interactúan mediante peticiones HTTP.

Jetpack Compose: Toolkit moderno de UI para aplicaciones Android desarrollado por Google.

Tailwind CSS: Framework de CSS orientado a utilidades que facilita el diseño responsive y personalizable en frontend web.

MySQL: Sistema de gestión de bases de datos relacional utilizado en producción.

H2: Base de datos en memoria ligera utilizada para entornos de pruebas y desarrollo.

SMTP (Simple Mail Transfer Protocol): Protocolo estándar de comunicación usado para el envío de correos electrónicos.