



# PROYECTO GOPASS

INTAKE 003  
2024



# PROYECTO GOPASS

## Documentación



### El problema:

La reventa de entradas falsas y la falta de organización en el transporte hacia los partidos de fútbol en Argentina son problemas serios que afectan a miles de fanáticos. La reventa ilegal no solo implica la pérdida económica para quienes adquieren entradas fraudulentas, sino también un impacto emocional significativo, como se vio en el caso del amistoso entre Argentina y Panamá en marzo de 2023. En ese partido, más de 400 personas fueron estafadas con entradas falsas, algunas de las cuales pagaron hasta 200 mil pesos por un ticket falso.

El problema persiste debido a la falta de un sistema robusto de verificación y el fácil acceso a canales de reventa no oficiales.

Otro reto es la falsificación masiva de entradas. Las plataformas digitales han facilitado el acceso a entradas ilegítimas, y muchas veces, estas falsificaciones no son detectadas hasta que los fanáticos intentan ingresar al estadio. Este tipo de estafas son recurrentes en eventos de alto perfil, como los partidos de la selección o de equipos populares, donde la alta demanda de entradas crea un mercado negro. La ausencia de un sistema centralizado de verificación y la venta descontrolada fuera de los canales oficiales contribuyen al problema.



# PROYECTO GOPASS

## Documentación



### La solución:

Para mitigar el problema de la reventa y las entradas falsas que enfrentan los aficionados, Igrowker propone GoPass, una solución web que garantiza la autenticidad de las entradas a eventos deportivos mediante el uso de códigos QR únicos y una API de Ticketmaster. Esta API permite la verificación de entradas en tiempo real, lo que asegura que cada entrada es genuina y está vinculada a una compra oficial. Como las APIs de los clubes de fútbol suelen ser privadas y de difícil acceso, GoPass utilizará la API pública de Ticketmaster para emular la validación, lo que permite una integración segura y eficiente sin depender de sistemas propietarios difíciles de integrar. El enfoque asegura que los aficionados puedan validar sus entradas de manera sencilla y evitar la compra de boletos fraudulentos.

Además, GoPass implementa una solución segura para la reventa de entradas, donde cada usuario que desee publicar un boleto para reventa debe contar con un ID único. Este ID se genera tras un proceso de verificación que incluye la confirmación del número de documento de identidad (DNI) y el número de teléfono del usuario, validado a través de un código de verificación. De esta manera, cualquier entrada puesta en venta estará vinculada a un usuario verificado, lo que reduce las posibilidades de fraude y asegura que tanto el comprador como el vendedor se sientan protegidos durante la transacción.

Con el uso de la API de Ticketmaster para validar la autenticidad de las entradas y un sistema de identificación único para los usuarios que revenden, GoPass se posiciona como una plataforma segura y escalable. Esta solución no solo mitiga el problema de las entradas falsas en el mercado secundario, sino que también centraliza el proceso de verificación y reventa, lo que brinda confianza a los organizadores de eventos y a los aficionados. Además, la simplicidad del proceso permite una rápida implementación por parte de los desarrolladores, lo que convierte a GoPass en una herramienta práctica y viable para combatir la falsificación de entradas en eventos deportivos.

# PROYECTO GOPASS

## Documentación



La presente documentación intenta cubrir todos los aspectos técnicos y funcionales para que puedan desarrollar el MVP sin mayores dificultades. Si bien Igrowker hace el esfuerzo por proveerles todo lo necesario para empezar, pueden existir áreas con poca definición o profundidad de detalle. Es parte del equipo juntarse y discutir cómo resolver este tipo de situaciones y determinar acciones concretas a seguir.

## Historias de usuario

Las siguientes historias de usuario, que se enmarcan en lo que llamamos “requerimientos funcionales”, son esenciales y conforman parte del MVP. El equipo es libre de agregar más funcionalidad, siempre y cuando no comprometa la entrega de las que aquí se exponen.

### **Historia 1: Verificación de entradas**

- **Título:** Verificación de entrada para un evento
- **Detalle:** **COMO** fanático de fútbol, **QUIERO** verificar la autenticidad de mi entrada, **PARA QUE** pueda asegurarme de que es válida y evitar estafas.
- **Criterios de aceptación:**
  - El sistema debe permitir escanear o ingresar manualmente el código QR de la entrada.
  - El sistema debe validar la entrada a través de la API de Ticketmaster.
  - El usuario debe recibir una confirmación clara de la validez o invalidez de la entrada.



### **Historia 2: Publicación de entradas para reventa**

- **Título:** Publicación de entrada para reventa
- **Detalle:** **COMO** usuario verificado, **QUIERO** publicar una entrada para reventa, **PARA QUE** pueda venderla de manera segura a otro usuario.
- **Criterios de aceptación:**
  - El sistema debe validar la identidad del usuario mediante DNI y número de teléfono.
  - Solo los usuarios verificados pueden publicar entradas para reventa.
  - Las entradas en reventa deben vincularse a un usuario específico y validarse antes de ser listadas.

### **Historia 3: Compra de entradas en reventa**

- **Título:** Compra de entrada en reventa
- **Detalle:** **COMO** comprador de entradas, **QUIERO** adquirir entradas verificadas en reventa, **PARA QUE** pueda asistir al evento sin riesgo de estafas.
- **Criterios de aceptación:**
  - El sistema debe mostrar solo entradas verificadas disponibles para reventa.
  - Al completar la compra, la entrada debe transferirse de manera segura al nuevo usuario.
  - El comprador debe recibir un código QR nuevo y exclusivo vinculado a su cuenta.



### **Historia 4: Registro de usuario verificado**

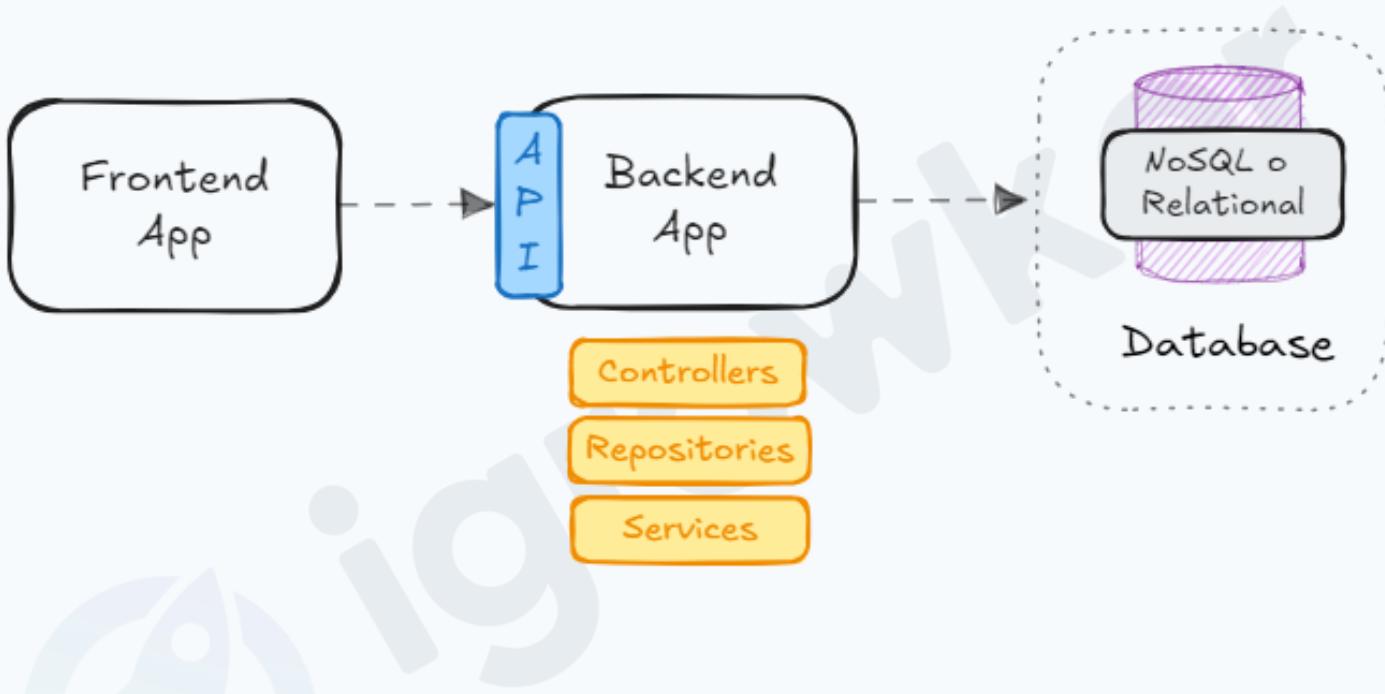
- **Título:** Registro de usuario con verificación de identidad
- **Detalle:** **COMO** usuario de la plataforma, **QUIERO** registrar mi cuenta con verificación de identidad, **PARA QUE** pueda vender y comprar entradas de manera segura.
- **Criterios de aceptación:**
  - El sistema debe requerir la verificación mediante DNI y número de teléfono.
  - El usuario debe recibir un código de verificación por SMS para confirmar su identidad.
  - Una vez completada la verificación, el usuario podrá publicar y comprar entradas.



## Requerimientos no funcionales

### Arquitectura de la aplicación

El esquemático de la arquitectura es el siguiente:



Para desplegar las aplicaciones front y back, el encargado del DevOps puede elegir cualquiera de las opciones gratuitas que existen.

Si el equipo no tiene DevOps, designar una persona responsable de hacer los despliegues.

# PROYECTO GOPASS

## Documentación



### Aplicación backend

#### a. Endpoints de la API REST

1. POST /api/verificar-entrada: Verifica la autenticidad de una entrada a través de la API de Ticketmaster.
2. POST /api/publicar-entrada-reventa: Permite a los usuarios verificados publicar una entrada para reventa.
3. POST /api/comprar-entrada-reventa: Permite a un usuario comprar una entrada en reventa.
4. POST /api/registro-usuario: Registra un nuevo usuario y realiza la verificación de identidad.

#### b. Controladores

1. VerificacionController: Controla la verificación de entradas utilizando la API de Ticketmaster.
2. ReventaController: Gestiona la publicación y compra de entradas en reventa.
3. UsuarioController: Maneja el registro y verificación de usuarios.

#### c. Servicios

1. VerificacionService: Realiza las llamadas a la API de Ticketmaster para verificar la autenticidad de las entradas.
2. ReventaService: Implementa la lógica de negocio para la publicación, compra y transferencia segura de entradas.
3. UsuarioService: Gestiona la lógica de verificación de identidad y registro de usuarios.



### d. Repositorios

1. EntradaRepository: Gestiona la persistencia de entradas publicadas y compradas.
2. UsuarioRepository: Maneja la persistencia de usuarios registrados y verificados.
3. ReventaRepository: Almacena las transacciones de reventa de entradas.

# PROYECTO GOPASS

## Documentación



### e. Modelo de datos

```
● ● ●  
public class Entrada {  
    public int Id { get; set; }  
    public string CódigoQR { get; set; }  
    public bool Verificada { get; set; }  
    public int UsuarioId { get; set; }  
}  
  
public class Usuario {  
    public int Id { get; set; }  
    public string Nombre { get; set; }  
    public string DNI { get; set; }  
    public string NumeroTelefono { get; set; }  
    public bool Verificado { get; set; }  
}  
  
public class Reventa {  
    public int Id { get; set; }  
    public int EntradaId { get; set; }  
    public int VendedorId { get; set; }  
    public int CompradorId { get; set; }  
    public DateTime FechaReventa { get; set; }  
    public decimal Precio { get; set; }  
}
```

# PROYECTO GOPASS

## Documentación



### Aplicación Frontend

#### a. Componentes

1. VerificarEntradaComponent: Permite a los usuarios escanear o ingresar manualmente el código QR de una entrada para verificar su validez.
2. PublicarReventaComponent: Componente para que los usuarios verificados publiquen entradas para reventa.
3. ComprarReventaComponent: Permite a los usuarios visualizar y comprar entradas en reventa.
4. RegistroVerificacionComponent: Componente que gestiona el registro de nuevos usuarios y su verificación mediante DNI y teléfono.

#### b. Servicios

1. VerificacionService: Se encarga de conectar con el backend para verificar la validez de las entradas a través de la API de Ticketmaster.
2. ReventaService: Gestiona la publicación y compra de entradas en reventa, incluyendo la transferencia segura de las mismas.
3. UsuarioService: Proporciona las funciones necesarias para el registro y verificación de los usuarios.

# PROYECTO GOPASS

## Documentación



### Base de datos

#### a. Entidades con sus atributos

- Entrada:
  - Id: int
  - CódigoQR: string
  - Verificada: bool
  - Usuarioid: int
- Usuario:
  - Id: int
  - Nombre: string
  - DNI: string
  - NumeroTelefono: string
  - Verificado: bool
- Reventa:
  - Id: int
  - Entradald: int
  - Vendedorld: int
  - Compradorld: int
  - FechaReventa: DateTime
  - Precio: decimal

#### b. Relaciones entre entidades

- Entrada tiene una relación 1 a 1 con Usuario (cada entrada está vinculada a un usuario).
- Reventa tiene una relación 1 a 1 con Entrada y relaciones 1 a 1 con Usuario tanto para el vendedor como para el comprador.



### c. Recomendación de base de datos

Una base de datos relacional es la opción más adecuada para este proyecto debido a las relaciones entre entradas, usuarios y transacciones de reventa. Bases de datos como SQL Server o PostgreSQL son recomendables, ya que permiten mantener la integridad de los datos y realizar transacciones seguras y eficientes, lo que es crucial para la verificación de entradas y las transacciones de reventa.

# PROYECTO GOPASS

## Documentación



### INFORMACIÓN PARA EL EQUIPO DE QA MANUAL

Tu rol es fundamental para garantizar que la plataforma funcione correctamente y proporcione una excelente experiencia de usuario.

Te detallamos acá la información necesaria para que puedas llevar a cabo tus tareas de manera eficiente.

**Claridad en la comunicación:** Documentar todos los hallazgos y comunicar problemas de buenas maneras al equipo de desarrollo. Incluir pasos para reproducir errores y capturas de pantalla.

**Atención al detalle:** Prestar atención a los pequeños detalles que pueden impactar la experiencia del usuario. Verificar que todos los elementos de la interfaz funcionen correctamente.

**Paciencia y persistencia:** Algunas pruebas van a requerir repetición y revisión minuciosa para asegurar el correcto funcionamiento de la plataforma. No apresurarse para evitar pasar por alto problemas.

**Colaboración:** Trabajar estrechamente con desarrolladores y diseñadores para entender los cambios y asegurar que todos los problemas sean abordados.

### Cosas que No pueden dejar pasar

**Errores en el proceso de registro y login:** Asegurarse de que el registro, inicio de sesión y autenticación de dos factores funcionen sin problemas.

**Problemas en el perfil de usuario:** Verificar que los usuarios puedan actualizar su perfil correctamente y que toda la información se muestre adecuadamente.

**Problemas de responsividad:** Asegurarse de que la plataforma funcione correctamente en móviles, tabletas y ordenadores, ajustándose adecuadamente a distintas resoluciones de pantalla.

# PROYECTO GOPASS

## Documentación



### Consideraciones técnicas:

Equipo:

Mientras trabajan en el desarrollo del proyecto, queremos asegurarles que tienen total libertad para usar herramientas de IA como ChatGPT, Gemini o Copilot para facilitar y mejorar la codificación.

La IA puede ser una excelente aliada para resolver dudas, obtener sugerencias y acelerar el proceso de desarrollo. Sin embargo, es fundamental que se aseguren de que cada línea de código que escriban sea eficiente, reutilizable y siga las mejores prácticas.

Recuerden siempre revisar y entender qué es lo que hace el código sugerido por la IA antes de integrarlo en el proyecto. La comprensión total de lo que están implementando les va a permitir garantizar la calidad y coherencia del código.

### Buenas prácticas de programación:

Código coherente: Asegurarse de que el código sea lógico y consistente. Revisá y refactorizá según sea necesario para mantener la claridad. Utilizá variables para evitar repetir código. Utilizá estados globales y hooks.

Reutilización: Identificá y extraé partes del código que puedan ser reutilizadas en diferentes módulos o componentes. Cada función debe dedicarse a una sola tarea.

Documentación: Comentá tu código de manera adecuada (no te pases, más es menos) y mantené la documentación (Postman, README) actualizada para facilitar la comprensión y mantenimiento.

Pruebas: Escribí testing unitario (en la medida que el tiempo alcance) y de integración para asegurar que tu código funciona correctamente y es robusto ante cambios.

Revisión de código: Si sobra tiempo pedile a un compañero que revise tu código para detectar posibles mejoras.

Estamos entusiasmados de ver tu creatividad y habilidades en acción. Confiamos en que, utilizando las herramientas manera consciente y aplicando principios de diseño sólidos, vas a crear un producto de alto nivel.

**¡Manos a la obra Juniors de Igrowker!**