

# Prueba Técnica – Fullstack Developer

---

## Introducción

La presente prueba busca evaluar tus habilidades como desarrollador fullstack en un escenario real. Trabajarás con un nuevo modo de juego llamado Survivor, donde deberás implementar tanto el backend (Node.js + Express + MongoDB) como un frontend en Flutter, basado en un diseño de referencia en Figma.

## Introducción Survivor

**Survivor** es un nuevo modo de juego basado en eliminación progresiva, donde el objetivo es **sobrevivir la mayor cantidad de jornadas posibles** eligiendo equipos que ganen sus partidos cada semana.

Cada jugador debe hacer un pick antes del inicio de cada jornada, seleccionando un equipo que crea que ganará su partido. Si el equipo elegido pierde o empata, el jugador pierde vidas. Si se queda sin vidas, queda eliminado.

El último jugador en pie será coronado como el ganador. Si más de uno sobrevive hasta el final del torneo, el premio se lo lleva el que tenga más vidas.

## Objetivo del Test

- Implementar un backend en Node.js/Express con MongoDB.
- Construir un frontend en Flutter que consuma dicho backend.
- Aplicar criterios de UI/UX tomando como base el diseño en Figma.
- Documentar el proceso de desarrollo y las decisiones tomadas.

## Duración

La prueba tiene una duración máxima de 6 horas.

## Parte 1 – Backend (3 horas)

Implementar un mini backend con Node.js, Express y MongoDB con las siguientes rutas mínimas:

1. GET / → Obtener Survivor.
2. POST /join/:survivorId → Unirse a un Survivor (crea registro en gambles\_survivor y predictions\_survivor).
3. POST /pick → Seleccionar equipo por jornada.

## Parte 2 – Frontend (2.5 horas)

Construir una app Flutter sencilla que consuma el backend:

- Pantalla de detalle de Survivor con partidos.
- Opción para unirse a un Survivor.
- Selección de equipo por jornada.
- Visualización de vidas restantes y resultados pasados.

## Parte 3 – UX & Documentación (1.5 horas)

1. Mejorar la experiencia de usuario con detalles como estados de carga y mensajes de error claros.
2. Entregar un README con instrucciones para correr el backend y frontend.
3. Explicar brevemente las decisiones tomadas durante el desarrollo.

### **Criterios de Evaluación**

- Backend (30%): correcta implementación de colecciones, rutas y validaciones.
- Frontend (30%): integración con backend, fidelidad al diseño y navegación fluida.
- UX (20%): claridad, consistencia y detalles de interacción.
- Organización/Documentación (20%): estructura clara, README y buenas prácticas.

### **Boilerplate proporcionado**

Se entrega un proyecto base con:

- Backend con datos iniciales

### **Entregables**

- Código fuente en un repositorio (GitHub o similar).
- Instrucciones de ejecución (README).
- Capturas de pantalla o video corto mostrando el flujo principal.