

# Prueba Técnica CTS Turismo

---

## 1. Forma de entrega

El desarrollo debe entregarse mediante un repositorio en GIT (GitHub, GitLab o Bitbucket).

El repositorio debe contener:

- Código fuente del backend y frontend.
- Archivo README.md con:
  - Instrucciones para instalar y ejecutar el proyecto (backend y frontend).
  - Explicación breve de las decisiones técnicas tomadas.
  - Listado de endpoints principales con ejemplos de request/response.
  - (Opcional) Screenshots o GIFs del funcionamiento.

## Datos de envío

El enlace al repositorio debe ser enviado a los correos:

- cristian.bustos@ctsturismo.cl

## Plazo de entrega

El plazo máximo de entrega es de 3 días corridos desde el momento del envío del correo con la prueba.

## 2. Lenguajes, frameworks y herramientas requeridas

### Backend

- Python 3.x
- Django
- Django Rest Framework
- Celery y Redis (para manejo de tareas asíncronas)

### Frontend

- VueJs o Nuxt

## 3. Problema a resolver

Debes desarrollar una aplicación Full Stack para gestionar un sorteo de San Valentín.

El premio es una estadía de 2 noches todo pagado para una pareja en un hotel.

El flujo del concurso debe ser el siguiente:

1. Inscripción: el usuario ingresa sus datos personales y se registra.
2. Validación de correo: si el email ya está inscrito, el sistema no permite un registro

duplicado.

3. Verificación de correo: el sistema envía un correo con un enlace de verificación.
4. Creación de contraseña: una vez verificado, el usuario crea su contraseña y activa su cuenta.
5. Confirmación de participación: al completar estos pasos, el usuario ya está participando y debe ver un mensaje claro.
6. Selección de ganador: el administrador puede seleccionar al azar un ganador entre los participantes válidos.
7. Notificación al ganador: se envía un correo automático al ganador.

## 4. Requerimientos del Frontend

El frontend debe incluir al menos 5 vistas principales:

1. Página de inscripción al concurso (pública):
  - Formulario de inscripción (nombre completo, correo electrónico y teléfono).
  - Validación para evitar registros duplicados.
  - Al finalizar, mostrar mensaje: "¡Gracias por registrarte! Revisa tu correo para verificar tu cuenta."
2. Pantalla de verificación de correo y creación de contraseña (pública):
  - Acceso desde enlace recibido en el correo.
  - Confirmación de verificación y creación de contraseña.
  - Mensaje final: "Tu cuenta ha sido activada. Ya estás participando en el sorteo."
3. Pantalla de login de administrador (pública):
  - Inicio de sesión exclusivo para administrador del hotel.
4. Panel de administración – Lista de concursantes (protegida):
  - Listado de concursantes (nombre, correo, estado de verificación).
  - Búsqueda y filtrado opcional.
5. Panel de administración – Sorteo de ganador (protegida):
  - Botón para generar un ganador aleatorio.
  - Mostrar en pantalla el nombre y correo del ganador.
  - Enviar correo automático de notificación al ganador.

## 5. Consideraciones importantes

- El sistema debe soportar un alto volumen de concursantes (rendimiento y escalabilidad).
- Los correos deben enviarse de manera asíncrona usando Celery + Redis.
- Buenas prácticas esperadas:
  - Organización clara del proyecto.

- Manejo de errores.
  - Validaciones robustas.
  - Seguridad (protección de rutas, contraseñas encriptadas).
- Tests unitarios o de integración serán considerados un plus.

## 6. Qué se evaluará

- Correctitud: que el flujo funcione según lo solicitado.
- Calidad del código: organización, legibilidad y buenas prácticas.
- Diseño de la API: endpoints claros, validaciones y manejo de errores.
- Frontend: experiencia de usuario y cumplimiento de requerimientos.
- Documentación: claridad en el README para ejecutar y probar el proyecto.