

## Prueba - Skate Park

- Para realizar esta prueba debes haber estudiado previamente todo el material disponible en el LMS correspondiente al módulo.
- Una vez terminada la prueba, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el .zip en el LMS.
- Puntaje total: 10 puntos
- Desarrollo prueba: Individual
- Para la realización de la prueba necesitarás el archivo disponible en el LMS llamado *Apoyo Prueba - Skate Park*.

### Habilidades a evaluar

- Crear una API REST con el Framework Express.
- Servir contenido dinámico con express-handlebars.
- Ofrecer la funcionalidad Upload File con express-fileupload.
- Implementar seguridad y restricción de recursos o contenido con JWT.

### Descripción

La Municipalidad de Santiago, ha organizado una competencia de Skate para impulsar el nivel deportivo de los jóvenes que desean representar a Chile en los X Games del próximo año, y han iniciado con la gestión para desarrollar la plataforma web en la que los participantes se podrán registrar y revisar el estado de su solicitud

En esta prueba deberás ocupar todos tus conocimientos para desarrollar un sistema que involucre tus habilidades como Full Stack Developer, consolidando tus competencias en el frontend y backend.

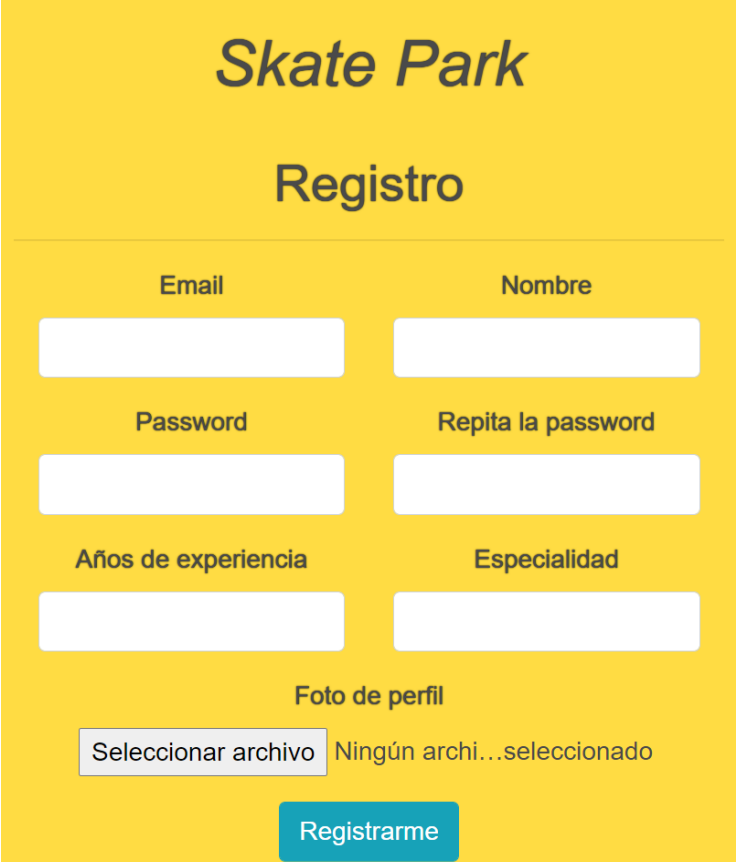
Las tecnologías y herramientas que deberás ocupar son las siguientes:

- Express
- Handlebars
- PostgreSQL
- JWT
- Express-fileupload

A continuación, se muestra la interfaz gráfica que tendrás disponible como material de apoyo. No obstante siéntete libre de maquetar a tu gusto siempre y cuando cumplas con los requerimientos:

Skate Park					
Lista de participantes					
#	Foto	Nombre	Años de experiencia	Especialidad	Estado
1		Tony Hawk	12	Kickflip	Aprobado
2		Evelien Bouilliant	10	Heelflip	Aprobado
3		Danny Way	8	Ollie	En revisión
<a href="#">Iniciar Sesión</a>   <a href="#">Registrarme</a>					

Imagen 1. Lista de participantes  
Fuente: Desafío Latam



**Skate Park**

**Registro**

---

Email

Nombre

Password

Repita la password

Años de experiencia

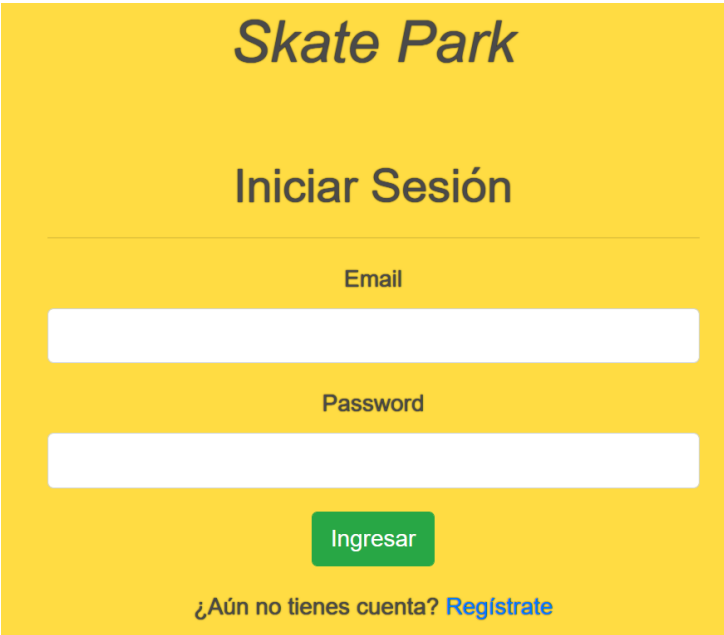
Especialidad

Foto de perfil

Seleccionar archivo Ningún archi...seleccionado

**Registrarme**

Imagen 2. Registro de participantes  
Fuente: Desafío Latam



**Skate Park**

**Iniciar Sesión**

---

Email

Password

**Ingresar**

¿Aún no tienes cuenta? [Regístrate](#)

Imagen 3. Inicio de sesión  
Fuente: Desafío Latam

## Skate Park

### Datos del perfil

Email	Nombre
tonyhawk@skate.com	Tony Hawk
Password	Repita la password
*****	*****
Años de experiencia	Especialidad
12	Kickflip

[Actualizar](#)

[Eliminar cuenta](#)

Imagen 4. Modificación de los datos del perfil  
Fuente: Desafío Latam

## Skate Park

### Administración



#	Foto	Nombre	Años de experiencia	Especialidad	Estado
1		Tony Hawk	12	Kickflip	<input checked="" type="checkbox"/>
2		Evelien Bouilliant	10	Heelflip	<input checked="" type="checkbox"/>
3		Danny Way	8	Ollie	<input type="checkbox"/>

Imagen 5. Administración  
Fuente: Desafío Latam

### Indicaciones:

- El sistema debe permitir registrar nuevos participantes.
- Se debe crear una vista para que los participantes puedan iniciar sesión con su correo y contraseña.
- Luego de iniciar la sesión, los participantes deberán poder modificar sus datos, exceptuando el correo electrónico y su foto. Esta vista debe estar protegida con JWT y los datos que se utilicen en la plantilla deben ser extraídos del token.
- La vista correspondiente a la ruta raíz debe mostrar todos los participantes registrados y su estado de revisión.
- La vista del administrador debe mostrar los participantes registrados y permitir aprobarlos para cambiar su estado.

Se debe persistir la información de los usuarios en PostgreSQL, por lo que deberás usar las siguientes sentencias SQL para la creación de la base de datos y la tabla de participantes.

```
CREATE DATABASE skatepark;
```

```
CREATE TABLE skaters (id SERIAL, email VARCHAR(50) NOT NULL, nombre  
VARCHAR(25) NOT NULL, password VARCHAR(25) NOT NULL, anos_experiencia  
INT NOT NULL, especialidad VARCHAR(50) NOT NULL, foto VARCHAR(255) NOT  
NULL, estado BOOLEAN NOT NULL);
```

### Requerimientos

1. Crear una API REST con el Framework Express (**3 Puntos**)
2. Servir contenido dinámico con express-handlebars (**3 Puntos**)
3. Ofrecer la funcionalidad Upload File con express-fileupload (**2 Puntos**)
4. Implementar seguridad y restricción de recursos o contenido con JWT (**2 Puntos**)