

## Desafío - Abracadabra

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera grupal.
  - Para la realización del desafío necesitarás apoyarte del archivo *Apoyo Desafío - Abracadabra*.

### Capítulos

El desafío está basado en los siguientes capítulos de la lectura:

- ¿Qué es Express?
- Express vs Node puro
- Creación de rutas
- Objetos Request y Response
- Middlewares
- Devolución de sitios web estáticos

### Descripción

Los sitios web en la actualidad son la cara visible de las empresas en el mundo digital, por ello, requieren disponer de un espacio de fácil navegación e interesante para los usuarios.


En este contexto, el instituto recreativo más conocido de la ciudad, contrató recientemente una asesoría de marketing para aumentar el tráfico en su sitio web. La consultora, en base al análisis realizado a la empresa contratante, les propone aumentar la interacción con los usuarios que visitan su sitio web, a través de la creación de juegos de azar con temáticas relacionadas a los diferentes shows que ofrece el instituto. Además, propone que estos juegos solo puedan realizarlos usuarios registrados, con el fin de generar más fidelidad con su público.

En este desafío, deberás desarrollar un servidor con Express que sirva un sitio web estático con temática de magia, donde se presenten 4 sombreros y al hacer click en uno de estos encontrar el conejo oculto.

El servidor deberá disponibilizar las siguientes rutas:

1. **/abracadabra/usuarios:** Se debe devolver un JSON con un arreglo de nombres alojado en el servidor.
2. **/abracadabra/juego/:usuario:** A través de un middleware, verificar que el usuario escrito como parámetro existe en el arreglo alojado en el servidor.
3. **/abracadabra/conejo/:n:** Basado en un número aleatorio del 1 al 4, devolver la foto del conejo en caso de coincidir con el número recibido como parámetro ó devolver la foto de Voldemort en caso de no coincidir.

Las siguientes imágenes representan las interacciones que debe tener el sitio web una vez terminado el desafío.



```
1 // 20201112235318
2 // http://localhost:3000/abracadabra/usuarios
3
4 {
5   "usuarios": [
6     "Juan",
7     "Jocelyn",
8     "Astrid",
9     "Maria",
10    "Ignacia",
11    "Javier",
12    "Brian"
13  ]
14 }
```

Imagen 1. Demostración del JSON que se debe obtener al consultar la ruta 1.  
Fuente: Desafío Latam



Imagen 2. Interfaz de la aplicación cliente que se debe recibir al consultar la ruta 2.  
Fuente: Desafío Latam



Imagen 3. Imagen que se debe devolver en caso de escribir en la ruta 2.  
Fuente: Desafío Latam

Si se hace click en un sombrero cuyo número no coincide con el número generado en el servidor de forma aleatoria, se debe devolver la imagen de Voldemort.



Imagen 4. Imagen si el número no coincide con el número generado de forma aleatoria.  
Fuente: Desafío Latam

Si se hace click en un sombrero cuyo número sí coincide con el número generado en el servidor de forma aleatoria, se debe devolver la imagen del Conejo.



Imagen 4. Imagen si el número coincide con el número generado de forma aleatoria.  
Fuente: Desafío Latam

Se debe devolver un mensaje diciendo “Esta página no existe” en caso de consultar una ruta que no esté definida en el enrutador de express.

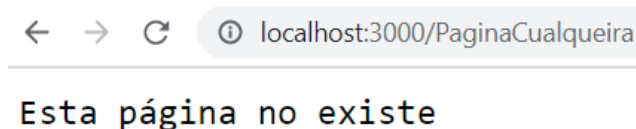


Imagen 5. Mensaje que se debe recibir al consultar una ruta que no existe en el servidor  
Fuente: Desafío Latam

En el **Apoyo Desafío - Abracadabra** se disponibiliza el index.html y las imágenes correspondientes. No obstante siéntete libre de modificar la interfaz del desafío o crear tu propia maqueta, siempre y cuando cumpla con los requisitos y continúe la temática planteada.

## Requerimientos

1. Crear un servidor con Express en el puerto 3000.
2. Definir la carpeta “assets” como carpeta pública del servidor.
3. Crear en el servidor un arreglo de nombres y devolverlo en formato JSON a través de la ruta **/abracadabra/usuarios**.
4. Crear un middleware con la ruta **/abracadabra/juego/:usuario** para validar que el usuario recibido como parámetro “usuario” existe en el arreglo de nombres creado en el servidor.

En caso de ser exitoso, permitir el paso a la ruta **GET** correspondiente, de lo contrario devolver la imagen “who.jpeg”.

5. Crear una ruta **/abracadabra/conejo/:n** que valide si el parámetro “n” coincide con el número generado de forma aleatoria.

En caso de ser exitoso, devolver la imagen del conejo, de lo contrario devolver la imagen de Voldemort.

6. Crear una ruta genérica que devuelva un mensaje diciendo “Esta página no existe...” al consultar una ruta que no esté definida en el servidor.