

## Desarrollo Web Full Stack Node

Ejercitación - M05C01

# Héroes digitales

### Introducción

En esta oportunidad queremos hacer un sitio usando la tecnología de Express.

Este sitio busca mostrar de manera sencilla la información de las *heroínas* y los *héroes tecnológicos* más representativos del ámbito de la programación.

Así que sin más preámbulos te dejamos aquí las consignas de este desarrollo.

### Requisitos

- **Buscar una dupla:** para poder iniciar con este desarrollo te solicitamos trabajar en la modalidad **PAIR PROGRAMMING**, pues estamos convencidos que cada quien puede aprender infinitudes de su par.
- **Leer TODO antes de codear:** una vez te hayamos designado una dupla les sugerimos que lean en conjunto las consignas (hasta la página 4). Pues si bien entendemos la "*ansiedad*" por comenzar, está bueno saber que tienen que hacer y así poder dividir tareas.

## Objetivo

Para la construcción de este pequeño sitio web se espera contar con la posibilidad de acceso a las siguientes URL's:

- /
- /heroes
- /heroes/id/profesion - (id = número, obligatorio)
- /heroes/id/resenia/tipo - (id = número, obligatorio / tipo = string, optativo)
- /creditos

**¡Atención!** Las anteriores deberán ser direcciones de acceso a la aplicación desde el navegador.

## Consignas

A continuación se detalla lo que deberá mostrar cada ruta.

### 1. Home (/)

Deberá mostrar un texto de bienvenida al sitio. El texto será: *Ni Superman, Iron Man o La Mujer Maravilla son tan importantes cómo las y los héroes de carne y hueso que encontrarás en este sitio. Esperamos que ellas y ellos te sirvan como inspiración para poder cumplir tus objetivos.*

### 2. Héroes (/heroes)

Deberá mostrar todo el contenido del archivo **heroes.json**. En el navegador se deberá ver algo así:

```

id: 1
nombre: "Ada Lovelace"
profesion: "Matemática e Informática"
pais: "Reino Unido"
▼ resenia: "Augusta Ada King, Condesa de Lovelace (Londres, 10 de diciembre de 1815–íd., 27 de noviembre de 1852), registrada al nacer como Augusta Ada Byron y conocida habitualmente como Ada Lovelace, fue una matemática, informática y escritora británica, célebre sobre todo por su trabajo acerca de la calculadora de uso general de Charles Babbage, la denominada máquina analítica. Entre sus notas sobre la máquina, se encuentra lo que se reconoce hoy como el primer algoritmo destinado a ser procesado por una máquina, por lo que se la considera como la primera programadora de ordenadores.123"

▼ 1:
id: 2
nombre: "Charles Babbage"
profesion: "Matemático y Científico de la Computación"
pais: "Reino Unido"
▼ resenia: "Charles Babbage FRS (Teignmouth, Devonshire, Gran Bretaña, 26 de diciembre de 1791–Londres, 18 de octubre de 1871) fue un matemático y científico de la computación británico.1 Diseñó y desarrolló parcialmente una calculadora mecánica capaz de calcular tablas de funciones numéricas por el método de diferencias. También diseñó, pero nunca construyó, la analítica para ejecutar programas de tabulación o computación; por estos inventos se le considera como una de las primeras personas en concebir la idea de lo que hoy llamaríamos una computadora, por lo que se le considera como «El Padre de la Computación». En el Museo de Ciencias de Londres se exhiben partes de sus mecanismos inconclusos. Parte de su cerebro conservado en formol se exhibe en el Royal College of Surgeons of England. Sitio en Londres.2"

```

### 3. Héroes - Detalle (/heroes/id/profesion)

Deberá mostrar un texto que diga: *"Hola, mi nombre es NOMBRE y soy PROFESION"*. En donde el NOMBRE y PROFESIÓN deberán ser reemplazados por la información correspondiente. En caso de que el id no exista en nuestra información deberás mostrar un texto que diga: *"No tenemos en nuestra base ningún héroe ni heroína con ese id"*.

Ejemplo: **/heroes/2** mostrará la información de la heroína o del héroe con *id* 2.

#### 4. Héroes - Detalle con reseña (/heroes/id/resenia/tipo)

Los detalles de esta vista son los siguientes:

- A. Si pasamos un id y el mismo no se encuentra en la información el texto a mostrar será nuevamente: *"No tenemos en nuestra base ningún héroe ni heroína con ese id"*.
- B. Si se encontró el héroe con ese id y **NO** vino exactamente la palabra **completa** o simplemente **NO vino nada** en la ruta, mostrar el nombre del héroe y solo las primeras 30 palabras de su reseña.
- C. Si se encontró el héroe con ese id y vino exactamente la palabra **completa**. Mostrar el nombre del héroe y su reseña completa.

#### 5. Créditos (/creditos)

Mostrar los créditos de ustedes como equipo y cualquier texto adicional que deseen mostrar.

Para lograr lo anterior y en búsqueda de que el proceso de desarrollo llegue en tiempo y forma al final de la clase. [Les dejamos estos archivos](#) que les servirán como punto de partida para lograr el objetivo.

Dentro del archivo **app.js**, encontrarán algunos espacios en blanco que deberán completar para que la aplicación funcione a la perfección.

## Refactorizando su código (extra)

Toda persona que se estime buena para la programación siempre va por más. Es por eso que en este punto y ya teniendo **TODO** lo anterior hecho y funcionando a la perfección. Les sugerimos darle una "vueltitita de tuerca" a lo hecho y refactorizar su aplicación para que la misma tenga:

1. un sistema de ruteo modularizado.
2. un sistema de controladores que administren las peticiones del cliente.

Para ello, la estructura de directorios que recomendamos seguir será la siguiente:

```
├── data
│   └── heroes.json
├── controllers
│   ├── heroesController.js
│   └── mainController.js
├── routes
│   ├── heroes.js
│   └── main.js
├── app.js
└── package.json
```

## Tips

- **Refactorear no es borrar:** quizás estén tentadas/os a tomar lo que ya hicieron y comenzar a modularizar... **¡Pero!**, quizás no sea lo mejor. Es por eso que nuestra sugerencia es crear otra carpeta y hacer la refactorización de la aplicación allí. De esa manera podrán comparar las dos versiones y sacar sus propias conclusiones acerca de cuál les parece la mejor opción 🤖✨.

## Refactorizando código nivel DIOS (para el hogar)

Si llegaste hasta aquí, es porque seguramente te ha encantado todo este maravilloso mundo de JavaScript, Node y Express. Es por ello que ahora te pedimos que generés la aplicación con **express-generator** y veas cuales son las ventajas de usar dicho generador.

Seguramente al terminar, te darás cuenta que al hacerlo con este generador vas a encontrar similitudes entre tu refactorización y lo hecho con **express-generator**.