

## Desarrollo Web Full Stack Node

Ejercitación - M03C06

# Digital Heroes

### Introducción

Hasta el momento, nuestros "inversores" han quedado maravillados con tu trabajo. Pues se han dado cuenta que has podido llegar a destino tal como se espera. Y si bien son concientes que te ha costado un poco, saben que el viejo dicho "*la práctica te da sabiduría*" se ajusta como anillo al dedo en este escenario. Es por ello que ahora ponen en tu camino otro pequeño desafío del cual están más que seguros que vas a cumplir.

En esta oportunidad quieren hacer un sencillo 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 programacional.

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 infinidades de su coequipera/o. Adicionalmente te sugerimos trabajar con alguien nuevo con quien no hayas compartido mucho. Siempre es un buen momento para conocer gente. Así que adelante, levántate de tu silla y busca una nueva amiga o un nuevo amigo.
- **Leer TODO antes de codear:** una vez hayás encontrado tu dupla. Les sugerimos que lean en conjunto las consignas (hasta la página 5). 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/detalle/id - (id = número, obligatorio)
- /heroes/bio/id/ok - (id = número, obligatorio / ok = 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 Heroes de carne y hueso que encontrarás en este sitio. Esperamos que ellas y ellos te sirvan como inspiración para poder cumplir tus objetivos. Recuerda: ¡nunca pares de creer en ti!.*

### 2. Héroe (/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/detalle/id)

Deberá mostrar un texto que diga: *"Hola, mi nombre es NOMBRE y soy PROFESION"*. En donde el NOMBRE y PROFESION deberán ser reemplazados por la información real. Ejemplo: `/heroes/2` mostrará la información de la Heroína o del Héroe con *id* 2 . Controlar el escenario en donde se pase un número **n** que no haga match con el **id** de un héroe.

### 4. Héroes - Detalle con bio (/heroes/bio/id/ok)

Aquí se desea mostrar la información así:

- A. Si no se encontró el héroe con ese id. El texto será: *"No encontramos un héroe para mostrarte su biografía"*. Esto, indistintamente si se envió o no el parámetro **"ok"**.
- B. Si se encontró el héroe con ese id y **NO** vino exactamente la palabra **ok** o simplemente **NO vino nada** en la ruta. Mostrar el nombre del héroe y un texto adicional que diga: *"Lamento que no desees saber más de mi :("*.
- C. Si se encontró el héroe con ese id y vino exactamente la palabra **ok**.  
Mostrar el nombre del héroe y su reseña.

## 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.

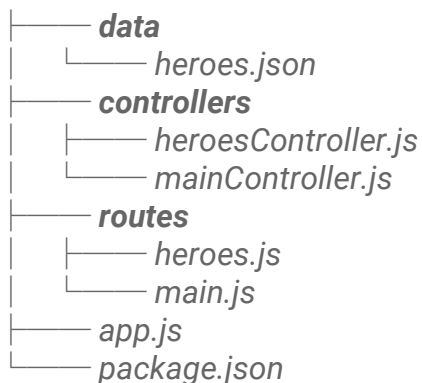
Dentro del archivo **app.js**, encontrarán algunos espacios en blanco que ustedes deberán completar para que la aplicación funcione a la perfección.

## Refactorizando su código

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



## 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 ello que nuestra sugerencia es crear otro folder y hacer en 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**.