

# Introducción a servidores: Héroes

## Práctica Integradora

### Introducción

En esta oportunidad, quieren hacer un sencillo sitio usando la tecnología de Express.

Este sitio busca mostrar de manera simple la información de las *Heroínas* y los *Héroes Tecnológicos* más representativos del ámbito de la programación.

Así que, sin más preámbulos, dejamos aquí las consignas para este desarrollo.

### Objetivo

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

- / -> Debe direccionar al recurso index.html.
- /babbage -> Debe direccionar al recurso babbage.html
- /berners-lee -> Debe direccionar al recurso berners-lee.html.
- /clarke -> Debe direccionar al recurso clarke.html.
- /hamilton -> Debe direccionar al recurso hamilton.html.
- /hopper -> Debe direccionar al recurso hopper.html.
- /lovelace -> Debe direccionar al recurso lovelace.html.
- /turing-> Debe direccionar al recurso turing.html.

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

# Consignas

## Micro desafíos - Paso 1

**Levantar un servidor web con Express que responda al puerto 3030.**

Como desarrolladores, este no es nuestro primer proyecto y debemos ser prolijos en la forma de ir guardando todos nuestros desarrollos. Armaremos una nueva carpeta para este proyecto, donde levantaremos un servidor web con Express.

## Micro desafíos - Paso 2

**Estructura de carpetas del proyecto.**

Para este desafío armaremos una estructura de carpetas simple, pero respetando los estándares que se utilizan para este tipo de sitios. Por eso, crearemos la siguiente estructura de archivos:



## Micro desafíos - Paso 3

### Ubicando recursos y sirviéndolos en nuestra aplicación.

1. Ya estamos en condiciones de ubicar nuestros archivos .html en su lugar. Son nuestras vistas. ¿Dónde los ubicamos?
2. Ahora, debemos volver a nuestro archivo app.js para introducir el código necesario para vincular las URLs con los recursos que acabamos de ubicar. Por ejemplo, debemos lograr que al introducir en el navegador "localhost:3030/home" se dé acceso, o se devuelva, al archivo index.html. Lo mismo deberá ocurrir con todos nuestros recursos conforme lo detallado en el "objetivo" de esta ejercitación.

**Nota:** Se recomienda el uso de 'path' para el armado de las rutas.

## Micro desafíos - Paso 4

### ¿Qué pasa con nuestros estilos y fotos?

La app ya funciona, pero sin los estilos CSS ni las fotos. A no preocuparse. ¡Vamos a solucionar esto!

1. Coloquemos nuestro archivo de estilos styles.css en su lugar. Para eso, habíamos creado una carpeta llamada "css" dentro de la carpeta "public".
2. También coloquemos las fotos de nuestros héroes, y el fondo de nuestro index (es decir, todos los archivos .jpg), en la carpeta "img".
3. ¡Listo! Pero sigue sin funcionar. Pues bien, solo falta aclarar en nuestro app.js la ubicación de nuestros archivos estáticos, con la línea:  
`app.use(express.static('public'));`

¡Felicitaciones! Levantamos un servidor web, vinculamos los recursos con las URLs y configuramos los archivos estáticos. Nada mal, ¿no? Con estos conocimientos ya estamos listos para adentrarnos al mundo del **HTML5** y **CSS** que nos espera en las próximas clases.

**¡Mucha suerte!**