



Curso de Back-End con Node.js (Inicial)

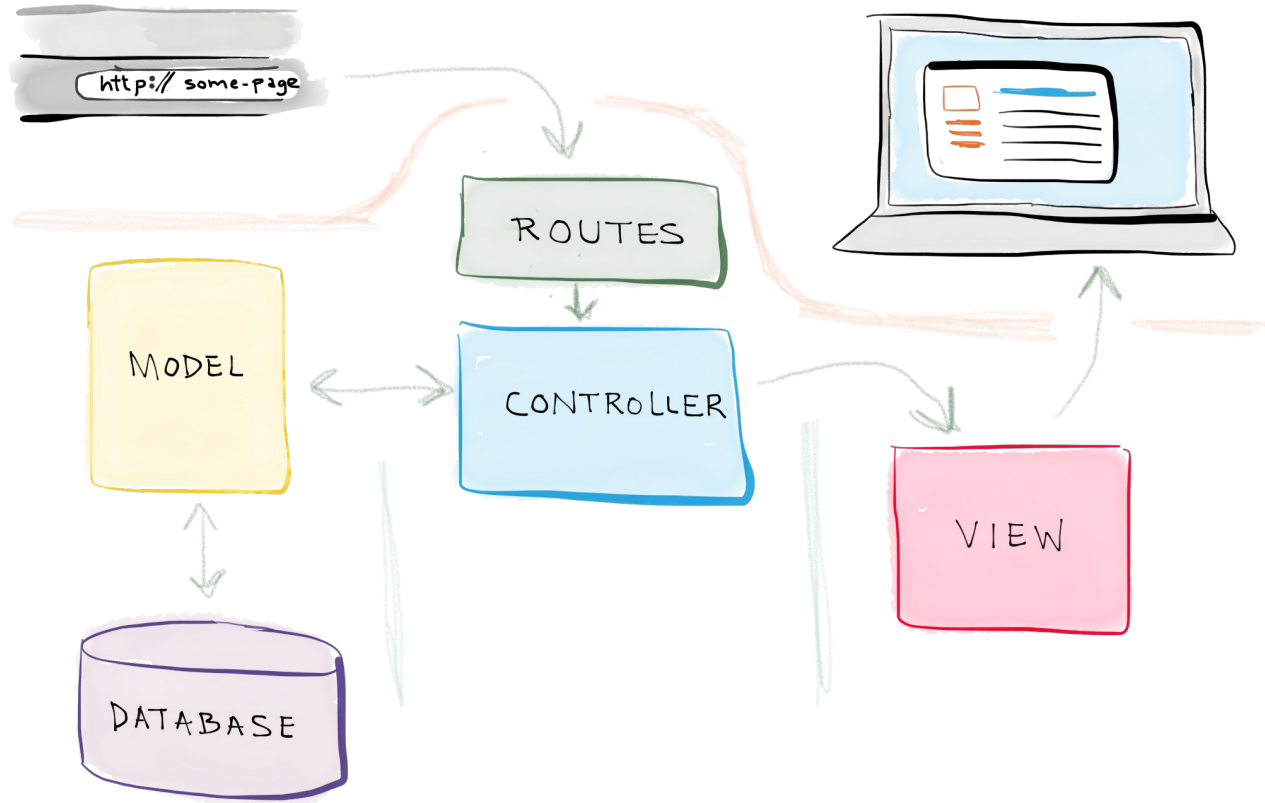
Proyecto Final



Antes... vale la pena repasar...

MVC

Resumen:





¿Qué hemos visto hasta ahora?

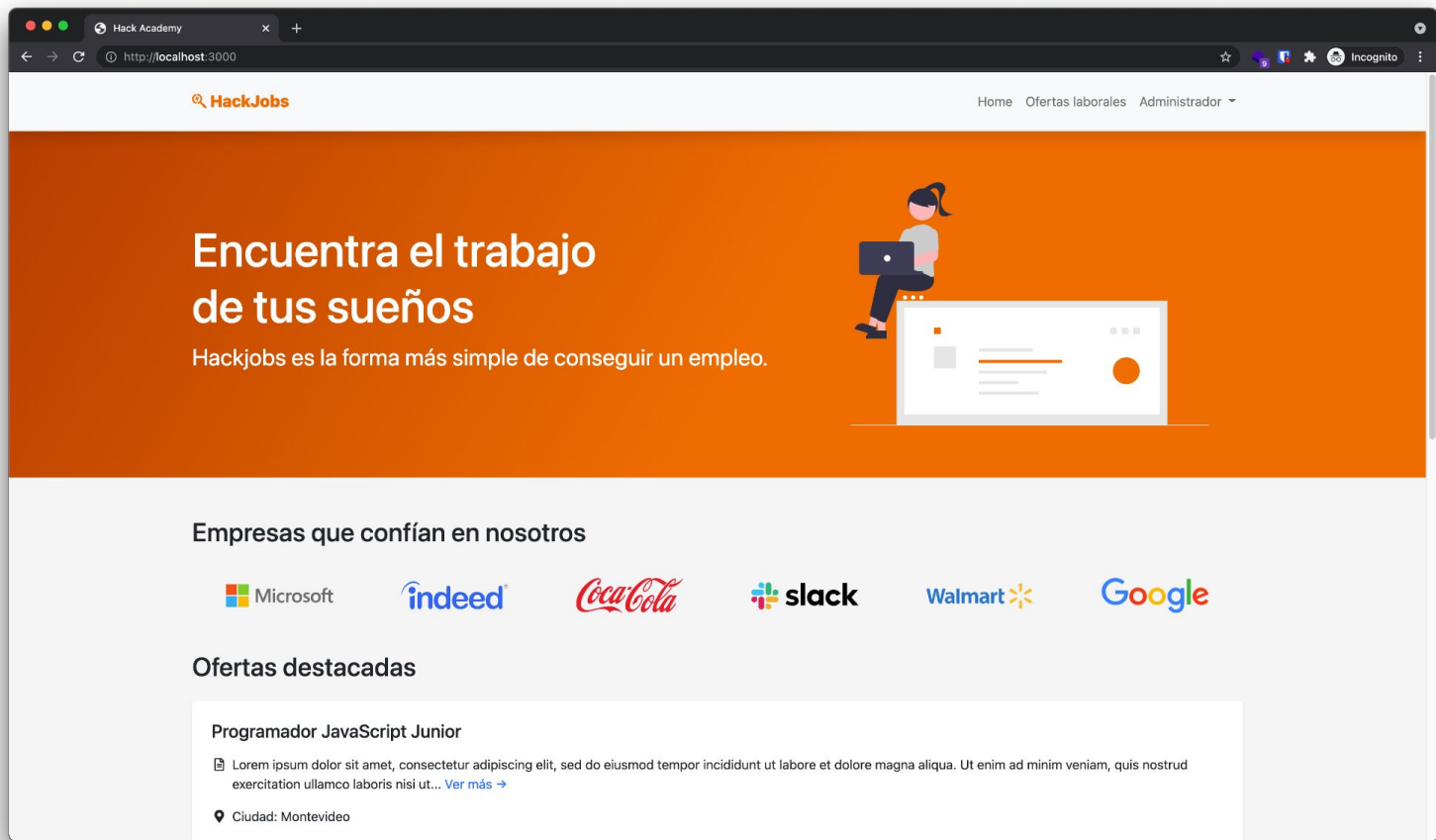
- ES6+
- Express.
- Rutas.
- Objetos `Request` y `Response`.
- Vistas (con Nunjucks).
- Controladores.
- Modelos.
- Variables de entorno (archivo `.env`).
- ORM (Sequelize).
- Formularios (con `POST` y `GET`).
- Validación de datos.
- Autenticación y Autorización.
- Middlewares.
- Control de Versiones – Git.
- File Uploads (Formidable).
- Deployment.

¿Qué es cada cosa?
¿Les gustaría repasar
algo en especial?



Proyecto Final

Sitio web de Ofertas Laborales





Hack Academy

http://localhost:3000/trabajos

Incognito

HackJobs

HomeOfertas laboralesAdministrador

Todas las ofertas laborales

Programador JavaScript Junior

📄

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut... [Ver más →](#)

📍

Ciudad: Montevideo

🏢

Empresa: ACME S.A.

Diseñador UI/UX

📄

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut... [Ver más →](#)

📍

Ciudad: Buenos Aires

🏢

Empresa: ACME S.A.

Experto en React.js

📄

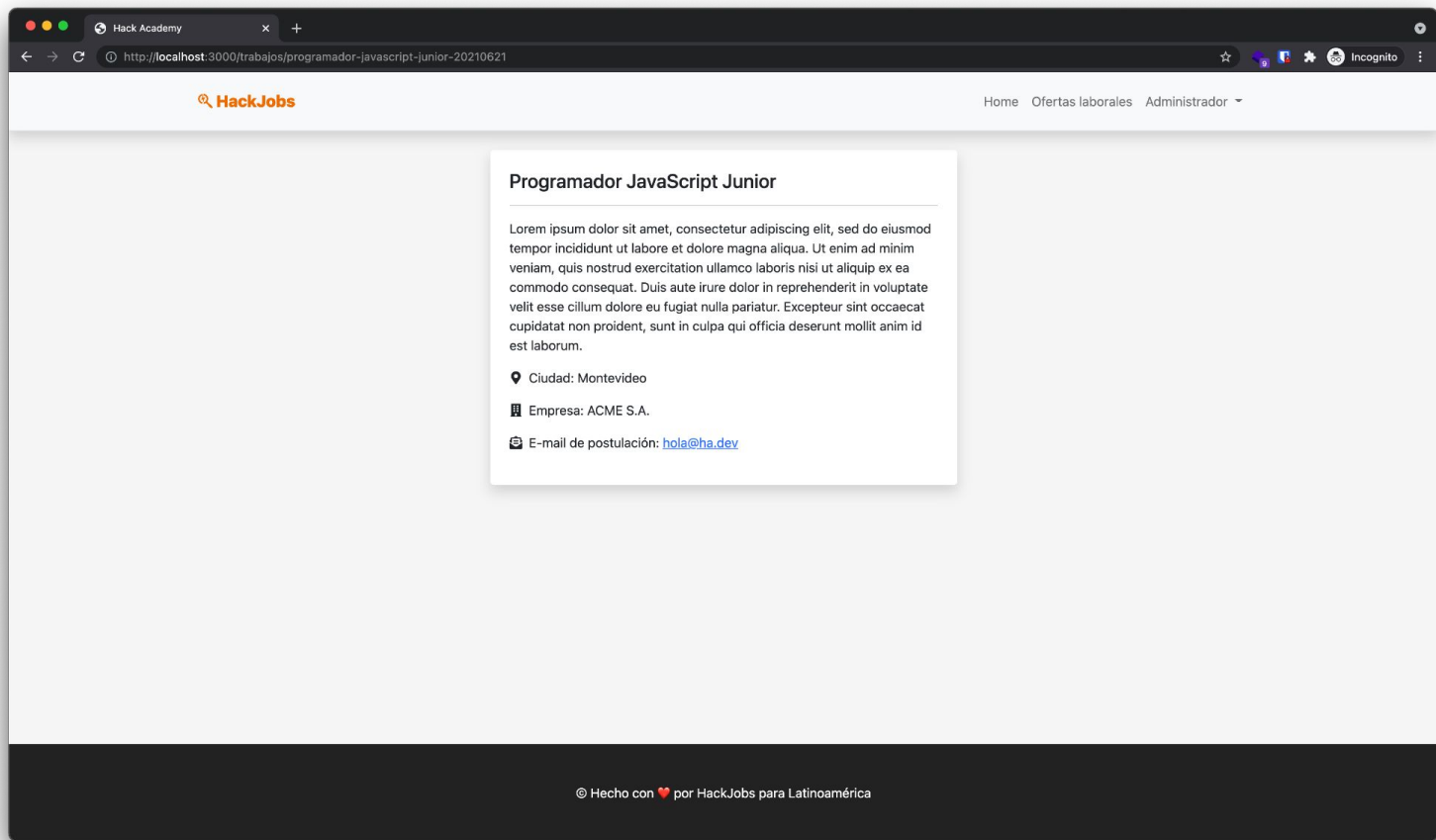
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut... [Ver más →](#)

📍

Ciudad: Bogotá

🏢

Empresa: ACME S.A.





Hack Academy

⌵ +

← → ↻

http://localhost:3000/admin/ofertas

☆

Incognito













⋮


HackJobs

Home Ofertas laborales Administrador

Admin – Ofertas laborales

Nuevo


#	Título	Destacada	Creado	Acciones
1	Programador JavaScript Junior	✓	2021-06-23	 
2	Diseñador UI/UX	✓	2021-06-23	 
3	Experto en React.js	✓	2021-06-23	 
4	Project Manager Senior	✗	2021-06-23	 
5	Data Scientist	✗	2021-06-23	 
6	Maquetador Senior	✓	2021-06-23	 

© Hecho con  por HackJobs para Latinoamérica



Hack Academy x +

← → ↻ http://localhost:3000/admin/ofertas/crear ☆ 🌐 ⚙️ Incognito

 HackJobs Home Ofertas laborales Administrador ▾

Crear oferta laboral

Título

Descripción

Empresa

Ciudad

Email para aplicar

Imagen

☐ Oferta destacada



Proyecto Final (1/2)

Durante el proyecto final se construirá un sitio de **ofertas laborales**.

Se les compartirá un un **proyecto base** específico para este ejercicio, el cual contiene:

- Estructura de carpetas necesaria para el proyecto.
- Vistas (archivos `.njk`).
- Paquetes de npm definidos en el archivo `package.json`.
- [Migraciones](#) para crear tablas en base de datos. 🤔 ¿Qué es esto?



Proyecto Final (2/2)

Lo primero que se deberá hacer al descargar el proyecto base es instalar las dependencias definidas en el archivo `package.json`:

```
npm install
```

Luego se deberá crear el archivo `.env` con las variables de entorno. Usar el archivo `.env.example` como referencia.



Modelos



Proyecto Final – Modelos

A priori, para este proyecto se deberán crear sólo dos modelos:

- Usuarios – Modelo: **User**.
- Trabajos (ofertas laborales) – Modelo: **Job**.
 - Id (BIGINT – Clave Primaria).
 - Título (STRING)
 - [Slug](#) (STRING) (se deberá generar automáticamente a partir del título).
 - Descripción (TEXT)
 - Empresa (STRING).
 - Ciudad (STRING).
 - E-mail para aplicar al trabajo (STRING).
 - Destacada (BOOLEAN).
 - Timestamps.



Rutas



Proyecto Final – Rutas (1/4)

Empezar creando las siguientes rutas:

URI (Ruta)	Método	Controlador	Función
/	GET	PagesController	showHome
/trabajos	GET	PagesController	showJobs
/trabajo/:slug	GET	PagesController	showJob
/login	GET	PagesController	showLogin
/register	GET	PagesController	showRegister

Crear el controlador `PagesController` y los métodos correspondientes que llamen a las vistas ya creadas.



Proyecto Final – Rutas (2/4)

Crear rutas de autenticación:

URI (Ruta)	Método	Controlador	Función
/register	POST	UserController	store
/login	POST	authController	login
/logout	GET	authController	logout

Crear los controladores `UserController` y `authController`.



Proyecto Final – Rutas (3/4)

Rutas del panel de administración para **usuarios**:

URI (Ruta)	Método	Controlador	Función
/admin/usuarios	GET	UserController	index
/admin/usuarios/crear	GET	UserController	create
/admin/usuarios/crear	POST	UserController	store
/admin/usuarios/editar/:id	GET	UserController	edit
/admin/usuarios/editar/:id	POST	UserController	update
/admin/usuarios/eliminar/:id	GET	UserController	destroy



Para acceder a todas estas rutas, es necesario que el usuario haya iniciado sesión.



Proyecto Final – Rutas (4/4)

Rutas del panel de administración para **ofertas laborales (trabajos)**:

URI (Ruta)	Método	Controlador	Función
/admin/ofertas	GET	jobsController	index
/admin/ofertas/crear	GET	jobsController	create
/admin/ofertas/crear	POST	jobsController	store
/admin/ofertas/editar/:id	GET	jobsController	edit
/admin/ofertas/editar/:id	POST	jobsController	update
/admin/ofertas/eliminar/:id	GET	jobsController	destroy



Para acceder a todas estas rutas, es necesario que el usuario haya iniciado sesión.



Validaciones



Proyecto Final – Validaciones

- Se deben validar los datos recibidos de todos los formularios. Es fundamental hacer las validaciones a nivel de Back-End por un tema de seguridad. Además, se pueden hacer validaciones a nivel de Front-End, para brindar una mejor experiencia de usuario (UX).
- Considerar que **Sequelize** ya dispone de un mecanismo de validaciones: <https://sequelize.org/master/manual/validations-and-constraints.html>.



Mejoras

Mejoras

- Crear un **buscador** de ofertas laborales.
- Crear un **filtro** de ofertas laborales. Por ejemplo: permitirle al navegante filtrar ofertas por ciudad o por empresa.

⚠ Existe un problema para poder crear el filtro anterior:

Actualmente, cada vez que se crea una nueva oferta laboral, el administrador debe escribir el nombre de la ciudad “a mano”. Esto implica que algunas ofertas podrían tener asignada la ciudad “Santiago” y otras “Santiago de Chile”, o “San Pablo” y “São Paulo”. Para resolver esta situación, se podría crear un modelo adicional llamado **City** y relacionar dicho modelo con el modelo **Job**. Esto se conoce como “relaciones” (o asociaciones) entre modelos.

Lo mismo sucede para las empresas. Un administrador podría escribir “ACME” y otro “ACME S.A.”. Para evitar esto se podría crear un modelo **Company** y las relaciones correspondientes.