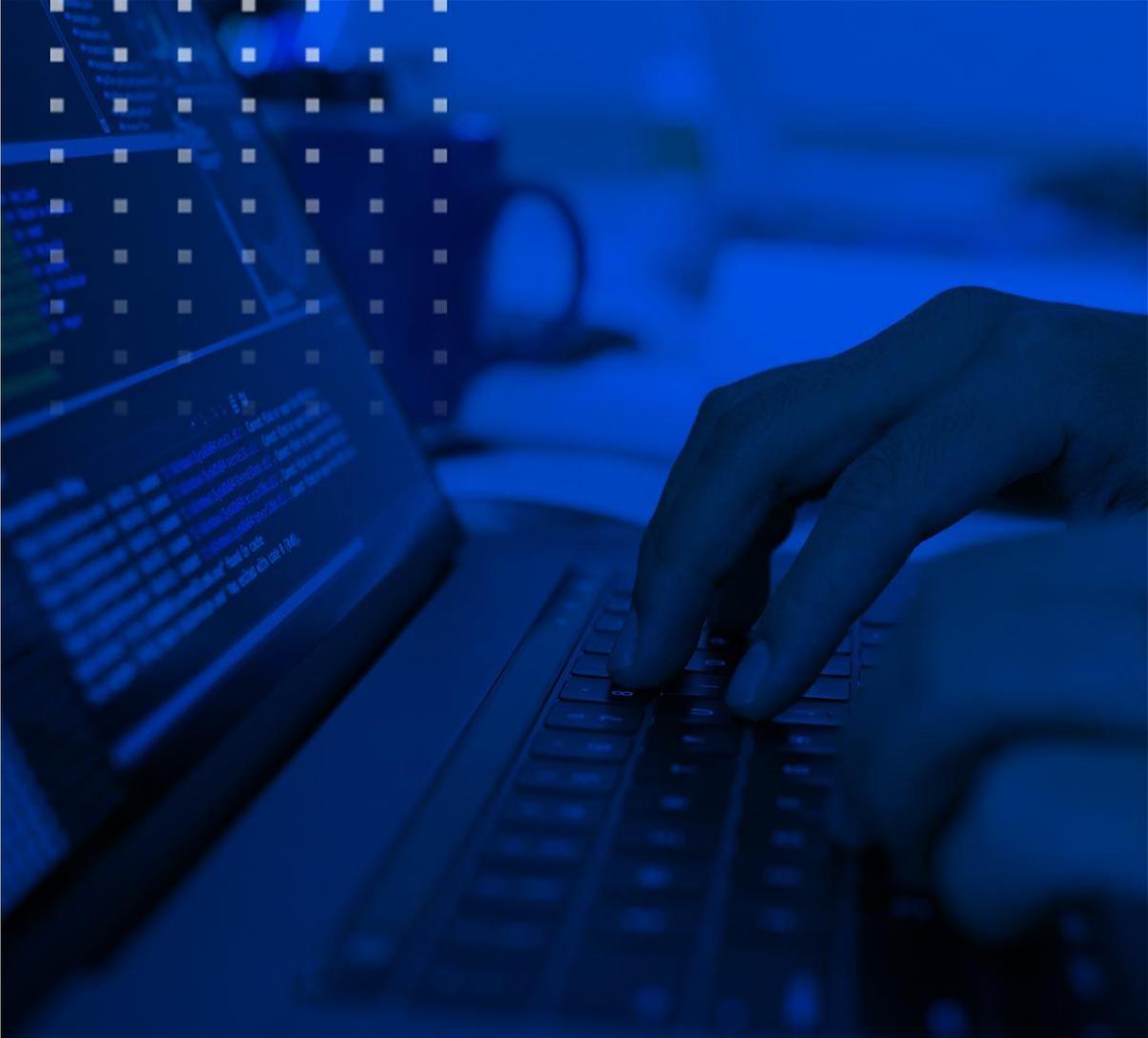


# CURSO

## FULL STACK DEVELOPER NIVEL INICIAL

UNIDAD 15  
Node JS: Formularios, sesiones y deploy



# Full Stack Developer Inicial

Node JS: Formularios, sesiones y deploy

# Cookies y sesiones

- HTTP es un protocolo sin estado. Cada petición es completamente independiente de otra y no hay forma de compartir información entre las peticiones.
- Las cookies nos permiten almacenar información en el navegador del usuario. Piensa en las cookies como un objeto de JavaScript.
- Una sesión Web es el tiempo que dura un visitante en nuestra aplicación antes de cerrar la pestaña, navegar a otra página, o estar inactivo por cierto tiempo.

# Sesiones

Una sesión inicia cuando el visitante ingresa por primera vez a un sitio o aplicación Web, o después de que ese mismo visitante ha finalizado una sesión previa.

Generalmente se define que la sesión termina después de 30 minutos de inactividad del visitante, pero dependiendo del sistema que utilicemos para almacenar la información de la sesión, es posible dejar abierta la sesión días o meses.

# Sesiones

- El objetivo de definir este concepto de sesión es poder almacenar información que sobrevive las peticiones HTTP de un visitante.
- La sesión involucra una cookie con una fecha de expiración, generalmente de 30 minutos.

# Sesiones

- Cada vez que el visitante hace una petición HTTP la cookie se actualiza para extender la sesión.
- La información de la sesión es un objeto de JavaScript que se puede almacenar en el cliente (dentro de una cookie) o en el servidor (en una base de datos o la memoria).

# express-session

Instalaremos el paquete express-session

- <https://www.npmjs.com/package/express-session>

# Middlewares

```
const session = require("express-session");
```

```
// MIDDLEWARES
```

```
app.use(express.json());
```

```
app.use(express.urlencoded({ extended: false }));
```

# Ejemplo formulario

```
app.use(session({  
  secret: '123456',  
  resave: true,  
  saveUninitialized: true  
}))
```

```
app.get("/", (req, res) => {  
  res.render("home");  
});
```

# Ejemplo formulario

```
app.post("/registro", (req, res) => {
  req.session.my_variable = req.body;
  res.redirect('/perfil')
});

app.get("/perfil", (req, res) => {
  const user = req.session.my_variable;
  delete req.session.my_variable;
  res.render("perfil", {
    user
  });
});
```

# ¡A practicar!

Creá un formulario, obtené los datos con get y recuperarlo por medio de sesiones.

# Despliegue - Deploy

# Configuraciones previas para despliegue

1. **Puerto:** Configurar una variable de entorno de puerto.

- Instalar paquete dotenv:

```
npm i dotenv
```

- Crear el archivo `.env` `PORT = 8081`
- Cargar variables de entorno en `app.js`

```
require('dotenv').config();
```

```
const port = process.env.PORT;
```

# package.json

## npm start

### 2. Especificar en los script el comando **START**

```
"scripts": {  
  "test": "echo \"Error: no test specified\" &&  
  exit 1",  
  "start": "node app.js"  
},
```

# gitignore

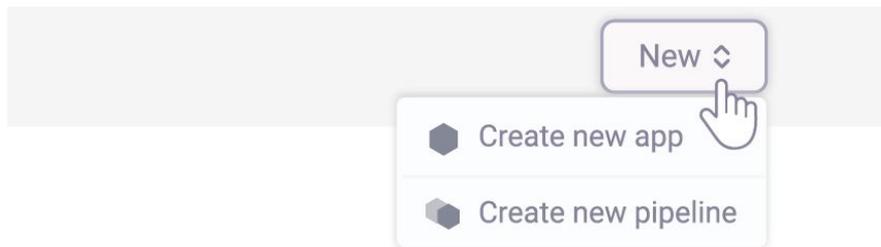
3. No dar seguimiento a los módulos de node.

Crear archivo **.gitignore**

node\_modules/

# Heroku

## 1. Crear aplicación



## 2. Dar nombre a la app

# Heroku

Create New App

App name

Choose a region

 United States

Add to pipeline...

Create app

# Heroku

Overview Resources Deploy Metrics Activity Access Settings

## Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

## Add this app to a stage in a pipeline to enable additional features



Pipelines let you connect multiple apps together and **promote code** between them.

[Learn more](#)



Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests.

[Learn more](#)

Choose a pipeline

## Deployment method



## Deploy using Heroku Git

Use `git` in the command line or a GUI tool to deploy this app.

## Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

# Heroku-cli

Heroku-cli

<https://devcenter.heroku.com/articles/heroku-cli>

# Heroku

Iniciar sesión en su cuenta de Heroku.

```
$ heroku login
```

Crea un nuevo repositorio de Git

```
$ cd my-project/
```

```
$ git init
```

```
$ heroku git:remote -a #####
```

Implementa tu aplicación

```
$ git add .
```

```
$ git commit -am "make it better"
```

```
$ git push heroku master
```

# Subir cambios

Por ejemplo,

```
$ git add .
```

```
$ git commit -am "make it better"
```

```
$ git push heroku master
```



**UTN.BA**

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES