



# Primer proyecto en Rails

Introducción a Rails (Parte I)

# ¿Qué aprenderemos en este módulo?

Lograremos adquirir las competencias teóricas y técnicas demandadas por la industria para el desarrollo de sistemas web con Rails.

Con el aprendizaje de este Stack, los y las estudiantes aplicarán los conocimientos adquiridos en **HTML, CSS, Bootstrap, Ruby y Bases de datos** para desarrollar aplicaciones web que contengan las capas de vista y backend, logrando conectar el cliente con el servidor para mostrar información dinámica.



***Aplicar procedimiento de creación de un proyecto nuevo en Ruby on Rails que incorpora las vistas y controladores utilizando la línea de comando, helpers y layouts para dar solución a un requerimiento.***

- Unidad 1:  
Primer proyecto en Rails
- Unidad 2:  
Acelerar la construcción de un proyecto
- Unidad 3:  
Almacenar información en una base de datos
- Unidad 4:  
Despliegue de una aplicación en internet utilizando Heroku.
- Unidad 5:  
Disponibilizar un servicio web utilizando la arquitectura REST



Te encuentras aquí



## ¿Qué aprenderás en esta sesión?

- *Aplica procedimiento de creación de un proyecto nuevo Ruby on Rails utilizando la línea de comando.*
- *Aplica procedimiento de levantamiento de un servidor de Ruby on Rails utilizando la línea de comando.*

¿Quién conoce acerca  
del perfil de  
desarrollador Full Stack?



# Perfil desarrollador Full Stack

El perfil Full Stack es un perfil híbrido capaz de crear y administrar sistemas web abordando las tecnologías del Front End (Cliente) y el Backend (Servidor). Este perfil domina un conjunto de tecnologías que comprenden:

1. HTML, CSS y JavaScript
2. Lenguaje de Programación Backend (**Ruby**)
3. Base de datos relacionales (**PostgreSQL**) o no relacionales (MongoDB)
4. Framework backend para el desarrollo de aplicaciones web (**Ruby on Rails**)
5. Entre otras tecnologías.

Estas son las tecnologías base al stack que aprenderemos a continuación.



# ***/\* Instalación de Rails \*/***

# Demostración "Instalando Rails"





# Sigamos los pasos

- **Paso 1:** Ingresar al siguiente [gist](#) para obtener la documentación.
- **Paso 2:** Comprobar la versión de rbenv.
- **Paso 3:** Del gist seleccionamos el ítem 4 “Instalando Rails”
- **Paso 4:** Ejecutamos el comando en terminal o consola de `sudo gem install rails`.
- **Paso 5:** Comprobamos la instalación con `rails - -version`.
- **Paso 6:** Ejecutamos el comando `rbenv rehash`.

**/\* Crear un proyecto Rails \*/**

# Demostración

## "Creando nuestro primer proyecto en Rails"



# Sigamos los pasos

- **Paso 1:** En terminal ejecutar el comando `rails new nombreProyecto`.
- **Paso 2:** Acceder desde la terminal al directorio del proyecto creado.
- **Paso 3:** Abrir el proyecto en Visual Studio code con el comando `code ..`



# rails new

- **rails new nombreProyecto**, es el comando que utilizamos cada vez que vayamos a crear un nuevo proyecto de Rails.
- En **nombreProyecto**, deberás indicar el nombre que quieras asignar a dicho proyecto.
- Este comando descarga una gran variedad de dependencias necesarias para el funcionamiento de rails.
- Inicia un repositorio **git** automáticamente.



# Demostración "Levantando el servidor en Rails"



# Sigamos los pasos

- **Paso 4:** Continuaremos trabajando en el proyecto del ejercicio anterior.
- **Paso 5:** En la terminal escribir el comando rails server para levantar el servidor.
- **Paso 6:** Acceder a la ruta `localhost:3000` en el navegador web.



# rails server

- Comando para levantar un servidor local.
- La manera resumida de este comando es **rails s**.
- Rails asigna por defecto el puerto 3000.
  - localhost:3000





# Detener el servidor

- Para detener el servidor debemos ejecutar en el teclado la combinación de teclas **ctrl + c**
- Esta combinación se realiza en la terminal o consola.
- Al ejecutar el servidor se detendrá.



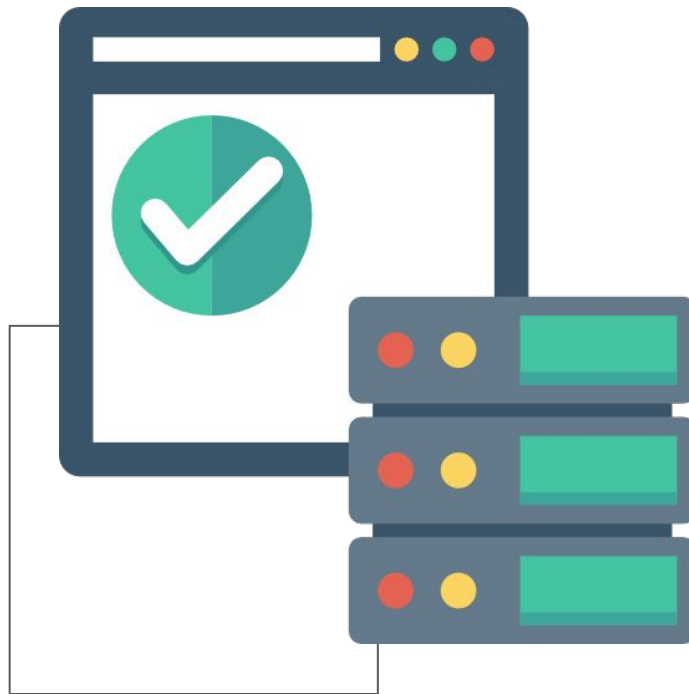
# ¿Qué es el servidor?

- Es un computador en específico al cual nos conectamos para ver una página web o descargar un archivo.
- También puede ser un programa que maneja una **petición** y entrega una **respuesta**.



# Los servidores web

- Son un tipo especial de servidores que resuelven peticiones web.
- Una petición consiste en entregar una dirección.
- El servidor a partir de esa dirección busca un archivo como una página web y devuelve el archivo HTML al cliente y este lo muestra.



***/\* Leyendo el output del servidor \*/***

# Demostración "Leyendo el output del servidor"



# Sigamos los pasos

- **Paso 7:** Levantemos el servidor y analicemos la información que retorna la terminal/consola.
- **Paso 8:** Acceder a la ruta dispuesta `localhost:3000`.
- **Paso 9:** Analizar la información en terminal/consola al acceder a la ruta.
- **Paso 10:** En el navegador acceder a la ruta `localhost:3000/home`.
- **Paso 11:** Analizar la información de la terminal/consola.



# ¿Qué es un servidor web?





## Próxima sesión...

- *Aplica procedimiento de creación de controllers y vistas utilizando generadores en Ruby on Rails para dar solución a un requerimiento.*



**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

