



Universidad Tecnológica de  
**TULA-TEPEJI**



## PORTADA

**Universidad Tecnológica de Tula-Tepeji (UTTT)**

**Tecnologías de la información Área Desarrollo de Software Multiplataforma**

**Edel Meza Hernández**

**Cristian Andres Cid Rodriguez**

**Brisa Adriana Mendoza Viguera**

**5TIDSM-G1**

**Aplicaciones Web para 14.0**

**Tema: Manual sobre Ruby on Rails**

**5/02/2024**



## Manual de Instrucciones para Desarrolladores de Ruby on Rails

### 1. Entorno y Configuración:

- **Objetivo:** Configurar el entorno de desarrollo de Ruby on Rails y personalizar la configuración de la aplicación.
- **Pasos a Seguir:**
  1. Instala Ruby y Ruby on Rails en tu sistema operativo.
  2. Crea un nuevo proyecto Rails utilizando el comando **rails new nombre\_de\_tu\_proyecto**.
  3. Explora y modifica el archivo **config/application.rb** para establecer variables globales y configuraciones generales.
  4. Personaliza la configuración específica para cada entorno (desarrollo, prueba, producción) en los archivos dentro del directorio **config/environments**.
  5. Utiliza el archivo **config/secrets.yml** para almacenar datos sensibles de forma segura, como claves API y tokens de acceso.

### 2. Hosting:

- **Objetivo:** Desplegar una aplicación Ruby on Rails en un servidor de alojamiento web.
- **Pasos a Seguir:**
  1. Elige un proveedor de alojamiento web que admita Ruby on Rails, como Heroku, AWS o DigitalOcean.
  2. Configura una cuenta en el proveedor de alojamiento y crea un nuevo servidor o instancia para tu aplicación.
  3. Utiliza herramientas como Git o FTP para cargar tu código fuente en el servidor.
  4. Configura la base de datos y el servidor web según las necesidades de tu aplicación.
  5. Apunta tu dominio a la dirección IP del servidor y verifica que la aplicación esté funcionando correctamente en línea.

### 3. Envío de Correo:

- **Objetivo:** Integrar la funcionalidad de envío de correos electrónicos en una aplicación Rails.
- **Pasos a Seguir:**
  1. Crea un nuevo mailer en tu aplicación utilizando el generador de Rails: **rails generate mailer NombreDelMailer**.
  2. Define métodos dentro del mailer para representar diferentes tipos de correos electrónicos, como confirmaciones de cuenta o notificaciones de usuario.
  3. Utiliza plantillas y vistas HTML para personalizar el contenido de los correos electrónicos.



4. Configura los ajustes de correo saliente en el archivo **config/environments/production.rb**.
5. Desde tus controladores o modelos, llama a los métodos del mailer para enviar correos electrónicos cuando sea necesario.

#### 4. Crear PDF:

- **Objetivo:** Generar archivos PDF dinámicamente desde una aplicación Rails.
- **Pasos a Seguir:**
  1. Agrega la gema Prawn o Wicked PDF a tu archivo **Gemfile** y ejecuta **bundle install** para instalarlas.
  2. Crea una nueva acción en uno de tus controladores para generar el PDF.
  3. Utiliza la gema seleccionada para construir el contenido del PDF utilizando Ruby y HTML.
  4. Configura una ruta en **config/routes.rb** para que los usuarios puedan acceder al PDF generado.
  5. Utiliza herramientas de estilo y formato para mejorar la apariencia del PDF, como CSS y la API de estilo de Prawn.

#### 5. Descargar Archivos del Servidor:

- **Objetivo:** Permitir a los usuarios descargar archivos desde la aplicación Rails.
- **Pasos a Seguir:**
  1. Almacena los archivos que deseas que los usuarios descarguen en un directorio accesible desde la aplicación, como **public**.
  2. Crea una nueva acción en uno de tus controladores para manejar las solicitudes de descarga.
  3. Utiliza el método **send\_file** para enviar el archivo solicitado como una respuesta HTTP al navegador del usuario.
  4. Agrega lógica de autenticación y autorización si es necesario para restringir el acceso a los archivos.

#### 6. Encriptación y Desencriptación:

- **Objetivo:** Proteger datos sensibles utilizando técnicas de encriptación en una aplicación Rails.
- **Pasos a Seguir:**
  1. Utiliza la gema **bcrypt** para almacenar contraseñas de usuario de forma segura en la base de datos.
  2. Configura la encriptación de datos sensibles utilizando claves de cifrado simétrico o asimétrico según sea necesario.



3. Implementa la descriptación de datos cuando sea necesario utilizando las mismas claves y algoritmos utilizados para la encriptación.
4. Mantén las claves de cifrado seguras y protegidas utilizando técnicas como el almacenamiento seguro y la rotación regular de claves.

#### 7. Creación de Menús y Submenús Dinámicamente:

- **Objetivo:** Generar menús de navegación personalizables en una aplicación Rails.
- **Pasos a Seguir:**
  1. Define un modelo de datos para representar la estructura del menú, incluyendo elementos como nombre, URL y permisos asociados.
  2. Utiliza lógica Ruby en tus vistas para iterar sobre los datos del menú y generar HTML dinámico.
  3. Implementa la lógica de autenticación y autorización para mostrar u ocultar elementos del menú según los roles y privilegios del usuario.
  4. Utiliza herramientas como Bootstrap o CSS para mejorar la apariencia y la usabilidad del menú de navegación.

#### 8. Almacenar Imágenes:

- **Objetivo:** Administrar el almacenamiento y la visualización de imágenes en una aplicación Rails.
- **Pasos a Seguir:**
  1. Utiliza gemas como CarrierWave o Active Storage para adjuntar imágenes a modelos de datos en tu aplicación.
  2. Configura el almacenamiento de archivos en el sistema de archivos local o en un servicio de almacenamiento en la nube como Amazon S3.
  3. Utiliza helpers de vista proporcionados por las gemas para mostrar las imágenes almacenadas en tus vistas.
  4. Implementa la carga y descarga de imágenes utilizando formularios HTML y controladores personalizados en tu aplicación.

#### 9. Debug:

- **Objetivo:** Identificar y corregir errores de código en una aplicación Rails.
- **Pasos a Seguir:**
  1. Utiliza herramientas de depuración integradas en Rails, como **byebug**, para detener la ejecución del código en puntos específicos.
  2. Inspecciona el estado de la aplicación, incluidas variables, parámetros y trazas de pila, para identificar la causa del error.



3. Utiliza herramientas de registro y monitoreo para capturar