

Checkpoint Test RoR

Objetivo:

El assessment tiene como objetivo evaluar las habilidades técnicas, capacidades de resolución de problemas, y competencias generales de los coders obtenidas por cada uno de ellos durante su entrenamiento en Riwi. Este assessment se realizará en un entorno controlado, permitiendo el uso herramientas de IA e internet de manera responsable y ética.

Se permitirá el uso de documentación oficial, ejemplos de código abierto y herramientas de IA como apoyo, siguiendo estrictamente los siguientes lineamientos:

Buen Uso de la IA

- **Asistencia en Código:** La IA se podrá utilizar para obtener sugerencias y correcciones en el código, pero no para resolver tareas completas de manera autónoma.
- **Aprendizaje y Recursos:** Los coders podrán utilizar herramientas de IA para acceder a materiales de aprendizaje y documentación relevante.
- **Optimización y Mejora:** Se permite el uso de la IA para optimizar y mejorar el código ya desarrollado por el coder, siempre y cuando se mantenga la integridad del proceso de aprendizaje.

Alcance de la IA

- **Código y Desarrollo:** La IA puede asistir en la escritura, depuración y optimización del código.
- **Investigación:** Los coders podrán utilizar la IA para buscar información y resolver dudas técnicas, puntuales y acotadas.



- **Documentación:** La IA puede ayudar a generar documentación y comentarios en el código.

Límites de la IA

- **Prohibido el Plagio:** No se permite la generación de soluciones completas o la copia de código de fuentes externas sin la debida atribución.
- **Autonomía del Coder:** La IA no debe sustituir la capacidad del coder para resolver problemas; debe actuar como una herramienta de apoyo.
- **Evaluación de Habilidades:** La IA no puede ser utilizada para responder a preguntas de evaluación de manera directa, ya que el objetivo es evaluar las competencias del coder.

Buen Uso de Internet

- **Investigación y Recursos:** Se permite el uso de internet para buscar información, acceder a documentación y utilizar recursos educativos.
- **Comunicación:** Los coders pueden utilizar internet para comunicarse con el equipo de evaluación o pedir asistencia técnica si es necesario.
- **Desarrollo Colaborativo:** El uso de plataformas de desarrollo colaborativo en línea (como GitHub) está permitido para la gestión de proyectos.
- **Alcance de Internet**
- **Acceso a Documentación:** Los coders podrán consultar documentación oficial y recursos educativos en línea.
- **Resolución de Problemas:** Se permite el uso de foros y comunidades técnicas para la resolución de dudas específicas y acotadas siempre que se referencie.
- **Actualizaciones y Herramientas:** Los coders pueden descargar e instalar herramientas y actualizaciones necesarias para el desarrollo del assessment.



Límites de Internet

- **Fuentes Confiables:** Se debe verificar la fiabilidad de las fuentes utilizadas. No se permite el uso de contenido no autorizado o pirata.
- **Prohibido el Fraude:** No se permite buscar o utilizar soluciones completas o respuestas directas a los problemas planteados en el assessment.
- **Seguridad y Privacidad:** Se debe garantizar la seguridad y privacidad de la información personal y de los datos del assessment. No se permite compartir información confidencial.
- **Monitoreo y Cumplimiento**
- **Supervisión:** El uso de IA e internet será monitoreado por el TL para asegurar el cumplimiento de los lineamientos.
- **Consecuencias:** Cualquier violación a estos lineamientos resultará en una evaluación adicional y posibles sanciones, incluyendo la descalificación del assessment.

Ética y Responsabilidad

- **Uso Ético:** Se espera que todos los coders usen la IA e internet de manera ética y responsable.
- **Responsabilidad Individual:** Cada coder es responsable de su propio trabajo y del cumplimiento de estos lineamientos.

Introducción:

Este assessment tiene como propósito que los coders demuestren su capacidad para desarrollar una aplicación funcional utilizando Ruby on Rails, aplicando prácticas modernas como el uso de Active Storage para la gestión de archivos y Tailwind CSS para el diseño. Además, se evaluará su habilidad para implementar validaciones robustas, internacionalización (i18n) para soportar múltiples idiomas, y el manejo de servicios en la nube como Cloudinary o AWS S3.



La aplicación debe reflejar un alto nivel de calidad técnica, alineándose con los requerimientos detallados, y debe incluir funcionalidades esenciales para la experiencia del usuario, como mensajes de éxito y error bien diseñados y traducidos. El proyecto pondrá a prueba tanto los conocimientos adquiridos durante el entrenamiento en RIWI como la capacidad para seguir lineamientos éticos y utilizar herramientas modernas de manera responsable.

Requerimientos:

Crea una aplicación Rails que permita subir imágenes al servicio en la nube Cloudinary o AWS S3 a través de un formulario sencillo, utilizando Tailwind CSS para darle estilo y aplicando validaciones para garantizar que solo se puedan subir archivos de imagen. Además, la aplicación debe ser completamente internacionalizada, soportando español e inglés, y debe mostrar mensajes de éxito o error según el resultado de la operación

1. Formulario para Subir Imágenes:

- Crea un formulario que permita a los usuarios seleccionar un archivo desde su dispositivo y enviarlo al servidor.
- Estiliza el formulario utilizando Tailwind CSS.
- Asegurate de establecer los campos tipo texto dentro del formulario

1. Subida a Cloudinary:

- Configura la app para que mediante Active Storage se permita subir las imagenes directamnete a Cloudinary.

2. Validaciones:

- Asegúrate de que solo se puedan subir archivos de tipo imagen (jpeg, png, etc.).
- Muestra un mensaje de error si se intenta subir un archivo inválido.
- Limita el peso maximo por imagen de 3MB
- Agrega validaciones de maximum para los campos tipo texto que uses en el formulario

3. Internacionalización:



- a. Traduce todos los textos visibles estaticos (etiquetas del formulario, botones, y mensajes) a español e inglés utilizando el sistema de internacionalización de Rails (I18n).
 - b. Asegúrate de que los mensajes de error y éxito también estén traducidos.
4. **Mostrar Resultado:**
- a. Después de subir la imagen, muestra la vista show que corresponda a donde queda la imagen almacenada
5. **Vistas:**
- a. La página principal debe mostrar el index de los formularios.
 - b. Debe existir una vista de creación de nuevo formulario
 - c. Debe existir la vista show del formulario donde se pueda visualizar la imagen adjunta
 - d. Debe existir un partial que contenga el navbar este debe contener la opción de seleccionar el idioma de la aplicación entre inglés y espa
6. **Tablas:**
- a. Única tabla en base de datos que nos permita trabajar con el formulario debe tener los siguientes campos mínimamente y tengan cuidado con las palabras reservadas en Rails como “form” usen otra opción para el nombramiento de ese modelo.
 - b. Campos minimos requeridos: title:string description:text
 - c. El campo de Active storage pueden nombrarlo como prefieran

Reglas Adicionales:

- Usa dotenv para manejar las claves de Cloudinary de manera segura.
- El diseño debe ser limpio y funcional, usando las clases de Tailwind CSS para lograr una buena experiencia visual.
- Implementa mensajes flash para mostrar los resultados de la subida (éxito o error).



Formulario test:

- En el siguiente enlace encontraras la ultima parte de la prueba un formulario donde podras encontrar las preguntas teoricas de seleccion multiple con unica respuesta

<https://forms.office.com/pages/responsepage.aspx?id=64fRSfXqCkOwnmocLu7ElozGQ05VAXtOrwLOW4VrA-1UN0tDQlhFVktUVEk2WVQ3WVBETzNYSUxaVi4u&origin=lprLink&route=shorturl>