

Checkpoint Test RoR

Objetivo:

Este checkpoint tiene como objetivo evaluar las habilidades técnicas adquiridas hasta el momento en la tecnología Ruby on Rails, 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:

Has sido contratado para desarrollar un proyecto de red social para escritores, donde los usuarios podrán compartir sus escritos, comentar en las publicaciones de otros y construir una comunidad de lectores y escritores. La plataforma está diseñada para ser un espacio exclusivo para autores, poetas y aficionados a la escritura que quieran conectar, recibir comentarios y crecer en su ámbito. La empresa que te contrata desea una plataforma intuitiva, que facilite la interacción entre los usuarios y fomente la participación en la comunidad.

Problemas que se buscan resolver:

1. **Dificultad para encontrar contenido de calidad y específico en temas de escritura:**
 - a. En otras plataformas, los escritores encuentran difícil destacar su trabajo debido a la gran cantidad de publicaciones de diverso contenido. Los usuarios necesitan una forma de navegar exclusivamente entre contenido de escritores.
2. **Falta de interacción significativa entre escritores y lectores:**
 - a. Los escritores a menudo buscan retroalimentación detallada para mejorar su trabajo, pero las herramientas actuales no favorecen un sistema de comentarios específicos que facilite un intercambio de ideas enriquecedor.
3. **Falta de gestión de contenido propio y organizado:**

Actualmente, los escritores no tienen una forma organizada de revisar sus publicaciones previas, ediciones y los comentarios recibidos. Esto dificulta la gestión y mejora continua de sus escritos.
4. **Ausencia de roles diferenciados para escritores y administradores:**

La plataforma necesita definir roles claros que faciliten una mejor organización y moderación. Sin un sistema de permisos adecuado, no hay control sobre quién puede publicar, moderar o administrar la comunidad.



Requerimientos de la red social:

La plataforma debe resolver estos problemas y mejorar la experiencia de los escritores y lectores. Los usuarios tendrán funcionalidades específicas según su rol, y deben poder interactuar de manera efectiva con el contenido de la plataforma. Para lograr esto, el sistema debe:

- **Permitir que diferentes usuarios accedan al sistema de forma independiente,** Habrá dos roles principales: escritor (usuarios que pueden publicar contenido y comentar) y administrador (usuarios con permisos para moderar y gestionar contenido).
- **Facilitar la creación y administración de publicaciones y comentarios:** Los escritores podrán crear publicaciones, actualizar sus escritos, y recibir comentarios. Los usuarios podrán ver las publicaciones y agregar comentarios.
- **Permitir el uso de etiquetas y categorías para organizar el contenido de escritura:** Los escritores podrán etiquetar sus publicaciones para mejorar la visibilidad y facilitar que los lectores encuentren contenido relevante.
- **Implementar funciones de categorización:** Los usuarios podrán buscar publicaciones por etiquetas, categorías, o palabras clave, ayudando a los lectores a encontrar rápidamente el contenido que desean leer.
- **Aplicar scopes y callbacks para garantizar la organización y calidad del contenido:**
 - Deben incluirse scopes para filtrar publicaciones por fecha o categoría. Los callbacks controlarán que solo los escritores puedan publicar y que los administradores aprueben el contenido.
- **Implementar pruebas de integración con una cobertura mínima del 20%:**
 - Las pruebas deben evaluar las funcionalidades de creación y administración de publicaciones.



Requerimientos específicos:

- Las pruebas deben evaluar las funcionalidades deben estar enfocadas en al menos uno de los modelos.
- Login y roles de usuarios:
La aplicación debe tener una página de inicio de sesión con autenticación, y los usuarios deben poder ver solo las funcionalidades permitidas por su rol.
- Visualización de contenido y categorías:
Implementar una vista principal de publicaciones, donde los usuarios puedan ver el contenido por categorías .
- Funciones de callback y validaciones para control de contenido:
Aplicar callbacks y validaciones para todos los campos que sean ingresados por los users restringiendo tipo y cantidad de caracteres para restringir la publicación de contenido y controlar que solo los escritores registrados publiquen o comenten.
- Interfaz sencilla para funcionalidades:
Las vistas no necesitan diseño avanzado, solo deben cumplir con la funcionalidad especificada para que los usuarios interactúen con la plataforma.
- Uso de una base de datos relacional (sin SQLite):
La aplicación debe utilizar PostgreSQL o MySQL, para la gestión de los usuarios, publicaciones , categorías y comentarios de forma escalable.

Roles y permisos:

Administrador:

- Puede ver, editar y eliminar cualquier publicación o comentario.
- Puede cambiar el status de las publicaciones.
- Puede crear, ver, editar y eliminar las categorías.



Escritor:

Puede crear publicaciones y comentar en publicaciones de otros usuarios.
No puede editar o eliminar publicaciones de otros.

Este proyecto no requiere un diseño de interfaz específico, pero debe cumplir con las funcionalidades mencionadas. Al finalizar, asegúrate de implementar una cobertura de pruebas adecuada para las funcionalidades clave de la plataforma y hacer uso de scopes y callbacks en los modelos según sea necesario.

Criterios de aceptacion :

Para considerar esta prueba como completada y aceptable, el proyecto debe cumplir los siguientes criterios:

- **Roles y Permisos:**
Se deben implementar roles de Escritor y Administrador con permisos específicos según lo indicado (crear, ver, editar, eliminar publicaciones y comentarios).
- Los escritores deben poder ver y comentar publicaciones, pero sin permisos de edición o eliminación en publicaciones de otros usuarios.
- Los administradores deben poder aprobar o cambiar el estado de las publicaciones.
- **Autenticación y Autorización:** Los usuarios deben poder iniciar sesión y ver solo las funcionalidades asignadas a su rol.
- La aplicación debe mostrar una interfaz de inicio de sesión y restringir el acceso de acuerdo con el rol asignado.
- **Gestión de Contenido y Publicaciones:** Los escritores pueden crear y administrar sus propias publicaciones.



Se deben utilizar callbacks y validaciones para asegurar que solo los escritores pueden publicar y comentar, y para validar los campos antes de guardarlos en la base de datos.

- Organización y Navegación de Contenido: Implementación de categorías para mejorar la visibilidad.
- La vista principal debe mostrar las publicaciones, con opciones para filtrar u observar por categoría y etiquetas.
- Pruebas e Integración: Implementar pruebas unitarias con una cobertura mínima del 20%
- Incluir scopes para filtros de publicaciones por fecha (ejemplo creados el ultimo mes) para categoría, y validaciones adecuadas.
- Base de Datos: Utilizar PostgreSQL o MySQL para la gestión de usuarios, publicaciones, categorías y comentarios.
- Recuerden las validaciones obligatorias que siempre se agregan con el fin de restringir ciertos comportamientos indebidos de los usuarios finales

Listado de entregable:

- Repositorio de Código: Un repositorio en Git (preferiblemente GitHub) con el código del proyecto completo,
- Archivo Readme detallado con instrucciones de configuración,
- Valores de variables de entorno en archivo o archivos .env para poder probar en otros entornos

Recursos oficiales:

- Repos Enlace: <https://guides.rubyonrails.org/>



Documentación oficial que cubre todos los aspectos de Ruby on Rails, desde la instalación hasta características avanzadas, ideal para aprender y mejorar habilidades en el framework.

- Listado de Gemas
Enlace: <https://rubygems.org/>
Repositorio principal para buscar, instalar y publicar gemas de Ruby, con estadísticas de descargas y documentación para cada gema.
- Ruby Toolbox
Enlace: <https://www.ruby-toolbox.com/>
Directorio que categoriza gemas y proyectos de Ruby según su propósito, facilitando la búsqueda y comparación de gemas populares para tareas específicas.