

Assessment de Empleabilidad

Reglas del Assessment:

- **Comunicación:** Está prohibido hablar o comunicarse de cualquier forma con otros estudiantes durante el examen.
- **Integridad Académica:** Cualquier forma de trampa, incluido el plagio, copia o uso de material no autorizado, resultará en una calificación de cero en el examen y puede llevar a sanciones adicionales según las políticas de RIWI.
- **Idioma:** Todo el código debe ser 100% en inglés incluyendo los comentarios, lo único que se permite en español es la información en sí de los registros en la base de datos
- **IA** bajo los lineamientos descritos en la política de uso de IA e internet
- **Permanencia en la sala:** Una vez iniciado el assessment, no se permite salir de la sala de entrenamiento hasta haber entregado el assessment y durante los descansos cada 3 horas (20 minutos c/u).
- **Entrega:** Una vez finalizada la prueba, se debe subir una carpeta comprimida a la plataforma Moodle. La carpeta comprimida debe nombrarse como **Assessment-NombreCompleto** del coder y debe contener:
 1. Enlace del o los repositorios que contienen el código fuente de la solución.
 2. Enlace del proyecto en Azure Devops (opcional)
 3. Credenciales de la o las bases de datos
 4. Diagramas UML del proyecto
 5. Casos de uso del proyecto



6. Valores de variables de entorno en archivo o archivos .env

- **Medio de Entrega:** El único medio para la entrega es la plataforma Moodle. Se recomienda enviar la prueba 10 minutos antes de la hora límite en caso de eventualidades.

La hora máxima de entrega es a las **14:00 horas**. Las pruebas enviadas después de esta hora **NO** serán consideradas.

Los cambios realizados en o los repositorios después de la hora de entrega no serán tomados en cuenta

Objetivo:

Este assessment de empleabilidad tiene un doble propósito:

1. Evaluación de habilidades y competencias: Medir el nivel de habilidades técnicas, capacidades de resolución de problemas y competencias generales que los coders han desarrollado durante su entrenamiento en Riwi.
2. Escalafón y priorización: Asignar un puntaje ponderado junto con el performance histórico del coder, que servirá para escalafón y priorización en la asignación de oportunidades laborales. La ponderación será de 40% para el puntaje obtenido en el assessment y 60% para el performance histórico.

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.

1. Para representar el cálculo del puntaje *assessment de empleabilidad* matemáticamente, se define la fórmula ponderada de la siguiente manera:

a. ***Puntaje de Empleabilidad = (0.4 × Puntaje Técnico) + (0.6 × Performance Histórico).***

Donde:

- **Puntaje Técnico:** Representa el puntaje obtenido en la evaluación de habilidades técnicas, capacidades de resolución de problemas y competencias generales en el assessment.
- **Performance Histórico:** Es el puntaje acumulado o promedio del desempeño histórico del coder a lo largo de su entrenamiento.

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.

Contexto del Caso Práctico: Gestión de Citas para Clínica

Como parte de este assessment, se te asigna un proyecto que simula un entorno real. Has sido contratado como parte de un equipo junior para desarrollar un sistema de gestión de citas para una clínica privada. A continuación, se detallan los problemas identificados en la clínica y los objetivos de la solución esperada:

Problemas que enfrenta la clínica:

1. **Conflictos de horario:**
 - a. En más de una ocasión, se han registrado **dos citas al mismo tiempo** para el mismo médico, lo que provoca que algunos pacientes deban ser **reprogramados** a último momento. Esto no solo genera frustración entre los pacientes, sino también afecta la **organización diaria** de los médicos, quienes se ven forzados a ajustar su horario sobre la marcha.

- b. Además, no existe un sistema que controle si un médico está **disponible** en un momento determinado, lo que genera incertidumbre al momento de agendar citas.

2. Citas duplicadas:

- a. La clínica ha notado que, en ciertos casos, algunos pacientes reciben **doble confirmación** de citas o terminan con **más de una cita** asignada para el mismo problema médico. Esto se debe a que, al gestionar manualmente, el personal puede perder el control sobre los **registros previos** de las citas. Como resultado, se está perdiendo tiempo valioso, y en algunos casos, citas importantes se **olvidan** o se **cancelan** sin previo aviso.

3. Falta de seguimiento del historial de citas:

- a. Actualmente, la clínica no cuenta con un registro detallado del historial de citas de cada paciente. Esto significa que, cuando un paciente regresa, el personal administrativo o los médicos no pueden ver fácilmente sus citas anteriores, lo que dificulta la revisión del historial médico relacionado con esas citas. En algunos casos, los médicos solicitan información repetida a los pacientes debido a la falta de este seguimiento.
- b. El personal administrativo ha intentado llevar un registro manual en hojas de cálculo, pero esto ha demostrado ser poco práctico y propenso a errores, especialmente cuando el volumen de pacientes aumenta. Necesitan una forma de acceder y revisar el historial de citas de cada paciente de manera sencilla y sin complicaciones.

4. Dificultad para visualizar la disponibilidad de los médicos:

- a. El personal de la clínica ha reportado que **no tienen una vista clara** del horario de los médicos. A menudo, tienen que revisar varias hojas de cálculo para saber si un médico está disponible o no. Esto **ralentiza el proceso** de agendamiento, y en más de una ocasión se han agendado citas en **horarios no disponibles**.
- b. También se ha observado que cuando un médico está **ausente** o debe cancelar una cita por un imprevisto, no hay una forma eficiente de **reprogramar** a los pacientes de manera automática, lo que genera una sobrecarga en la gestión manual.



La clínica está buscando una **solución tecnológica** que elimine los problemas actuales y haga que la gestión de citas sea **más eficiente** y **menos propensa a errores**. El sistema debe permitir a diferentes personas (médicos y pacientes) interactuar con la información de las citas de forma sencilla y clara. Para lograr esto, la solución debe:

- **Permitir que diferentes usuarios accedan al sistema de forma independiente**, desde distintos lugares dentro de la clínica o incluso desde fuera de ella, dependiendo de su rol.
- **Mantener la información organizada** de manera centralizada, de modo que no haya confusión o duplicación de citas, y cada usuario pueda ver lo que necesita sin tener que consultar otras personas.
- **Facilitar la visualización de la disponibilidad de los médicos** y las citas programadas de manera rápida, clara y fácil de actualizar cuando sea necesario.

Es importante que el sistema sea fácil de usar para cualquier persona, sin importar su experiencia técnica. Además, debe ser **flexible y preparado para crecer**, para que el cliente pueda agregar nuevas funcionalidades o integrar otras herramientas en el futuro si fuera necesario, sin tener que cambiar todo el sistema.

Criterios de Aceptación

- Poder registrarme, loguearme e interactuar con base a mi rol
- No debe permitir agendar dos citas en el mismo horario para un médico
- Debe actualizar la disponibilidad en tiempo real cuando se agenda una cita
- Debe mostrar advertencias cuando se intenta agendar en un horario ocupado
- Incluir información sobre el motivo de cada cita
- Permitir filtrar citas por fecha, especialidad o motivo
- Permitir agregar notas o comentarios a cada cita
- Los usuarios solo pueden ver la información pertinente a su rol



Listado de entregables

- Documentación (readme, diagramas, casos de uso, etc.)
- Enlace al/los repositorios
- Valores de variables de entorno en archivo o archivos .env
- Base de datos desplegadas y pobladas

Adicionales (Opcional)

- **Notificaciones en tiempo real:** Implementar notificaciones en tiempo real mediante WebSockets o una herramienta como Pusher para alertar a los usuarios sobre eventos importantes (nuevas citas, cambios en el horario de un médico, cancelaciones o recordatorios). La funcionalidad debe permitir notificaciones push que lleguen al navegador o al dispositivo móvil, diferenciando por rol (por ejemplo, avisos de confirmación para pacientes o notificaciones de disponibilidad a médicos).
- **Pruebas avanzadas:** Crear un conjunto robusto de pruebas, incluyendo:
 - **Pruebas de integración y de extremos a extremos (E2E),** que abarquen el flujo de usuario completo, como el registro, programación de citas, filtrado y visualización de disponibilidad.
- **Despliegue en la nube:** Configurar el sistema para que escale automáticamente y mantenga su disponibilidad, usando monitoreo para revisar su estado y rendimiento.
- **Tablero de gestión en Azure DevOps:** Crear un tablero que permita a los desarrolladores y stakeholders gestionar el ciclo de vida del desarrollo, facilitando el seguimiento de tareas y prioridades en tiempo real. Incluir métricas como progreso de desarrollo, estado de los despliegues y bugs identificados. Dicho proyecto en azure devops debe ser nombrado como **Assessment-Clan(NombreCompleto)**, ejemplo:

Assessment-Ritchie(PepitoPerez)

Assessment-BernersLee(PepitoPerez)



Assessment-JeffBezoz(PepitoPerez)

Recursos de apoyo

<https://github.com/Riwi-io-Medellin>

<https://nodejs.org/docs/latest/api/>

<https://docs.nestjs.com/>

<https://typeorm.io/>

<https://www.python.org/doc/>

<https://www.django-rest-framework.org/>

<https://dev.azure.com/riwi-intro>

<https://learn.microsoft.com/es-es/training/dotnet/>

<https://learn.microsoft.com/es-es/dotnet/>