



Check Point Globant – React Native CLI

Team Leader: Ramiro Grisales Noguera

Plataforma de Gestión y Mantenimiento de Flotas de Vehículos

1. 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 sobre el uso de IA
- **Permanencia en el Aula:** Una vez iniciado el assessment, no se permite salir del aula hasta haber entregado el assessment y descanso cada 3 horas.





- **Entrega:** Una vez finalizada la prueba, se debe subir una carpeta comprimida a la plataforma Moodle. La carpeta comprimida debe nombrarse como CheckPoint-ReactNative-NombreCompleto del coder y debe contener:
 - Enlace del o los repositorios
 - Enlace del proyecto en Azure Devops (opcional)
 - Diagramas del proyecto
 - Casos de uso del proyecto
 - Aplicación APK
 - 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 22:00 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

2. Objetivo

Este checkpoint tiene como objetivo evaluar las habilidades técnicas adquiridas hasta el momento en la tecnología React Native CLI, se permitirá el uso de documentación oficial, ejemplos de código abierto y herramientas de IA como apoyo, siguiendo estrictamente los siguientes lineamientos:

3. 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.





4. 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.

5. Introducción

Has sido contratado para desarrollar un proyecto de gestión de vehículos, donde los usuarios podrán controlar su flota de vehículos y mantenimientos. La empresa que te contrata desea una plataforma intuitiva, que facilite el control de los vehículos y fomente el uso de la aplicación.

6. Problemática

Empresa: Belatrix S.A., una empresa de transporte y logística que opera a nivel nacional enfrenta serios desafíos en el mantenimiento y control de su flota de vehículos, compuesta





por más de 500 unidades, entre camiones, furgonetas y vehículos de reparto. La falta de un sistema eficiente de registro de mantenimiento ha ocasionado problemas recurrentes:

- **Incremento en el tiempo de inactividad** de los vehículos debido a mantenimientos no planificados.
- **Costos elevados** derivados de reparaciones inesperadas y desgaste prematuro de las unidades.
- **Falta de visibilidad** en los datos históricos de cada vehículo, dificultando la toma de decisiones informadas sobre cuándo es necesario renovar o dar de baja una unidad.

Ante esta situación, la empresa ha decidido implementar una **plataforma digital** que permita a los administradores y técnicos realizar un seguimiento detallado del historial de mantenimiento de cada vehículo, optimizando la planificación de servicios preventivos y reduciendo tiempos de inactividad.

7. Objetivo del Proyecto

Desarrollar una aplicación para dispositivos Android que interactúe con una API backend para gestionar la flota de vehículos de Belatrix S.A., ofreciendo a los usuarios una interfaz para:

- **Registrar nuevos vehículos** y asociarlos a la flota.
- **Realizar el seguimiento de los servicios de mantenimiento** de cada vehículo, registrando el tipo de mantenimiento, la fecha, el kilometraje y otros datos relevantes.
- **Consultar el historial de mantenimientos** de cada vehículo, con opciones de filtrado y paginación, para facilitar el análisis de patrones de mantenimiento.
- **Verificar el estado de cada vehículo** y programar mantenimientos preventivos según el historial de uso y mantenimiento registrado.

8. Tecnologías y patrones requeridos





Framework y/o Tecnologías:

React Native CLI 0.75 o superior y todo el ecosistema de React Native

Patrones de componentización a utilizar:

- **Componentes funcionales:** Componentes pequeños y enfocados en una sola funcionalidad.
- **Componentes contenedores (Custom Hooks):** Componentes que manejan la lógica de negocios y el estado de la aplicación.
- **Componentes de presentación:** Componentes puramente visuales, sin lógica de negocios.
- **Componentes de enrutamiento:** Componentes que manejan la navegación entre diferentes vistas de la aplicación.

Aplicar mejores prácticas de React:

- **Uso de props y estado:** Pasar datos a los componentes a través de props y manejar el estado interno de cada componente de manera eficiente.
- **Ciclo de vida de los componentes:** Entender y utilizar correctamente los métodos del ciclo de vida de los componentes (mounting, updating, unmounting).
- **Separación de concerns:** Mantener una clara separación entre la lógica de negocios, la presentación y el manejo de estado.

Programación Funcional:

La interfaz debe estar desarrollada usando componentes funcionales de React y hooks

9. Requerimientos específicos

Autenticación y Gestión de Sesión:



www.riwi.io



301 732 53 27



Cl. 16 # 55 - 129



- Los usuarios deben poder iniciar sesión y mantenerse autenticados en toda la aplicación.
- Mantener el token disponible para usarlo en las rutas protegidas
- Opción de cerrar sesión y manejar la expiración de la sesión.

Registro y Gestión de Vehículos:

- Los usuarios deben poder registrar nuevos vehículos, proporcionando detalles como marca, modelo, año, y número de placa.
- Permitir la carga de una imagen del vehículo en el proceso de registro, almacenándola y mostrando una vista previa de la imagen en la interfaz.
- Permitir a los usuarios editar la información de los vehículos o eliminar vehículos de la flota.

Historial de Mantenimiento:

- Los usuarios deben poder registrar y consultar el historial de mantenimiento de cada vehículo.
- El historial debe mostrar los registros de mantenimiento ordenados por fecha, desde el más reciente al más antiguo.

Carga de listados a demanda(paginado) y Filtrado en Listados:

- Los listados de vehículos y mantenimientos deben ser paginados y ofrecer controles de filtrado para facilitar la búsqueda, según los atributos clave (marca, modelo, año, etc.).

Experiencia de Usuario:

- Asegúrate de que la interfaz sea intuitiva y responsiva para que los usuarios puedan acceder y gestionar la información de manera rápida y eficiente.
- Muestra notificaciones o alertas de éxito/error en el registro, edición, o eliminación de





10. Criterios de aceptación

- Repositorio de Código: Un repositorio en Git (preferiblemente GitHub) con el código del proyecto completo, instrucciones de uso y/o instalación, y un archivo README detallado.
- Diagramas UML: Entidad Relación, Casos de uso, Flujo de la app
- APK 100% funcional con **todos los requerimientos mínimos solicitados e instalable para dispositivos Android**, adicional debe tener su propio icono y splash screen

11. Entregables por Moodle

- Aplicación APK
- Archivo TXT con los valores del .env (si aplica)
- Enlace al repositorio de Github - el repositorio debe ser publico
- Diagramas UML

Todo debe ser entregado en un solo archivo comprimido, **NO** se aceptarán archivos independientes.

12. Anexos

- [Link Coleccion de postman](#)

13. Adicionales (Opcionales)

Funcionalidad de Escaneo QR para Identificación Rápida

- Generar un código QR único para cada vehículo, que pueda ser escaneado para acceder rápidamente a su historial de mantenimiento y estado actual.





- Permitir la actualización del registro de mantenimiento mediante el escaneo del código QR, facilitando el trabajo en campo de los técnicos.

Sistema de Notificaciones Avanzado

- Implementar notificaciones push en tiempo real para alertar sobre la necesidad de mantenimiento preventivo o cuando se presente un problema crítico.
- Agregar opciones de configuración de notificaciones para que los usuarios puedan personalizar las alertas según sus preferencias (ej., recordatorios de mantenimiento, alertas de kilometraje).

Generación de reportes y envió por email

- Implementar la generación de reportes en PDF y envió del mismo por medio del sistema de correos del dispositivo.

14.Recursos

<https://reactnative.dev/docs/environment-setup>

<https://www.npmjs.com/package/react-native-image-picker>

<https://www.npmjs.com/package/react-native-pdf>

<https://www.npmjs.com/package/react-native-mail>

<https://icon.kitchen/ii/H4slAAAAAAAAA6tWKkvMKU0tVrKqVkpJLMoOyUjNTVWYySkvMKU6t1VHKzU8pzQHJRi5qUU5WemKOkZeYXA8ny1CSI2FoApT8%2BHkAAAAA%3D>

<https://www.digitalocean.com/community/tutorials/android-splash-screen>