DESARROLLO DE APLICACIONES OPEN SOURCE (SI729)
EXAMEN FINAL
2022-2

Sección: SW51, SW52, WS51, WV51, WX51
Profesores: Asencio Zelada, Víctor Manuel
Flores Moroco, Juan Antonio
Velásquez Núñez, Ángel Augusto
Duración: 170 minutos
Indicaciones:

1. El examen consta de 1 pregunta, y tendrá 170 minutos para resolverlas.
2. La pregunta es de tipo Proyecto de Software y la entrega de su respuesta es a través de envío de archivo empaquetado **.zip** con nombre *upc-pre-202202-si729-<sección>-eb-u<código-estudiante>.zip*, conteniendo el proyecto de software, en la Actividad para el Examen final.
3. Cada examen cuenta con un equipo académico, el cual estará conectado durante los primeros **15 minutos del examen**.
4. El alumno debe dedicar los primeros 15 minutos a revisar las preguntas del examen y de presentarse alguna duda enviar un correo al(los) profesor(es)
Velásquez Núñez, Ángel Augusto, correo pcisavel@upc.edu.pe
5. De no recibir respuesta del equipo académico, o tener algún inconveniente adicional pasado los primeros 15 minutos, puede comunicarse con el profesor
Requejo Chaname, Walter Juan, correo pcsiwreq@upc.edu.pe
6. Los profesores en mención, solo recibirán correos provenientes de las cuentas UPC, de ninguna manera se recibirán correos de cuentas públicas.
7. Ante problemas técnicos, debe de forma obligatoria adjuntar evidencias del mismo, como capturas de pantalla, videos, fotos, etc. Siendo requisito fundamental que, en cada evidencia se pueda apreciar claramente la fecha y hora del sistema operativo del computador donde el alumno está rindiendo el examen.
8. Los problemas técnicos se recibirán como máximo 15 minutos culminado el examen.

Enunciado:

Caso EcoDrive, inc.

Su cliente, EcoDrive, inc. se embarcó tiempo atrás en el proyecto de desarrollar vans híbridas que ofrezcan mayor eficiencia en términos de consumo de combustible y emisión de gases. La tecnología que desarrollaron se desarrolló y puso a prueba, pero lamentablente falló en producir los resultados esperados cuando las vans entraron en operación. Algo no estaba siendo considerado.

Después de un análisis profundo, Mark Roberts, fundador de EcoDrive, inc., se dio cuenta que la razón por la que las vans convertidas no llegaban a alcanzar su potencial no era era un aspecto tecnológico – todo se reducía a la forma como eran conducidas. Llegó a la conclusión de que, no importa que tan optimizado esté un vehículo, si es conducido de forma ineficiente, su rendimiento siempre será ineficiente.

Esta idea llevó a un cambio de rumbo de la compañía, dejando de lado el tema de vans híbridas y estableciendo como nuevo objetivo el mejorar el rendimiento de los choferes para asegurar que cualquier flota de vehículos disfrute de la experiencia de reducción de costos, reducción de riesgo y menor impacto en el medio ambiente.

Es por ello que EcoDrive acude a usted con la idea de desarrollar un sistema de *fleet management* (gestión de flotas), contando con el apoyo de un equipo de expertos en driver-behaviour. El sistema debe brindar a los clientes el beneficio de tener flotas más seguras, más eficientes y menos costosas de operar.

EcoDrive deberá comunicarse directamente con el conductor a través del EcoDrive IoT device en el tablero de instrumentos en la cabina, brindando comentarios visuales y verbales en tiempo real para fomentar una conducción más suave. A medida que mejora su rendimiento de conducción, los conductores podrán participar en competencias, ganar premios, competir en tablas de clasificación y disfrutar de descuentos exclusivos a través de la EcoDrive App. De esta forma, EcoDrive no solo proporcionará a los conductores de la flota del cliente las herramientas para crear mejores hábitos de manejo, sino que también los recompensa e incentiva por hacerlo, lo que garantizará que el cliente experimente resultados duraderos dentro de su flota.

El in-cab EcoDrive IoT device estará completamente conectado a la onboard computer del vehículo, lo que permitirá que la intelligent technology lea y comprenda los datos del motor. Esto significa que EcoDrive puede obtener una imagen real de cómo se opera el vehículo, teniendo en cuenta factores como la selección de gear selection (cambio de velocidad), revs (las revoluciones), engine load (la carga del motor), payload (la carga útil), si el vehículo va uphill (cuesta arriba) o downhill (cuesta abajo) e incluso si está remolcando.

Esta información permitirá que el dispositivo brinde asesoramiento en tiempo real basado en el desempeño real del conductor, guiando a sus conductores hacia un estilo de conducción que utilice el motor de la manera más productiva posible, lo que llevará a una eficiencia y economía de combustible óptimas.

La capacitación a conductores se brindará a través del *EcoDrive Nudge System*, un *embedded system* para el EcoDrive device, cuyo desarrollo contará con el apoyo de científicos del comportamiento y psicólogos, que hará uso de de alertas audibles y semáforos en el display del EcoDrive device para que los conductores se den cuenta cuando se están alejando del punto óptimo del motor, o el punto donde el vehículo está entregando el mejor kilometraje posible.

Con el apoyo de este sistema, los conductores aprenderán a ajustar su estilo de conducción en el momento, haciendo cambios intuitivos incluso antes de que ocurran casos de mala conducción. El resultado que se busca es una flota rentable en la que las mejoras se mantienen a lo largo del tiempo a medida que la conducción más suave y segura se convierte en la norma, reemplazando los malos hábitos como el exceso de velocidad, el frenado brusco y la aceleración rápida.

EcoDrive convertirá el rendimiento del conductor en una competencia a través del EcoDrive App para Drivers. Se contará con una plataforma de rewards en línea que ayudará a los conductores a crear mejores hábitos de conducción de por vida a través del refuerzo positivo, ya que reciben puntajes altos, premios y reconocimiento por un manejo más fluido del vehículo. La aplicación también destacará áreas de mejora, lo que permitirá a los conductores ser autónomos mientras trabajan para lograr un estilo de conducción más eficiente.

El equipo de EcoDrive le comenta que desde su punto de vista el objetivo es simple: cada conductor tiene como objetivo alcanzar el estándar de "Elite Driver" manteniendo su EcoDrive device en verde con la mayor frecuencia posible. Esto no significa conducir de manera eficiente el 100 % del tiempo; eso simplemente no es factible en el manejo en el mundo real. En cambio, el objetivo se establece en el 85%, lo que permite un amplio margen para maniobras esenciales, como aumentar la velocidad en las vías de acceso o alejarse rápidamente en las rotondas. Sin EcoDrive, alrededor del 3 % de los conductores alcanzarían este estándar; con EcoDrive, el 80 % de los conductores de flota alcanzarán el objetivo.

Los conductores podrán controlar su puntaje y monitorear su desempeño a través de la aplicación, además de ver cómo se comparan con otros conductores a través de in-app league tables (tablas de clasificación) y leader boards. Hay muchas opciones, que incluyen pre-set league tables (tablas de liga preestablecidas) entre diferentes departamentos y regiones, y los usuarios de la aplicación podrán incluso crear sus propias ligas, lo que generará una competencia sana y alegre entre amigos y colegas.

Pregunta 1 (20 p.).

Usted se integra al backend software developer team, a cargo de la creación de un **RESTful API** que brinde soporte a las operaciones de ecodrive.com.

El ecosistema de ecodrive.com requiere que el Fleet Management Application, el Embedded IoT System y el Driver App Application cuenten con **Endpoints** en el RESTful API, para el manejo de la información de *Scores* conformadas por los atributos id (Long), driverId (Long), value (Float), registeredAt (Date).

Durante la etapa de desarrollo, le asignan trabajar en específico sobre un Endpoint:

Driver Driving Scores Endpoint (/api/v1/drivers/{driverId}/scores)
Debe implementar **solo dos** operaciones en el RESTful API: agregar un **driving score** (POST). Los valores de **id** no se ingresan al momento de agregar, pues son autogenerados al momento de almacenar la información; la otra operación es consultar los scores (GET). En el caso de la consulta se puede incluir el parámetro scope como request parameter (@RequestParam) en el request. A continuación los posibles valores de scope y el response esperado.

scope	Interpretación	Request example	Response

0	Obtener el valor promedio general de score para el driver	http://localhost:8090/api/v1/drivers/5/scores?scope=0	JSON object con objeto score conteniendo el valor promedio con respecto a todos los objetos score almacenados para el driver
1	Obtener el máximo valor de score obtenido hasta el momento por el driver	http://localhost:8090/api/v1/drivers/5/scores?scope=1	JSON object con objeto score conteniendo el máximo valor con respecto a todos los objetos score almacenados para el driver

Como reglas de negocio, ecodrive.com:

- Establece que la información que se desea preservar de las ***Scores*** incluye **id** (**Long**, **obligatorio**, **autogenerado**, **llave primaria**), **driverId (Long, obligatorio, no vacío)**, **value (Float, obligatorio, no vacío)**, **registeredAt (Date, obligatorio, autogenerado en base a fecha y hora actual al momento del registro)**,
- En el caso de responses que retorna el API con información de **Score**, debe incluir **value** y **driverId**.
- En caso de no incluir el parámetro **scope** debe retornar un response con el **HTTP Status** de *Bad Request* (HttpStatus.BAD_REQUEST) con el mensaje *"Scope value not specified"*.

Incluya como parte del desarrollo la implementación de todas las reglas de negocio.

Technical constraints

1. Elabore la solución con Java 19 y Spring Boot Framework.
2. Cree su proyecto de software con el nombre ***si729ebu<código-estudiante>*** (por ejemplo, *si729ebu201621873*).
3. La información debe ser persistente en una base de datos relacional (MySQL / PostgreSQL), en un esquema **ecodrive**.
4. Los packages de su solución deben tener como nombre raíz **com.ecodrive.platform.u<código-estudiante>** (por ejemplo, com.*ecodriver.platform.u201621873*).
5. Considere que el concepto **Score** pertenece al bounded context **behaviour**.
6. Aplique buenas prácticas de Arquitectura de Software, enfoque Domain-Driven, principios y patrones de diseño, convenciones de nomenclatura en inglés, así como buenas prácticas de nomenclatura en Java (entre ellas Upper-Camel-Case para Clases, Lower-Camel-Case para atributos y métodos) y buenas prácticas para nomenclatura de objetos de Base de Datos (entre ellas snake case, tablas en plural, sin mnemónicos).
7. El puerto de escucha del API en **localhost** debe ser el puerto **8090**.
8. Utilice minúsculas para los nombres de URL para todos los endpoints.
9. Utilice la biblioteca Lombok para el manejo de métodos constructores y de acceso en las clases POJO[1].
10. Utilice la biblioteca ModelMapper para el Object Mapping.
11. Incluya documentación de Endpoints con OpenAPI.
12. Considere la gestión de excepciones en la aplicación.

[1] POJO: Plain Old Java Object. Se refiere a aquellas clases que en su mayoría no contienen lógica, solo atributos y métodos de acceso a los mismos (get y set).

13. Empaquete su solución como un archivo **.zip**. (único formato válido) con el nombre ***upc-pre-202202-si729-<sección>-eb-u<código-estudiante>.zip*** (por ejemplo, *upc-pre-202202-si729-sw51-eb-u201621873.zip*).
14. Suba su archivo de solución en la Actividad indicada para el Examen final.

NO forma parte del alcance del proyecto:

1. Auditoría
2. Soporte de CORS.
3. Security.
4. Testing.

Rúbrica de calificación

Criterio de Calificación	Sobresaliente (S)	Esperado (E)	Necesita Mejorar (M)	Insuficiente (I)	Calificación
C01. Building y ejecución	Al abrir el proyecto y ordenar la ejecución, ésta se inicia sin problemas. El API es accesible en la ruta indicada.	La aplicación no llega a iniciar y ejecutarse, sin embargo el proceso de building llega a concluir.	Al cargar el proyecto el proceso de building presenta errores y no llega a concluir.	No elabora solución.	
	2.0 puntos	**1.0 punto**	**0.5**	**0 puntos**	
C02. Driver Scores Endpoint	El RESTful API expone el endpoint /api/v1/drivers/{driverId}/scores. Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones. Se evidencia la persistencia de los objetos Score, cumpliendo la estructura según enunciado, en una tabla *scores* en base de datos relacional indicada en un esquema *ecodrive*. Se incluye documentación del Endpoint con OpenAPI.	El RESTful API expone el endpoints /api/v1/drivers/{driverId}/scores. Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones, o se evidencia parcialmente la persistencia en base de datos relacional, o se evidencia parcialmente la persistencia de los objetos Idea con estructura según enunciado, en una tabla *scores* en base de datos relacional indicada en un esquema *ecodrive*.	La aplicación implementa y expone el endpoint /api/v1/drivers/{driverId}/scores, pero no cumple con la ruta especificada o no se puede registrar elementos.	La aplicación no implementa o expone el endpoint /api/v1/drivers/{driverId}/scores.	
	6.0 puntos	**4.0 puntos**	**2.0 puntos**	**0 puntos**	
C04. Business Rules	El desarrollo incluye la implementación de reglas de negocio, cubriendo de forma completa las condiciones y escenarios establecidos, siendo éstas ejecutables, con adecuado manejo de excepciones, implementando éstas en las capas más adecuadas, aplicando convenciones y buenas prácticas.	El desarrollo incluye la implementación de la mayoría de reglas de negocio, cubriendo de forma parcial las condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en la mayoría de casos, ó aplicando parcialmente convenciones y buenas prácticas.	El desarrollo incluye la implementación de algunas de las reglas de negocio, incumpliendo la mayoría de condiciones y escenarios establecidos, siendo éstas ejecutables, implementándolas en las capas adecuadas en muy pocos casos, ó con poca evidencia de aplicar convenciones y buenas prácticas.	No implementa reglas de negocio, o no cubre escenarios más allá de operaciones CRUD básicas, o éstas no son ejecutables.	
	4.0 puntos	**3.0 puntos**	**1.5 puntos**	**0 puntos**	
C05. Code Organization	El desarrollador organiza el código y los elementos de backend de la solución, aplicando buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design, agrupando los elementos de la solución según convenciones, manteniendo organización de paquetes y carpetas recomendadas por el fabricante y buenas prácticas de la industria de software.	El desarrollador aplica en la mayoría de casos para el backend convenciones, recomendaciones y buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design.	El desarrollador aplica en algunos casos para el backend convenciones, recomendaciones y buenas prácticas de Java, Spring Boot Framework y Domain-Driven Design.	No se evidencia un criterio de organización para los elementos de la solución.	
	3.0 punto	**2.0 puntos**	**1.0**	**0 puntos**	
C06. Code Quality	Utiliza para el backend el lenguaje de programación Java La codificación tiene un estilo claro, indentando los bloques de código según los estándares de programación correspondientes al lenguaje, aplicando una lógica consistente en los métodos, condicionales sin escenarios no contemplados, uso adecuado de reutilización de código para evitar redundancia. Aplica patrones de arquitectura y patrones de diseño. Distribuye el código en los niveles correspondientes, asignando lógica de persistencia, lógica de negocio, lógica de control, lógica de mapping y transferencia a las interfaces y clases que corresponden. Cumple de forma completa con los technical constraints.	Utiliza para el backend el lenguaje de programación Java. La codificación es funcional, aplica en la mayoría de casos los estándares de indentación de bloques de código, ó existen algunas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica parcialmente patrones de arquitectura y patrones de diseño, o existe en algunas partes una distribución de la lógica en los niveles incorrectos. Cumple con la mayoría de technical constraints.	Utiliza para el backend el lenguaje de programación Java. La codificación es funcional, pero solo aplica algunos de los estándares de indentación de bloques de código, ó existen muchas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica algunos patrones de arquitectura y patrones de diseño, o existe en muchos casos una distribución de la lógica en los niveles incorrectos. Cumple con solo algunos de los technical constraints.	No utiliza el lenguaje de programación Java para el backend, ó la codificación es funcional pero no se evidencia aplicación de estándares ó criterios de eficiencia en la codificación, con ausencia de comentarios, ó no aplica patrones de arquitectura ni patrones de diseño, o la codificación no es funcional.	
	3.0 puntos	**2.0 punto**	**1.0**	**0 puntos**	
C07. Naming Standards	El desarrollador aplica en todos los nombres de objetos de programación y base de datos como paquetes, componentes, interfaces, clases, objetos, variables, constantes, métodos, tablas, columas la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, tablas, columnas, así como los recursos.	El desarrollador aplica en la mayoría de casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica en muy pocos casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador no aplica nomenclatura en inglés para los objetos de programación ó recursos.	
	2.0 puntos	**1.0 punto**	**0.5 puntos**	**0 puntos**	
Total	**20 puntos**	**13.0 puntos**	**6.5 puntos**	**0 puntos**	

Lima, 30 de Noviembre del 2022

Anexos

Anexos A. Referencias

Comprimir y descomprimir archivos: https://support.microsoft.com/es-es/windows/comprimir-y-descomprimir-archivos-8d28fa72-f2f9-712f-67df-f80cf89fd4e5

REST API Tutorial: https://restfulapi.net/

Project Lombok: https://projectlombok.org/

ModelMapper: http://modelmapper.org/

springdoc-openapi: https://springdoc.org/

Spring Data JPA - Reference Documentation: https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#reference