BACK-END

# DEVELOPER TECHNICAL SKILLS

Lea la prueba y teniendo en cuenta su disponibilidad de tiempo, informe el día y hora en la que máximo enviará las respuestas. Note que en el transcurso del tiempo podrá enviar preguntas que le serán respuestas con la mayor prontitud. El cumplimiento a los propios compromisos es el ADN de RAPPI.

**CODING CHALLENGE (70 puntos)**

La primera parte de la prueba consiste en un Coding Challenge tomado de hackerrank. Lo puede encontrar en el siguiente link: <https://www.hackerrank.com/challenges/cube-summation/problem> .

Puede escribir el programa en el lenguaje de programación en el que se sienta más cómodo, aunque se prefiere el uso de los siguientes lenguajes: Java, Node, Kotlin, Scala, o PHP. Puede usar cualquier mecanismo para la entrada y salida de datos del programa. Una vez haya terminado el código describa en un documento brevemente:

1. Las capas de la aplicación (por ejemplo, capa de persistencia, vista, de aplicación, etc.) y qué clases pertenecen a cuál.
2. La responsabilidad de cada clase creada.

**Recibirá puntos extra si:**

1. El mecanismo usado para la entrada y salida de datos es Web (5 puntos).
2. Usa Bitbucket (preferido), git o SVN y la historia del programa muestra su progreso en el desarrollo, usando commits con unidades de funcionalidad (5 puntos).
3. Usa pruebas unitarias y/o de validación (acceptance) que demuestren el buen funcionamiento del programa (10 puntos).

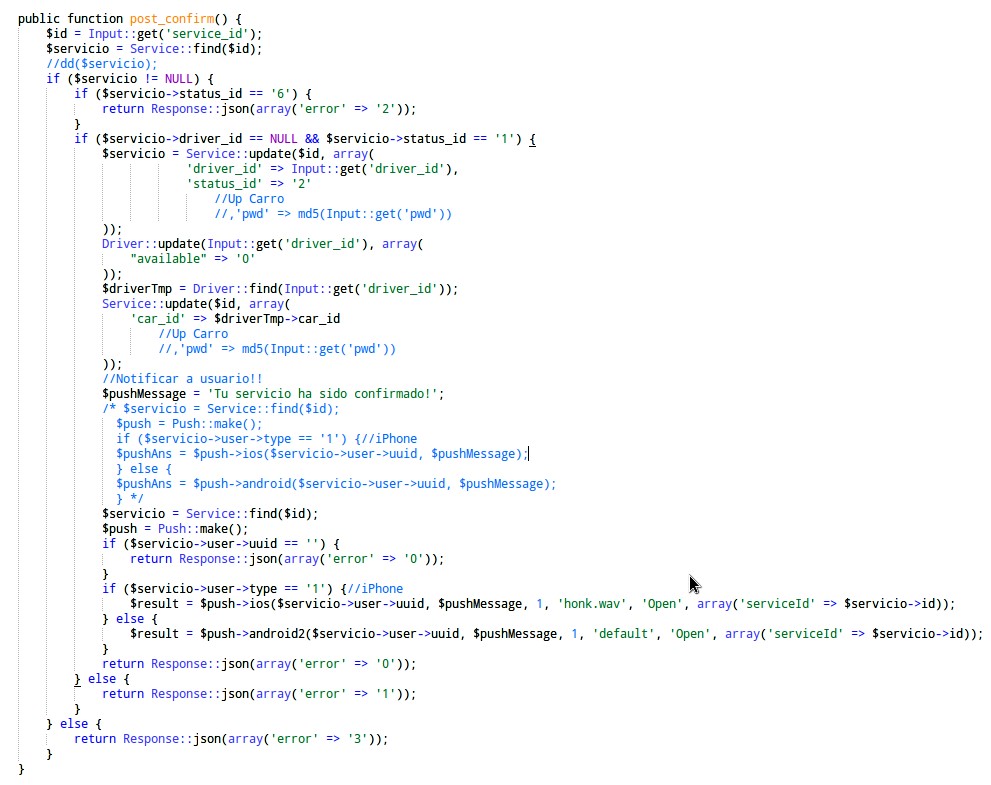
## CODE REFACTORING (20 puntos)

El siguiente código muestra el método de un controlador que:

1. Recibe dos parámetros por POST: El id de un servicio, el id de un conductor.
2. Cambia el estado de varias entidades en la base de datos basado en la lógica del negocio.
3. Envía notificaciones y retorna una respuesta.

**Refactorice y envíe el código y en un documento explique:**

1. Las malas prácticas de programación que en su criterio son evidenciadas en el código.
2. Cómo su refactorización supera las malas prácticas de programación.



## PREGUNTAS (10 puntos)

Responda y envíe en un documento las siguientes preguntas:

1. ¿En qué consiste el principio de responsabilidad única? ¿Cuál es su propósito?
2. ¿Qué características tiene según su opinión “buen” código o código limpio?
3. ¿Qué es un microservicio?, ventajas y desventajas de los micorservicios