

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50% en cada sección. Se puede recuperar por partes (teoría, objetos, arquitectura), pero no para la aprobación directa; en ese caso se deberá rendir el parcial completo nuevamente. Arquitectura se recuperará en conjunto primer y segundo parcial (Arq. parte I, Arq. parte II). Se podrá recuperar dos veces cada eje de la asignatura (objetos, datos, arquitectura).

## ***RunnersTrack***



### **Contexto general**

En esta oportunidad vamos a diseñar y desarrollar **RunnersTrack**, una aplicación móvil que tiene la finalidad de permitir a los **Runners** visualizar y recorrer caminos no convencionales de forma recreativa llamado desde ahora “travesías”. La plataforma está pensada para la comunidad de **Runners** que busca nuevas aventuras y rutas alternativas, proporcionando una experiencia única y personalizada en sus fondos.

### **Gestión de Runners**

De los runners nos interesa su nombre y apellido, peso en kgs, altura, edad, tipo de sangre, y si posee lesiones junto a una especificación cada una.

### **Gestión de Zapatillas**

El Sistema debe permitir que cada **Runner** gestione sus **Zapatillas**. Para ello los runners deberán seleccionar aquellas zapatillas que tengan, directamente desde el catálogo de zapatillas que ofrecerá el Sistema; el cual será gestionado por los administradores de la plataforma. De cada tipo de zapatilla interesa conocer, mínimamente, su marca, modelo y fotos; y de cada zapatilla en particular: el año de adquisición, número de talla (en distintas unidades), fotografía/s y cantidad de kilómetros que se recorrieron con la misma. Este último dato nos interesa pues las zapatillas tienen una vida útil en kilómetros aproximada.

### **Gestión de Travesías**

Los administradores de la plataforma deberán poder cargar Travesías. De cada travesía nos interesa, mínimamente, tener la siguiente información: nombre, descripción detallada, fotografías de la travesía (cada una con descripción), nivel de dificultad (principiante, intermedio, avanzado, extremo), tipo de terreno (asfalto, ripio, montañas, etc.), duración estimada de la travesía (puede ser en horas y/o en días), distancia total (en kms), punto de partida y punto de llegada.

### **Puntos de apoyo**

El Sistema debe permitir que los administradores carguen puntos de apoyo sobre las **travesías**. Estos puntos de apoyo pueden ser: zonas de descanso, puestos de hidratación, estaciones de servicio, baños, entre otros. Además, es necesario que de cada punto de apoyo se cuente con una descripción, ubicación geográfica y fotografías.

### **Planificación de Fondos**

Cada runner debe poder indicar que va a realizar un “**fondo**” en una fecha determinada, seleccionando una travesía. Además, deberán poder especificar sus necesidades particulares para el fondo, tales como: zonas de descanso, puestos de hidratación, estaciones de servicio, baños, etc. Es importante que se indique con qué zapatillas se va a realizar el fondo, debiendo ofrecer el par de zapatillas “favoritas” por defecto.

### Sugerencias Personalizadas

El Sistema deberá enviar sugerencias al runner 48 horas antes de la fecha del fondo, basándose, entre otras cosas, en las necesidades indicadas. Las sugerencias deben incluir:

- Puntos de apoyo a visitar: Deben tenerse en cuenta las necesidades establecidas por el runner para ese fondo.
- Precauciones de seguridad: El Sistema deberá enviar las medidas de seguridad que los administradores dieron de alta para dicha travesía.

Por otro lado, el Sistema debe enviarle al runner, cada mes, una recomendación de nuevas zapatillas para que renueve las suyas. Para lograr esto el Sistema debe importar los modelos de zapatillas que ofrece la plataforma TECH - Shoes, la cual expone una API REST como mecanismo de integración. Es necesario que se le recomienden zapatillas similares a las que posee el runner actualmente, y solamente considerando aquellas cuya vida útil está por alcanzar el 80%. A la derecha se muestra un ejemplo de respuesta de la API.

```
{
  "shoes": [
    {
      "shoe_id": 101,
      "brand": "Nike",
      "model": "Air Zoom Pegasus 37",
      "year": 2022,
      "sizes": {
        "US": 10,
        "EU": 44,
        "UK": 9
      },
      "photos": [
        "url_to_photo1.jpg",
        "url_to_photo2.jpg"
      ]
    }
  ]
}
```

### ¿Cuánto puedo cargar en mi mochila?

El Sistema debe poder calcular el peso máximo estimado que un runner puede llevar en su mochila durante un fondo. Este cálculo se debe realizar teniendo en cuenta varios factores, incluyendo la capacidad física del runner, la dificultad del terreno (que coincide con el nivel de dificultad de la travesía), y la duración de la travesía. El cálculo de este dato se debe poder realizar por alguna de las siguientes opciones (van aparecer más a futuro), según elija el runner en cada caso.

| Cálculo optimista   | Cálculo pesimista  |
|---|--|
| <p>El cálculo optimista asume condiciones ideales donde el runner está en su mejor condición física, el terreno es favorable, y la duración de la travesía no afecta significativamente su rendimiento.</p> <p>Sea <math>p</math> el peso máximo estimado en la mochila, <math>c</math> la capacidad física del runner en kg, <math>t</math> la dificultad del terreno (1 para principiante, 2 para intermedio, 3 para avanzado, 4 para extremo), y <math>d</math> la duración de la travesía en horas.</p> <p>La integral triple que representa el peso máximo es:</p> $\int_0^c \int_0^t \int_0^d \frac{1}{t \cdot d} dx \cdot dy \cdot dz = \frac{c}{t \cdot d} = p$ | <p>Asume que la capacidad física del runner puede variar debido a factores como la fatiga, el clima adverso, y el desgaste durante la travesía. Sea campo escalar <math>f(x, y, z)</math> donde</p> <ul style="list-style-type: none"> <li>• <math>x</math> es el peso inicial,</li> <li>• <math>y</math>: la dificultad del terreno.</li> <li>• <math>z</math>: la duración de la travesía.</li> </ul> <p>La función que resuelve el cálculo es:</p> $f(x, y, z) = \frac{c}{t \cdot d} \left(1 - \frac{x \cdot y \cdot z}{c \cdot t \cdot d}\right)$ <p>La integral triple que resuelve este caso es:</p> $\int_0^c \int_0^t \int_0^d \frac{c}{t \cdot d} \left(1 - \frac{x \cdot y \cdot z}{c \cdot t \cdot d}\right) dx dy dz = \frac{7c}{8} = p$ |

### **Recomendación de Artículos para la Mochila**

Para garantizar que los runners estén bien preparados para sus travesías, el Sistema debe recomendarle los artículos que llevarán en sus mochilas, optimizando la selección en función del peso máximo calculado (en la sección anterior). Es decir, en ningún caso el Sistema debe recomendar una serie de artículos que superen el peso máximo que puede cargar el runner en su mochila.

Los artículos a sugerir deben ser cargados por los administradores de la plataforma y de cada uno interesa: su nombre, peso (en kg) y la categoría del mismo.

### **Alcance y requerimientos**

El Sistema deberá permitir:

1. Que cada Runner gestione sus Zapatillas.
2. Que los administradores gestionen las Travesías.
3. Que los administradores gestionen Puntos de apoyo.
4. Que los runners gestionen sus Fondos.
5. Enviar sugerencias a los runners, 48 horas antes de cada fondo planificado, mediante WhatsApp y notificaciones push
6. Enviar sugerencias de recomendación de “*nuevas zapatillas*”, de forma mensual.
7. Que los runners puedan conocer su peso máximo (en kilogramos) permitido para cargar en la mochila.
8. Que los administradores gestionen los artículos disponibles para las recomendaciones de cargas de mochila.

### **Punto 1 - Modelo de Dominio (45 puntos)**

- A. **(12.5 puntos)** Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(12.5 puntos)** Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. También puede optar por justificar mediante código, pseudocódigo o algún otro diagrama complementario.
- C. **(10 puntos)** Código/pseudo de un req. en particular detallando capa de controladores, servicios (opcional), repositorios y entidades de dominio involucradas de alguna de las siguientes requerimientos:
  - a. Gestión de Puntos de apoyo
  - b. Gestión de travesías
- D. **(10 puntos)** Nos avisaron, un poco tarde, que el Sistema debe soportar **Feedback de Travesías**: Una vez que el Runner haya finalizado un fondo, habiendo transitado una travesía, podrá dar su feedback del mismo. Para ello deberá seleccionar de las opciones: Excelente, Bueno, Intermedio, Debería mejorar.

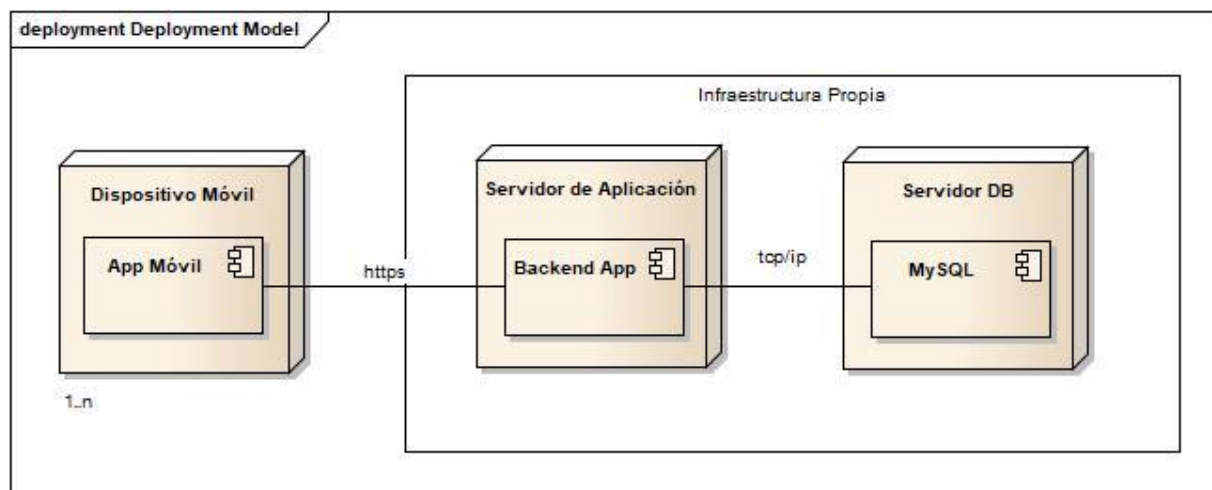
Además, con la finalidad de recolectar la mayor cantidad de información posible para mejorar las recomendaciones, la información de los puntos de interés y las descripciones de las Travesías, el Sistema deberá realizarle una serie de preguntas las cuales pueden ser, por ejemplo:

- ¿Qué te pareció la travesía?
- ¿La recomendación del clima fue precisa?
- ¿Crees que necesitabas más preparación física? Seleccione todas las que correspondan.

Los administradores deberán poder cambiar estas preguntas con facilidad.

## Punto 2 - Arquitectura parte I (45 puntos)

- 1) **(15 puntos)** Dado el siguiente Diagrama de Despliegue del Sistema, analícelo y describa mejoras, faltantes y puntos de mejora a considerar.



- 2) **(15 puntos)** Los Administradores deben recibir al final de cada día un reporte con las métricas del sistema, por ejemplo la cantidad de nuevos usuarios ese día. ¿Cómo resolvería la generación de estos reportes?
- 3) **(15 puntos)** Una empresa de descuentos nos propuso una alianza estratégica con nuestra aplicación. En ese sentido, el sistema de descuentos nos tiene que avisar cada vez que un Runner realiza una compra en algún comercio adherido. El sistema nos enviará el aviso de una novedad con un ID específico. Luego nuestro sistema deberá recuperar la operación y asignarla a un Runner. La misma tiene el DNI del Runner, el id del comercio, descripción corta del comercio y monto de la operación. ¿Qué estrategia propone para la integración entre ambos sistemas? ¿Para incorporar esta integración es necesario cambiar algo del modelo de objetos? Justifique adecuadamente su respuesta.

## Punto 3 - Teoría (10 puntos)

Indique si cada una de las siguientes afirmaciones es verdadera o falsa. En ambos casos justifique su respuesta brevemente. **(2 puntos c/u)**

- 1) El atributo de calidad **Mantenibilidad** es el más importante en todo sistema ya que permite que los desarrolladores produzcan código de manera más eficiente y eficaz.
- 2) El patrón de diseño Adapter siempre se utiliza cuando debemos integrarnos con un componente externo.
- 3) Para obtener un buen diseño, es ideal diseñar con alto acoplamiento y baja cohesión.
- 4) Una clase que solamente tiene atributos y no métodos (sin considerar los setters y getters) es una Data Class.
- 5) Cuando se tiene un método con muchos parámetros, los cuales pueden ir aumentando en el tiempo, se debe considerar refactorizarlo utilizando un Value Object como único parámetro.