

## Segunda Entrega

### Objetivo General

En esta entrega pondremos a prueba algunas de las decisiones de diseño tomadas en la anterior, al incorporar nuevos guardarropas e introducir factores climáticos dentro del algoritmo de sugerencias, que nos permitirán dar recomendaciones más precisas sobre los atuendos.

Para esto, incorporaremos integraciones con sistemas externos, que nos permitirán conocer la temperatura dada una ubicación geográfica, para lo cual introduciremos el concepto de adaptación de interfaces, y presentaremos a alto nivel el protocolo HTTP y las APIs REST.

Finalmente, trabajaremos el patrón command e implementaremos colas de trabajo que permitirán generar sugerencias para eventos que el usuario cargará.

### Dominio

En esta instancia, nos enfocaremos en lograr que cada uno de los atuendos que sugiramos tengan sentido en base al clima. La idea es combinar las prendas de forma tal que satisfagan un nivel de abrigo acorde, es decir, que el usuario no resulte ni demasiado desabrigado, ni excesivamente abrigado. Para lograrlo, necesitaremos integrarnos con una API que nos informe la temperatura en el día y lugar en que nuestro cliente se encuentre al momento de vestirse. Esta información la obtendremos conociendo de antemano los eventos a los que el usuario planea asistir. Estos eventos pueden ser actividades tales como “Ir a trabajar”, “Salir con amigos”, etc. y son cargados por el usuario especificando cuándo y dónde ocurrirán. Por el momento, solo consideraremos eventos que ocurran una única vez.

Dado que las APIs de pronóstico climático que conocemos son inestables y no siempre funcionan correctamente, decidimos implementar la integración con dos de ellas, a fin de poder, a futuro, alternar entre ellas en caso de fallos.

Asimismo, sabiendo que las temperaturas dependerán de la ubicación<sup>1</sup> del usuario, tendremos que comenzar a contemplar temperaturas no tan cálidas.

Es por ello que ahora sí será necesario que nuestras combinaciones puedan incluir varias prendas superpuestas, como usar una camisa con una remera debajo, o ponerse una campera de abrigo y un suéter arriba de la remera. Todo esto recordando que las combinaciones no deben dejar de tener sentido (por ej, no tendría sentido ponerse dos camisas una sobre la otra). Para simplificar, asumiremos que todas las prendas del mismo tipo se comportan igual, es decir, “abrigan” en la misma medida.

---

<sup>1</sup> Por simplicidad supondremos que todos los usuarios estarán localizados en C.A.B.A

Otra funcionalidad que deberemos implementar será el soporte para distintos tipos de usuario: premium y gratuitos. Estos últimos tendrán la particularidad de que sus guardarropas se verán limitados en su capacidad, es decir, tendrán un número máximo de prendas que pueden albergar.

Además, será necesario permitir que, en el guardarropas, al agregar una prenda al mismo, se pueda cargar también una foto de la misma, para facilitar su reconocimiento. Para evitar el uso excesivo de memoria en el servidor para el almacenamiento de las imágenes, sería recomendable que las mismas se guarden normalizadas con una resolución y tamaño fijos, sin tener en cuenta los de la foto original.

Por último, debemos permitir que, ante la ocurrencia de un evento, el usuario pueda elegir entre las sugerencias generadas cuál es aquella que prefiere vestir para esa ocasión. No olvidemos que al hacer esto, el usuario podría aceptar una sugerencia por error, por lo cual deberemos permitir que esta acción sea deshecha.

## Requerimientos Detallados

Para esta entrega se deberán satisfacer los siguientes requerimientos:

1. Soportar combinatorias con prendas superpuestas.
2. Creación y manejo de guardarropas limitados.
3. Carga de eventos para una cierta fecha, y generación de sugerencias para los mismos cuando están próximos.
4. Aceptación de sugerencias para los eventos. Esta funcionalidad deberá posibilitar aceptar, rechazar y deshacer la última operación. Es importante saber qué sugerencias aceptó y rechazó un usuario para, a futuro, adaptarse a sus preferencias.
5. Integración con dos sistemas externos diferentes para obtener los datos de la temperatura de Buenos Aires. Esta integración deberá estar lo suficientemente desacoplada del dominio como para poder:
  - a. Intercambiar o incorporar nuevos proveedores de información climática
  - b. Probar de forma unitaria al sistema utilizando *mocks*
  - c. Al menos una de estas implementaciones deberá interactuar con un API HTTP
  - d. **Opcional:** Integrar con una segunda API HTTP
6. **Opcional:** Posibilitar sugerencias flexibles, que habiliten que los atuendos sugeridos sean aproximados. Es decir, dados climas extremos (o climas en los cuales haya pocas o una sola combinación de prendas para resolver esa sugerencia), el sistema va a dar atuendos muy similares. Lo que buscamos con este requerimiento, es que se pueda flexibilizar el nivel de abrigo para que si bien no es exactamente el atuendo ideal para esa temperatura, esté cerca de este, y te de mas opciones para usar.

7. Carga de imágenes para las prendas a partir de un archivo local. (**Opcional:** almacenarlas normalizando su tamaño y resolución independientemente de cual sea la imagen original subida)
8. La sugerencia de atuendos debe contemplar las reglas de la entrega anterior e incluir:
  - a. Contemplar la temperatura al momento del evento.
  - b. Tipos de prendas, refactorizando los mismos de ser necesario, a fin de manejar cuanto abriga cada una.
  - c. Tener capas de prendas para aumentar el nivel del abrigo.