WEBIR

Curso 2020

Informe Final

Autores: Grupo 06
Santiago Acevedo
CI: 4.996.001-3
Santiago Alles Conde
CI: 4.853.251-8
Oscar Muñoz
CI: 4.778.943-9
Damian Sancristobal
CI: 4.955.160-2

Docente:
Libertad Tansini

Contents

1 Introducción al informe

La consigna recibida fue pensar e implementar una aplicación que realice recuperación, manejo e integración de datos, para darle valor al usuario. Se utilizaron los conceptos vistos en el curso para abordar una problemática que afecta a todo el mundo en estos días, como lo es el COVID-19.
En particular, se trató el tema del turismo en épocas de coronavirus.

2 Problema

Con eso en mente se notó que se han generado muchos cambios en la manera de realizar turismo. Teniendo que buscar mucha información sobre precios, lugares a los que se puede viajar, estado actual de la situación sanitaria en distintos países y las distintas normas existentes en cada uno.
Debido a lo tedioso y difícil de organizar un viaje en los tiempos actuales se buscó cómo mejorar la experiencia de los turistas.

3 Enfoque de la solución

Con el fin de facilitar a las personas el encontrar la información y adaptar sus viajes en torno a esta, se abordó la problemática de recuperar información relevante sobre restricciones de viaje y casos de COVID-19 mediante una aplicación web.
La aplicación permite visualizar los países del mundo y al seleccionar uno cualquiera se muestra la información disponible sobre el mismo.
El hecho de unificar, en una sola aplicación, distintas fuentes de datos y recuperar información de distintos tipos y en distintos formatos, agrega valor a la aplicación, que de lo contrario, se debería gastar mucho tiempo buscándola por separado.

4 Diseño

En esta sección comentaremos las tecnologías y componentes que fueron utilizados durante el desarrollo de la aplicación, así como el diagrama de la misma.
En la figura 1 se presenta una imagen de la arquitectura de la aplicación:

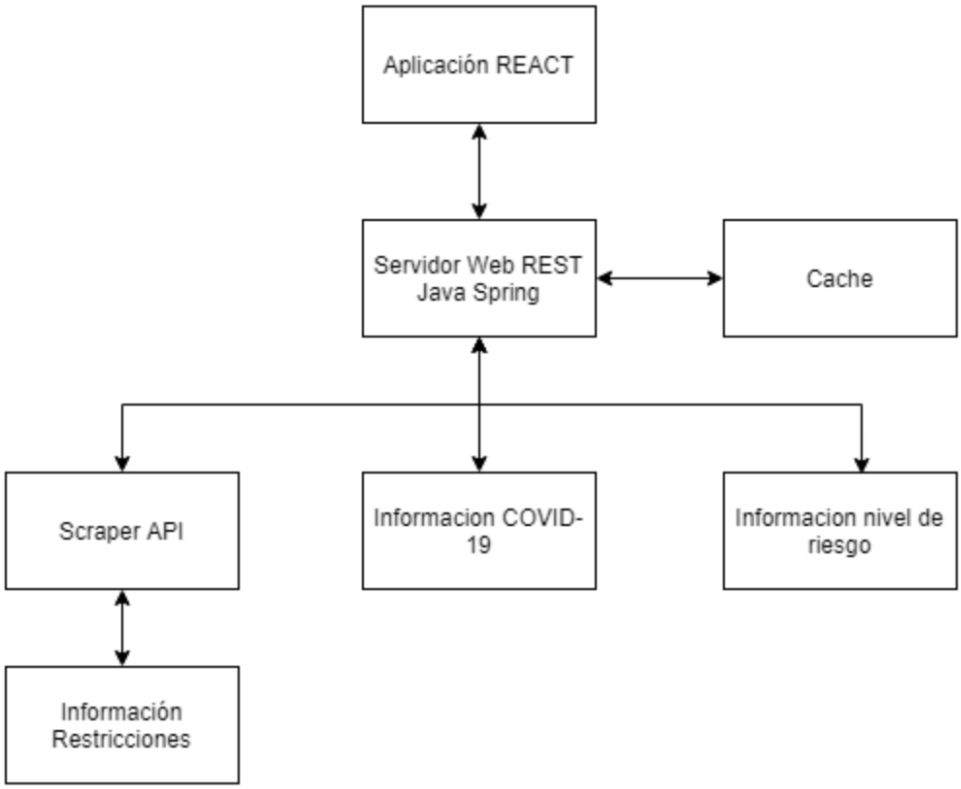

Figure 1: Arquitectura de la aplicación

4.1 Front end

Se dispone de una aplicación en React que obtiene los datos provistos mediante una API expuesta por nuestro Backend. En la app se utilizan distintas librerías para facilitar la implementación de ciertos subsistemas.
La App, como se muestra en la figura 2, cuenta con un mapa de colores que despliega información según el interés del usuario y la interacción del mismo con la aplicación. Este flujo será ampliado en la sección 6.

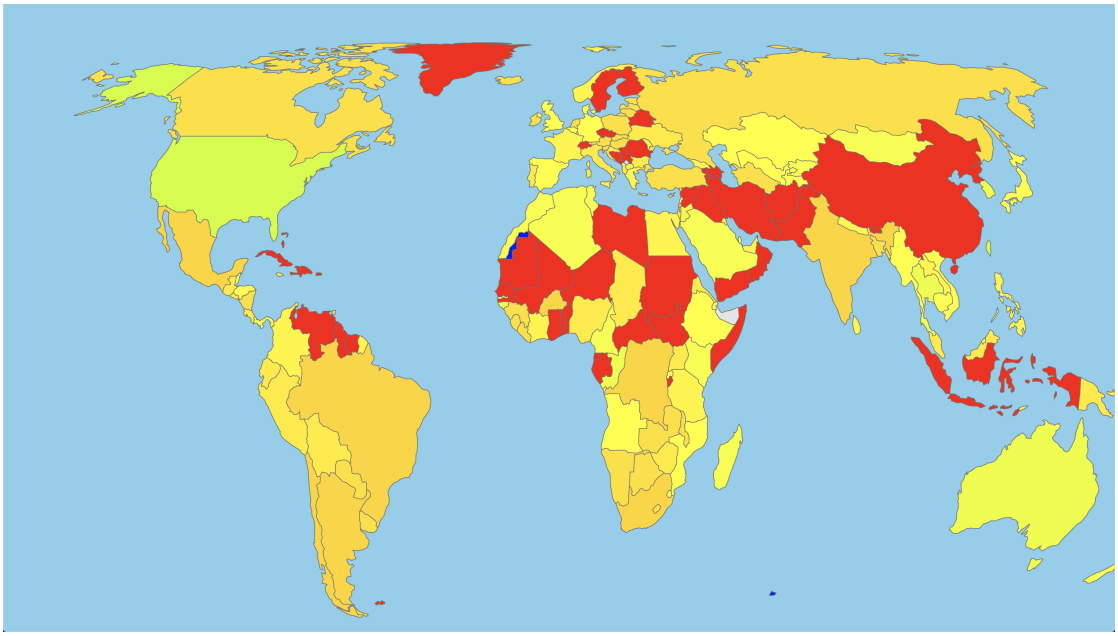

Figure 2: Captura de la Aplicación

4.2 Back end

Se implementó el servidor backend con Java 11 Spring Boot, ya que algunos miembros del equipo tienen mucha experiencia en el lenguaje. Spring Boot permite implementar servidores REST [4] de forma sencilla. De esta forma, se expusieron tres endpoints para acceder mediante pedidos HTTP para obtener la información deseada.
En cada pedido es necesario explicitar el código ISO del país del cual se pide la información. En la sección 5.1 se encuentra información ampliada acerca del uso del estándar ISO 3166.

4.3 Cache local Spring Boot

Se implementó una cache local en el servidor backend ya que brinda varios beneficios que se detallarán a continuación.
La Cache sirve para brindar los datos previamente accedidos, de manera de ahorrar pedidos a la API y mejorar la experiencia de usuario, ya que al ser una gran cantidad de datos, el tiempo que demora en responder, es elevado. Además es

relevante mencionar que muchas de las API's accedidas no son de acceso libre, por lo que la cantidad de pedidos a las mismas es limitado.

4.4 Scraper API

Se utilizó la API Scraper [1] de manera de obtener el código fuente HTML de la página de información de riesgos. Una vez obtenida se parseó la información relevante para ser almacenada en el backend y ser consultada directamente por la App. Fue necesario este mecanismo ya que se encontró información muy valiosa para el usuario acerca de los riesgos de viajar a cada país, pero dicha información no estaba disponible mediante una API o un archivo JSON público, por lo que se necesitó de esta herramienta auxiliar.

5 Implementación

Como se mencionó anteriormente, para lograr la recolección de información se utilizaron distintas API's:

- Para los datos de casos se utilizó una api de información de coronavirus que está disponible en: `https://about-corona.net/documentation`.
- Para la información acerca del nivel de riesgos de viaje se utilizó una API del gobierno de Estados Unidos con los criterios definidos por dicho país. Esta api brinda un JSON en la siguiente url: `https://www.travel-advisory.info/api`.
- Para la situación de frontera aérea de los países, se recolectó con Scraper API información de la siguiente fuente: `https://www.kayak.com/travel-restrictions`.

Para la metodología de trabajo, se decidió realizar en primera instancia un prototipo mínimo de la app. Una vez realizado este, se puso el esfuerzo en realizar el servidor backend explicado anteriormente, con el uso de las API's provistas y se utilizó el prototipo para probar su correcto funcionamiento.
La implementación terminó con la presentación los datos de manera limpia y prolija, y agregando features a la app de manera individual e incremental.

5.1 Estándares

Para la cohesión de datos y pensando en escalar la aplicación se decidió utilizar el estándar ISO 3166-1 alfa-2 [3]. Este ISO es utilizado por 2 de las API's utilizadas. Para la otra, se creó un mapeo que transforma la información de dicha API al estándar ISO 3166.
Se cree fundamental la utilización de dicho estándar ya que permite escalar de manera muy sencilla, nuevos países e integrar nuevas API's a futuro, en donde la utilización de ISOs es muy frecuente.

5.2 Desafíos enfrentados

En la realización de la aplicación, tanto en frontend como en backend, se sortearon varios desafíos.

El primero de ellos fue la realización de un backend que provea de datos a la aplicación web, teniendo como limitante la cantidad de pedidos disponibles a las API's. Las API's utilizadas cuentan con una versión gratuita, con una cantidad limitada de pedidos, por lo que fue necesario implementar una caché local. Como se mencionó anteriormente, esta caché también es utilizada para mejorar la UX del usuario final de la aplicación.

El segundo desafío fue la estandarización de códigos de países. Este problema fue encontrado al integrar la información de varias fuentes ya que no todas utilizaban el mismo formato para los pedidos. Fue necesario implementar un mapeo para estandarizar todos los códigos de países al formato ISO previamente mencionado.

Por ultimo, otro de los desafíos enfrentados de gran impacto fue el hecho de intentar mostrar una gran cantidad de información de manera ordenada y en diferentes pasos. Se cree que la forma de presentación de los datos es tan importante como los datos en sí. La experiencia de usuario deber ser buena para que una aplicación sea utilizada con frecuencia.
Esto se resolvió mostrando la información en 3 pasos o formas. Primero el mapa con colores da una idea fácil y rápida del estado de riesgo de cada país. Si el usuario esta interesado en tener mas información sobre uno en particular puede poner el

mouse arriba y activar un efecto *on hover* que le mostrará información básica. Si esta no es suficiente, puede hacer click y abrir la modal con la información completa. Cabe destacar que dicha modal también esta dividida en 2 pasos de forma de separar la información en texto puro, de la información visual.

6 Funcionalidades y uso

En esta sección se presentarán los distintos flujos y funcionalidades de la aplicación.

Al ingresar a la misma, se puede ver un mapa mundial con los diferentes paises con diferentes colores, como se mostró en la figura 2.

Al hacer hover, se muestra la información del nivel de riesgo con mas detalle. Esto se puede apreciar en la siguiente figura:

Figure 3: Hover Riesgo

La siguiente funcionalidad implica ver los datos específicos de un país. Al hacer click en un país se manda un pedido HTTP al backend para mostrar la información detallada del mismo. El evento del click abre una modal donde se puede ver dicha información.
Los detalles se dividen en dos secciones. La primera refiera información estadística del COVID-19. En esta sección se encuentra la cantidad de casos activos, los

casos confirmados y la cantidad de muestres. La segunda sección refiere a las restricciones impuestas por el gobierno de dicho país a la hora de viajar al territorio. En esta información también podrá verse si las fronteras del país se encuentran cerradas o no.
Al día 16/11/2020 Uruguay cuenta con fronteras cerradas, como se puede ver en el punto 3.

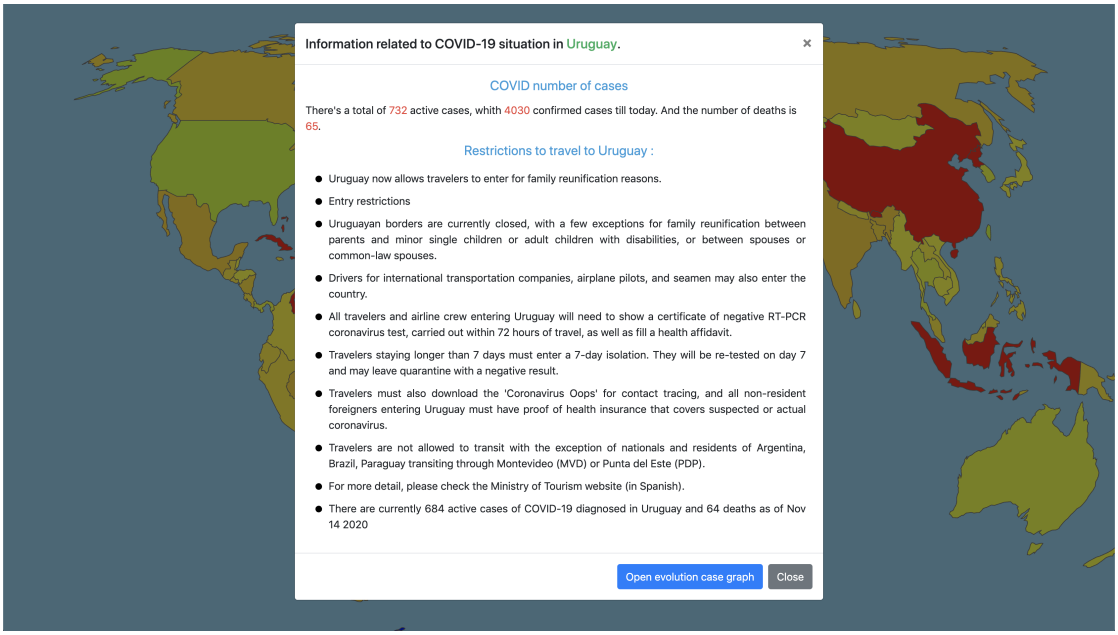

Figure 4: Modal con información detallada de riesgos y recomendaciones

Por último, también se considera que es importante no solo ver la fotografía actual a la hora de considerar un viaje, si no el panorama completo. Por esto mismo es que se cuenta con un botón en la Aplicación el cual abre una gráfica con la evolución de casos desde el comienzo de la pandemia mundial.

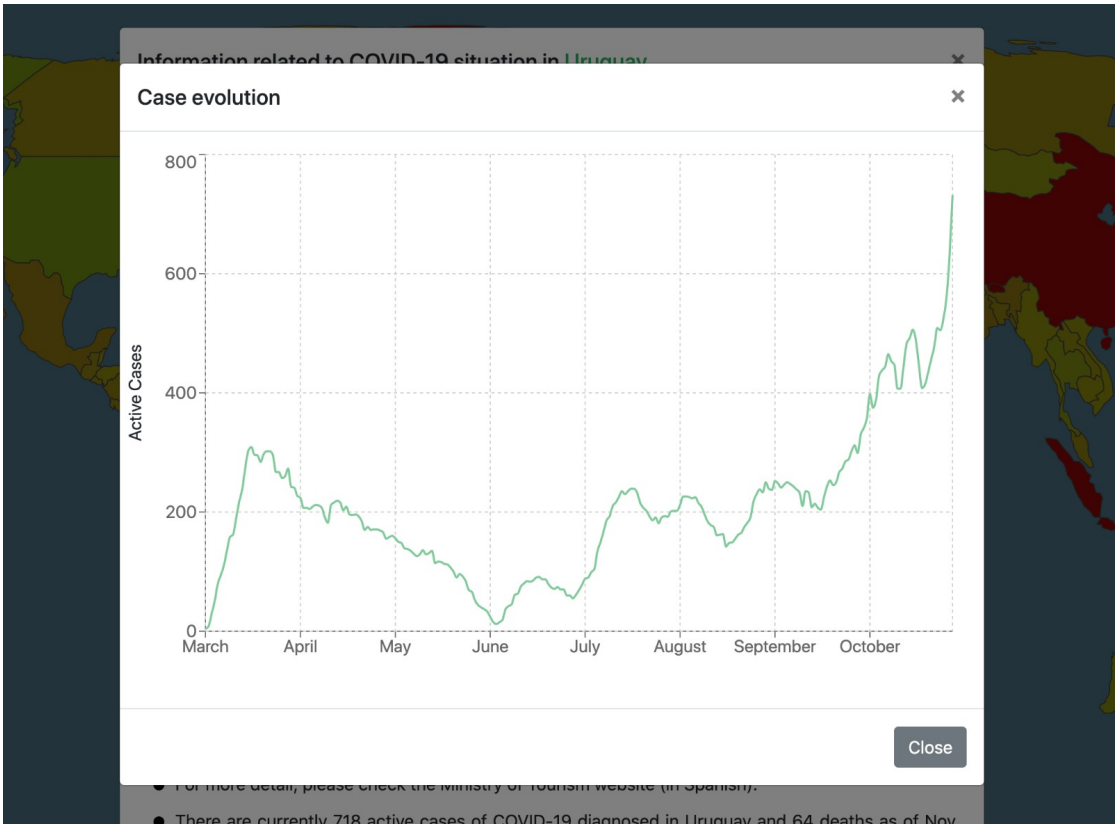

Figure 5: Gráfica con la evolución de casos

7 Trabajo Futuro

Como trabajo futuro se identificó fundamentalmente el hecho de integrar más API's de forma de aumentar los datos provistos al usuario para facilitar la decisión de viajar o no a un país.
En la misma línea se podría cambiar alguna de la API's utilizadas actualmente, por una versión paga o de fuentes oficiales. Como ejemplo concreto la API que brinda información acerca del coronavirus podría ser sustituida por la que provee la OMS (Organización mundial de la salud), la cual es paga.

Otro aspecto que puede mejorar a la aplicación es el hecho de integrarse con una aplicación como Booking [2]. De esta forma se podrá ver los precios de vuelos y alojamientos, al mismo tiempo que el riesgo y recomendaciones para el viaje.

8 Resultados y Conclusiones

Como conclusiones generales se considera que los resultados obtenidos cumplieron con el problema presentado en primera instancia. Si bien existen cosas a mejorar, se pudo realizar todo lo que estaba planificado y diseñar una aplicación que sea amena y fácil de usar para los usuarios. También destacar que no es un problema trivial el hecho de mostrar gran cantidad de información de manera organizada y se cree que el resultado obtenido es muy bueno.

Como conclusiones individuales, creemos que la realización de este curso y en particular de este trabajo, aportó conocimientos de gran valor en el área profesional. Se trataron tecnologías y conceptos nuevos que pueden ser de gran utilidad en un futuro.

References

[1] Scrapper API. Scrapper API. https://www.scraperapi.com/documentation. Último acceso: Noviembre 2020.

[2] Booking. Vuelos, Hoteles y Alojamientos. https://www.booking.com. Último acceso: Noviembre 2020.

[3] International Organization for Standardization. ISO 3166. https://www.iso.org/iso-3166-country-codes.html. Último acceso: Noviembre 2020.

[4] Wikipedia. Información acerca de REST api. https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional. Último acceso: Noviembre 2020.