**Explicación de técnicas:**

Durante este desafío técnico utilice las técnicas o pruebas que ya todos conocemos que a pesar de ser populares no carecen de potencial. El enfoque durante el desarrollo de pruebas para el desafío técnico fue:  
Pruebas unitarias

* Pruebas de integración
* Pruebas de sistema
* Pruebas de caja negra
* Test de aceptación

**Ejercicio 1:**

Durante el primer ejercicio me enfoque en pruebas de aceptación y de caja negra, debido a que se el mismo ejercicio un planteamiento teórico resulta que solo tendremos parámetros de aceptación y resultados esperados. El resto lo podemos interpretar como un error o como algo indefinido, un ejemplo de esto puede ser la consulta de una fecha de un año pasado o futuro. Al ser un planteamiento teórico mi enfoque fue en los casos bordes para la fechas que se puedan ingresar y en la combinación posible en los campos de Country y City.

Para el caso de las fechas me pregunte como seria el comportamiento de la API en función de los casos bordes, como es siempre para fechas debemos pensar en el fin de un mes, de un año o el caso especial de un día extra en febrero (año bisiesto). Pasando los casos de borde intente que la información traída por la API fuera consistente al consultar un día y al siguiente esperando ver valores iguales. Finalmente termine la prueba sobre la fecha validando el formato de entrada, aplicando el uso correcto y un caso de uso incorrecto.

Para los campos de Country y City la prueba fue más fácil de implementar debido a que los valores correctos ya están definidos, solo debía hacer las posibles combinaciones posibles aplicando el test de aceptación. De igual forma con Case Sensitive del campo y verificar que otro posible valor es un error de ciudad o país.

Finalmente para darle fin a la primera parte del desafío técnico pensé en cómo reproducir los errores más conocidos al probar una API. Los errores 404, 503 0 400 por solo mencionar, así de esta forma le doy atención a su característica de servicio web

**Ejercicio 2:**El implementar las pruebas sobre un caso práctico fue más libre debido a que el resultado era más palpable de una forma. Me enfoque en pruebas de integración para el API post, test de aceptación para API user y API albums,

Para realizar las pruebas empecé por la parte funcional, realizar un request get y ver el body del response, luego preguntarme cómo sería un criterio de aceptación sobre la API. Luego pasé a pensar cómo podría aprovechar las ventajas que me daba Postman como herramienta de prueba. Finalmente resultó en casos de pruebas donde hago validaciones sobre los campos como puede ser el Email de un usuario. Los campos requeridos para la creación de un usuario como puede ser el nombre, celular o dirección.

Debido a la simplicidad de las API expuestas sobre la carpeta de Albums sólo se hallan pruebas de validación y pruebas funcionales sobre una API. Sobre la carpeta Users verán pruebas de aceptación sobre un usuario y una prueba de validación de datos entre el response body de la API contra un csv cargado (adjunto en la carpeta del github). Así de esta forma puedo simular que la información fue cargada o migrada de forma ok con la herramienta de postman, o simular la ejecución de varios casos de pruebas variando las variables de entrada de mi API.

Finalmente para las pruebas sobre API post son de integración, dado que simuló una creación por el proceso POST, consultar la creación de un nuevo registro, un update por el proceso PUT y una baja o eliminación con proceso de DELETE. Pude agregar casos sobre el status code del PUT y DELETE.

Así finalizó el desafío técnico sobre pruebas en un framework el cual conocía pero no había utilizado llegado a exprimir a su totalidad, sin dudas se pueden realizar pruebas automatizadas y flujos de prueba sobre esta herramienta.