**Reto técnico pruebas de servicio con Postman**

**Primera parte**

Valentin Despa es un desarrollador de software que está trabajando en un proyecto, en el cual hace un servicio Rest, para poder ver libros disponible, así como poder hacer ordenes de libros, así como actualización de estas órdenes, y hacer un borrado de una orden en caso de ser necesario. Por este motivo después de terminado el desarrollo del servicio el desarrollador quiere probar el correcto funcionamiento de este y para esto contrata a la empresa Choucair Testing.

Por este motivo el desarrollador le proporciona a Choucair Testing la documentación de la Api en el siguiente link: <https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md>.

Ya después de haber leído la documentación, con el cliente se llega a un acuerdo de que las pruebas realizadas se harán en Postman, y que deben estar organizadas en colecciones que nos permitan ver de mejor manera que endpoint se está probando. Por otra parte lo que el cliente espera que tenga nuestras pruebas en la herramienta de Postman es lo siguiente:

1. Se debe crear un ambiente de pruebas para hacer la ejecución de nuestras pruebas
2. También se hace necesario el uso de variables desde el ambiente creado.
3. Poder hacer un crud con cada uno de los endpoints del servicio (incluye probar caminos felices y caminos no felices)
4. Cada una de las validaciones que se hacen deben estar respaldadas con aserciones que lo justifiquen, no solo validaciones de código de respuesta.

**Segunda parte**

Para esta segunda vamos a trabajar sobre la siguiente Api <https://rickandmortyapi.com/documentation> , la cual es muy usada para hacer pruebas de servicio, para este ejercicio se plantea hacer solicitudes sobre el endpoint de “Character” y se espera puedan hacer lo siguiente con este endpoint:

1. Hacer un listado de todos los personajes, y hacer un test con la información de un personaje que aparezca en esa respuesta (Tener en cuenta que deben hacer una búsqueda entre los diferentes objetos que hay embebidos en la respuesta)
2. Hacer un listado de personajes por página (en la documentación se explica cómo hacer paginación), y hacer un test por aparición de personaje en esta página.
3. Traer un personaje por id, y hacer un test correspondiente.
4. Al momento de buscar múltiples Id, ingresando id’s que no existen nos da una respuesta, la cual es un array vacío, hacer un test en donde se valide que se tiene una respuesta vacía