| **LABORATORIO 10 -** Arquitectura de Software |
| --- |
| **TEAM:** |
| **Proyecto:** Bot - MonitorMach |
| El bot no tiene ningún componente de AI solo interpreta comandos pre-definidos. |
| **Bot - Monitor Mach**   |  | **Comandos:**  **CheckLatency**  **<module> <start-date> <end-date>**  **CheckAvailability**  **<module> -[Last5Days, Last7Days]**  **RenderGraph -[Availability, Latency}**  **<module> -[Last5Days, Last7Days]** | | --- | --- | |
| **Parte I ( 5 ptos )**   * Implementar el siguiente diseño:     **(\*) Lenguaje libre ( Solo se debe implementar backend )**  **(\*) Cada Microservicio/Modulo debe tener su propio manejo de logs.**  **(\*\*) Los logs deben tener una sintaxis o nomenclatura estandar para todos los microservicios. {Fecha}{Modulo}{API}{Funcion} Message**  **(\*\*) Los logs deben identificar claramente tiempo y a que microservicio pertenecen.**  **(\*\*) Para medir la latencia asumir que debe medir el tiempo de inicio y fin de cada bloque de codigo.**  **URLS de Fuente de Datos:**  **POKE\_API: https://pokeapi.co/api/v2/pokemon/{id or name}/**  **POKE\_STATS:** [**https://www.kaggle.com/datasets/abcsds/pokemon/code**](https://www.kaggle.com/datasets/abcsds/pokemon/code)  **POKE\_IMG:** [**https://www.kaggle.com/datasets/hlrhegemony/pokemon-image-dataset**](https://www.kaggle.com/datasets/hlrhegemony/pokemon-image-dataset) |
| **Entregable:**  **Repo:**  **JMeter(jmx) o Locust:**  **(\*) Este jmeter debe tener un minimo de 1000 llamadas y un maximo de 10000 llamadas. Esto es necesario para generar los logs.**  **Screenshot de Response:**   |  | | --- | |
| **Parte II ( 5 ptos )** |
| * Implementar comandos del Bot.   + **CheckLatency:** Muestra la latencia de la aplicación de un modulo en un periodo determinado.   + **CheckAvailaibilty:** Muestra la disponibilidad del servicio durante X días.     - **Availability =** Tasa de Error / (Tasa de Éxito + Tasa de Error)   **Tasa de Éxito =** Número de Requests con HTTP Code 200.  **Tasa de Fallo =** Número de Requests con HTTP Code 500.   * **RenderGraph**: Renderiza una grafica lineal con las tendencias de disponibilidad/latencia para un modulo en un periodo de X dias.   (\*) El diagrama puede ser en consola. ASCII ART:  <=======]}======  --. /|  \_\"/\_.'/  .'.\_.\_,.'  :/ \{}/  (L /--',----.\_  | \\  : /-\ .'-'\ / |  snd \\, || \|  \/ || || |
| **Use Cases:**  **CheckAvailability PokeStats -Last5Days**  01/10 99.9%  02/10 89.9%  03/10 89.8%  04/10 90.5%  05/10 87.8%  **CheckLatency PokeImage -01/10 -03/10**  01/10 3000ms  02/10 4000ms  03/10 1500ms  **RenderGraph -Latency -Last3Days**  \*\*4000\*\*  \*\*3000\*\*  \*\*1500\*\*  01/10 02/10 03/10 |
| **Entregables:**  **Repo: <código de bot>**  **Grabar un video haciendo una Demo del Uso de su Bot** |