

1. Solución Prueba Técnica Especialista Programabilidad

A continuación, se presentan los pasos, scripts y soluciones implementadas para dar solución a los requerimientos descritos en el documento “*Prueba Técnica Especialista Programabilidad*”, recibido dentro del proceso de selección de proceso Especialista Programabilidad

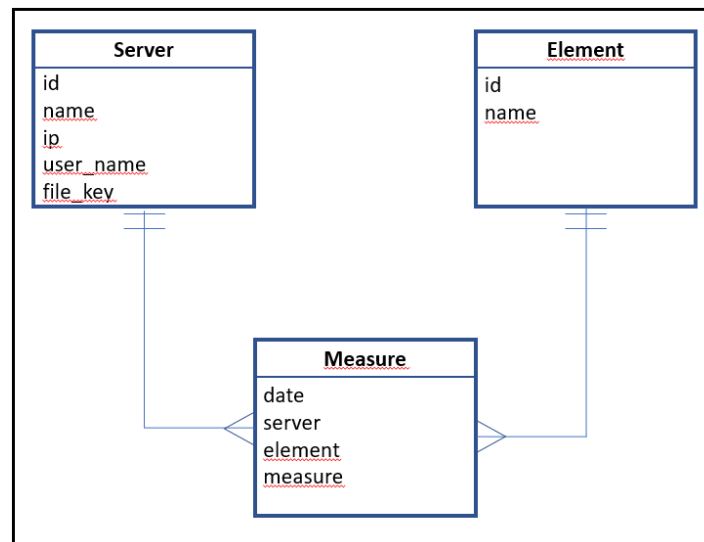
1.1. Creación de servidores

Usando el servicio “Amazon Lightsail”, se crean dos servidores Linux en los cuales se simulará la granja de servidores y servirán para ejecutar los scripts desarrollados dando cumplimiento a los requerimientos indicados en el documento.



1.2. Definición modelo de datos

A continuación, se presenta el modelo de datos donde se almacenará la configuración de los servidores y los datos capturados para cada uno de los elementos (CPU – Memoria - Disco), en los momentos en los que se ejecute el script. La ejecución del script se puede programar para que se ejecuta con la periodicidad requerida con la herramienta “*cron*” o “*crontab*” de Linux.



1.3. Script ([main_load_info.py](#))

El script “*main_load_info.py*”, obtiene via ssh la información del estado actual de los servidores para cada uno de los elementos definidos.

El script carga los servidores que estén definidos en la tabla “SERVER”, obteniendo la dirección ip y el nombre del archivo que contiene la llave pública de conexión. Para establecer la conexión, se hace uso de la librería “*paramiko*” y almacena la información en la tabla “MEASURE”.

Con la librería “*telegram*”, se usa el bot definido en Telegram al cual se enviarán los mensajes en caso de superar los umbrales del 75%.

1.4. Script ([mainExpone.py](#))

El script “*main_Expone.py*”, carga la información de la tabla “MEASURE” y la expone mediante un método POST, para que sea consumido por las aplicaciones que lo requiera. Para el caso del presente ejercicio, es consumido por la aplicación web que grafica las mediciones efectuadas por el script *main_load_info.py*

Para su ejecución se requieren los frameworks “FastAPI y uvicorn”.

1.5. Aplicación web

El archivo “**ExampleTIGO.zip**”, contiene los archivos fuente de la aplicación web, la cual usa el framework Django, para consumir el Api expuesto por el script *mainExpone.py*.

Para el presente ejercicio se desplegó la aplicación en el servidor: <http://3.227.170.105:8001/>, el cual presenta en modo gráfico la información para los servidores definidos.