

Prueba Técnica:

1. Dadas las bases de datos **ejercicio_b1.xlsx** Hoja1 y **ejercicio_b2.txt**, desarrolle un API web, con Python y Flask en la que se exponga un servicio get que realice el cruce de estas dos bases de datos alojadas en una ruta específica parametrizable mediante el archivo de configuración; para producir una tabla con las siguientes columnas y características.

Nombre	Apellido	Cedula	Nacimiento	Nombre Completo	Edad	Tipo de pedido	Numero de pedido
Nombre de la persona en mayusculas	Apellido de la persona en mayusculas	Valor númerico	Fecha en formato separado por "/"	Nombre y apellido de la persona solo con la primera letra en mayuscula ej: "Andres Perez"	Valor entero positivo en años entre la fecha de nacimiento y la fecha 25/11/2014	Valor estandarizado en mayusculas de los distintos tipos de pedido	Total de pedidos hechos por el usuario, del mismo tipo de pedido

2. Desarrolle un entorno de FrontEnd con un layout que contenga los siguientes componentes:
 - a. El logo de la compañía CALA-Analytics (disponible en nuestra página web oficial)
 - b. Un label con la ruta donde se encuentran alojados los archivos (este label debe ser dinámico y cambiar cada vez que se modifique la ruta de los archivos en el archivo de configuración del API)
 - c. Un botón "Mostrar Pedidos"
 - d. Un botón "Mostrar Análisis"
3. Realice la interacción del botón "Mostrar Pedidos", de tal manera que al pulsar sobre él se muestre en pantalla la tabla desarrollada en el punto uno (consumiendo el API), dicha tabla debe mostrarse al usuario con la posibilidad de ordenar, filtrar y buscar por cada uno de los campos, la tabla debe estar paginada a un máximo de 10 registros por página.
4. Realice la interacción del botón "Mostrar Análisis", de tal manera que al pulsar sobre él se muestre un gráfico de barras de los nombres completos vs el Numero de pedido, discriminado por tipo de pedido.
5. Genere un documento con las evidencias (imágenes del aplicativo ejecutado en local) donde se pueda visualizar el layout, y las interacciones de los botones.
6. Relate los pasos, y desarrolle un cronograma de actividades mediante los cuales se pueda poner en producción este aplicativo sobre un entorno de AWS sea lo más detallado posible.

Consideraciones generales:

1. Se espera el desarrollo por un lado de una aplicación OnPremise bajo la arquitectura Cliente servidor o Microservicios.
2. El lenguaje debe ser Python y Flask para el BackEnd en cuanto al FrontEnd trabaje con las tecnologías de su preferencia.
3. El layout del FrontEnd debe contar con los elementos descritos en el paso dos, en cuanto al orden, tamaño y características graficas de los componentes se dejan a su criterio y creatividad.
4. Las librerías para procesamiento de datos y fabricación de gráficos se dejan a su criterio.

5. Se debe subir esta información a un repositorio GitLab o GitHub público con una rama master y una develop, junto con un ReadMe con los pasos para realizar el correspondiente despliegue en local. Dentro del histórico del repositorio deben verse varios commit y pull request desde develop y hacia master.
6. Con respecto al ReadMe del repo, procure ser lo más detallado posible.