



Prueba Técnica Practicante Analista de Datos

Instrucciones:

- Realizar el proceso de limpieza y depuración de la información en un lenguaje de programación o consulta de Base de datos que prefiera. (Python o SQL)
- Para el desarrollo de los análisis de reporting y/o dashboard que realice, utilizar herramientas de preferencia: Power BI o Tableau (Public).
- La duración de la prueba es de 48 horas a partir de la fecha de envío.
- Las respuestas podrán ser enviadas en un PDF, un notebook, o con un link a tu repositorio de GitHub.
- Debe presentar los scripts y documentos que realice en el análisis del ejercicio



Manejo de base de datos.

Dado el siguiente diseño de base de datos de una empresa de regalos.

Novios <u>codigo_casamiento</u> nombre_novio apellido_novio	Regalos <u>codigo_articulo</u> <u>codigo_casamiento</u> cantidad_requerida regalado? (S/N)
Casamiento <u>codigo_casamiento</u> fecha comentario	Ordenes_envio <u>codigo_envio</u> <u>codigo_casamiento</u> fecha_envio
Articulos <u>codigo_articulo</u> precio cantidad_stock nombre_articulo	Detalle_envio <u>codigo_envio</u> codigo_articulo cantidad_enviada

Escribir las instrucciones de SQL que permitan resolver las consultas detalladas más abajo.

Observaciones:

1. **Sí** se pueden utilizar funciones genéricas del tipo “EDAD(fechanacimiento)”, “AÑO(fecha)”, “MES(fecha)”.

Sugerencia: Para el desarrollo de este ejercicio, no es necesario crear una base de datos y complementar información, el objetivo del ejercicio es que a través de las relaciones plantee las consultas apropiadas para responder las preguntas. El crear una base de datos le permitirá validar las consultas creadas.

Preguntas:

- a) Para cada casamiento con fecha posterior al 2022-01-01, el nombre de los artículos pedidos, la cantidad requerida y el indicador si fue regalado o no para los artículos con menos de 10 unidades en stock.
- b) Los nombres de los artículos cuya cantidad total requerida pendiente de regalar (regalado = N)



es mayor a 100 y que tiene un precio mayor a \$500.

- c) Para cada casamiento desde el 2021-06-01, la cantidad de envíos.
- d) EL nombre del artículo de menor Stock



Lógica de Programación.

Responder las siguientes preguntas de entendimiento de lógica de programación, no es necesario escribir código en lenguaje, salvo que se diga lo contrario.

- a. Sea la siguiente función, cuya entrada es una lista de números enteros:

```
def func (list_):  
    i = 0  
    while i < len (list_) and list_[ i ] == 0 :  
        i = i +1  
    a = i  
    while i < len (list_) and list_[ i ] == 1 :  
        i = i + 1  
    return i == len(list_) and 2*a == i
```

Describir en lenguaje cotidiano cuál es el resultado de aplicar está función sobre una lista (es decir, explicar para qué entradas devuelve True y para qué entradas devuelve False.

- b. **¡Sin usar librerías!** Construya una función recursiva que reciba como parámetros dos enteros positivos p y q, donde p representa el numerador y q el denominador de la fracción p/q (q != 0). La función debe retornar la fracción en su mínima expresión.

Ejemplo p = 85 y q = 45, la respuesta a imprimir es **17/9**

Tip: Considere que existe una función **verificar_primo(x)** que retorna True si el número x es primo, o False en caso contrario.

- c. **¡Sin usar librerías!** Construya una función en python que solucione el siguiente ejercicio:

Definimos super numero de un número entero positivo dadas las siguientes reglas:

1. Dado un número entero, tenemos que encontrar el súper número de ese número entero.
2. Si el número solo tiene un dígito, ese es el súper número.
3. En caso contrario, entonces su super número es igual al super número de la suma de los dígitos.

Por ejemplo si el numero es P= 148148148, su super numero es:

```
super_numero(P) = super_numero(148148148)  
                 = super_numero(1+4+8+1+4+8+1+4+8)  
                 = super_numero(39)  
                 = super_numero(3+9)  
                 = super_numero(12)  
                 = super_numero(1+2)  
                 = super_numero(3) = 3
```



Manejo de Análisis de Datos.

FIFA 2019

Los jugadores de fútbol tienen muchas habilidades diferentes. Para cada posición, se necesitan habilidades diferentes en comparación con otras posiciones. Para ello se desea analizar los conjuntos de habilidades clave más necesarios para la posición de cada jugador. Para este caso, se trabajará con el conjunto de datos completo de jugadores de FIFA 19 de Kaggle. (Dataset Adjunto)

Para ello se desea realizar el siguiente análisis:

- a. **Data Cleaning;** Realizar un proceso de limpieza del dataset, el cual permita hacer análisis exploratorios y estadísticos.
- b. **Exploratory Analysis:** Realizar un análisis univariado y bivariado que permite identificar:
 - i. Jugadores con características atípicas dentro de la muestra con respecto a salarios o edad, etc.
 - ii. Identificar el top 5 de jugadores de mejores salarios, valoración general y potencial.
 - iii. Realizar un comparativo entre clubes de la distribución de salarios y edad de los jugadores
 - iv. Ver en general si existe una correlación entre la reputación internacional, el valor del mercado actual y la cláusula de liberación (value, international reputation, release clause).
 - v. Realizar un exploratorio de las características de un jugador con base en su posición. (<https://fifauteam.com/fifa-21-positions/>) **nota:** Puede ser útil recategorizar las posiciones para facilitar el análisis.
- c. **Modelling:** Construir una segmentación de jugadores con base en sus características para entender su posición. Realice un comparativo entre los hallazgos encontrados y el exploratorio realizado en el literal anterior.