

## PRACTICAS PROCEDIMIENTO Y FUNCIONES

Un número primo es aquel que sólo es divisible por el mismo y la unidad (1).

1. Realiza una función llamada `EsPrimo` que determine si un número que le pasamos como argumento es primo (TRUE) o no (FALSE).
2. Utilizando la función anterior, crea un procedimiento (`p_primo`) que pasándole un número como parámetro muestre todos los números primos menores o iguales a dicho número.
3. Modifica el procedimiento anterior (`p_primo2`) para que los número primos los guarde en una tabla llamada `primo` que solo tendrá un campo llamado número de tipo entero. Antes de insertar debe vaciar la tabla. El procedimiento debe mostrar el contenido de la tabla antes de salir.
4. Crea una función llamada `Factorial` que dado un número de entrada devuelva su factorial: Ejemplo:  $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$ .

Realice los siguientes procedimientos y funciones sobre la base de datos `jardineria`.

5. Crea una Función: `calcular_precio_total_pedido` que dado un código de pedido la función debe calcular la suma total del pedido. Tenga en cuenta que un pedido puede contener varios productos diferentes y varias cantidades de cada producto. La función recibirá como parámetros de entrada el `codigo_pedido` (INT) y devolverá el precio total del pedido (FLOAT).
6. Crea una función `calcular_suma_pedidos_cliente` que dado un código de cliente la función debe calcular el precio total del pedido cuyo código de pedido sea el menor, es decir, de todos los pedidos que ha hecho ese cliente el del código más pequeño. Deberá hacer uso de la función `calcular_precio_total_pedido` del apartado anterior. Recibirá como argumento el `codigo_cliente` (INT) y devolverá la suma total de todos los pedidos del cliente (FLOAT).
7. Crea una función `calcular_suma_pagos_cliente` que dado un código de cliente la función debe calcular la suma total de los pagos realizados por ese cliente. Recibirá como argumento el `codigo_cliente` (INT) y devolverá la suma total de todos los pagos del cliente (FLOAT).
8. Crea un procedimiento `calcular_pagos_pendientes` que deberá calcular los pagos pendientes de un cliente. Para saber si un cliente tiene algún pago pendiente deberemos calcular cuál es la cantidad de todos los pedidos y los pagos que ha realizado. Si la cantidad de los pedidos es mayor que la de los pagos entonces ese cliente tiene pagos pendientes. Se deberá insertar en una tabla llamada `clientes_con_pagos_pendientes` los siguientes datos:
  1. `id_cliente`
  2. `suma_total_pedidos`

3. suma\_total\_pagos
4. pendiente\_de\_pago

9. Escribe un **procedimiento** llamado obtener\_numero\_empleados que reciba como parámetro de entrada el código de una oficina y devuelva el número de empleados que tiene.
  - Escribe la sentencia SQL que realiza la llamada al procedimiento realizado para comprobar que se ejecuta correctamente.
10. Escribe una **función** llamada cantidad\_total\_de\_productos\_vendidos que reciba como parámetro de entrada el código de un producto y devuelva la cantidad total de productos que se han vendido con ese código.
  - Escriba una sentencia SQL que realice una llamada a la función realizada para comprobar que se ejecuta correctamente.