

Escriba los programas en **PLSQL** y construya los procedimientos, funciones y paquetes. Salve cada script de creación con el mismo nombre del objeto que va a crear.

```
SQL> set echo on
SQL> set serveroutput on
SQL> spool lab2.txt
SQL> start terna
SQL> start esprimo
SQL> start goldbach
SQL> start setprimo
SQL> start demuestre
SQL> start gold
SQL> spool off
```

Conjetura de Golbach

En [teoría de números](#), la **conjetura de Goldbach** es uno de los [problemas abiertos](#) más antiguos en [matemáticas](#). A veces se le califica del problema más difícil en la historia de esta ciencia. Concretamente, [G.H. Hardy](#) en [1921](#) en su famoso discurso pronunciado en la Sociedad Matemática de Copenhage¹ comentó que probablemente la conjetura de Goldbach no es sólo uno de los problemas no resueltos más difíciles de la teoría de números, sino de todas las matemáticas. Su enunciado es el siguiente:

Todo número par mayor que 2 puede escribirse como la suma de dos números primos.

[Christian Goldbach](#) (1742)

BASE DE DATOS

1. Haga una tabla PRIMOS con una columna VALOR number(10). Inserte los valores 2,3,5 y 7.
2. Construya el tipo de datos TERNA

create or replace type terna as object

(iteracion number(5), primo1 number, primo2 number)

FUNCIONES (30 pts)

EsPrimo(n): Retorna TRUE si n es primo, FALSE si no lo es.

Goldbach(n): Recibe un número entero y retorna el objeto TERNA (con 3 números)

Examina si se cumple la Conjetura de Goldbach con n.

Si n es impar, retorna la terna con valores NULL.

Si n es par, y no se cumple la conjetura de Golbach, retorna ceros en primo1, primo2 y la cantidad de iteraciones (número de veces que entra al ciclo).

Si n es par y se cumple la conjetura, retorna los primeros dos números primos que suman ese número y las iteraciones.

PROCEDIMIENTOS (40 pts)

SetPrimo(n): Inserta en la tabla PRIMOS todos los números primos que se encuentren: desde el último que está en la tabla, hasta **n**. Despliega en una secuencia textual, los números primos insertados (utilice la función EsPrimo)

Demuestre(n): Si **n** es impar,

Si **n** es primo, busque su cardinalidad (posición de número primo)

Si **n** no es primo, busque 2 múltiplos que lo producen

Si **n** es par mayor a 2, realice la prueba de Golbach(**n**) y obtenga la terna

Despliegue un mensaje con **n**, lo que está buscando y el resultado.

Por ejemplo:

demuestre(8)

Resultado: 8 es un número par resultado de la suma de los números primos 3+5 en 1 iteraciones

Demuestre(11)

Resultado: 11 es el número primo en la posición: 5

Demuestre(15)

Resultado: 15 es un número impar producto de 3*5

PAQUETES (10 pts)

Haga un Application Package llamado **gold** que contenga: Terna, Goldbach(), setPrimo(), Demuestre().

REPORTE (20 pts)

Haga un spool a "gold.txt" y Ejecute:

gold.Setprimo(15)	gold.Setprimo(102)	gold.Setprimo(1000)
gold.Demuestre(696)	gold.Demuestre(878)	gold.Demuestre(983)
gold.Demuestre(999)	gold.Demuestre(1001)	

Envíe:

- spool **lab2.txt**: con la creación del objeto, funciones, procedimientos y paquete
- spool **gold.txt**: con las ejecuciones de prueba