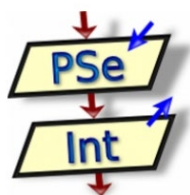


PROGRAMACIÓN DESDE CERO

# SUBPROGRAMAS EN PSEINT



Universidad de  
**LA PUNTA**



GOBIERNO DE  
**SAN LUIS**



**EGG**

## EJERCICIOS PRÁCTICOS

Para cada uno de los siguientes ejercicios realizar el análisis del problema e indicar cuáles son los datos de entrada y cuáles son los datos de salida. Escribir luego el programa en PSeInt utilizando funciones y/o procedimientos.



**VIDEOS:** Te sugerimos ver los videos relacionados con este tema, antes de empezar los ejercicios, los podrás encontrar en tu aula virtual o en nuestro canal de YouTube.

### Subprogramas: Funciones

1. Realizar una función que calcule la suma de dos números. En el algoritmo principal le pediremos al usuario los dos números para pasárselos a la función. Después la función calculará la suma y lo devolverá para imprimirlo en el algoritmo.
2. Realizar una función que valide si un número es impar o no. Si es impar la función debe devolver un verdadero, si no es impar debe devolver falso. **Nota:** la función no debe tener mensajes que digan si es par o no, eso debe pasar en el Algoritmo.
3. Crea una función *EsMultiplo* que reciba los dos números pasados por el usuario, validando que el primer número múltiplo del segundo y devuelva verdadero si el primer número es múltiplo del segundo, sino es múltiplo que devuelva falso.
4. Realizar un programa que pida al usuario una frase y una letra a buscar en esa frase. La función debe devolver la cantidad de veces que encontró la letra. **Nota:** recordar el uso de la función `Subcadena()`.
5. Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número es primo o no. Un número es primo cuando es divisible sólo por 1 y por sí mismo, por ejemplo: 2, 3, 5, 7, 11, 13, 17, etc. **Nota:** recordar el uso del MOD.
6. Realizar una función que calcule y retorne la suma de todos los divisores del número n distintos de n. El valor de n debe ser ingresado por el usuario.
7. Diseñar una función que reciba un numero en forma de cadena y lo devuelva como numero entero. El programa podrá recibir números de hasta 3 dígitos. **Nota:** no poner números con decimales ni letras. Ejemplo: ingresando "100"(carácter) debe convertirse en 100(entero).
8. Crear una función llamada "Login", que recibe un nombre de usuario y una contraseña y que devuelve Verdadero si el nombre de usuario es "usuario1" y si la contraseña es "asdasd". Además, la función calcula el número de intentos que se ha usado para loguearse, tenemos solo 3 intentos, si nos quedamos sin intentos la función devolverá Falso.
9. Los empleados de una fábrica trabajan en dos turnos: Diurno y Nocturno. Se desea calcular el jornal diario de acuerdo con las siguientes reglas:
  - a) La tarifa de las horas diurnas es de \$ 90
  - b) La tarifa de las horas nocturnas es de \$ 125
  - c) En caso de ser feriado, la tarifa se incrementa en un 10% si el turno es diurno y en un 15% si el turno es nocturno.

El programa debe solicitar la siguiente información al usuario: el nombre del trabajador, el día de la semana, el turno (diurno o nocturno) y la cantidad de horas trabajadas. Además,

debemos preguntarle al usuario si el día de la semana (lunes, martes, miércoles, etc.) era festivo o no, para poder calcular el jornal diario. Utilice una función para realizar el cálculo.

10. Realizar una función que calcule la suma de los dígitos de un número.

**Ejemplo:**  $25 = 2 + 5 = 7$

**Nota:** Para obtener el último número de un dígito de 2 cifras o más debemos pensar en el resto de una división entre 10. Recordar el uso de la función Mod y Trunc.

11. Realizar una función que reciba un número ingresado por el usuario y averigüe si el número tiene todos sus dígitos impares (ejemplo: 333, 55, etc.). Para esto vamos a tener que separar el número en partes (si es un número de más de un dígito) y ver si cada número es par o impar. **Nota:** recordar el uso de la función Mod y Trunc(). No podemos pasar el número a cadena para realizar el ejercicio.

12. Realizar una función que permita obtener el término  $n$  de la sucesión de Fibonacci. La sucesión de Fibonacci es la sucesión de los siguientes números:

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Donde cada uno de los números se calcula sumando los dos anteriores a él. Por ejemplo:

La sucesión del número 2 se calcula sumando  $(1+1)$

Análogamente, la sucesión del número 3 es  $(1+2)$ ,

Y la del 5 es  $(2+3)$ ,

Y así sucesivamente...

La sucesión de Fibonacci se puede formalizar de acuerdo a la siguiente fórmula:

$\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$  para todo  $n > 1$

$\text{Fibonacci}(n) = 1$  para todo  $n \leq 1$

Por lo tanto, si queremos calcular el término " $n$ " debemos escribir una función que reciba como argumento el valor de " $n$ " y que calcule la serie hasta llegar a ese valor.

Para conocer más acerca de la serie de Fibonacci consultar el siguiente link:

<https://quantdare.com/numeros-de-fibonacci/>

13. Realizar una función que reciba un número ingresado por el usuario y averigüe si el número es capicúa o no (Por ejemplo: 12321). **Nota:** recordar el uso del MOD y el Trunc. No podemos transformar el número a cadena para realizar el ejercicio.