

# Programación

Introducción a la programación

Ejercicios prácticos I

(Python)

# Programación

## Ejercicios

### Ejercicios prácticos I

# Programación

## Ejercicio 1

### Ejercicio:

**Desarrollar un juego que consista en adivinar un número aleatorio entre 1 y 100.** Para ello, la aplicación generará el número a adivinar y luego irá pidiendo números al usuario, indicando si son mayores o menores que el número a adivinar. Cuando el usuario acierte el número, la aplicación mostrará la puntuación (puede ser el número de intentos) y un mensaje de despedida. La aplicación debe empezar mostrando un título y un texto explicativo del desarrollo del juego.

# Programación

## Ejercicio 2

### Ejercicio:

**Desarrollar un juego para adivinar una palabra.** La aplicación irá pidiendo palabras y cada vez que se fallé dará una pista. El juego terminará cuando el usuario acierte la palabra o cuando haya llegado al número máximo de intentos. En ese momento, la aplicación mostrará la puntuación (puede ser el número de intentos) y un mensaje de despedida. La aplicación debe empezar mostrando un título y un texto explicativo del desarrollo del juego.

**Importante:** Se valorará el hecho de que las pistas estén almacenadas en un array.

# Programación

## Ejercicio 3

### Ejercicio:

El alumno creará una aplicación que permita calcular el interés simple y compuesto generado por un determinado capital. El usuario tendrá que introducir los datos correspondientes al capital inicial, el interés y el tiempo. El funcionamiento de la aplicación será el siguiente:

1. Aparecerá un mensaje de bienvenida.
2. Aparecerá un menú de opciones (Calcular interés simple o compuesto).
3. Si se selecciona una operación:
  1. Pedirá el capital inicial.
  2. Pedirá el interés.
  3. Pedirá el tiempo.
  4. Calculará el resultado de la operación.
  5. Devolverá el resultado de la operación y pedirá al usuario si volver al menú o salir de la aplicación.
    1. Si se selecciona volver al menú, se limpiará la pantalla y volverá a aparecer el menú de la aplicación.
    2. Si se selecciona la opción Salir, el programa se cerrará con un mensaje de despedida.
4. Si se selecciona la opción Salir:
  1. El programa se cerrará con un mensaje de despedida.

# Programación

## Ejercicio 4

### **Ejercicio:**

Escribe un programa que permita calcular el área y el volumen de una serie de figuras geométricas (cubo, paralelepípedos, cilindro, cono, pirámide, etc.).

# Programación

## Ejercicio 5

### Ejercicio:

**Desarrollar un juego elegido por el alumno.** El desarrollo del juego puede ser libre. Al terminar, la aplicación mostrará la puntuación y un mensaje de despedida. La aplicación debe empezar mostrando un título y un texto explicativo del desarrollo del juego.

**Nota:** se valorará el uso de comentarios explicativos en el código fuente y la interacción con el usuario

# Programación

## Ejercicio 6

### **Ejercicio:**

Se desea desarrollar una aplicación que permita leer las líneas de un archivo de texto “palabras.txt” y muestre por pantalla el número de líneas leídas y el número de letras que hay en cada línea.

**Nota:** se valorará el uso de comentarios explicativos en el código fuente y la interacción con el usuario



# Programación

## Ejercicio 7

### **Ejercicio:**

Se desea desarrollar una aplicación que permita leer las líneas de un archivo de texto “entrada.txt” y las escriba en otro archivo de texto “salida.txt” pero eliminando los cinco primeros y los cinco últimos caracteres de cada línea.

**Ayuda:** se recomienda crear un método procesarLinea() que reciba como parámetro la línea leída, la procese y devuelva la línea procesada.

# Programación

## Ejercicio 8

### Ejercicio:

**Desarrollar un juego para adivinar una palabra.** La aplicación irá pidiendo palabras y cada vez que se fallé dará una pista. El juego terminará cuando el usuario acierte la palabra o cuando haya llegado al número máximo de intentos. En ese momento, la aplicación mostrará la puntuación (puede ser el número de intentos) y un mensaje de despedida. La aplicación debe empezar mostrando un título y un texto explicativo del desarrollo del juego. La palabra a adivinar y las pistas se han de leer de un archivo de texto “pistas.txt”.

**Ayuda:** Antes de empezar el juego, se recomienda leer la palabra a adivinar y las pistas del archivo de texto y cargarlas en un array.

# Programación

## Ejercicio 9

### **Ejercicio:**

Se desea desarrollar una aplicación que permita leer una serie de números enteros de un archivo de texto “numeros.txt” y muestre por pantalla la cantidad total de números leídos, cuántos son pares e impares y la suma total de todos ellos.

# Programación

## Ejercicio 10

### **Ejercicio:**

Se desea desarrollar una aplicación que permita leer una serie de números enteros de un archivo de texto “enteros.txt” y los escriba en otro archivo de texto en orden inverso.

**Ayuda:** se recomienda cargar los enteros leídos en un array. Una vez cargado el array, hay que recorrerlo en orden inverso escribiendo los números en el archivo de salida.

# Programación

Fin

# Programación

Introducción a la programación

Ejercicios prácticos II

(Python)