



## Examen Práctico - Programación 1ª Evaluación

Profesor: **Luis Javier López López**

### Ejercicio 1 (3 puntos)

Se desea crear un programa , que reciba como parámetro una cadena de texto y muestre por consola el número de signos de puntuación que tiene el texto en cuestión. A continuación se muestra un ejemplo de prueba del programa:

Cuando el programa recibe como parámetro el siguiente texto:

*En las ciudades de Piltover y Zaun, se palpa el desasosiego en el ambiente: inventores, ladrones, políticos y señores del crimen buscan liberarse de las ataduras de una sociedad fragmentada. Mientras la rebelión va cobrando fuerza, dos hermanas roban un artefacto de poder inimaginable. Los descubrimientos y el peligro son el trasfondo sobre el que nacerán héroes y se romperán vínculos. ¿Servirá este poder para cambiar el mundo o lo llevará a la ruina? Este es el mundo de Arcane.*

El programa mostrará por consola:

*Signos de puntuación encontrados:*

*Número de comas : 4*

*Número de puntos : 3*

*Número de interrogaciones: 1*

*Número de dos puntos: 1*

Nota: Hay que tener en cuenta que los signos de puntuación posibles son los siguientes:

,	la coma	.	el punto
;	el punto y coma	“ ”	las comillas
:	los dos puntos	. . .	los puntos suspensivos
( )	el paréntesis	ü	la diéresis
[ ]	los corchetes	-	el guión
¿ ?	la interrogación	¡ !	la admiración

Y que los paréntesis , interrogaciones , exclamaciones , etc, no se cuentan por separado sino conjuntamente, es decir, un texto que contenta “¿?” contará como 1 interrogación no como 2.



## Examen Práctico - Programación 1ª Evaluación

Profesor: **Luis Javier López López**

### Ejercicio 2 (5 puntos)

Se desea realizar un programa que simule el juego del ahorcado. Para ello el programa recibe como parámetro una palabra.

Una vez que se ejecuta el programa empieza a preguntar al usuario por una letra, guardando en una lista las letras que va diciendo.

Cada vez que se dice una letra el programa comprueba que no esté dicha para que el usuario no pueda repetir la letra.

Si la letra que dice el usuario está en la palabra, el programa muestra un mensaje diciendo “Has acertado” y a continuación muestra la palabra sólo mostrando las letras acertadas.

El programa siempre muestra el número de vidas que le quedan al usuario, inicialmente son 5 y cada vez que dice una letra errónea se va restando una vida. Cuando las vidas del usuario llegan a 0, el programa finaliza.

A continuación se muestra una traza de ejecución:

Imaginamos que la palabra que recibe el algoritmo es “patinete”:

CONSOLA:

-----  
*Vidas : 5*

*Letras citadas: [ ]*

*Palabra: \_ \_ \_ \_ \_*  
-----

*Dime una letra : m*  
-----

*Vidas : 4*

*Letras citadas: [ m]*

*Palabra: \_ \_ \_ \_ \_*  
-----

*Dime una letra : e*  
-----

*Vidas : 4*

*Letras citadas: [ m , e]*

*Palabra: \_ \_ \_ \_ \_ e \_ e*  
-----

*Dime una letra :*



**Examen Práctico - Programación 1ª Evaluación**

**Profesor:** *Luis Javier López López*

**Ejercicio 3 (5 puntos)**

Se desea realizar un programa que reciba una **lista** de diccionarios de esta estructura:

```
[{
    "nombre" : "Enrique"
    "apellidos" : "García, Migueza"
    "dni" : "12345678K"
    "email" : "egarciamigueza@safareyes.es"
},
{
    "nombre" : "Paloma"
    "apellidos" : "Machado, López"
    "dni" : "12345678Z"
    "email" : "pmachadolopez@hotmail.es"
},
{
    "nombre" : "Antonio"
    "apellidos" : "Romero, Domínguez"
    "dni" : "12345678A"
    "email" : "aromerodominguez@safareyes.es"
}]
```

- a) **(2 puntos)** Y que devuelva la **lista** de alumnos cuyo dominio de email pertenece al centro (tiene el dominio "safareyes.es"), sabiendo que todos los email que se pasen al algoritmo siguen el patrón: prefijo@dominio
- b) **(2 puntos)** Que devuelva al primer alumno de la lista, tomando como orden el primer apellido en el orden alfabético natural.
- c) **(1 punto)** Que devuelva al primer alumno de la lista, tomando como orden la letra del dni en el orden alfabético natural.



## Examen Práctico - Programación 1ª Evaluación

Profesor: **Luis Javier López López**

### Ejercicio 4 (1 puntos)

Se desea realizar un programa que reciba como parámetro una lista de notas y calcula la media de todas las notas pasadas en la lista.

Además una vez obtenida la media el método devolverá una cadena de texto en función a la nota media obtenida, siguiendo la referencia:

Condición	Cadena que devuelve el método
$\text{Nota media} < 5$	“Suspenso”
$5 > \text{Nota media} < 7$	“Aprobado”
$7 > \text{Nota media} < 9$	“Notable”
$\text{Nota media} > 9$	“Sobresaliente”

Por ejemplo para la entrada:

[ 5.6 , 7 , 6.2 , 8 ]

La media calculada sería : 6.7

El método devolvería: “**Aprobado**”

### Ejercicio 5 (2 puntos)

Se desea realizar un programa de comprobación de contraseñas seguras. Para ello el programa recibe una cadena de texto, y ha de comprobar que esa cadena de texto:

- Empiece y termine por una letra mayúscula.
- Contenga algún valor numérico.
- Contenga alguno de los siguientes símbolos( punto, guión bajo, hashtag )

El programa devuelve True si la contraseña cumple dichas condiciones y False si no se cumple alguna de ellas.