

# Programación para la Inteligencia Artificial

## Práctica 5 (Parte II)

### Cifrado y descifrado del César

El *Cifrado del César* consiste en sustituir cada letra por la que se encuentra 3 posiciones más adelante en el alfabeto. Por ejemplo, la codificación de la cadena “Estamos en un examen de PIA”, sería “Hvwdprv hq xq hadphq gh SLD”. Este rudimentario sistema de encriptación se puede generalizar con un desplazamiento variable. A la hora de descifrar, hay que realizar un análisis de frecuencias de las letras que aparecen en el texto encriptado y compararlo con las frecuencias habituales de cada letra en el idioma de partida, para sustituir. Se pide definir las siguientes funciones para obtener una aplicación que realice el cifrado y descifrado del César, usando las funciones de E/S para realizar un pequeño menú.

1. Definir el tipo **Mensaje** como un renombramiento de **String**.
2. Definir las dos funciones:
  - **minuscualaAint**, que devuelva el entero correspondiente a la letra minúscula que tome como entrada, es decir, un 0 si toma como entrada el carácter 'a', un 4 si toma como entrada el carácter 'e' o un 25 si toma como entrada el carácter 'z'.
  - **mayusculaAint** que devuelva el entero correspondiente a la letra mayúscula que tome como entrada, es decir, un 0 si toma como entrada el carácter 'A', un 4 si toma como entrada el carácter 'E' o un 25 si toma como entrada el carácter 'Z'.

Y sus inversas:

- **intAminuscula** que devuelva la letra minúscula correspondiente al entero que tome como entrada.
- **intAmayuscula** que devuelva la letra mayúscula correspondiente al entero que tome como entrada.

Para todas ellas son útiles las funciones predefinidas **ord** y **char** de la librería **Data.Char**.

3. Definir la función **desplaza** que tome como entrada un desplazamiento **n** y un carácter **c** y devuelva el carácter obtenido desplazando **n** caracteres el carácter **c**. Por ejemplo, **desplaza 3 'a'** devolvería 'd' y **desplaza (-3) 'B'** devolvería 'Y'.

4. Definir la función `codifica` que tome como entrada un desplazamiento `n` y un Mensaje `ms` y devuelva el mensaje resultante de codificar `ms` con un desplazamiento `n`. De esta forma,

```
ms == codifica (-n) (codifica n ms)
```

5. Calcular la función `porcentaje`, que devuelva, dados dos números `n` y `m` como entrada, el porcentaje de `n` sobre `m`. Por ejemplo, `porcentaje 2 5 == 40.0`.
6. Calcular la función `soloLetras`, que tome una cadena `xs` como entrada y devuelva la cadena obtenida al quitarle a `xs` todos los caracteres que no son letras. Por ejemplo, `soloLetras "Hoy es martes." == "Hoyesmartes"`.
7. Calcular la función `ocurrencias`, que tome como entrada un carácter `c` y una cadena `xs` y devuelva el número de ocurrencias de `c` en `xs`. Por ejemplo, `ocurrencias 'a' "Galicia" == 2`.

8. Calcular la función `frecuencias`, que tome como entrada una cadena `xs` y devuelva una lista de longitud 26, que contenga en cada posición la frecuencia de la letra correspondiente del alfabeto dentro de la cadena `xs`, con independencia de que dicha letra aparezca en minúscula o mayúscula. Por ejemplo:

```
frecuencias "Inteligencia Artificial Z" devolvería
```

```
[13.0,0,8.7,0,8.7,4.3,4.3,0,26.1,0,0,8.7,0,8.7,0,0,0,4.3,0,8.7,0,0,0,0,0,4.3]
```

9. Una medida de la discrepancia entre una distribución observada `os` y la esperada `es` es la *prueba*  $\chi^2$ :

$$\chi^2 = \sum_{i=0}^{n-1} \frac{(os_i - es_i)^2}{es_i}$$

de forma que la discrepancia es menor cuanto menor es el resultado de la fórmula anterior.

Definir una función `chiCuad` que implemente la fórmula anterior aplicada a dos distribuciones diferentes. Por ejemplo:

```
chiCuad [7,8,9] [7,8,9] == 0.0
```

```
chiCuad [7,8,9] [6,5,4] == 8.21666
```

10. Calcular la función `rota`, que tome como entrada un número `n` y una cadena `xs` y devuelva la lista obtenida rotando `n` posiciones los elementos de `xs`. Por ejemplo, `rota 3 "carpanta" == "pantacar"`.

11. Calcular la función `descifra`, que tome como entrada un mensaje encriptado `ms` y devuelva el mensaje descifrado. Esto puede hacerse decodificando la cadena `ms` por el anti-desplazamiento que produce una distribución de letras con la menor desviación *chi cuadrado* respecto de la tabla de distribución de las letras en castellano. Por ejemplo:

```
codifica 5 "Todo Para Nada" devolvería
```

```
"Ytit Ufwf Sfif", y
```

```
descifra "Ytit Ufwf Sfif", devolvería
```

```
"Todo Para Nada"
```

La constante `tabla` siguiente es la lista de la frecuencias de las letras en castellano. Por ejemplo, la frecuencia de la 'a' es del 12.53%, la de la 'b' es 1.42%:

```
tabla :: [Float]
```

```
tabla = [12.53, 1.42, 4.68, 5.86, 13.68, 0.69, 1.01, 0.70, 6.25, 0.44,  
0.01, 4.97, 3.15, 6.71, 8.68, 2.51, 0.88, 6.87, 7.98, 4.63,  
3.93, 0.90, 0.02, 0.22, 0.90, 0.52]
```