

Introducción a la programación

Guía 8: Archivos y diccionarios

1c2025

Guía 8 - Ejercicio 3.16

problema calcular_promedio_por_estudiante (in notas:
 $seq\langle seq\langle Char \rangle \times \mathbb{R} \rangle$) : Diccionario $\langle seq\langle Char \rangle, \mathbb{R} \rangle$ {
 requiere: {El primer componente de las tuplas de notas no es
 una cadena vacía}
 requiere: {El segundo componente de las tuplas de notas está
 en el rango $[0, 10]$ }
 asegura: {Todas las claves de *res* son nombres que aparecen
 en *notas* (primer componente)}
 asegura: {Todos los nombres de *notas* (primer componente)
 son clave en *res*}
 asegura: {El valor de cada clave de *res* es el promedio de
 todas las notas que obtuvo el estudiante (segunda
 componente de *notas*)}
}

Guía 8 - Casos de Test Ejercicio 3.16

Elaboremos casos de test para la solución.

Caso	Entradas	Esperado
Un solo estudiante	[(" Juan",10)]	{" Juan":10}
Varios estudiantes	[(" Juan",10), (" Pedro",5)]	{" Juan":10, " Pedro":5}
Varios exámenes	[(" Juan",10), (" Juan",5)]	{" Juan":7.5}
Varios estudiantes y varios exámenes	[(" Juan",10), (" Juan",5), (" Pedro",5)]	{" Juan":7.5, " Pedro":5}

Guía 8 - Ejercicio 19.1

problema contar_lineas (in nombre_archivo: $\text{seq}\langle\text{Char}\rangle$) : \mathbb{Z} {
 requiere: {*nombre_archivo* es el path con el nombre de un
 archivo existente y accesible}
 asegura: {*res* es igual a la cantidad de líneas que contiene el
 archivo indicado por *nombre_archivo*}
}

Guía 8 - Casos de Test Ejercicio 19.1

Caso	Entradas	Esperado
Archivo vacío	archivo_vacio.txt	0
Archivo con una línea	archivo_una_linea.txt	1
Archivo con cinco líneas	archivo_cinco_lineas.txt	5
Archivo con una línea vacía	archivo_una_linea_vacia.txt	1

Guía 8 - Ejercicio 22

```
problema clonar_sin_comentarios (in nombre_archivo_entrada:
seq<Char>, in nombre_archivo_salida: seq<Char>) {
  requiere: {nombre_archivo_entrada es el path con el
             nombre de un archivo existente y accesible}
  requiere: {nombre_archivo_salida es el path con el nombre
             de un archivo que, si existe, se puede modificar, y si no
             existe, se puede crear}
  asegura: {El archivo indicado por nombre_archivo_salida
            contiene las mismas líneas y en el mismo orden que el
            archivo nombre_archivo_entrada, excepto aquellas que
            comienzan con el carácter #}
}
```

Guía 8 - Ejercicio 22

Ejemplo:

```
# esto es un comentario
    # esto tambien
esto no es un comentario # esto tampoco
```

Guía 8 - Casos de Test Ejercicio 22

Caso	Entradas
Archivo vacío	archivo_vacio.txt
Archivo sin comentarios	archivo_sin_comentarios.txt
Archivo con un comentario	archivo_con_un_comentario.txt
Archivo con varios comentarios	archivo_con_varios_comentarios.txt
Archivo con espacios y un comentario al final	archivo_con_comentario_al_final.txt

Guía 8 - Ejercicio 20

problema agrupar_por_longitud (in nombre_archivo:
 $seq\langle Char \rangle$) : Diccionario $\langle \mathbb{Z}, \mathbb{Z} \rangle$ {
 requiere: {*nombre_archivo* es el path con el nombre de un
 archivo existente y accesible}
 asegura: {Para cada longitud n tal que existe al menos una
 palabra de longitud n en el archivo indicado por
 nombre_archivo, $res[n]$ es igual a la cantidad de
 palabras de esa longitud}
 asegura: {No hay otras claves en *res* que no correspondan a
 longitudes de palabras presentes en el archivo}
}

Guía 8 - Ejemplo Ejercicio 20

Ejemplo del diccionario de salida:

```
{  
    1: 2,  
    2: 10,  
    5: 4  
}
```

indica que se encontraron 2 palabras de longitud 1, 10 palabras de longitud 2 y 4 palabras de longitud 5. Para este ejercicio se consideran como palabras todas aquellas secuencias de caracteres delimitadas por espacios en blanco.

Guía 8 - Casos de Test Ejercicio 20

Caso	Entradas	Esperado
Archivo vacío	archivo_vacio.txt	{}
Archivo con una palabra de longitud 1	archivo_una_palabra.txt	{1:1}
Archivo que contiene una palabra de longitud 3 dos veces	archivo_varias_palabras.txt	{3:2}

Guía 8 - Ejercicio 21

```
problema la_palabra_mas_frecuente (in nombre_archivo:
seq<Char>) : seq<Char> {
  requiere: {nombre_archivo es un archivo existente y
             accesible que tiene, por lo menos, una palabra}
  asegura: {res es una palabra que aparece en el archivo
            nombre_archivo}
  asegura: {No hay ninguna palabra contenida en el archivo
            nombre_archivo que aparezca más veces que la palabra
            res }
}
```

Guía 8 - Casos de Test Ejercicio 21

Caso	Entradas	Esperado
Archivo vacío	archivo_vacio.txt	[]
Archivo cuya palabra más frecuente es "Manzana"	archivo_una_palabra.txt	["Manzana"]
Archivo cuyas palabras más frecuentes son "Manzana", "Pera", "Üva"	archivo_varias_palabras.txt	["Manzana", "Pera", "Üva"]