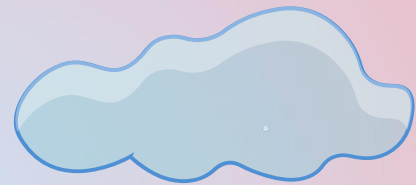


Auxiliar 6

Listas

Introducción a la Programación 2022-2



+ P1

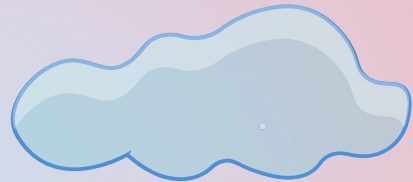
Función enRango



Función enRango

```
from lista import *
L=lista(20,lista(10,lista(5,lista(15,None))))
#enRango: lista(num) num num -> int
#cantidad de numeros en L en el rango entre x e y
#ej: enRango(L,15,20)->2, enRango(L,10,12)->1
def enRango(L,x,y):
    assert esLista(L) and (type(x) == int or type(y) == float) and (type(x) == int or type(y) == float)
    assert 0<=x <= y
    if vacia(L):
        return 0
    else:
        assert type(cabeza(L)) == int or type(cabeza(L)) == float
        if cabeza(L) >= x and cabeza(L) <= y:
            return 1 + enRango(cola(L),x,y)
        else:
            return enRango(cola(L),x,y)
assert enRango(L,15,20)==2
assert enRango(L,10,12)==1
assert enRango(listaVacia,10,11)==0
```





+ P2

Listas con tiempo



Módulo Tiempo

```
#tiempo: horas(int) minutos(int)
import estructura
estructura.crear("tiempo","horas minutos")
#esTiempo: tiempo -> bool
#True si x es un tiempo correcto (horas y minutos enteros positivos y minutos<60)
#ejs: esTiempo(tiempo(13,30)) ->True, esTiempo(tiempo(12,90)) ->False
def esTiempo(x):
    assert type(x)==tiempo
    assert type(x.horas)==int and type(x.minutos)==int
    return x.minutos<60
assert esTiempo(tiempo(13,30))
assert not esTiempo(tiempo(12,90))

#sumar: tiempo tiempo -> tiempo
#x + y
#ej: sumar(tiempo(10,15),tiempo(1,30)) -> tiempo(11,45)
def sumar(x,y):
    assert esTiempo(x) and esTiempo(y)
    minutos=x.minutos+y.minutos
    if minutos>=60:
        return tiempo(1+x.horas+y.horas,minutos-60)
    else:
        return tiempo(x.horas+y.horas,minutos)
assert sumar(tiempo(10,15),tiempo(1,30))==tiempo(11,45)
```



Función suma

```
from lista import *
from tiempo import *
L=lista(tiempo(14,30),lista(tiempo(10,15),lista(tiempo(8,30),lista(tiempo(16,15),None))))
#suma: lista(tiempo) -> tiempo
#suma de tiempos de lista L
#ej: suma(L) -> tiempo(49,30)
def suma(L):
    assert esLista(L)
    if vacia(L):
        return tiempo(0,0)
    else:
        assert esTiempo(cabeza(L))
        return sumar(cabeza(L),suma cola(L))
assert suma(L)==tiempo(49,30)
```



Función tarde

```
from lista import *
from tiempo import *
L=lista(tiempo(14,30),lista(tiempo(10,15),lista(tiempo(8,30),lista(tiempo(16,15),None))))
#tarde: lista(tiempo) -> lista(tiempo)
#lista con tiempos de L mayores a mediodía
#ej: tarde(L)->lista(tiempo(14,30),lista(tiempo(16,15),None))
def tarde(L):
    assert esLista(L)
    if vacia(L):
        return None
    else:
        assert esTiempo(cabeza(L))
        if cabeza(L).horas>=12:
            return lista(cabeza(L),tarde(cola(L)))
        return tarde(cola(L))
assert tarde(L)==lista(tiempo(14,30),lista(tiempo(16,15),None))
```



Thanks!



Do you have any questions?
Preguntas al foro o al correo

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**

Please keep this slide for attribution