

# Algoritmos 2

Licenciatura en Ciencia de Datos - UNSAM

## Examen parcial

**Formato de entrega:** Código fuente de módulos y/o paquetes python

*Resolver los ejercicios utilizando los principios del Paradigma Orientado a Objetos, principalmente herencia.*

*Entregar cada ejercicio en su propio paquete o módulo.*

*Se puede heredar en los casos que consideren necesarios del código de cualquier TAD visto en clase de árboles. En dicho caso, entregar también los archivos de los TADs utilizados.*

*El uso de cualquier herramienta de inteligencia artificial será penalizado.*

### CONSIGNA:

La empresa de desarrollo de software **EvilCorp** se encuentra actualizando su **organigrama**, ya que su personal rota frecuentemente. Lo necesita ya que es necesario **pagar los salarios** de este mes.

En la **jerarquía** se encuentra el **director general**. De él dependen el **director de IT**, el de **finanzas** y el de **recursos humanos**. No se busca tener más directores ya que el resto de las tareas se realizan por medio de consultoría externa. A su vez, **del director de IT** hay siempre **uno o más líderes técnicxs**. Un **líder técnicx** tiene **unx o más desarrolladorxs**, que de acuerdo a su categoría pueden ser **senior**, **junior** o **trainee**. Un desarrollador es **trainee** si tiene menos de 1 año de experiencia, es **junior** si tiene entre 1 y 5, y **senior** si tiene más de 5. También, se contratan **pasantes**, que están a cargo de un **desarrollador senior**, pero no necesariamente hay pasantes en todo momento.

En cuanto a los **sueldos**, se calculan de la siguiente manera:

-Lxs **directores de área**, cobran \$2.500.000 más \$5.000 por cada año de **antigüedad**

-Lxs **líderes técnicxs** cobran \$100.000 más \$3.000 por cada año de **antigüedad**

-Lxs **desarrolladorxs**:

1) Si son **senior**, cobran \$700.000 más \$2.000 por cada año de **antigüedad**

2) Si son **junior** cobran \$500.000 más \$1.500 por cada año de **antigüedad**.

3) Si son **trainees** cobran \$300.000

Además, como **beneficio**, la empresa, en caso de tener algún **empleadx** al menos 1 hijx, les da un **bono** de \$100.000 (en total).

-Los **pasantes no cobran sueldo** ya que realizan sus tareas para ganar experiencia.

Se pide:

- 1) Desarrollar **todas las clases** que consideres necesarias para modelar el problema (2 puntos)
- 2) Desarrollar un **método recursivo** dentro de la **clase organigrama** que, dado un organigrama, determine **si cumple con las condiciones para ser válido** para **EvilCorp**. Utilizar **todos los tipos de empleadx** que existen.  
Firma: `def es_valido(self) -> bool:` (2,5 puntos)
- 3) Crear el **método** dentro de la **clase** correspondiente para calcular el **monto total a pagar sin discriminar por tipo de empleadx utilizando recursividad** teniendo en cuenta el **sueldo de todos los empleadx** de Evilcorp para informarle la **suma** del monto al **Director General**. Firma: `def calcular_sueldo_total(self) -> int:` **NOTA: EL DIRECTOR GENERAL NO COBRA SUELDO, SINO QUE EL RESULTADO DE LA FUNCIÓN DEBE DEVOLVER EL MONTO TOTAL QUE EL DIRECTOR DEBE CONOCER PARA PAGAR A SUS EMPLEADXS** (2,5 puntos).
- 4) En la **función main()** desarrollar **casos de prueba**. Crear **dos árboles**, uno **válido** y uno **inválido**; para **ambos** testear **es\_valido**. Para el **válido**, **también** testear **calcular\_sueldo\_total** (3 puntos)