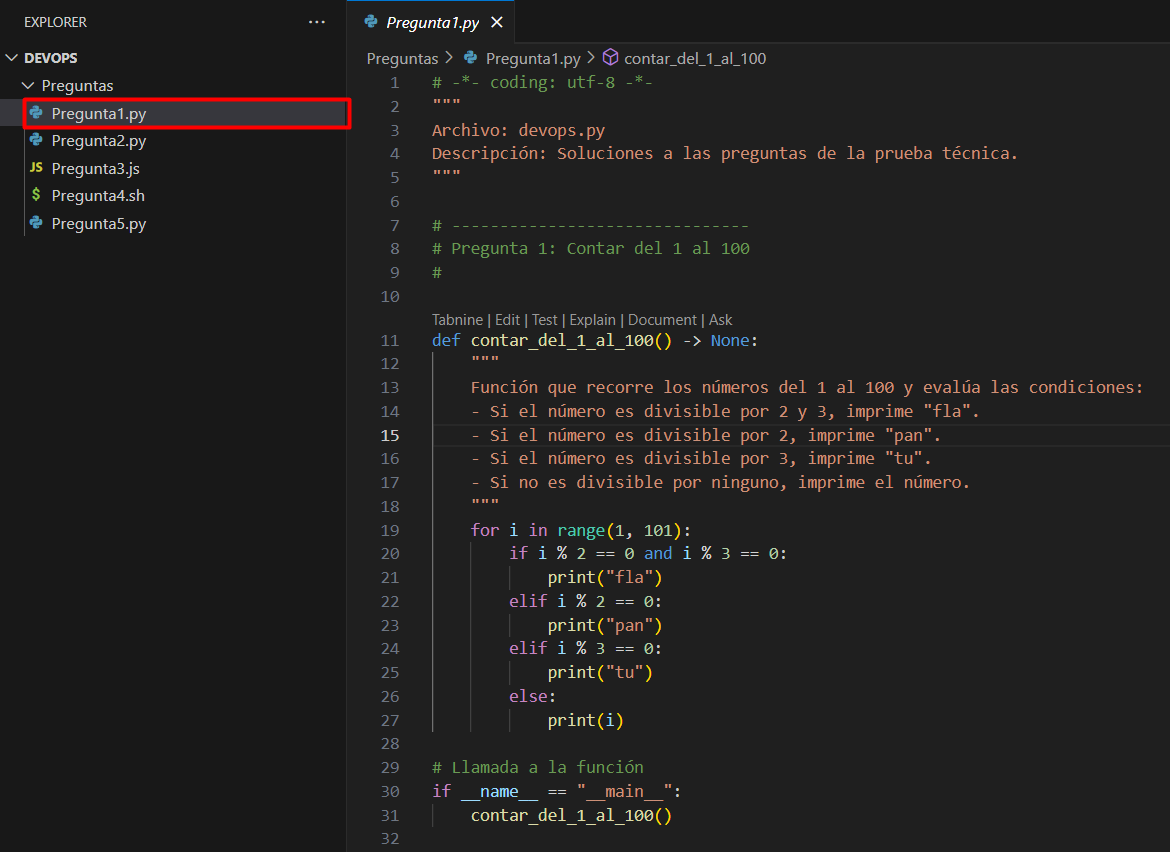
Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

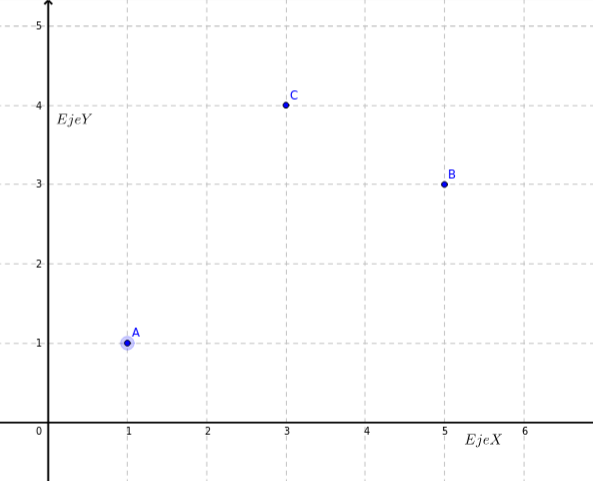
Pregunta 1:

Escriba un método en el lenguaje que más le acomode lo siguiente: una función que cuente del 1 al 100 y que por cada número vaya evaluando lo siguiente:

* Si el número es divisible por 2 mostrar en pantalla “pan”.
* Si el número es divisible por 3 mostrar en pantalla “tu”.
* Si el número es divisible por 2 y 3 mostrar en pantalla “fla”.



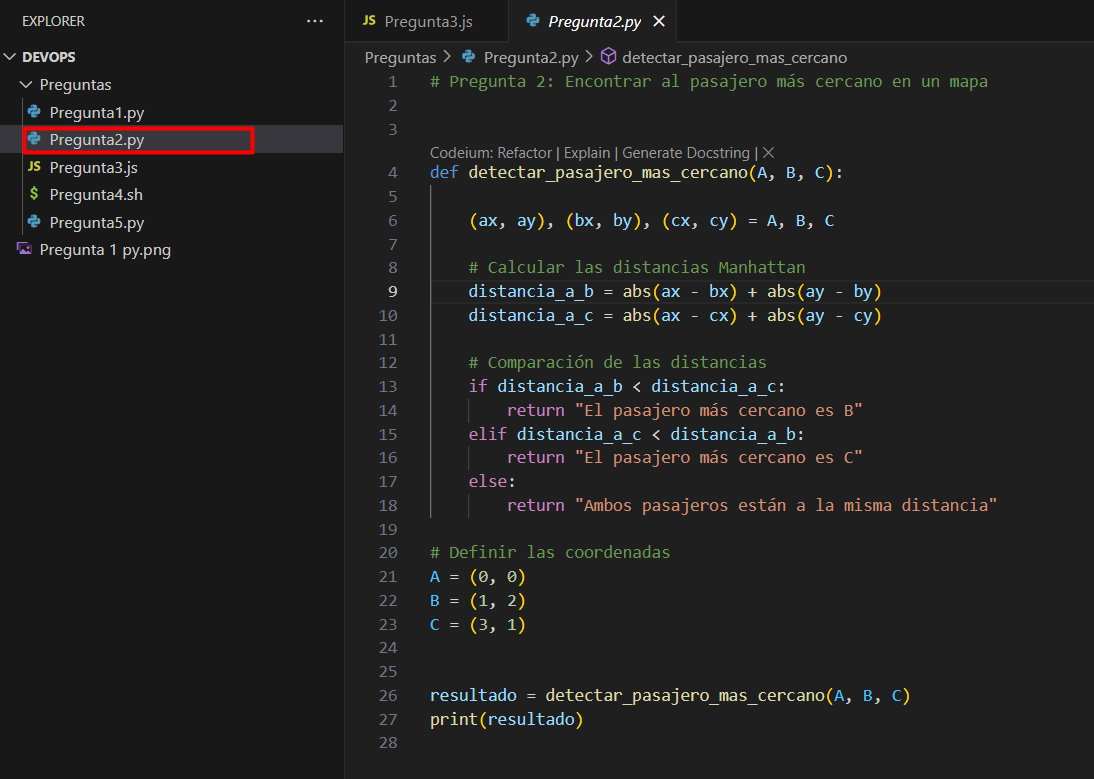
Pregunta 2.  
  
Imagine usted que está desarrollando una aplicación similar a uber y esta debe encontrar al pasajero más cercano en un mapa, para ello se le plantea lo siguiente:  
Dado un plano cartesiano con 3 coordenadas, A(conductor), B( pasajero 1) y C(pasajero 2), debe encontrar el camino más corto desde el punto de origen A y mostrar por pantalla si el pasajero 1 o el pasajero 2 está más cercano al punto de origen.



Para dar una solución a este problema usted debe escribir un método en el lenguaje de programación que más le acomode:

-Para calcular la distancia considere moverse por coordenadas tanto horizontal como vertical por números enteros, imagine que cada coordenada de números es una “cuadra” de calles.

No se consideran los movimientos diagonales.  
-Debe recibir como parámetros 3 puntos del plano: A(x,y) , B(x’,y’), C(x’’,y’’)  
-Debe calcular la distancia entre el conductor y los pasajeros  
-Debe imprimir por pantalla con un mensaje al pasajero más cercano, ejemplo: “El pasajero más cercano es B”



Pregunta 3.

Existen 2 liebres compitiendo en una carrera, la primera (liebre a) comienza la carrera en la posición A y tiene una velocidad Va que representa los cuadrantes que avanza en cada salto desde su posición inicial, la segunda (liebre b) comienza la carrera en la posición B y va a una velocidad Vb que representa los cuadrantes que avanza en cada salto desde su posición inicial.

Dada la siguiente función con los parámetros de entrada descritos anteriormente:

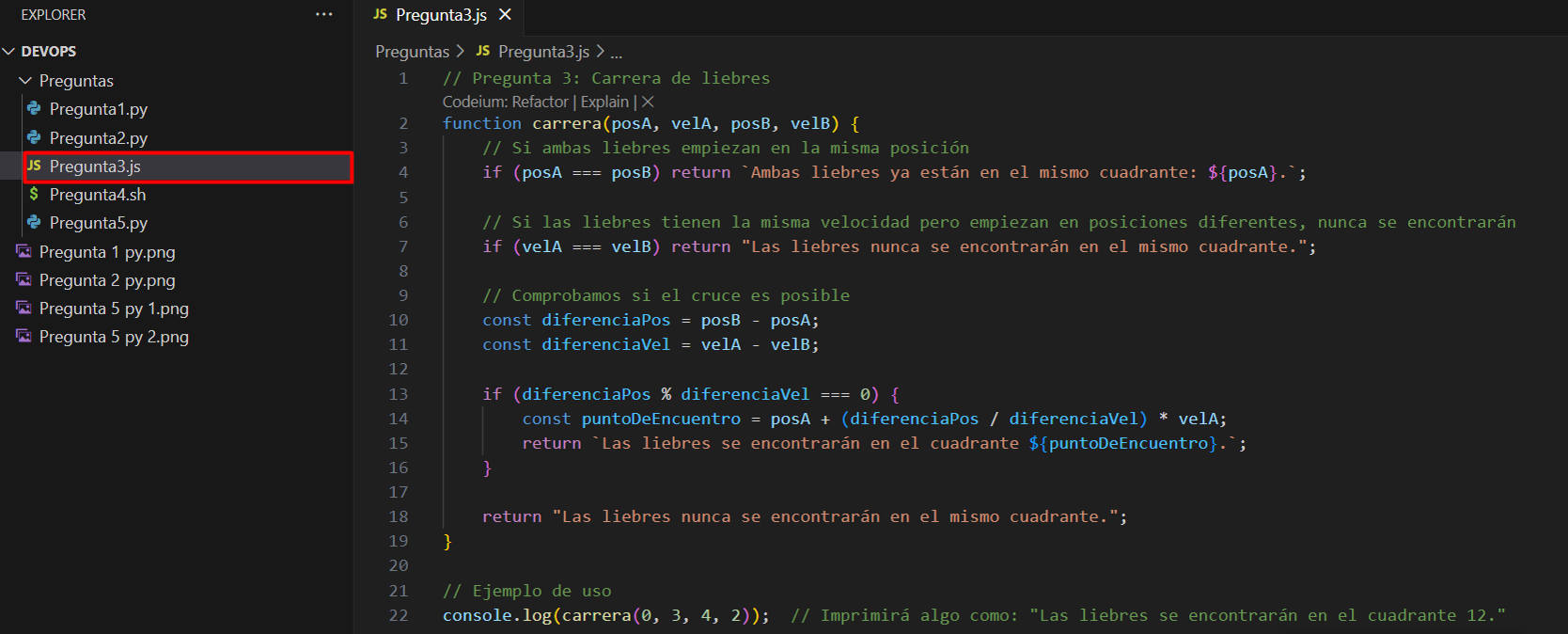
*Carrera (int liebre\_a\_posicion, int liebre\_a\_velocidad,*

*int liebre\_b\_posición, int liebre\_b\_velocidad) {*

*..*

*}*

Implementar la función en el lenguaje que más le acomode que indique o retorne en qué posición ambas liebres se encuentran en el mismo cuadrante.

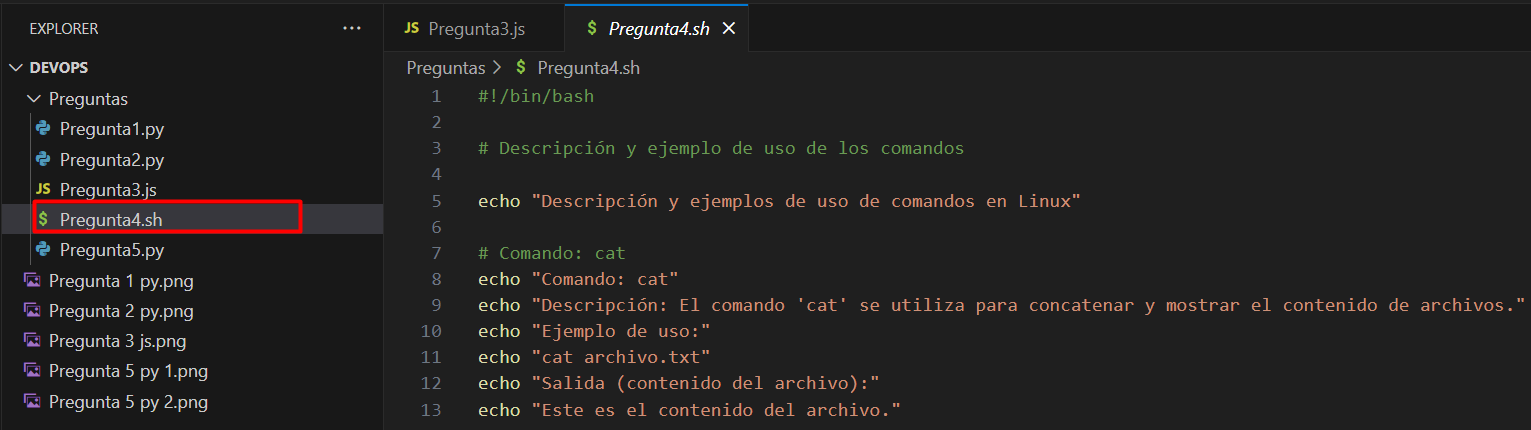


Pregunta 4:

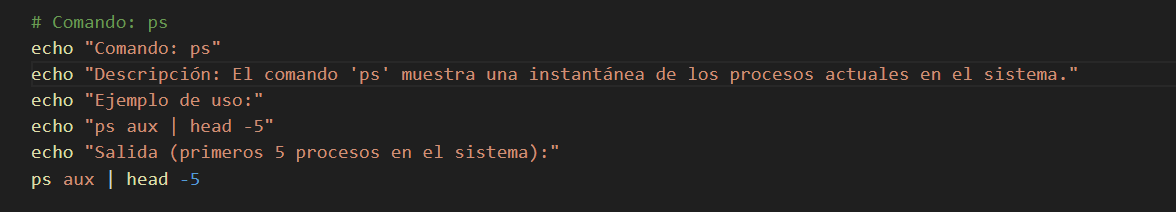
**Linux:**

Describir para qué sirve y dar un ejemplo de uso de los siguientes comandos:

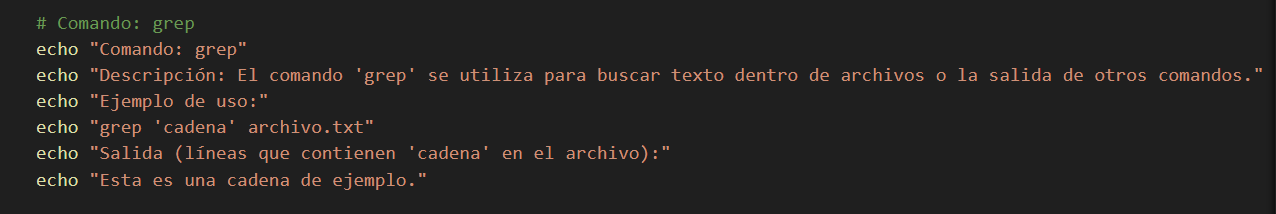
cat

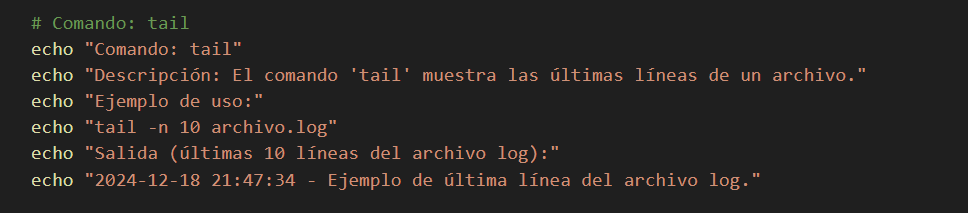


ps



grep



tail

Pregunta 5:

**Electrónica básica:**

Tenemos un circuito del que se desea, desde las salidas de relé de dos sistemas distintos, cerrar ese circuito (activarlo) en caso que cualquiera de los 2 relés conectados se activen. ¿Qué conexión deberá utilizarse?

1. Salida Normal Abierto (NO), conexión en serie
2. Salida Normal Cerrado (NC), conexión en serie
3. Salida Normal Abierto (NO), conexión en paralelo
4. Salida Normal Cerrado (NC), conexión en paralelo

