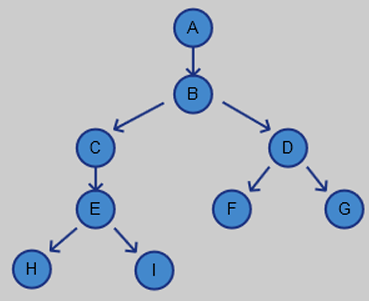
# Case Test PYTHON

Este documento contiene los casos de pruebas realizados en PYTHON

## **Enunciado 1:**

Generar el siguiente árbol de procesos en Python:



En cuanto a la creación, todos los procesos hijos deberán crearse con el enfoque de programación concurrente, es decir NO sería una solución válida crear un proceso y esperar a que éste finalice para crear el siguiente.

**Nota**: Pausar o retrasar la finalización del programa para que el equipo docente pueda verificar la correcta creación del árbol.

## **PROMPT 1:**

* **LENGUAJE DE PROGRAMACION**:
  + Utilizar lenguaje Python
* **PLATAFORMA:**
  + Windows
* **FUNCIONALIDAD A IMPLEMENTAR:**
  + Desarrollar un sistema que permita generar un árbol de procesos de acuerdo al diseño de la imagen enviada.
* **REQUISITOS ESPECIFICOS:**
  + Imprimir en consola el árbol una vez que finalice su creación
  + Agregar puntos para poder pausar o retrasar la finalización del programa para poder verificar la correcta creación del árbol.
* **RESTRICCIONES:** 
  + Todos los procesos hijos deben generarse utilizando el enfoque de programación concurrente, es decir para crear un proceso no se debe esperar a que otro proceso finalice para crear el siguiente.

## **RESULTADOS OBTENIDOS**

## **CURSOR**

## Se generó correctamente el código con el siguiente output por pantalla:

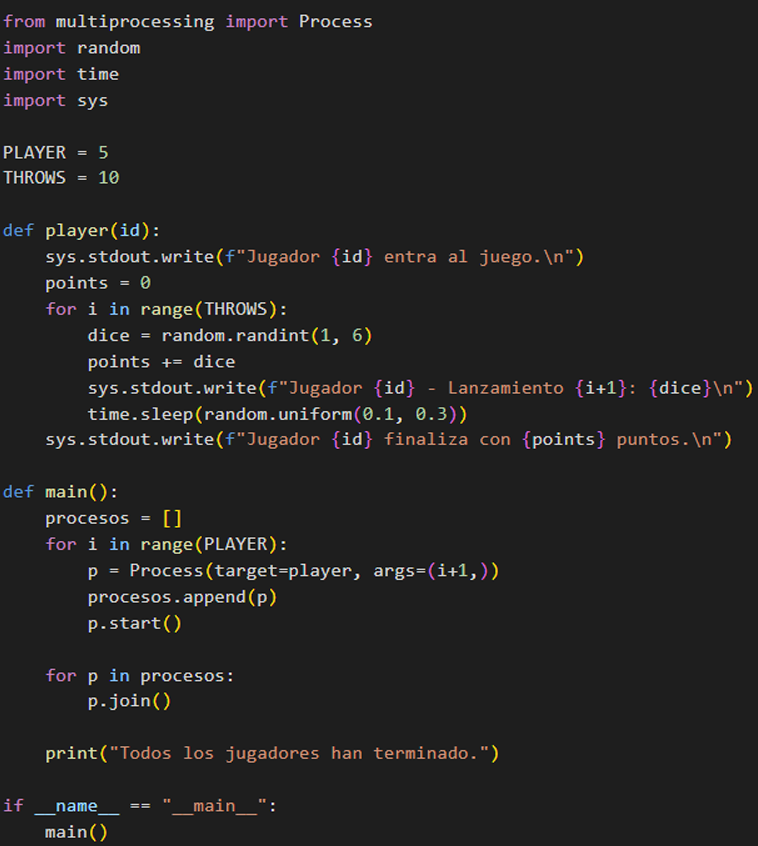
## **GPT**

## Se generó correctamente el código con el siguiente output por pantalla:

## **Enunciado 2:**

Dado el código que se muestra en la siguiente imagen:

1. Explique con sus palabras que es lo que hace.
2. Desarrolle en Python el mismo programa pero utilizando fork() en lugar de Process.



## **PROMPT 1:**

* **LENGUAJE DE PROGRAMACION**:
  + Utilizar lenguaje Python
* **PLATAFORMA:**
  + Windows
* **FUNCIONALIDAD A IMPLEMENTAR:**
  + Explicar que hace el código fuente y reemplazar la funcionalidad de Process por Fork()
* **REQUISITOS ESPECIFICOS:**
  + Agregar una descripción general al comienzo del código fuente explicando que hace
  + Agregar comentarios en el código fuente para explicar que hace.
* **RESTRICCIONES:** 
  + Mantener la estructura del código en la medida que sea posible.

## **RESULTADOS OBTENIDOS**

## **CURSOR**

## Se generó correctamente el código con el siguiente output por pantalla:

**Enunciado 3:**

La mansión Derceto, construida en el año 1818, es una conocida fuente de actividad paranormal y fenómenos inexplicables. Los famosos investigadores paranormales Ed y Lorraine Warren han sido convocados para investigar la mansión. La pareja lo contrata a usted, experto en programación concurrente, para realizar un programa que pueda monitorear sus múltiples cámaras de última generación y reportar cualquier tipo de actividad extraña. A continuación se citan las especificaciones de diseño:

* El lenguaje será Java.
* Las ZONAS a monitorear son las siguientes:
  + Sótano 2
  + Ático Año 2025 Universidad Nacional de La Matanza Programación Concurrente
  + Cocina
  + Dormitorio
  + Jardín
  + Mausoleo
* Cada CÁMARA será un proceso independiente.
* Cada cámara monitorea una zona y cada cierto tiempo informa alguno de los siguientes EVENTOS:
  + Sin actividad
  + Movimiento detectado
  + Anomalía térmica
  + Sombra extraña
  + Ruido detectado
* Los mensajes se imprimen por consola indicando; ID de CÁMARA, ZONA y EVENTO.
* La duración del monitoreo de las cámaras (segundos) deberá ser tomado como parámetro.
* La frecuencia con que reportan las cámaras (segundos) deberá ser tomada como parámetro.
* Al finalizar, cada cámara informará la cantidad de eventos paranormales detectados.
* El proceso principal esperará a que todas las cámaras finalicen para luego finalizar.
* A fin de realizar la prueba del programa, genere aleatoriamente los EVENTOS en cada cámara, queda a criterio del grupo la probabilidad de ocurrencia de cada EVENTO.

## **PROMPT 1:**

* **LENGUAJE DE PROGRAMACION**:
  + Utilizar lenguaje Python
* **PLATAFORMA:**
  + Windows
* **FUNCIONALIDAD A IMPLEMENTAR:**
  + Desarrollar un sistema que permita generar un árbol de procesos de acuerdo a los requisitos y restricciones indicados en los siguientes puntos. El contexto de la aplicación es que unos investigadores paranormales han sido convocados para investigar una mansión y somos contratados para realizar un programa que pueda monitorear sus múltiples cámaras de última generación y reportar cualquier tipo de actividad extraña.
* **REQUISITOS ESPECIFICOS:**
  + Las ZONAS a monitorear son las siguientes:
    - Sótano 2
    - Ático Año 2025 Universidad Nacional de La Matanza Programación Concurrente
    - Cocina
    - Dormitorio
    - Jardín
    - Mausoleo
  + Cada CÁMARA será un proceso independiente.
  + Cada cámara monitorea una zona y cada cierto tiempo informa alguno de los siguientes EVENTOS:
    - Sin actividad
    - Movimiento detectado
    - Anomalía térmica
    - Sombra extraña
    - Ruido detectado
  + Los mensajes se imprimen por consola indicando; ID de CÁMARA, ZONA y EVENTO.
  + La duración del monitoreo de las cámaras (segundos) deberá ser tomado como parámetro.
  + La frecuencia con que reportan las cámaras (segundos) deberá ser tomada como parámetro.
  + Al finalizar, cada cámara informará la cantidad de eventos paranormales detectados.
  + El proceso principal esperará a que todas las cámaras finalicen para luego finalizar.
  + A fin de realizar la prueba del programa, genere aleatoriamente los EVENTOS en cada cámara, queda a criterio del grupo la probabilidad de ocurrencia de cada EVENTO.

## **RESULTADOS OBTENIDOS**

## **CURSOR**

## Se generó un código pero no funcionaba correctamente.

## Hubo que refinarlo con 2 iteraciones donde lo corrigió automaticamente.

## Al final anduvo, pero tenía problemas con los íconos.

## Se generó correctamente el código, un manual de usuario y un archivo de prueba con el siguiente output por pantalla:

## **GPT**