Proyecto final – El maestro del ajedrez. Robots vs Humano

Simulación computacional.

MG. Carlos Andrés Delgado Saavedra.

Presentado por:

Álvaro Javier Quintero Peña – 201556009.

Kevin Santiago Lemos López – 201556168.

Stiveen Correa Escobar – 201556134.



Universidad del valle.

Sede Tuluá.

13 de septiembre de 2019.

**Documentación del modelo conceptual**

**Variables de entrada:** Tiempo de llegada de los robots, Cantidad de tiempo que durará la simulación, tamaño del tablero, tiempo entre llegadas de los robots, tipo de solución robot (para el tiempo de servicio), tipo de solución profesor (para el tiempo de servicio),

**Variables de estado:** tamaño cola robots, estado del profesor (si en partida o esperando a un robot), número de robots atendidos, tiempo transcurrido, número de partidas perdidas, número de partidas ganadas, ganancias del profesor

**Variables de desempeño:** Tamaño promedio de la cola, tiempo de solución promedio del profesor y del robot por cada n, promedio de n, tiempo promedio de espera en la cola, porcentaje de victorias, porcentaje derrotas

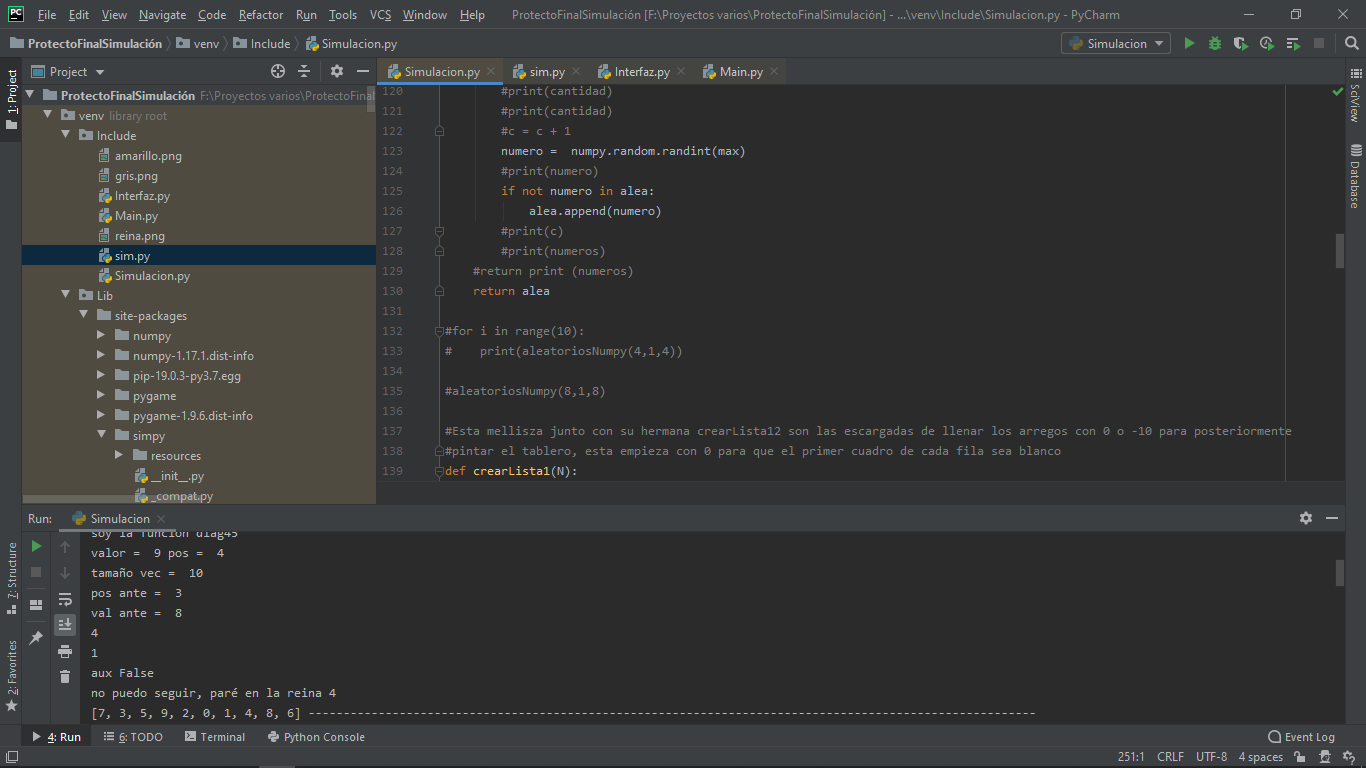
**Especificación de eventos (diagrama de flujo):**

**Especificación de la simulación:**

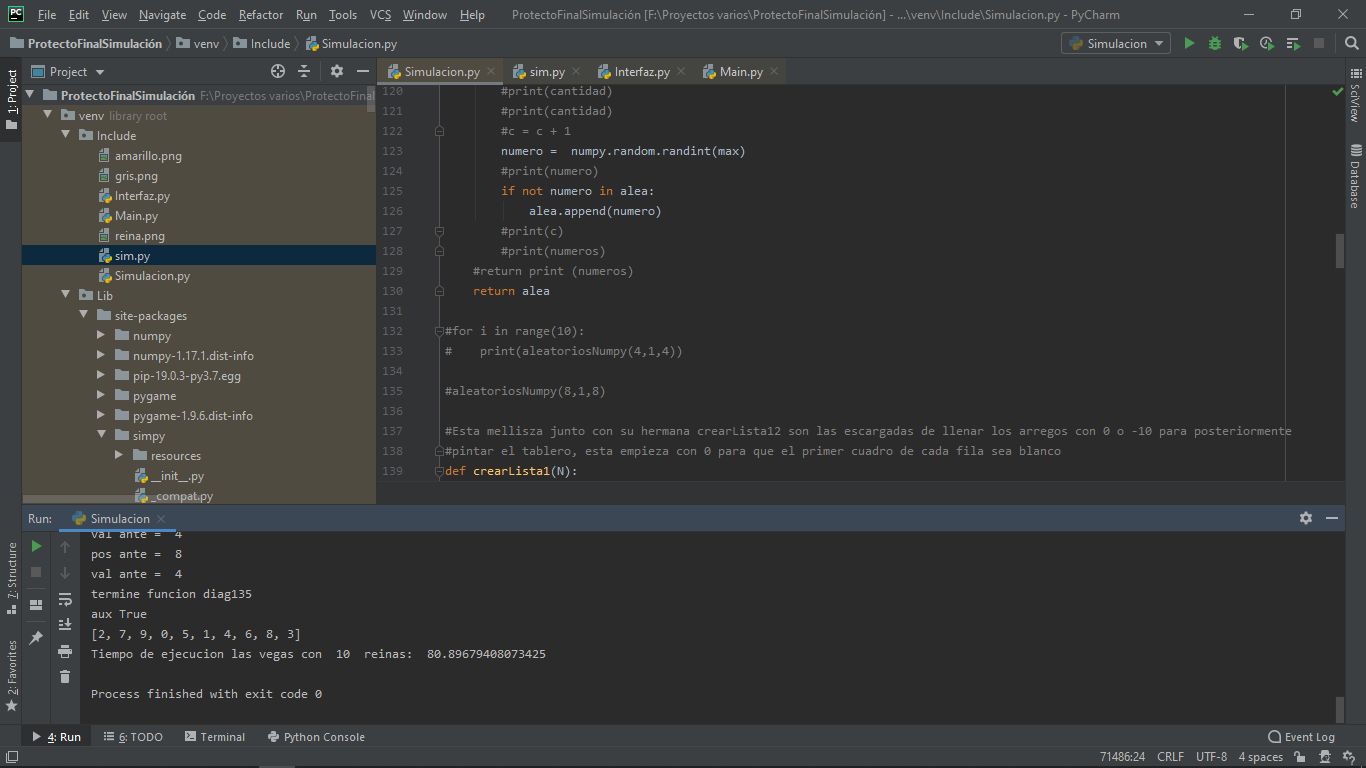
**Implementación Las Vegas**

Para nuestra solución tipo las vegas la cual será usada para la solución de los robots usamos 4 funciones principales, de las cuales 2 son las que se encargan de validar que una reina si pueda ser colocada en una posición i del arreglo, validando las diagonales hacia atrás de la posición, otra función es la encargada de generar las posiciones aleatorias en las que se ubicarán las reinas, esta genera un número aleatorio y verifica que no esté usado anteriormente (en otras palabras genera un número aleatorio entre los disponibles) y finalmente la función que llama a todas la funciones principales y auxiliares que se repite hasta dar solución al problema.

En cada iteración de esta última función, si el algoritmo las vegas falla, este admite que falló y retorna hasta que reina fue capaz de poner e intenta nuevamente llenar el tablero y cuando termina devuelve un vector con las posiciones de la reina



*Imagen si el algoritmo tipo las vegas falla*



*Imagen si el algoritmo tipo las vegas termina*

**Implementación Determinista (Back Tracking)**

Para nuestra solución determinista la cual será usada para la solución del profesor usamos 3 funciones, la primera encarga de todas las validaciones para que las reinas no se amenacen, la segunda que a medida que va avanzando en el arreglo va poniendo las reinas y finalmente la función recursiva que se repite hasta dar solución al problema.

La estrategia usada fue **Back Tracking**, la cual va poniendo reinas en la una posición factible, en caso que llegue a un punto en el cual no pueda seguir, este se devuelve y pone la reina anterior en la siguiente posición factible, así sucesivamente hasta encontrar una posición factible para todas las reinas.

Este algoritmo siempre termina con una solución

**Implementación Simpy**

Realizaremos una simulación en Python usando Simpy la cual es una librería Python para simular eventos discretos (simulación de llegadas y servicio).

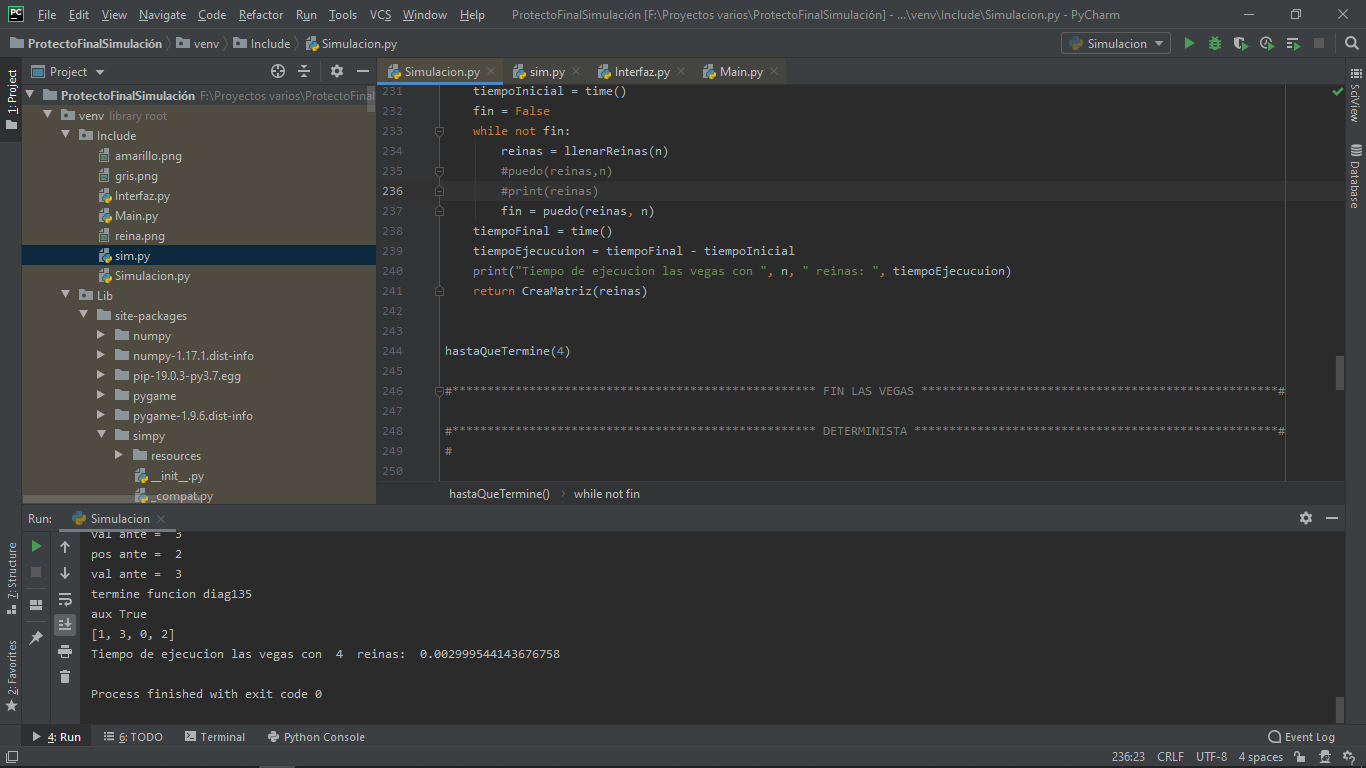
En este caso simularemos una maratón de ajedrez, específicamente en el problema de las n reinas, en el cual se enfrentará un profesor contra los robots quienes le han quitado el trabajo al profesor.

Las funciones utilizadas con esta librería fueron:

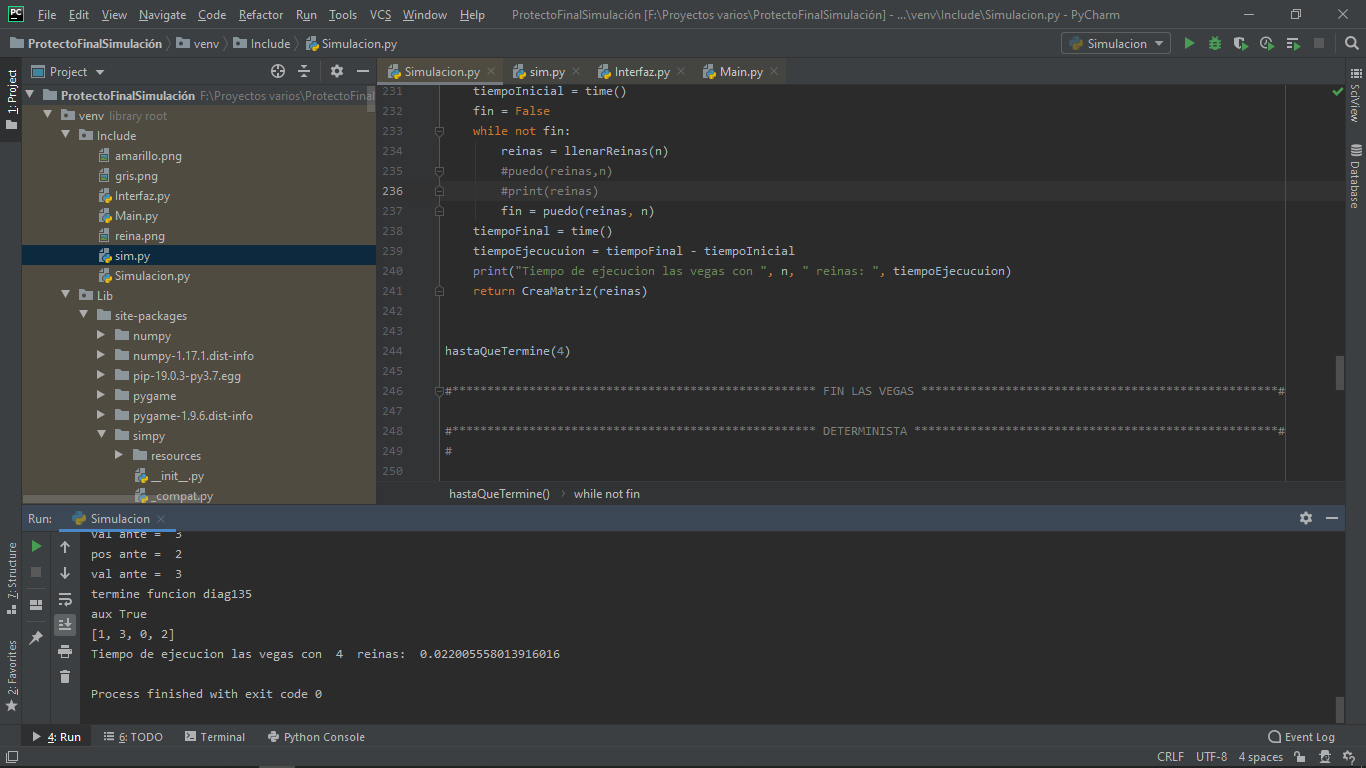
* Función llegada: Simula la llegada de un robot con una distribución uniforme (10, 30)
* Función cliente: simula la llegada de un tablero para que los dos algoritmos jueguen

**Tiempos de ejecución en segundos**

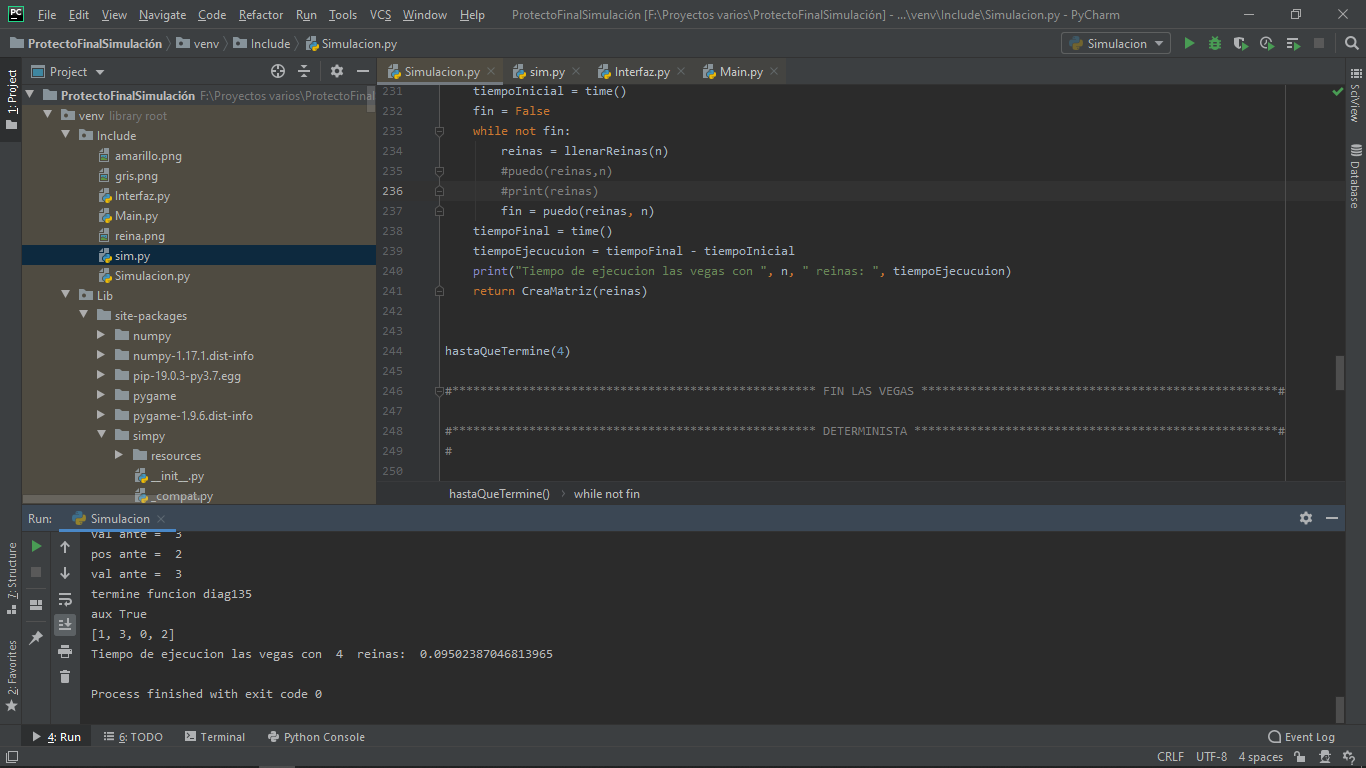
**N = 4 Las vegas**



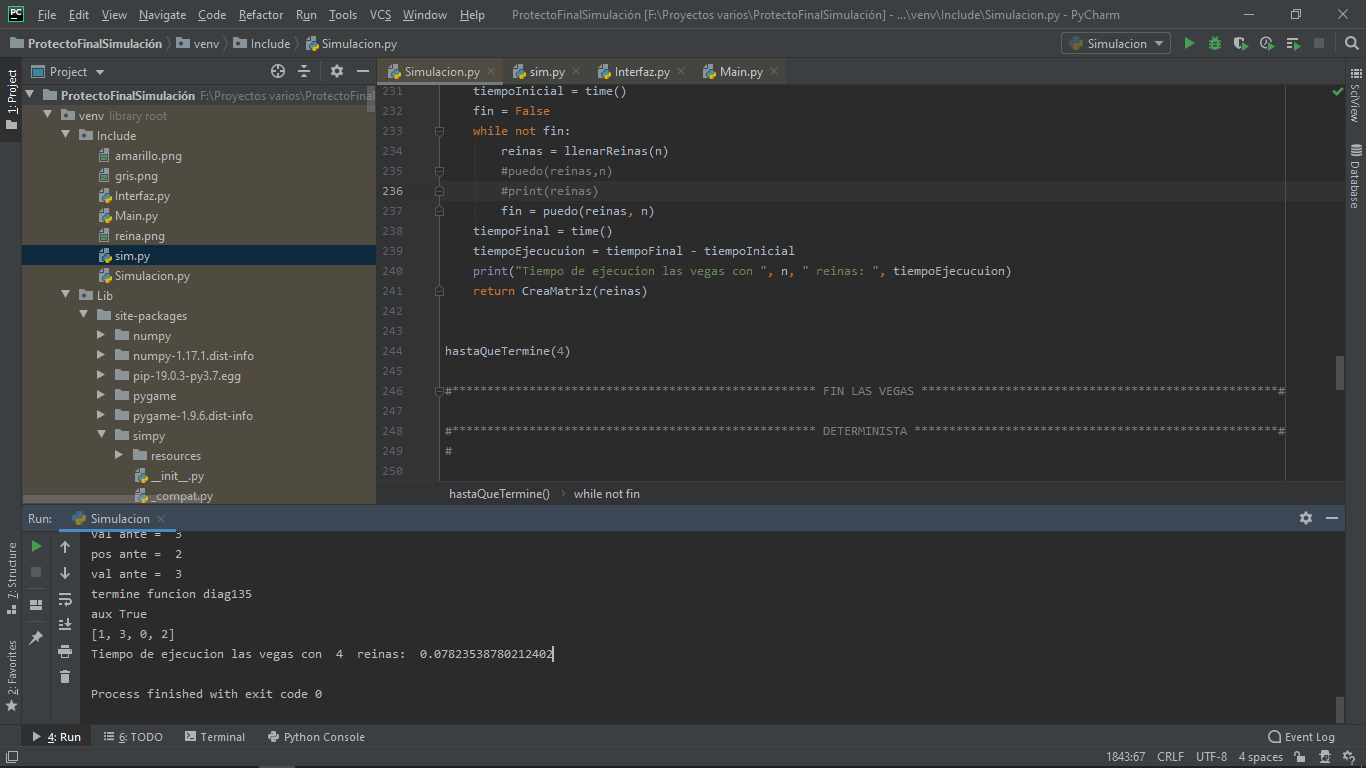
T1 = 0.002999544143676758



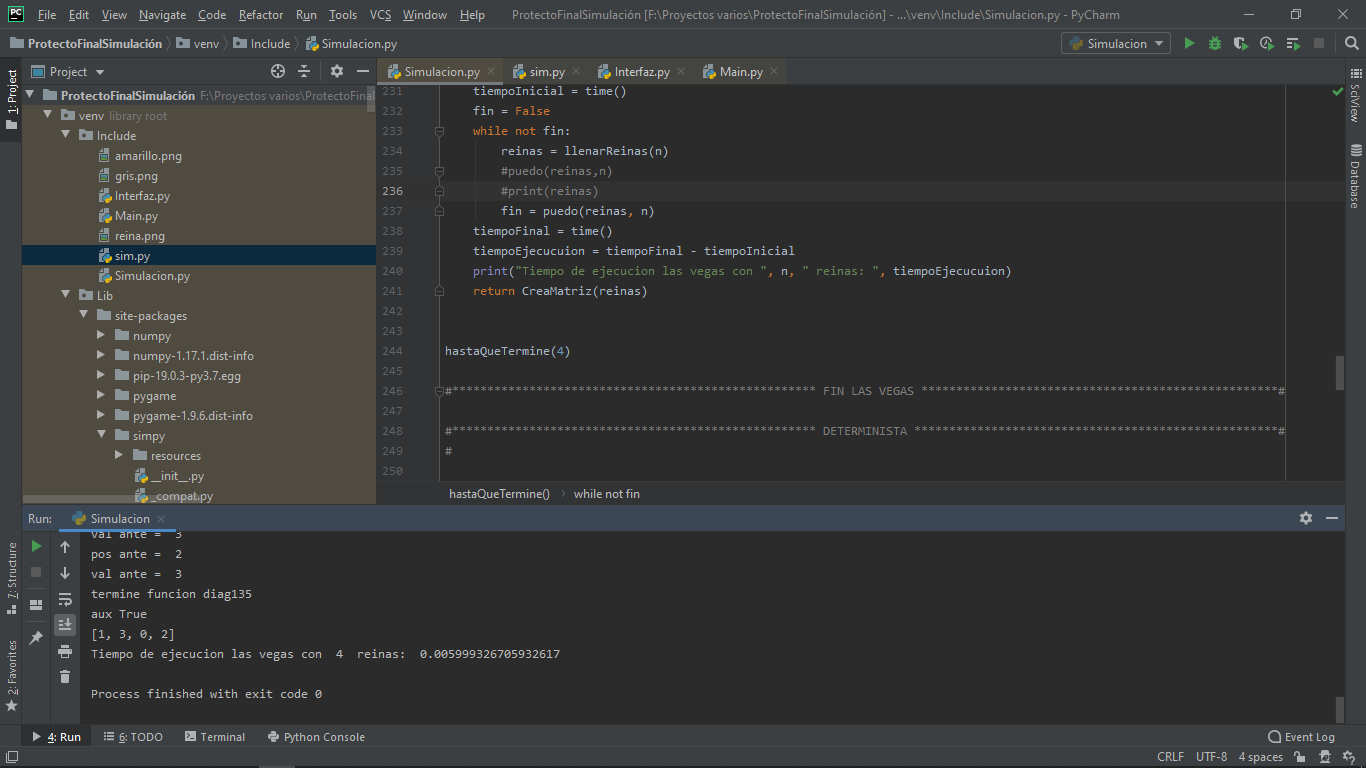
T2 = 0.022005558013916016



T3 = 0.09502387046813965



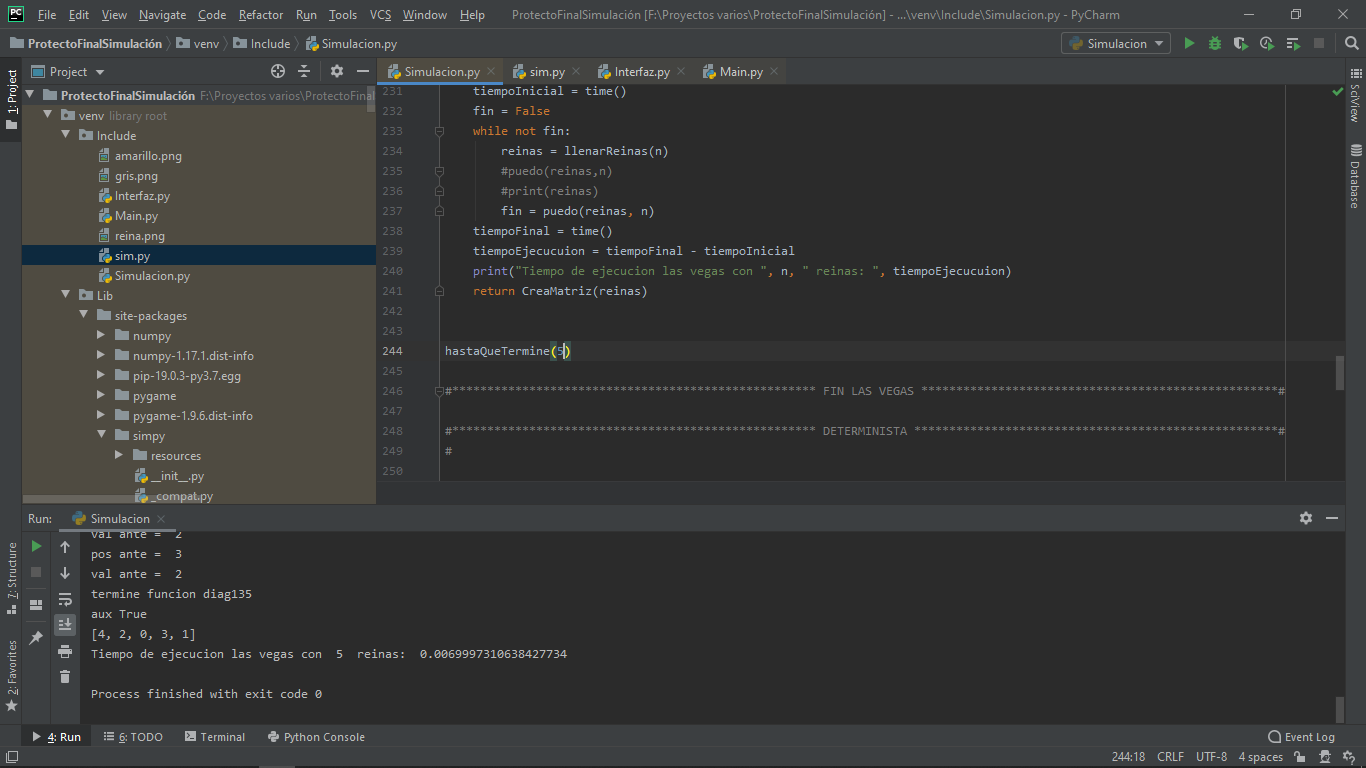
T4 = 0.07823538780212402



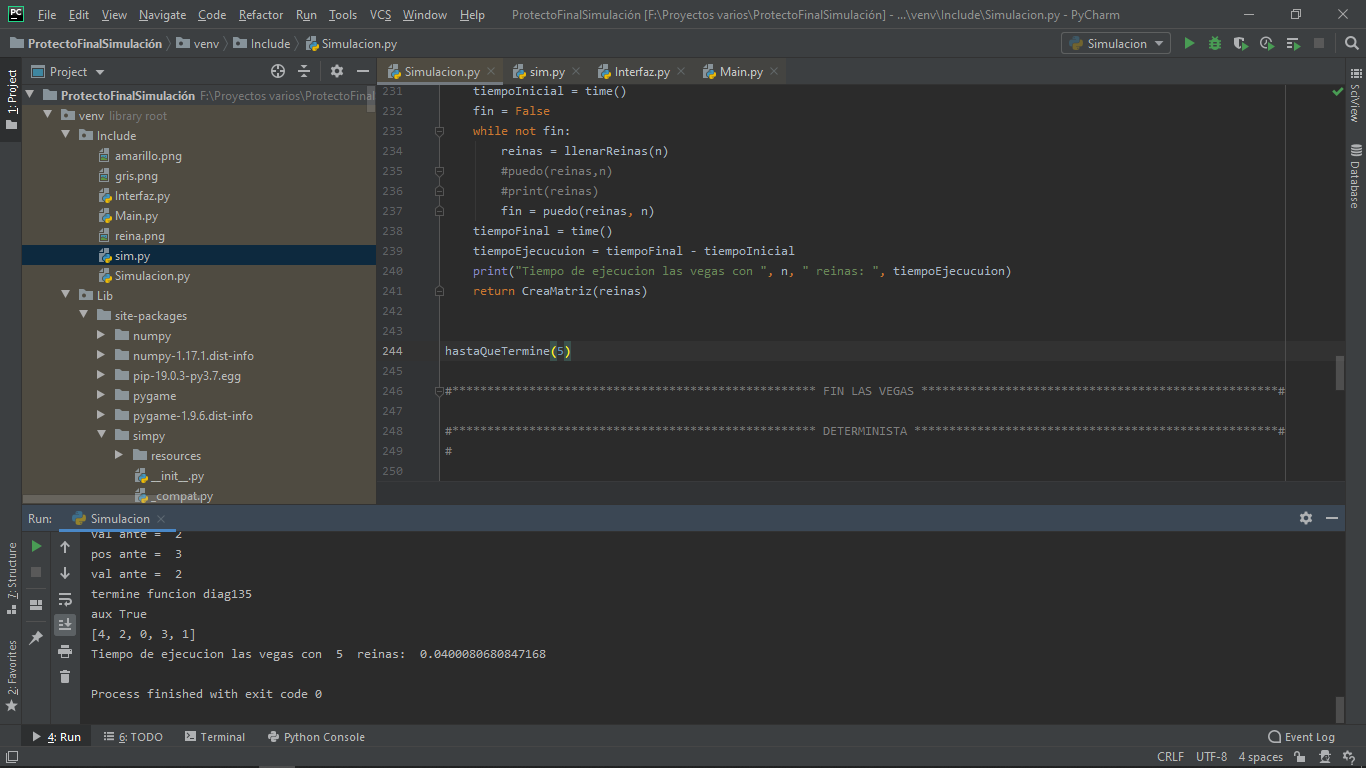
T5 = 0.005999326705932617

**Promedio tiempos las vegas con N = 4:** 0.04085273742 segundos

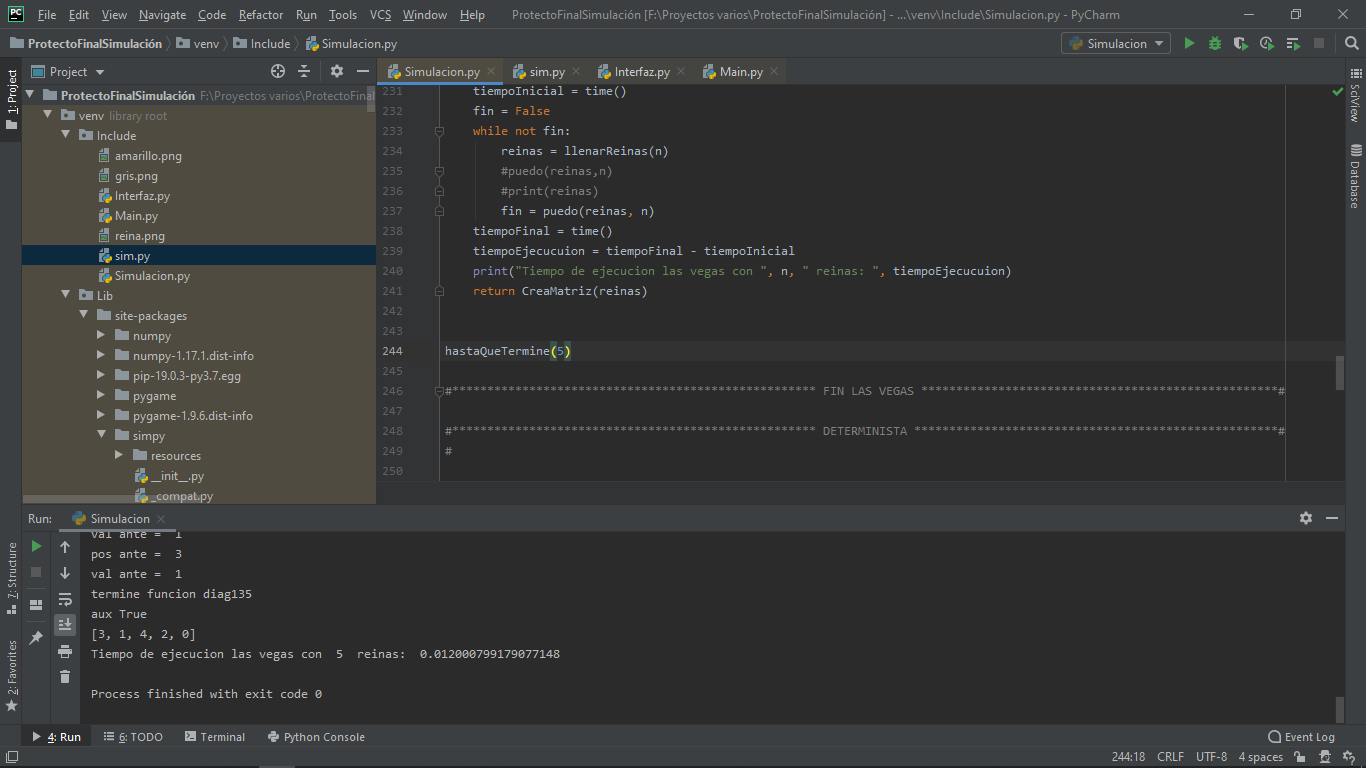
**N = 5 Las vegas**



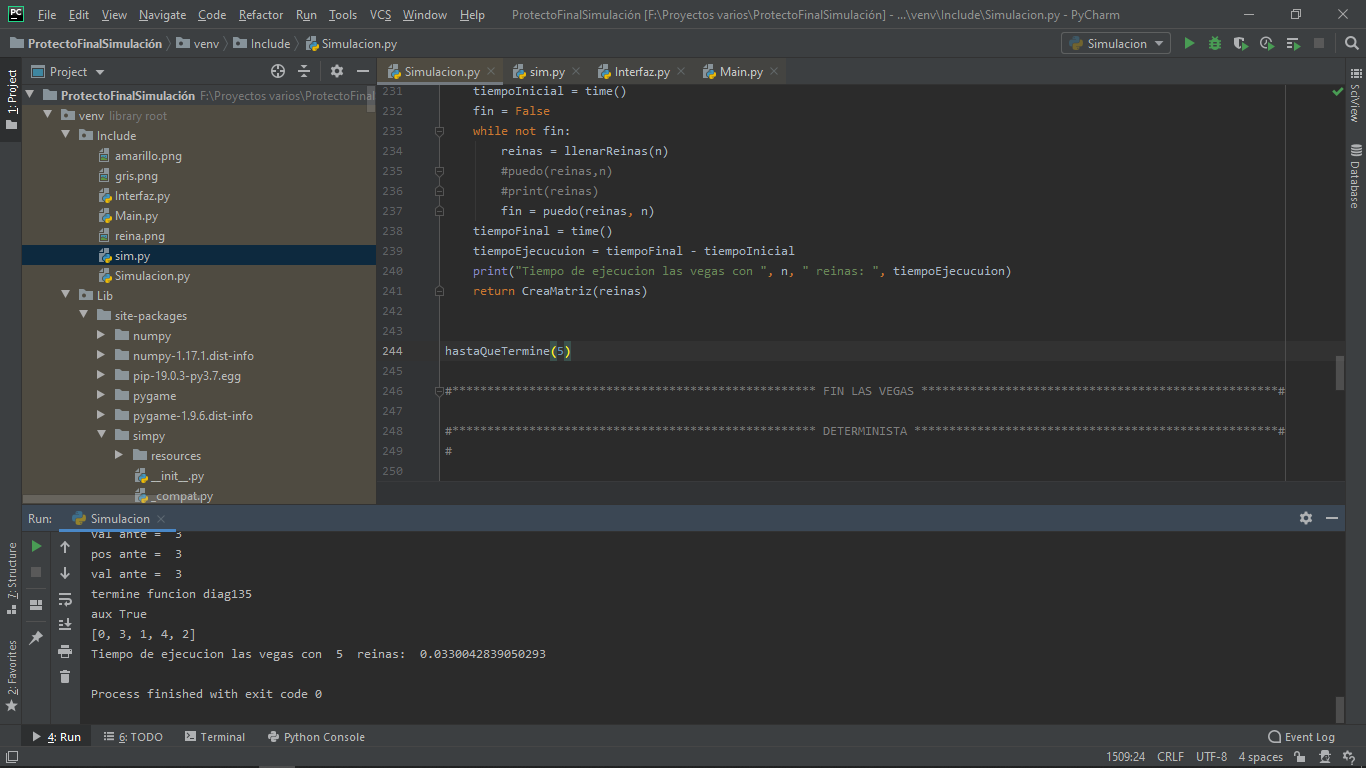
T1 = 0.0069997310638427734



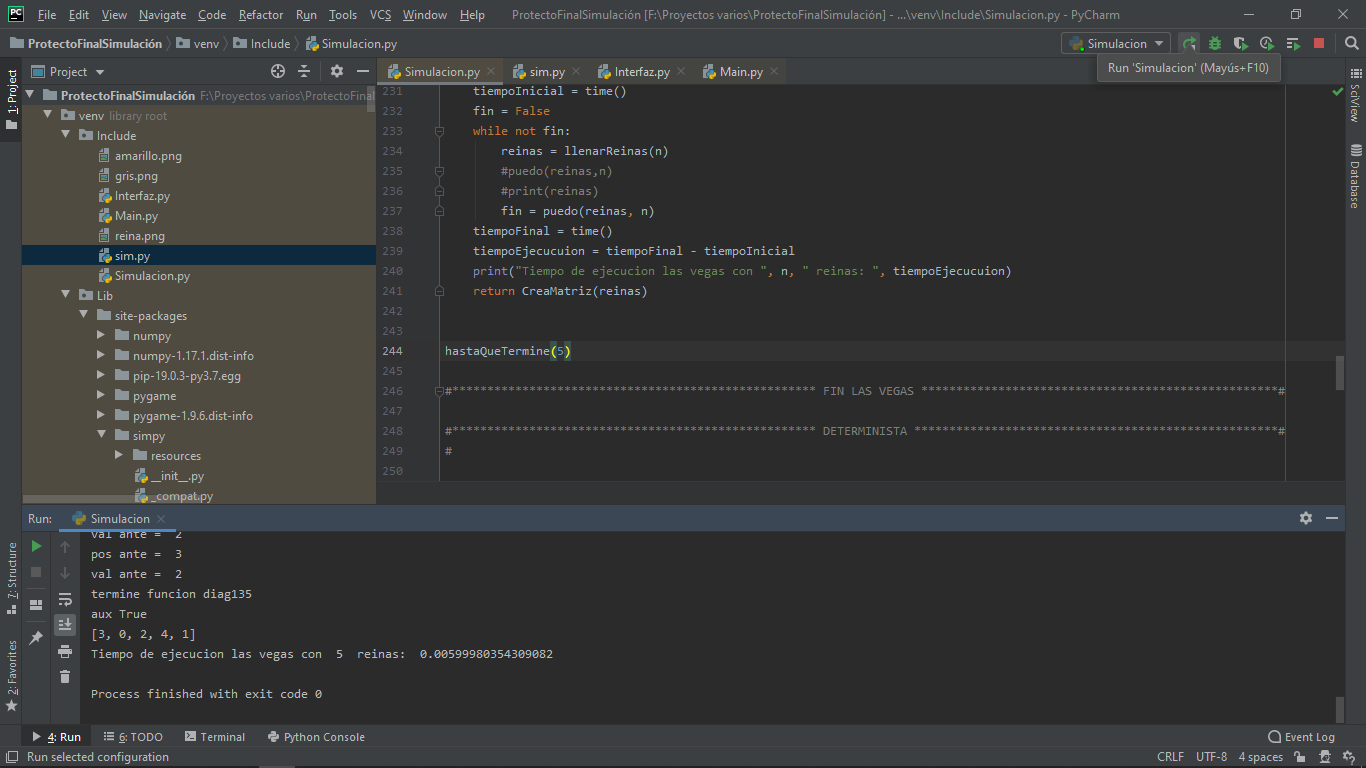
T2 = 0.0400080680847168



T3 = 0.012000799179077148



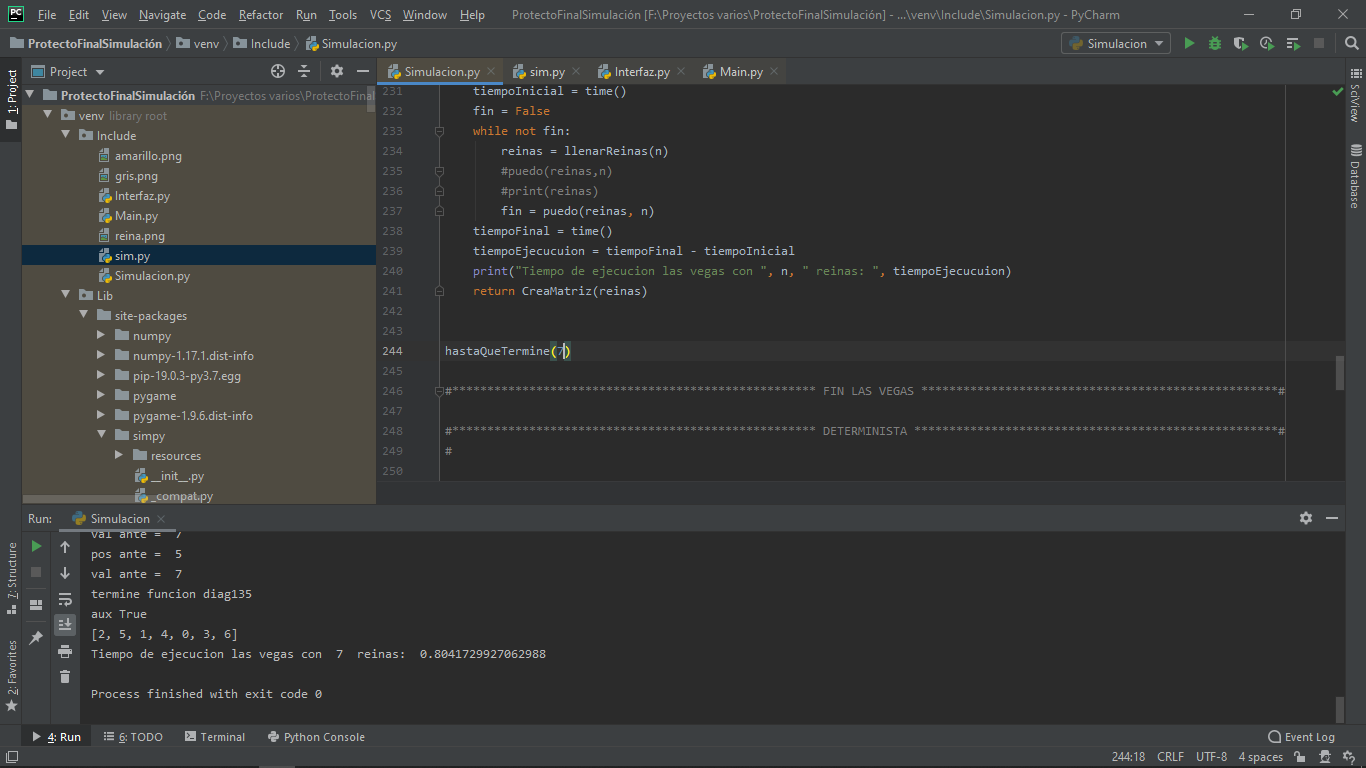
T4 = 0.0330042839050293



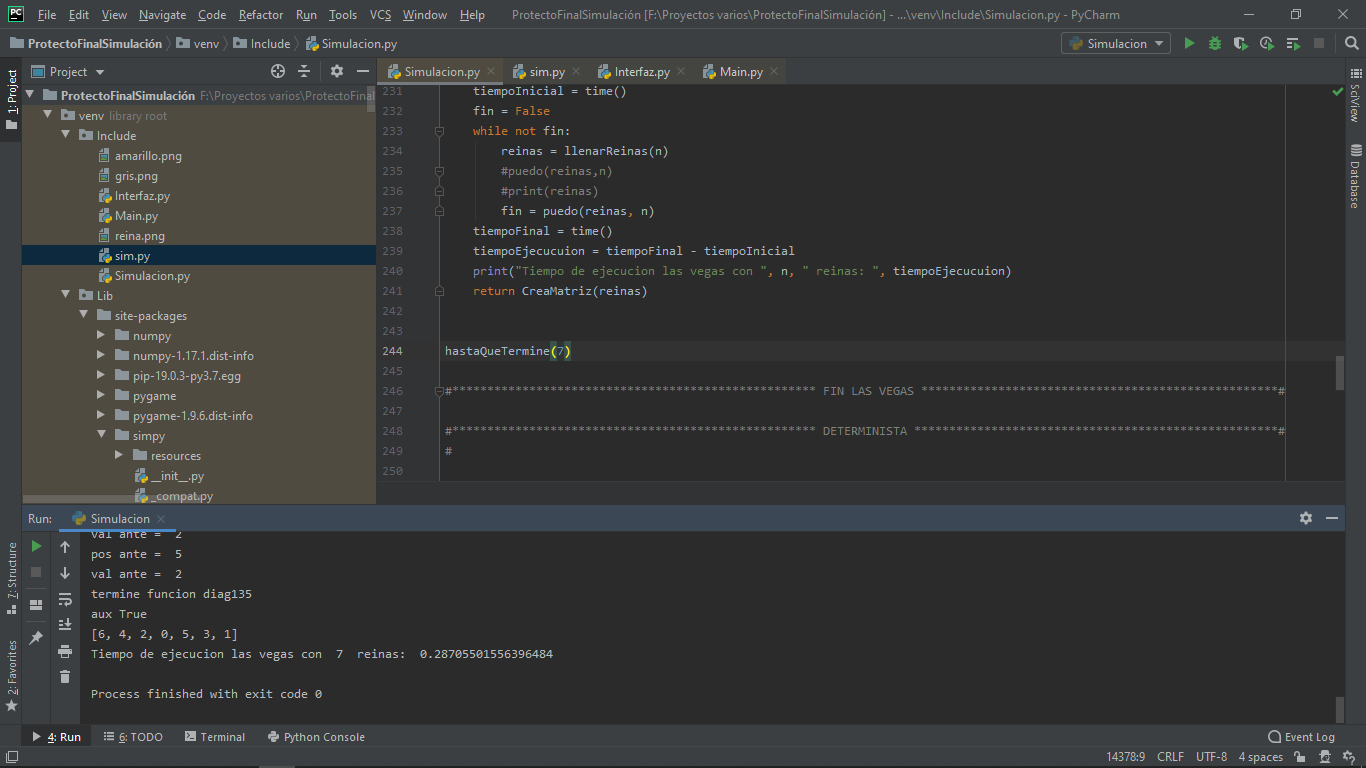
T5 = 0.00599980354309082

**Promedio Tiempos las vegas con N = 5:** 0.01960253715 segundos

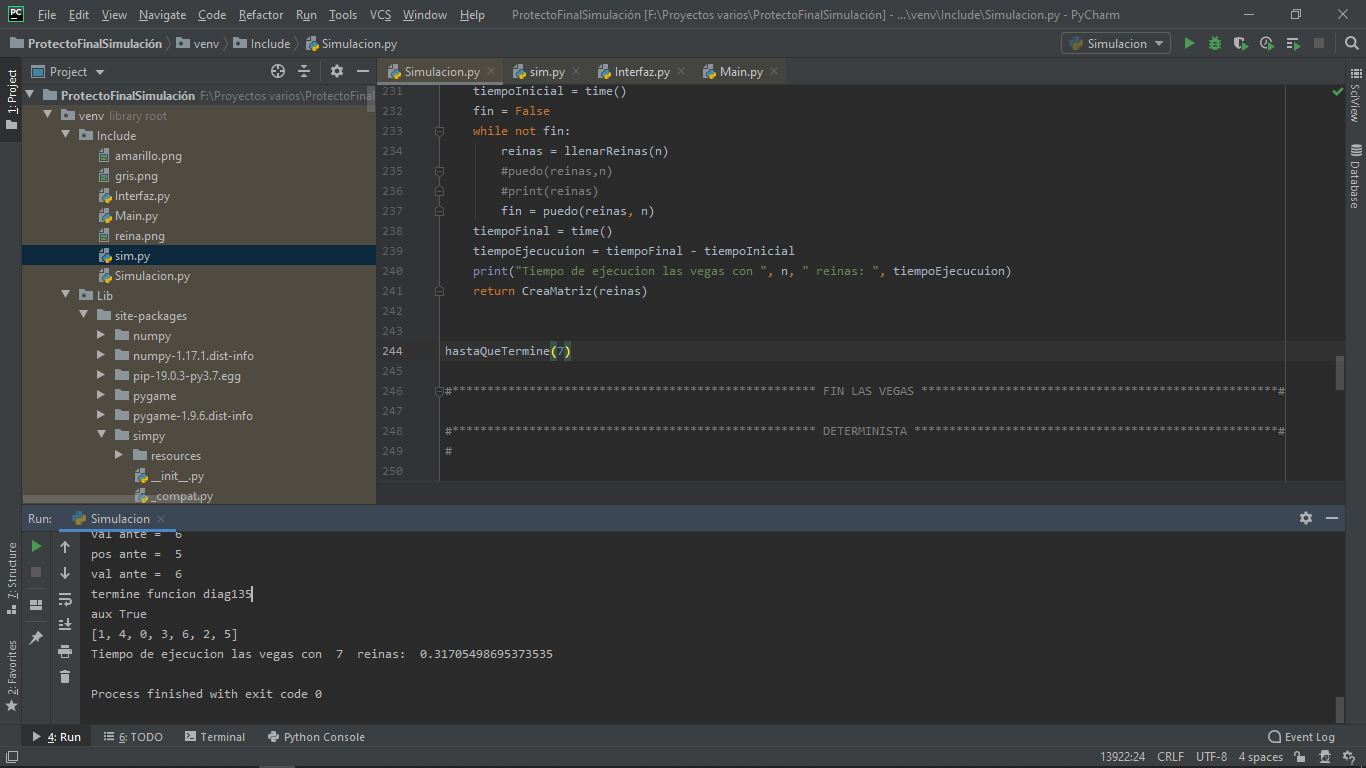
**N = 7 Las Vegas**



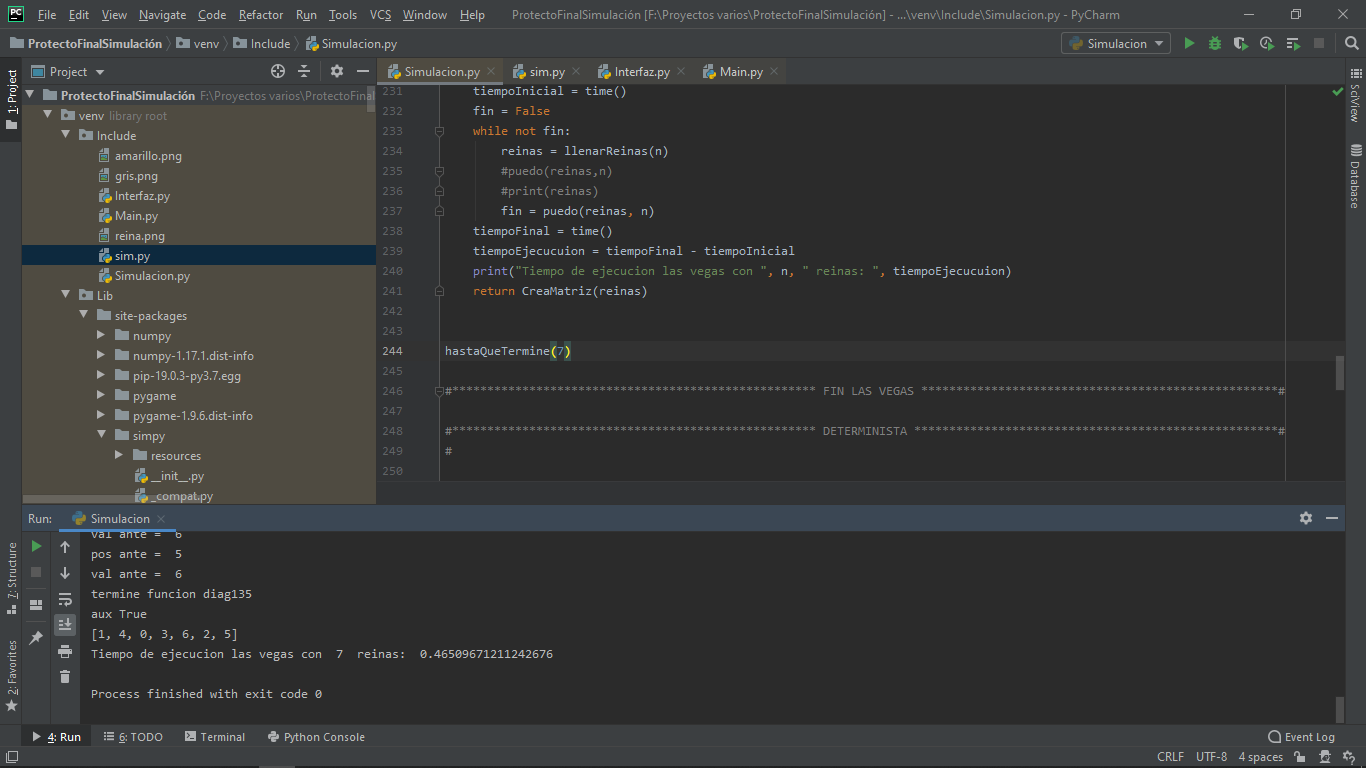
T1 = 0.8041729927062988



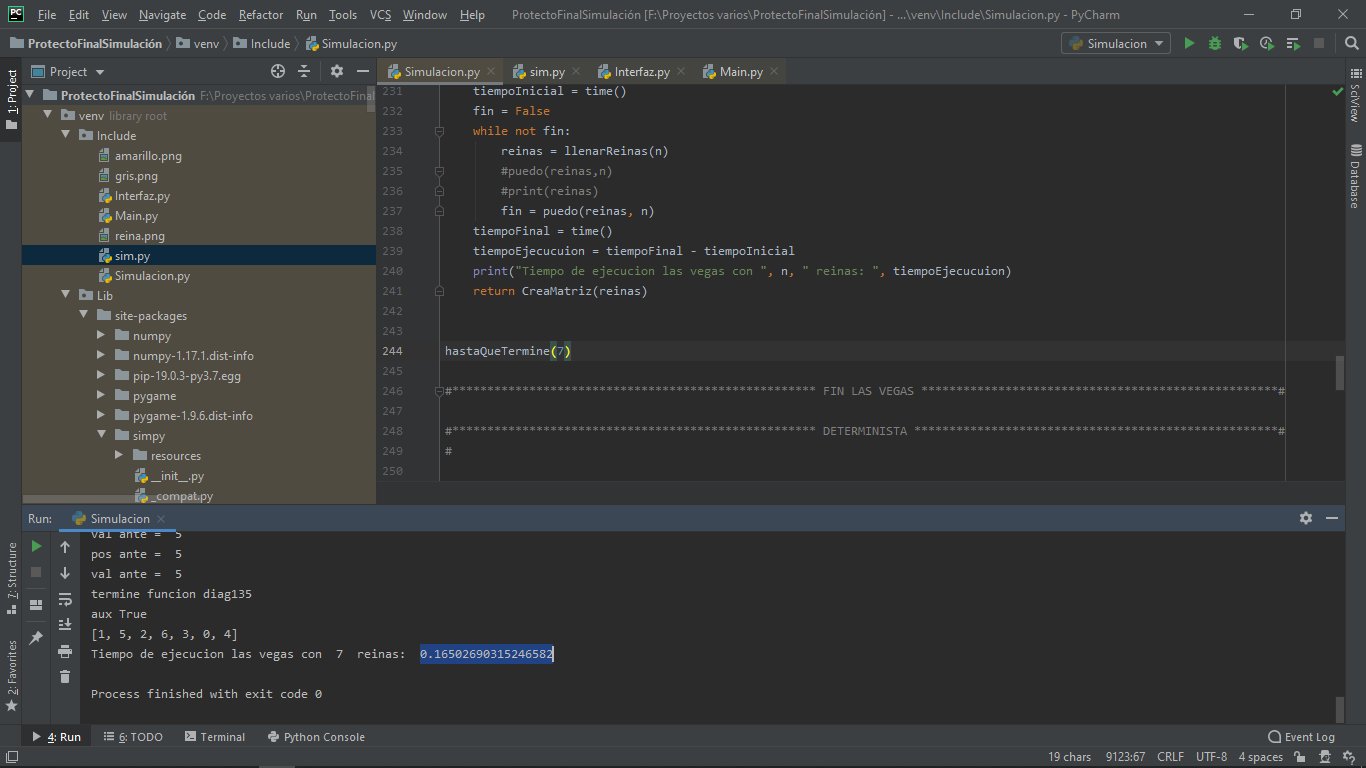
T2 = 0.28705501556396484



T3 = 0.31705498695373535



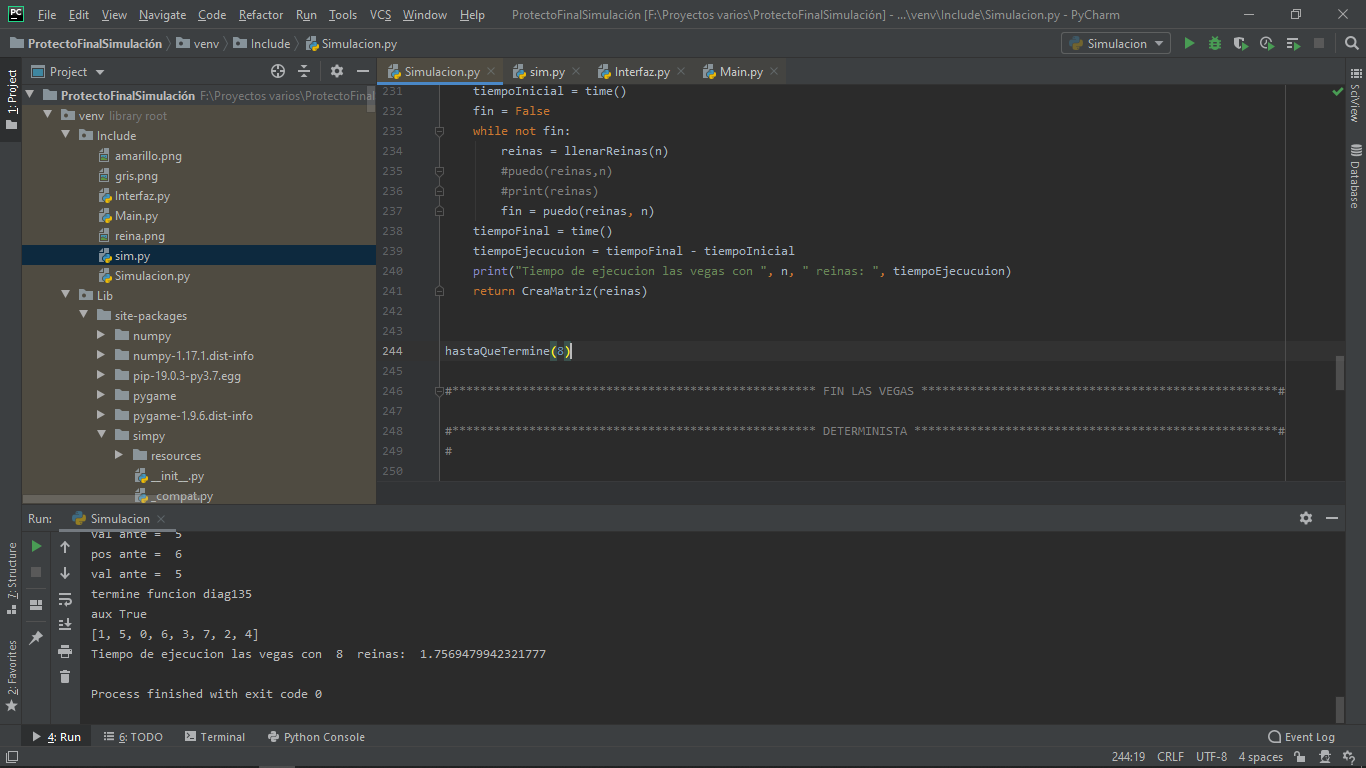
T4 = 0.46509671211242676



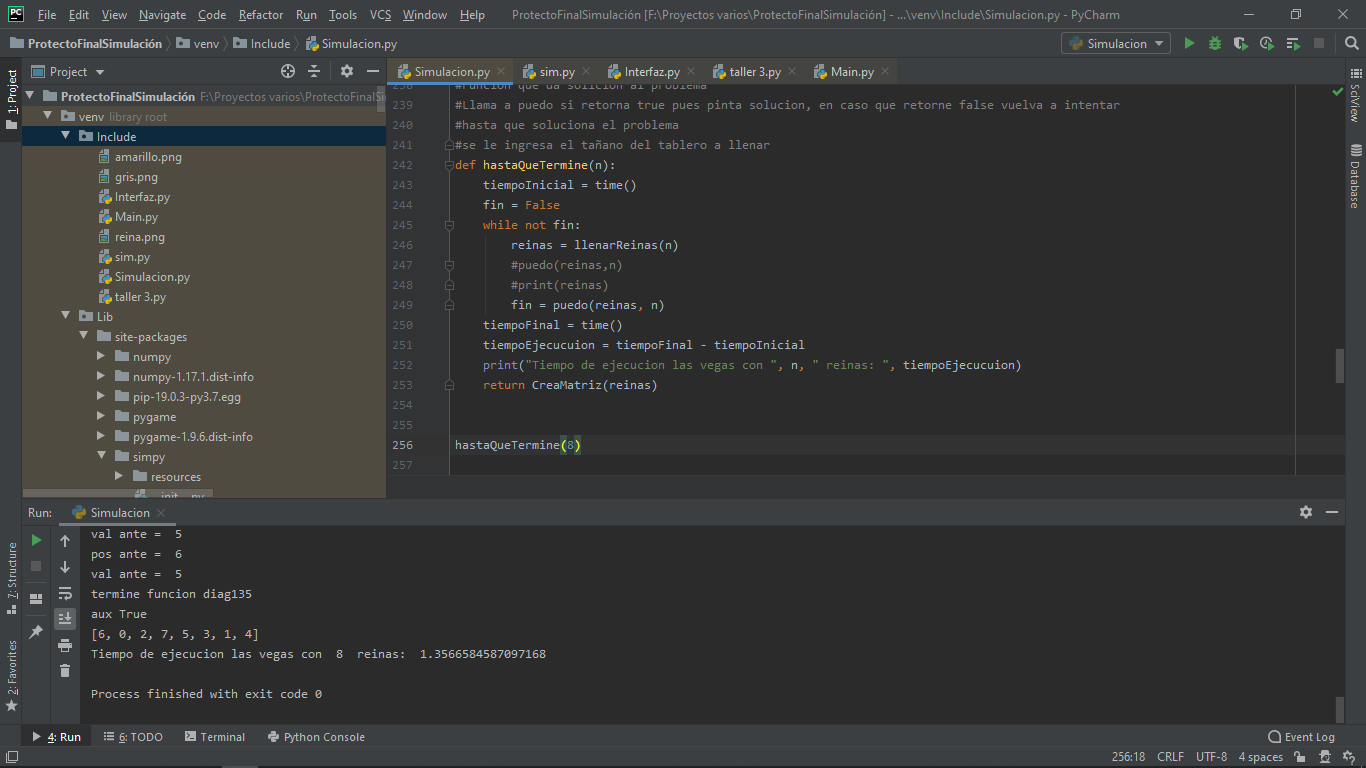
T5 = 0.16502690315246582

**Promedio tiempos las vegas con N = 7:** 0.40768132209 segundos

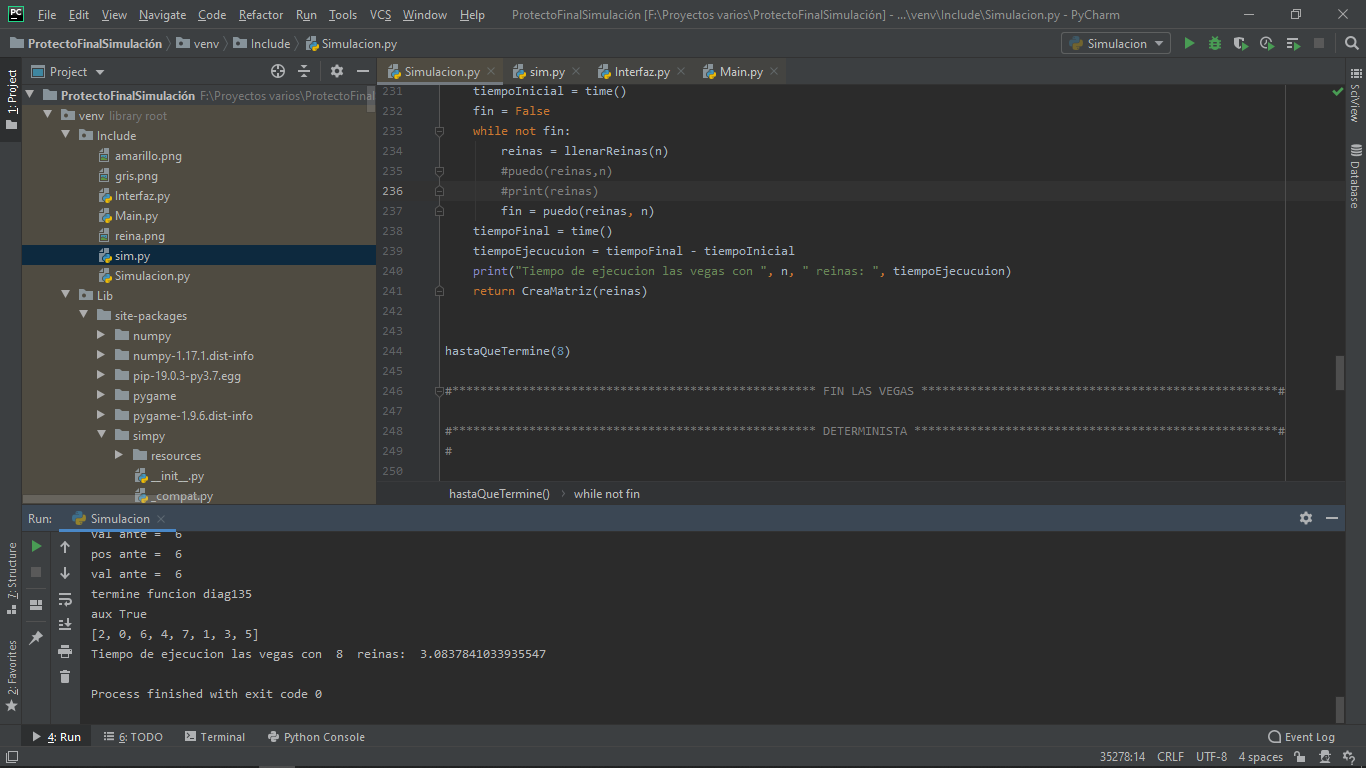
**N = 8 Las Vegas**



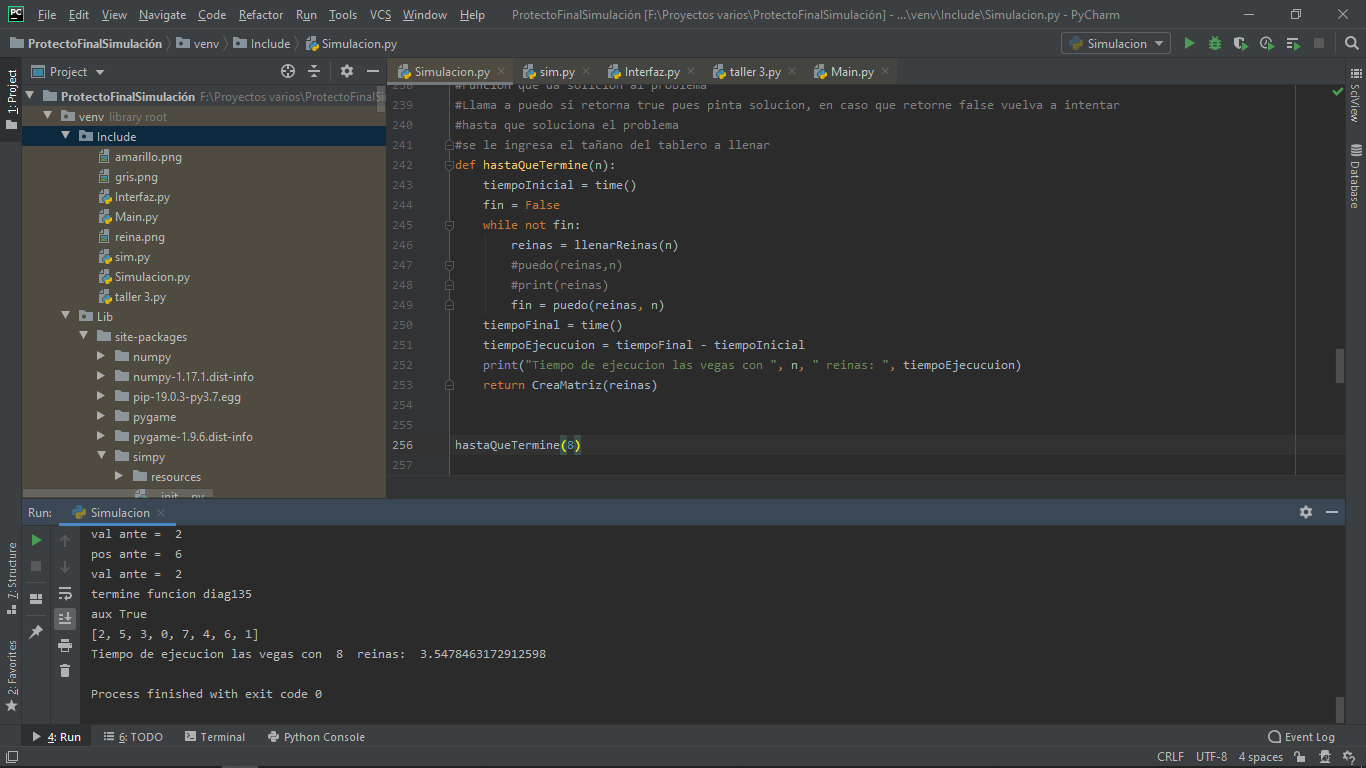
T1 = 1.7569479942321777



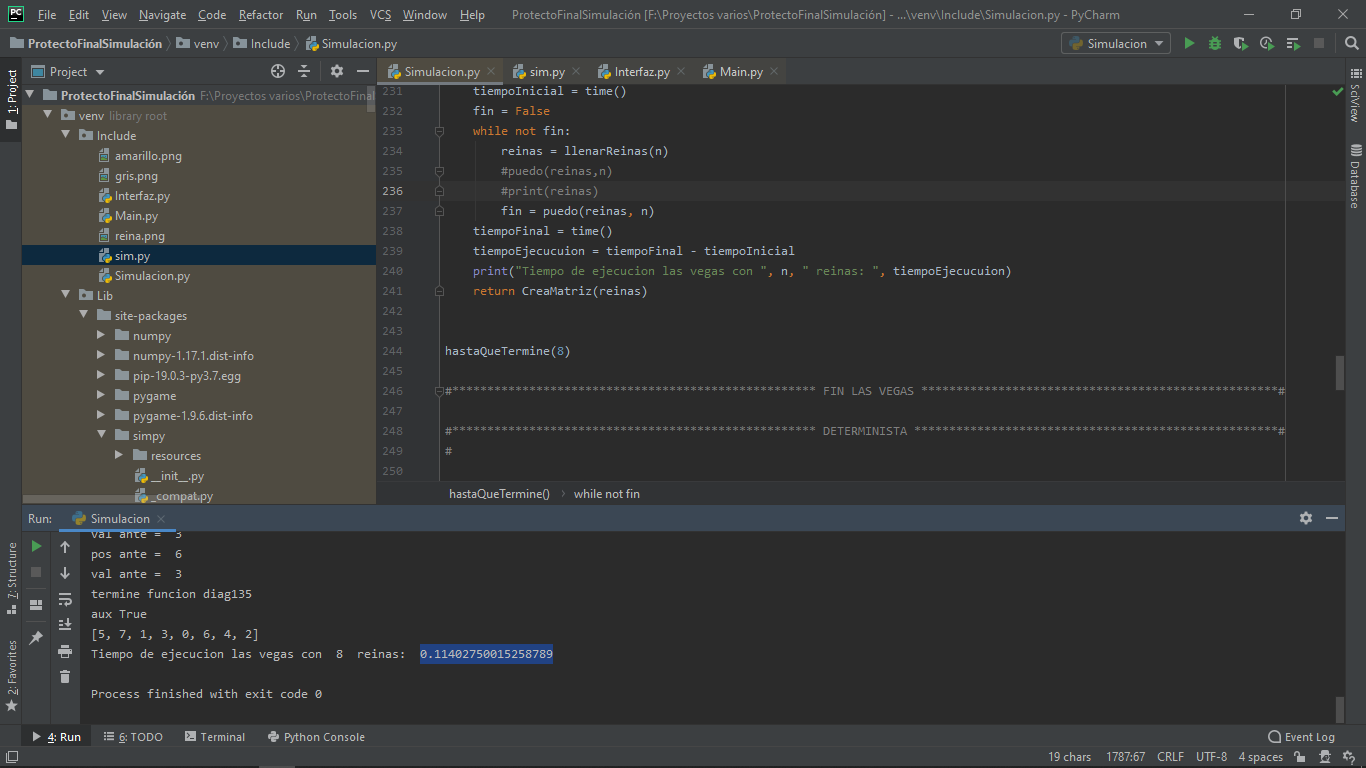
T2 = 1.3566584587097168



T3 = 3.0837841033935547



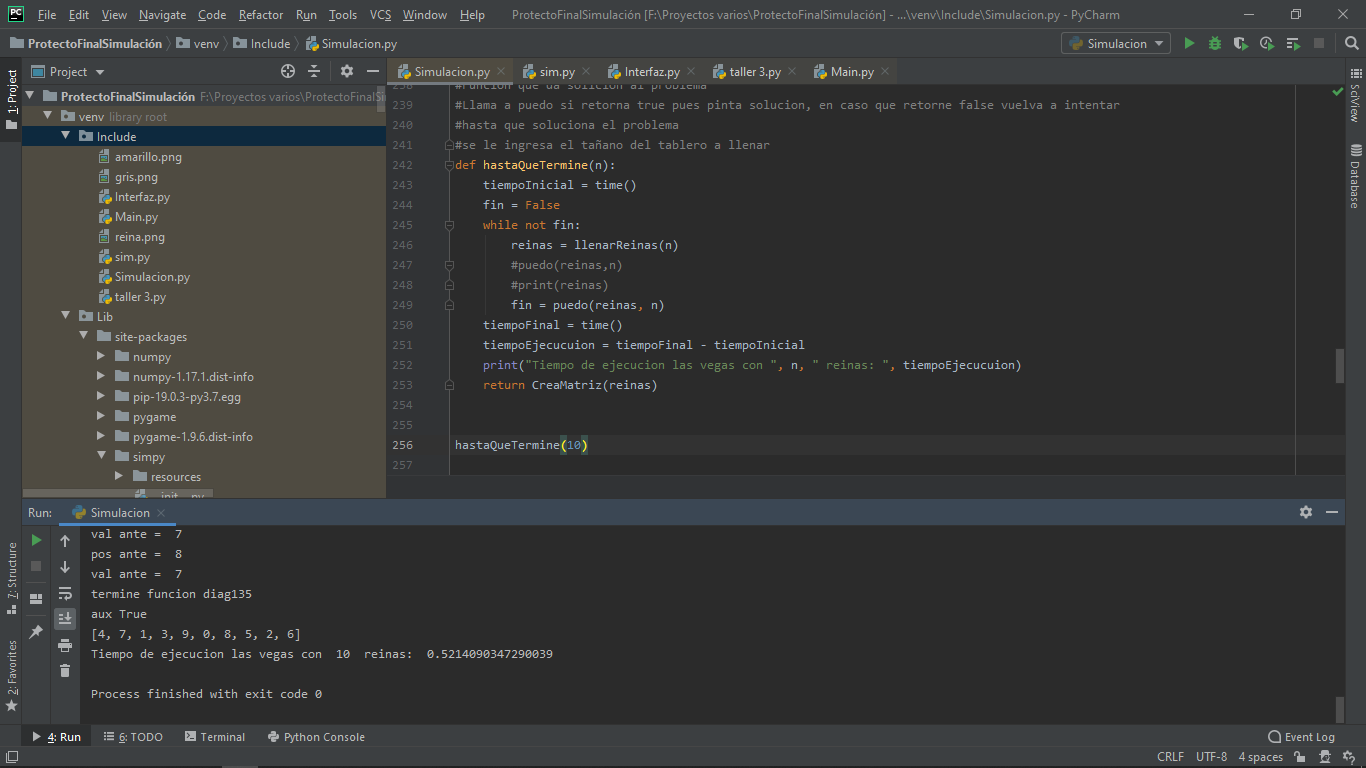
T4 = 3.5478463172912598



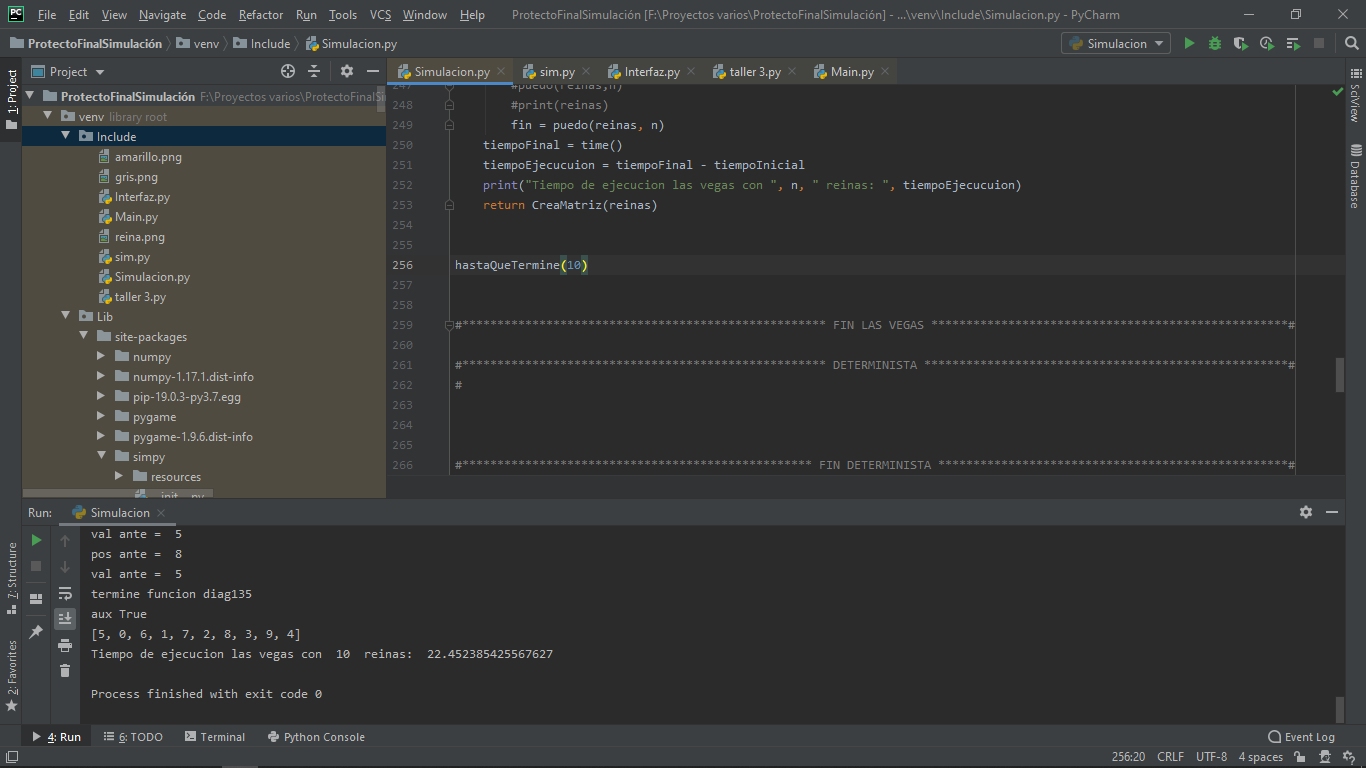
T5 = 0.11402750015258789 OOOMMMMGGGG

**Promedio tiempos las vegas con N = 8:** 1.97185287476 segundos

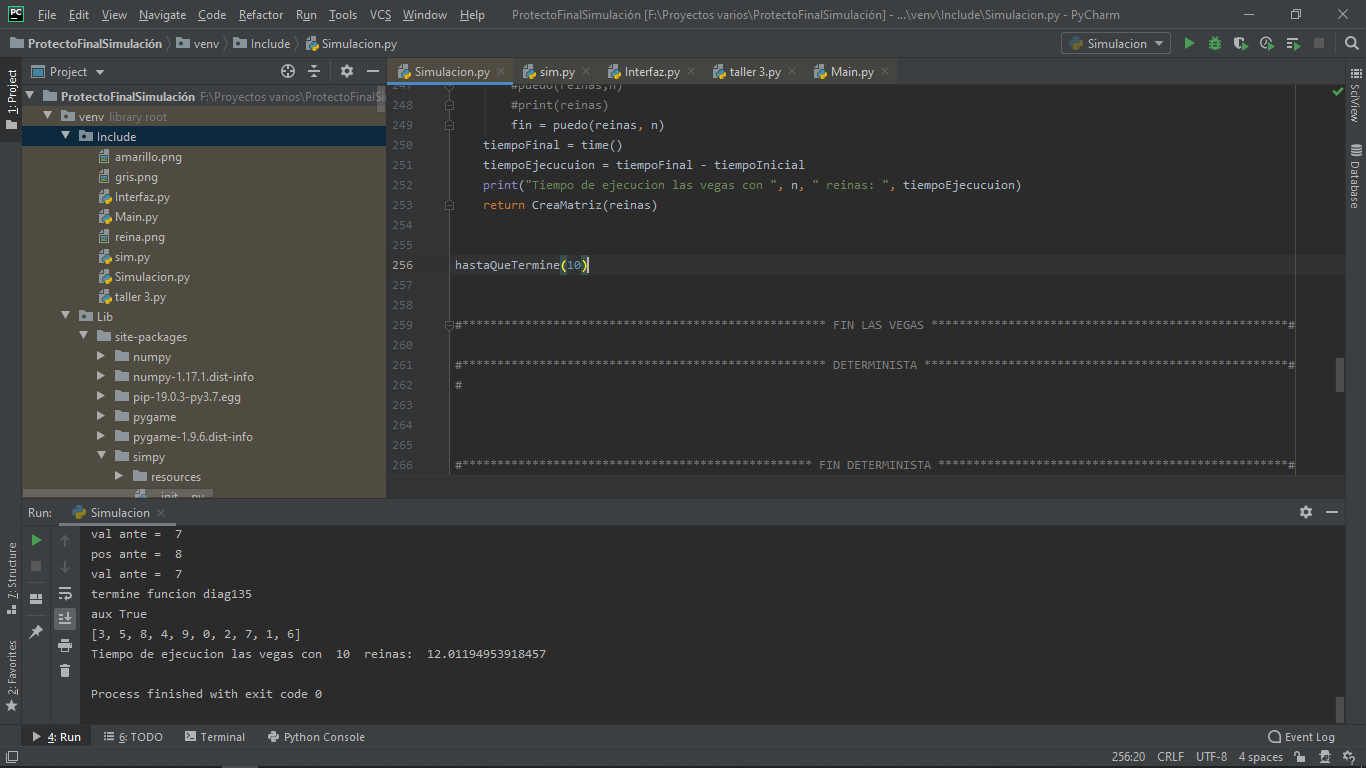
**N = 10 Las vegas**



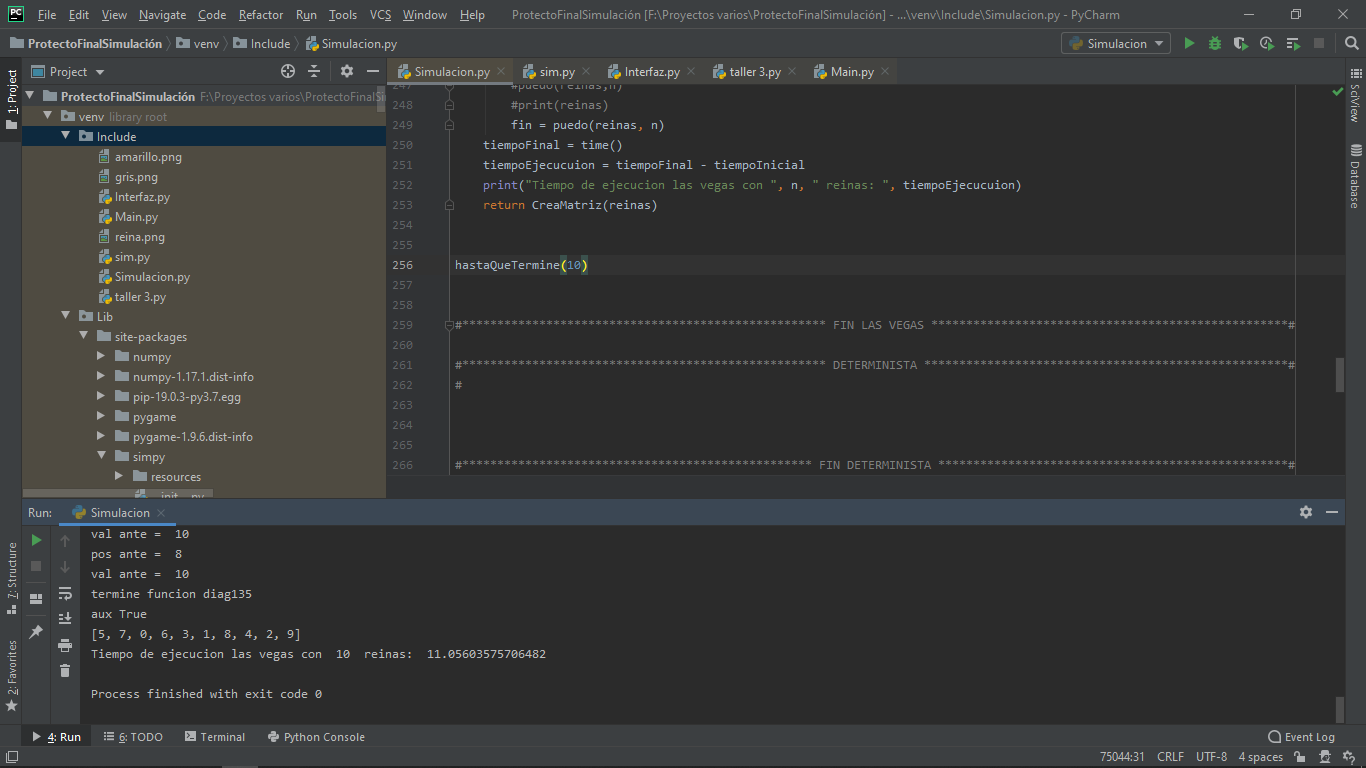
T1 = 0.5214090347290039



T2 = 22.452385425567627



T3 = 12.01194953918457



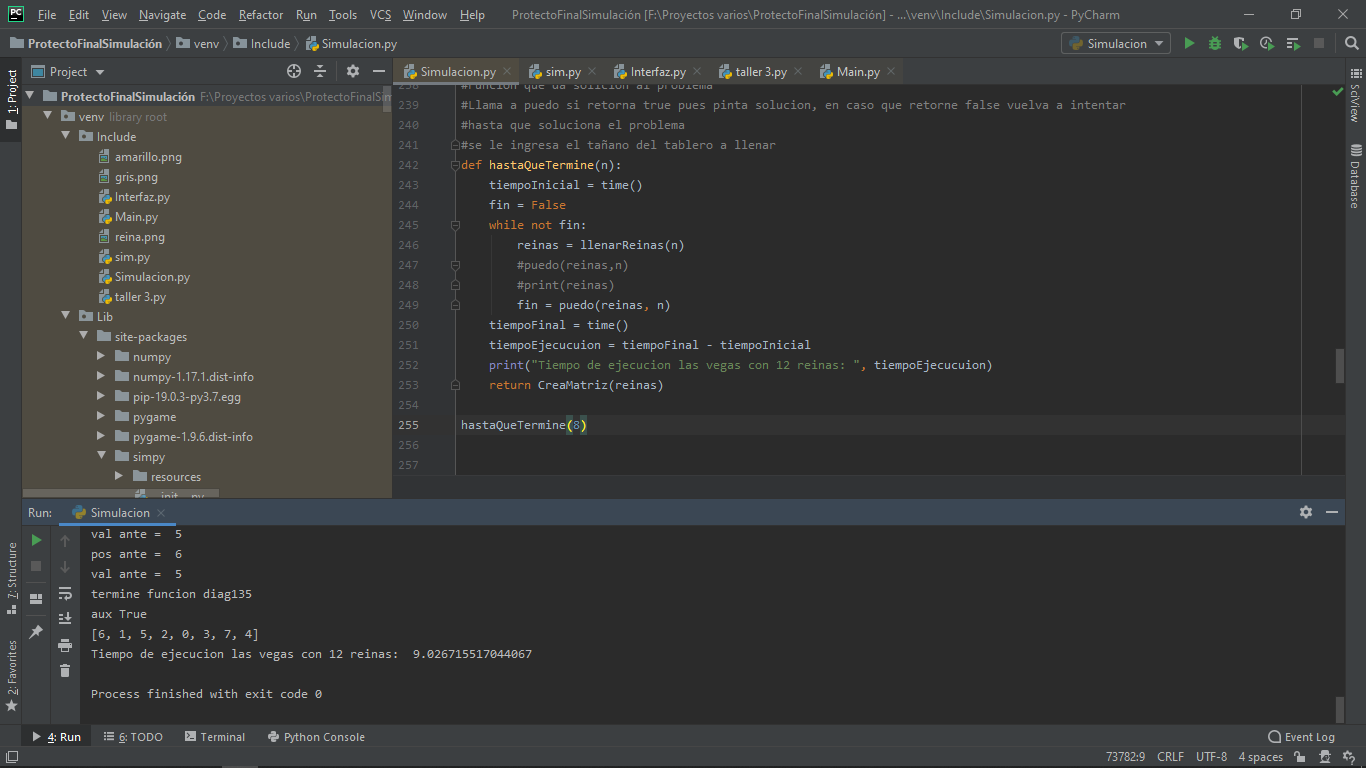
T4 = 11.05603575706482



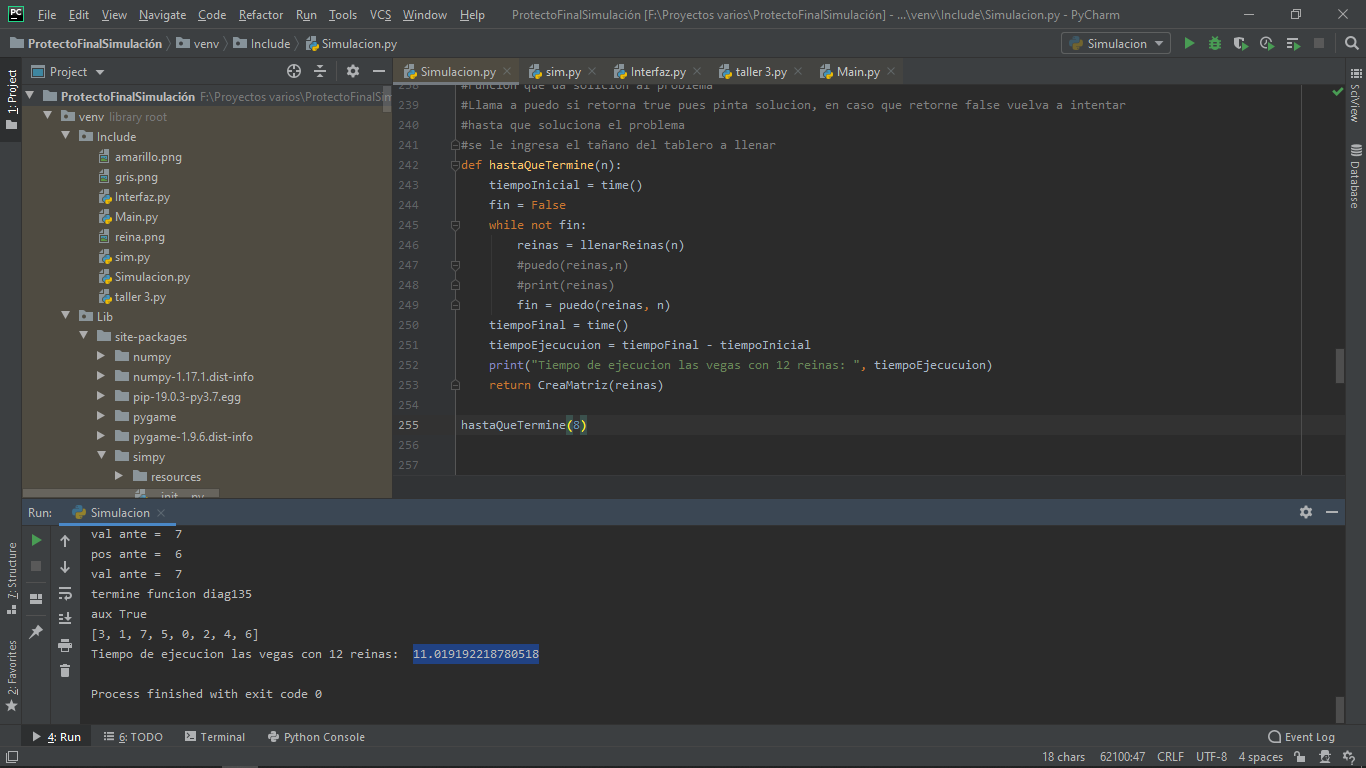
T5 = 16.70023274421692

**Promedio tiempos las vegas con N = 10:** 12.5484025002 segundos

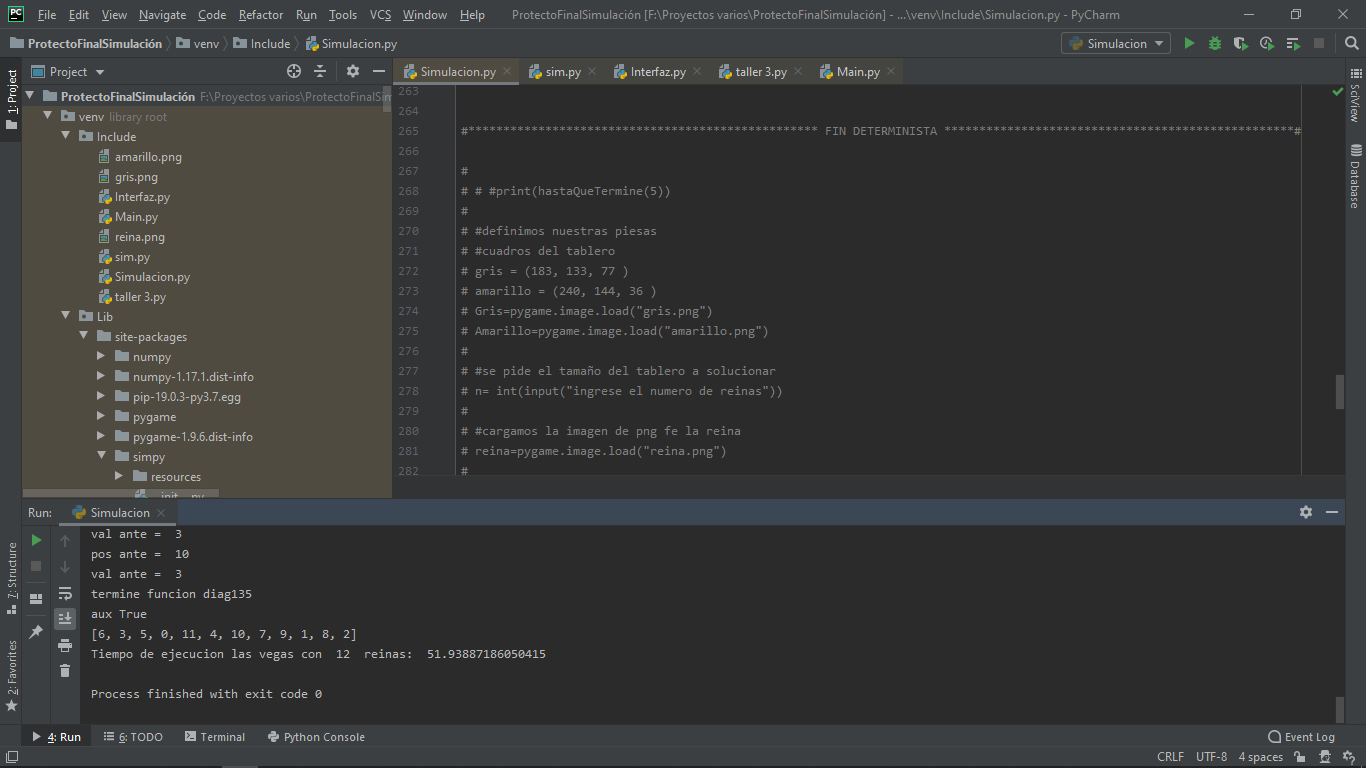
**N = 12 Las vegas**



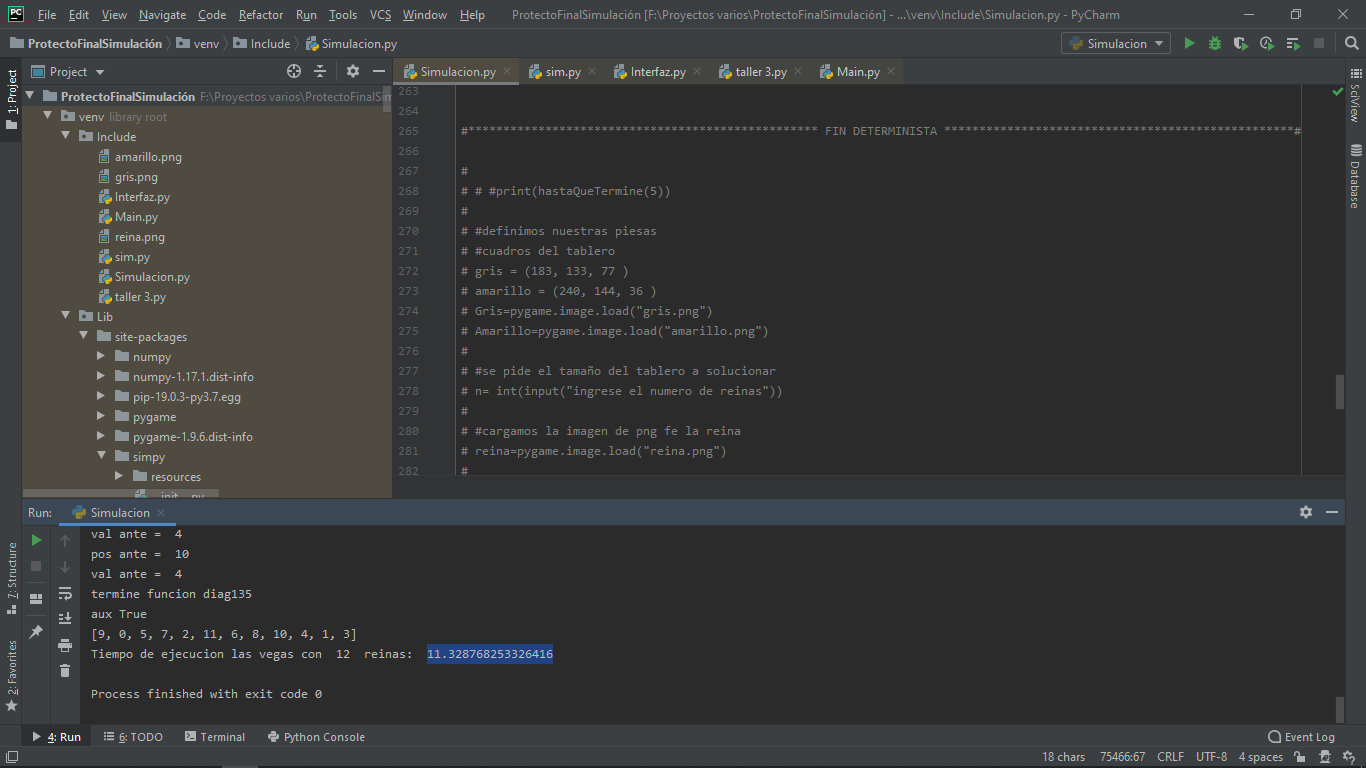
T1 = 9.026715517044067



T2 = 11.019192218780518



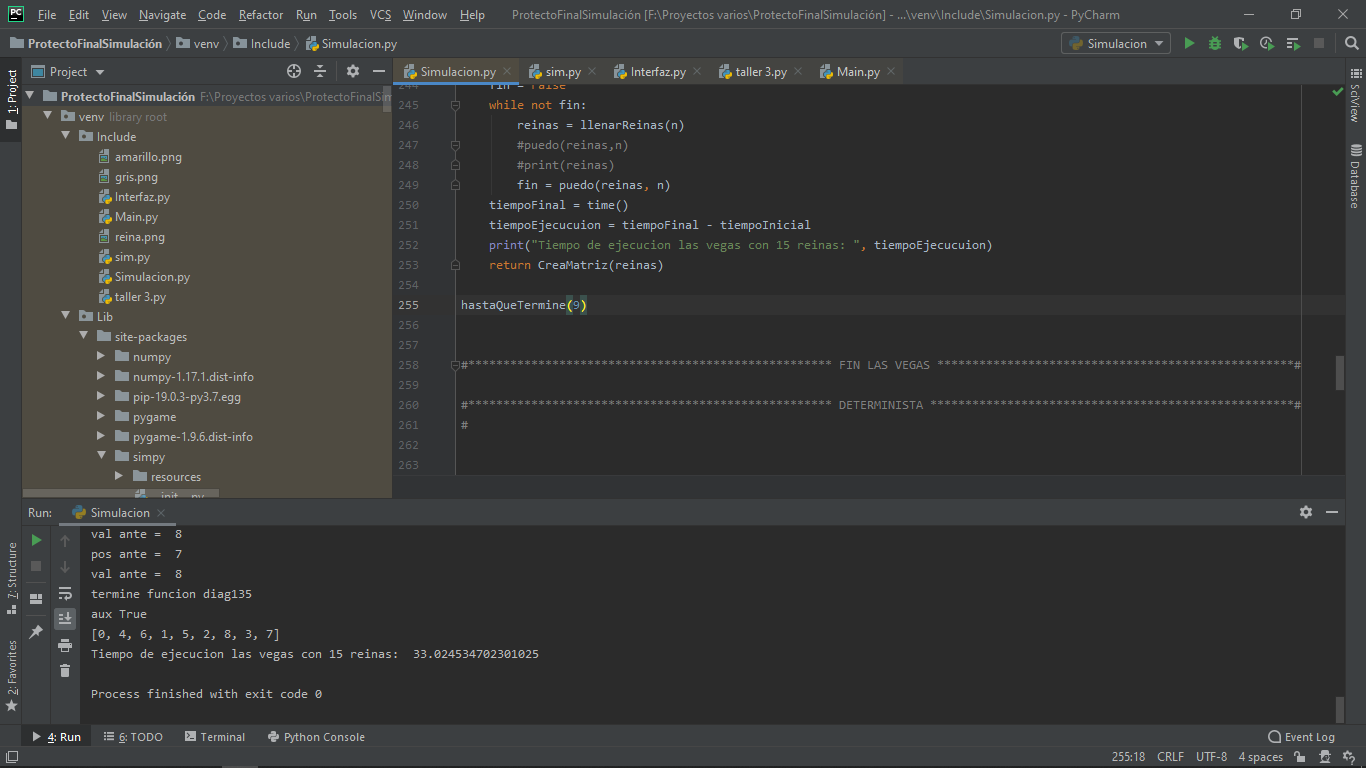
T3 = 51.93887186050415



T4 = 11.328768253326416

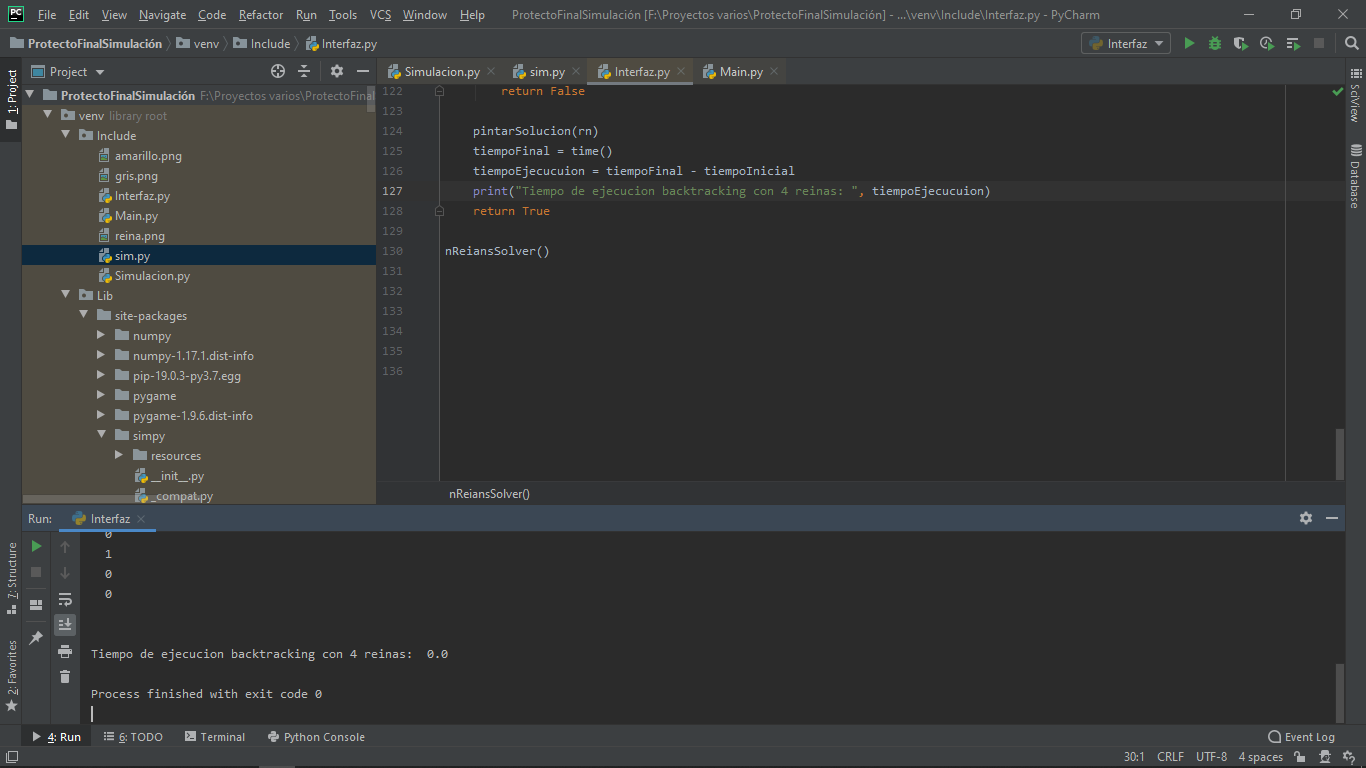
**Promedio tiempos las vegas con N = 12:** 20.8283869624 segundos

**N = 15 Las Vegas**

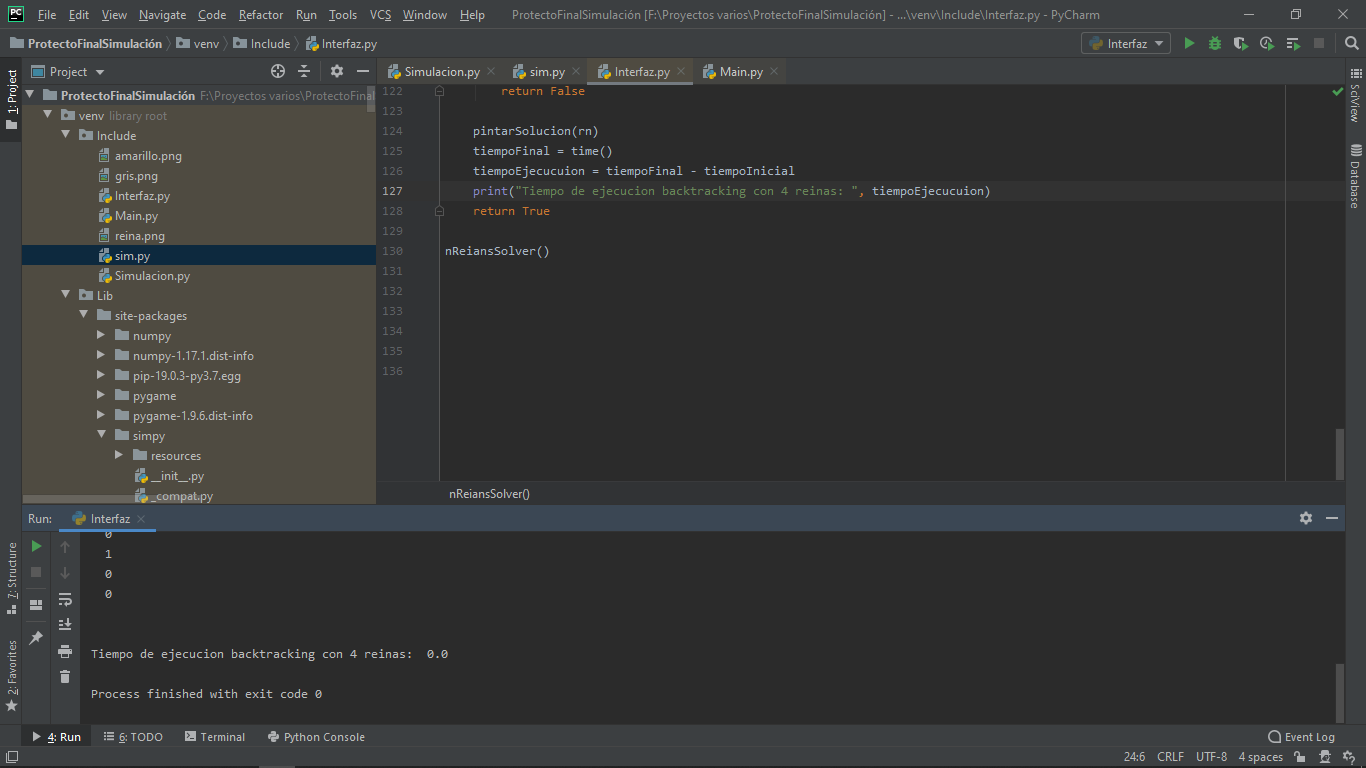


T1 = 33.024534702301025

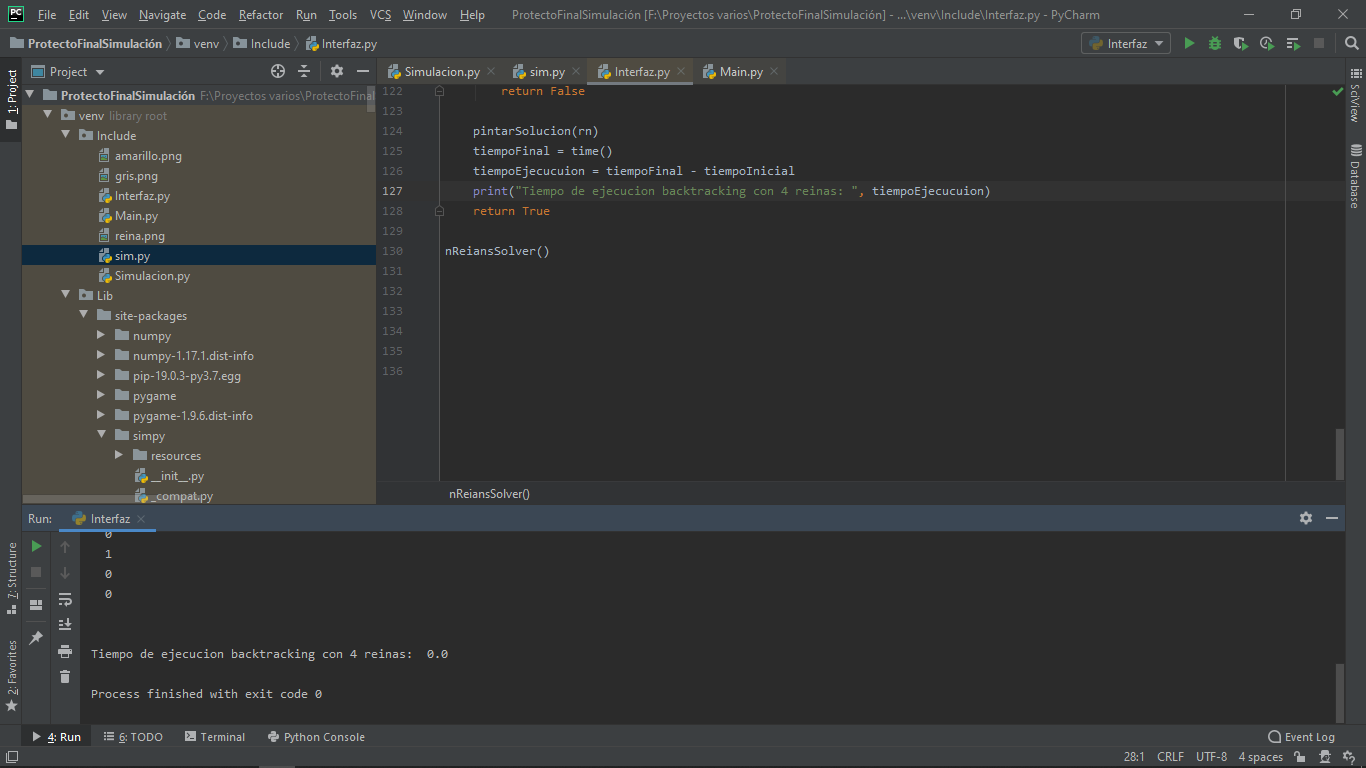
**N = 4 Determinista**



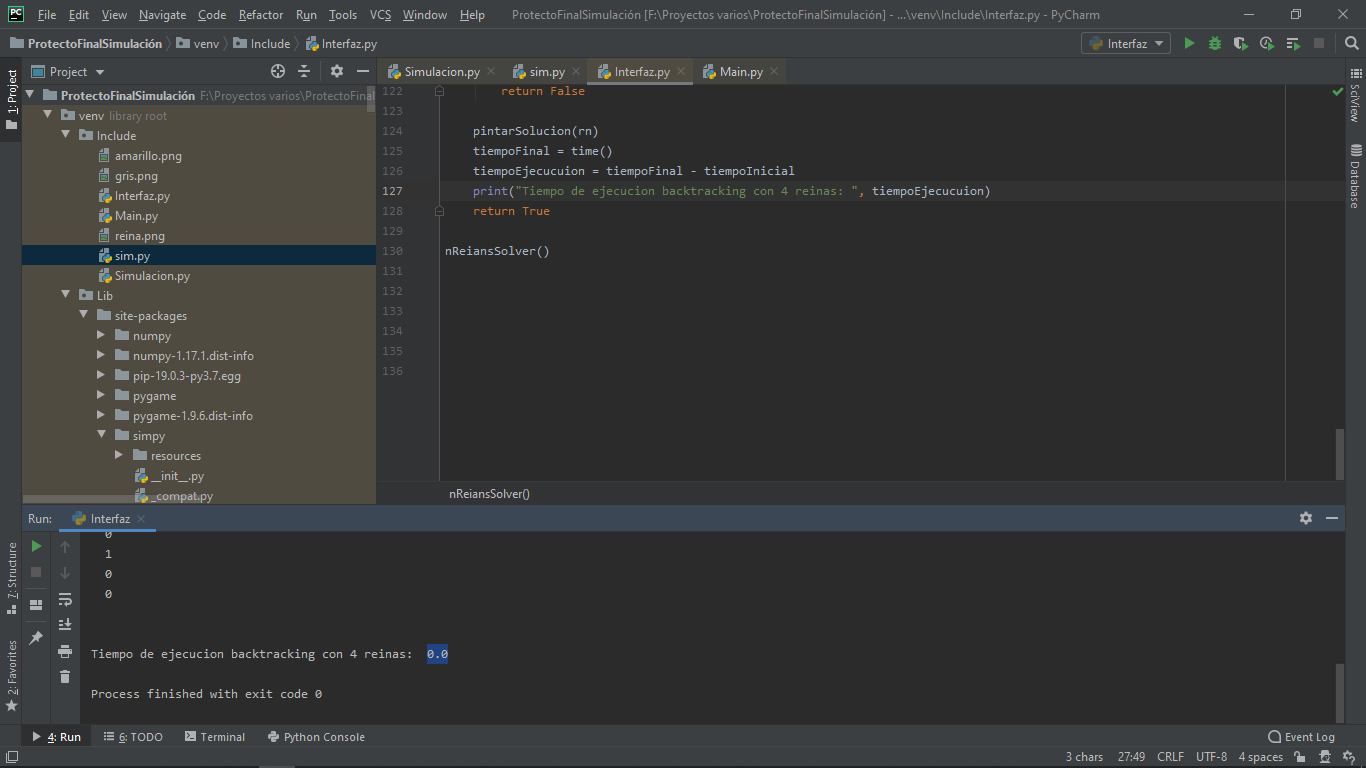
T1 = 0.0



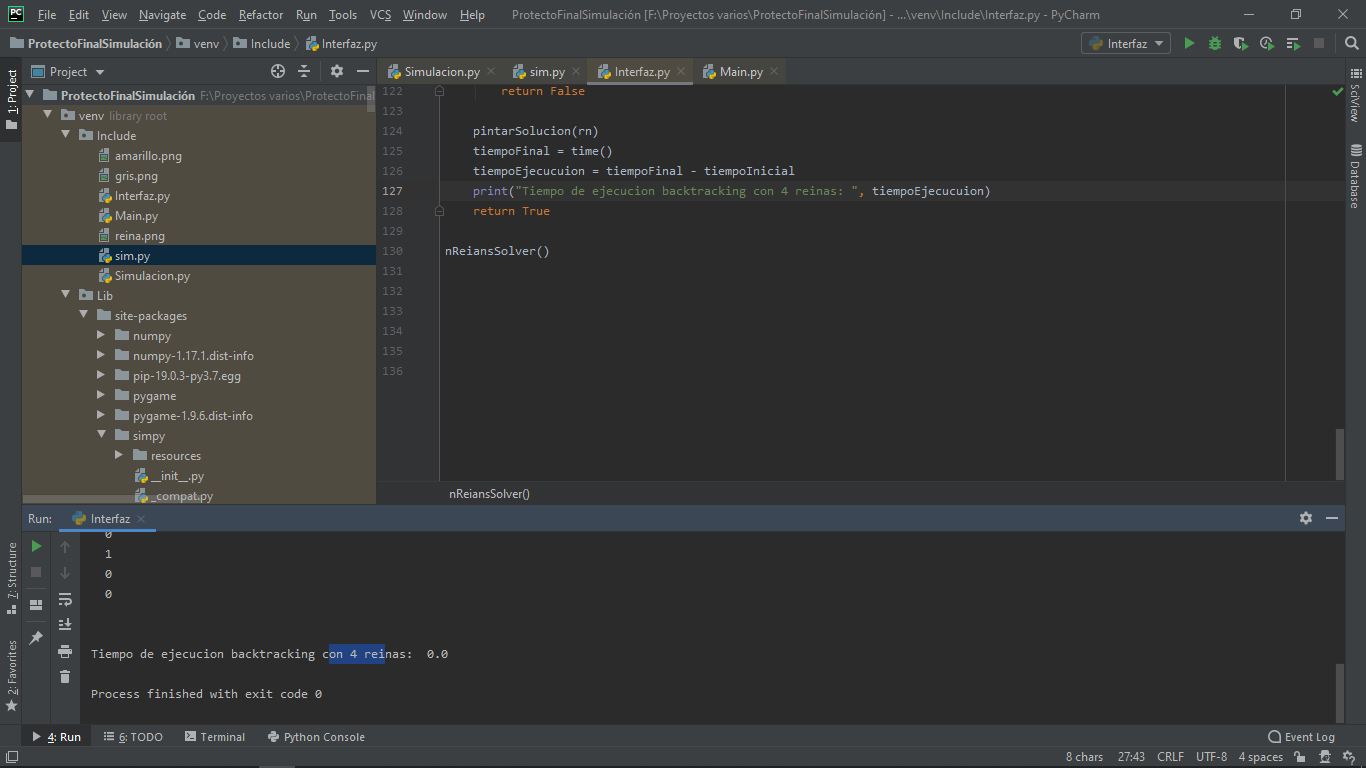
T2 = 0.0



T3 = 0.0

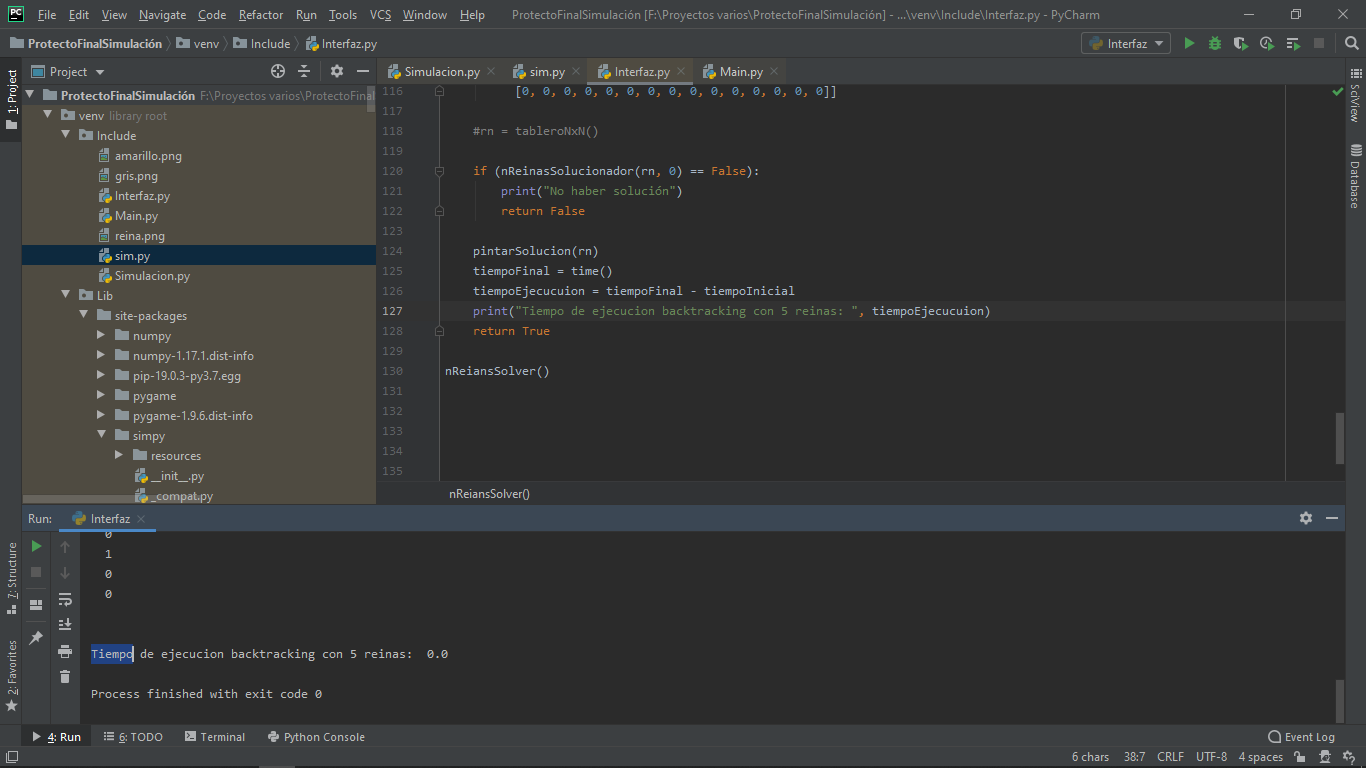


T4 = 0.0

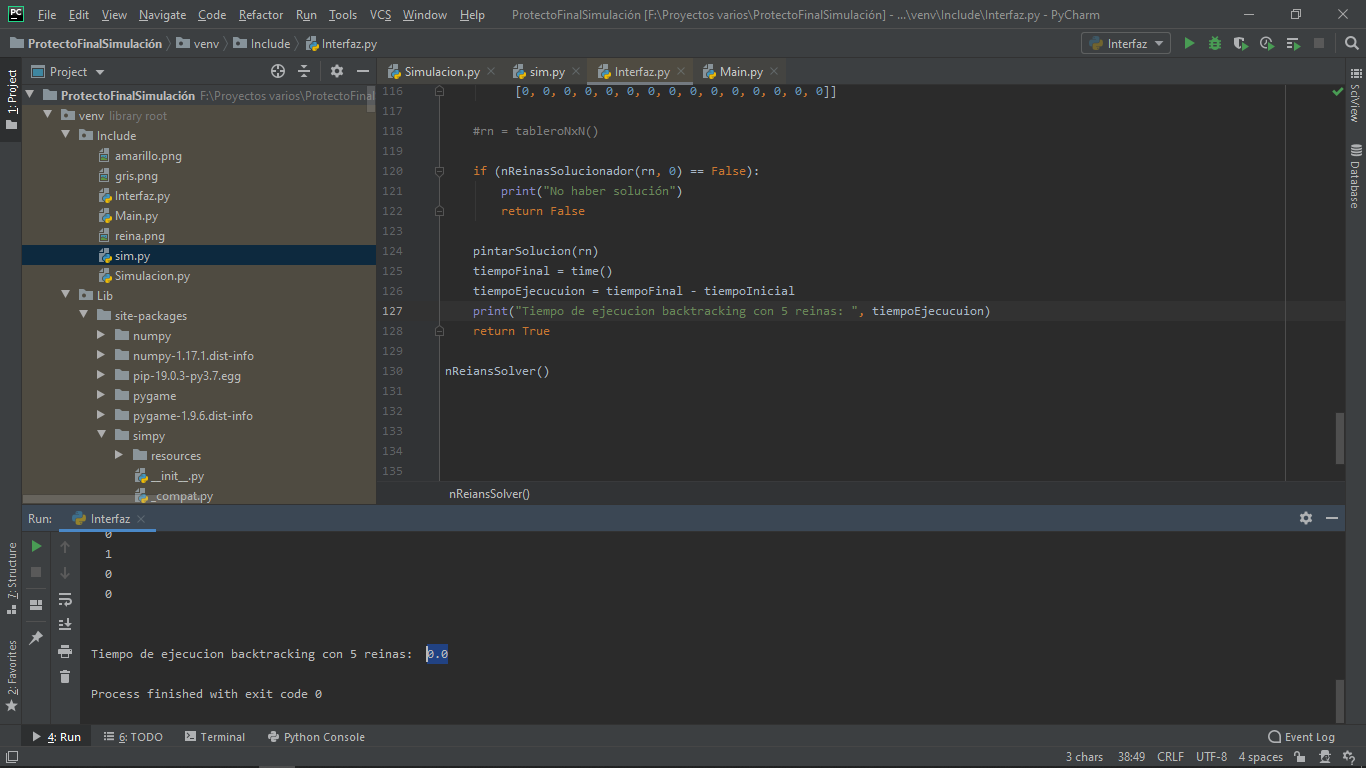


**Promedio tiempos back tracking con N = 4:** 0.0 segundos

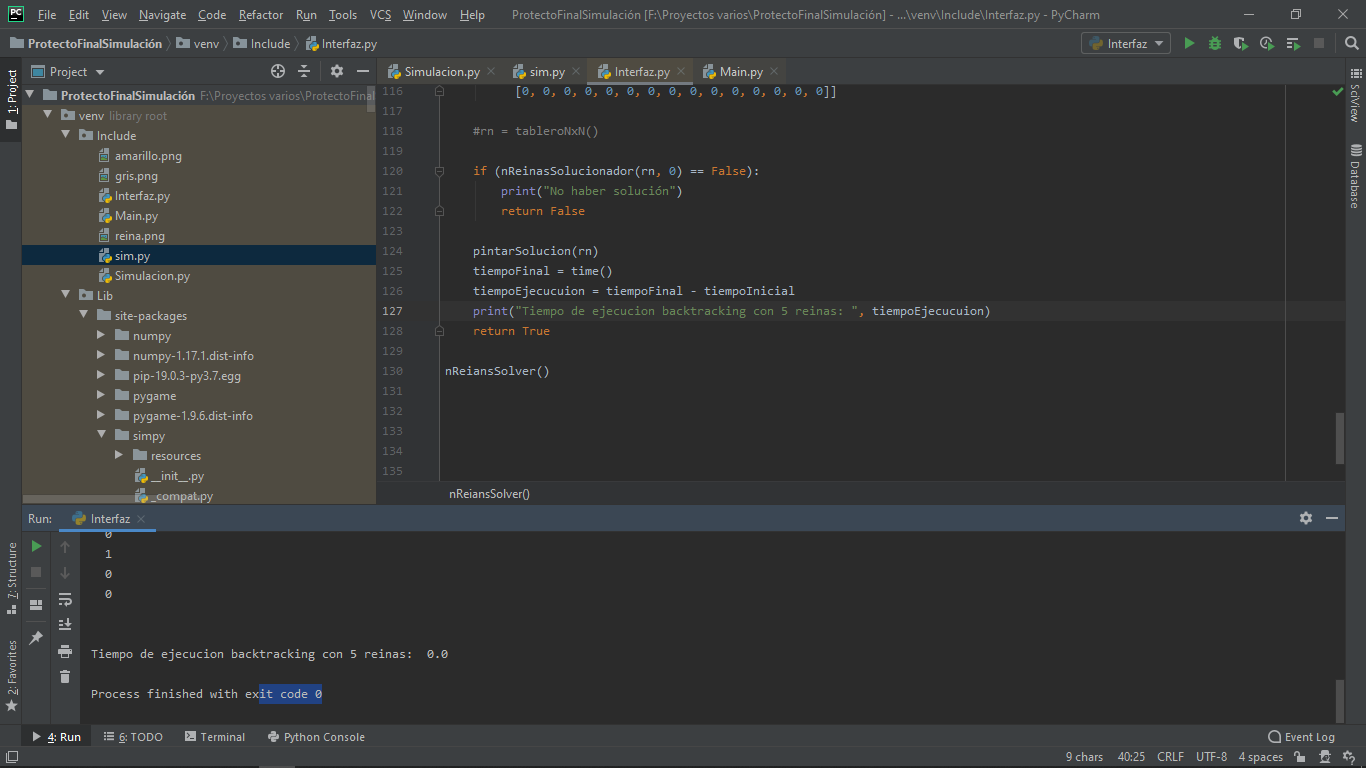
**N = 5 Determinista**



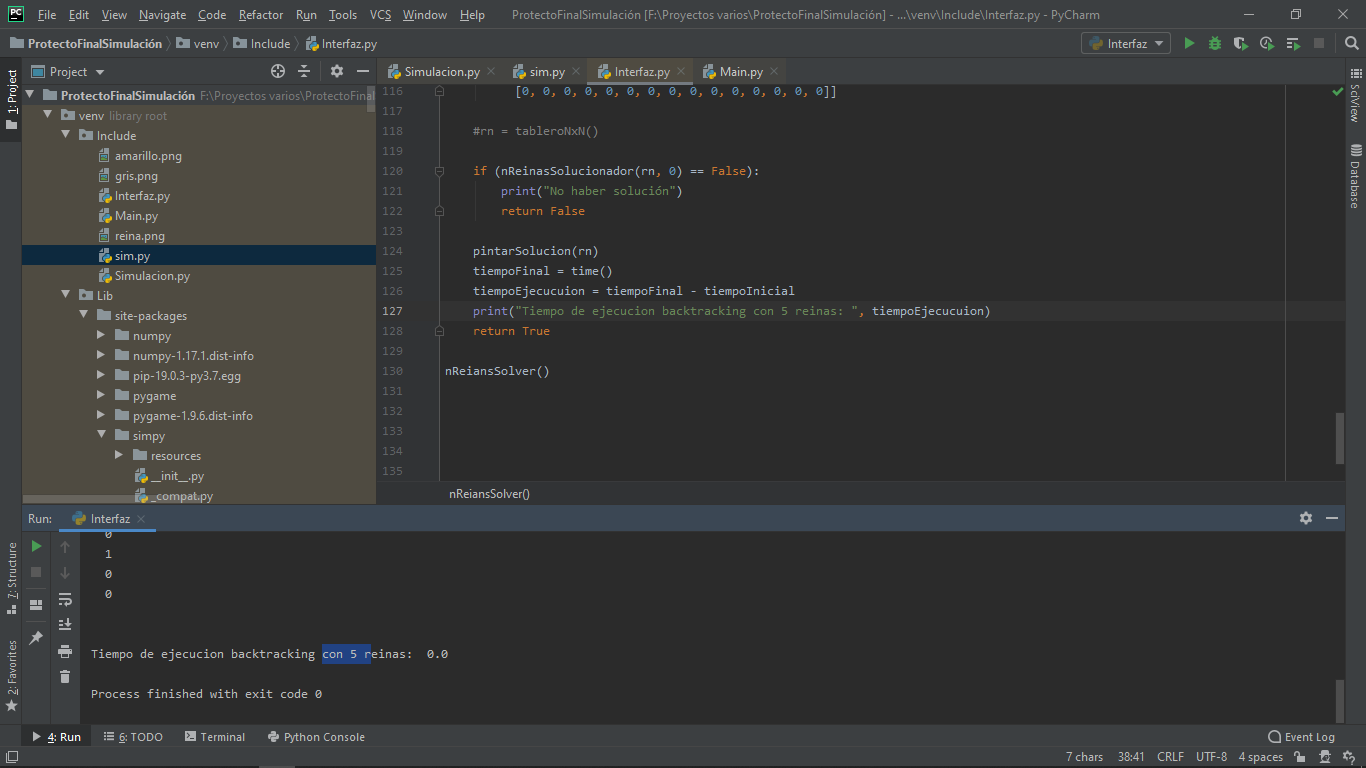
T1 = 0.0



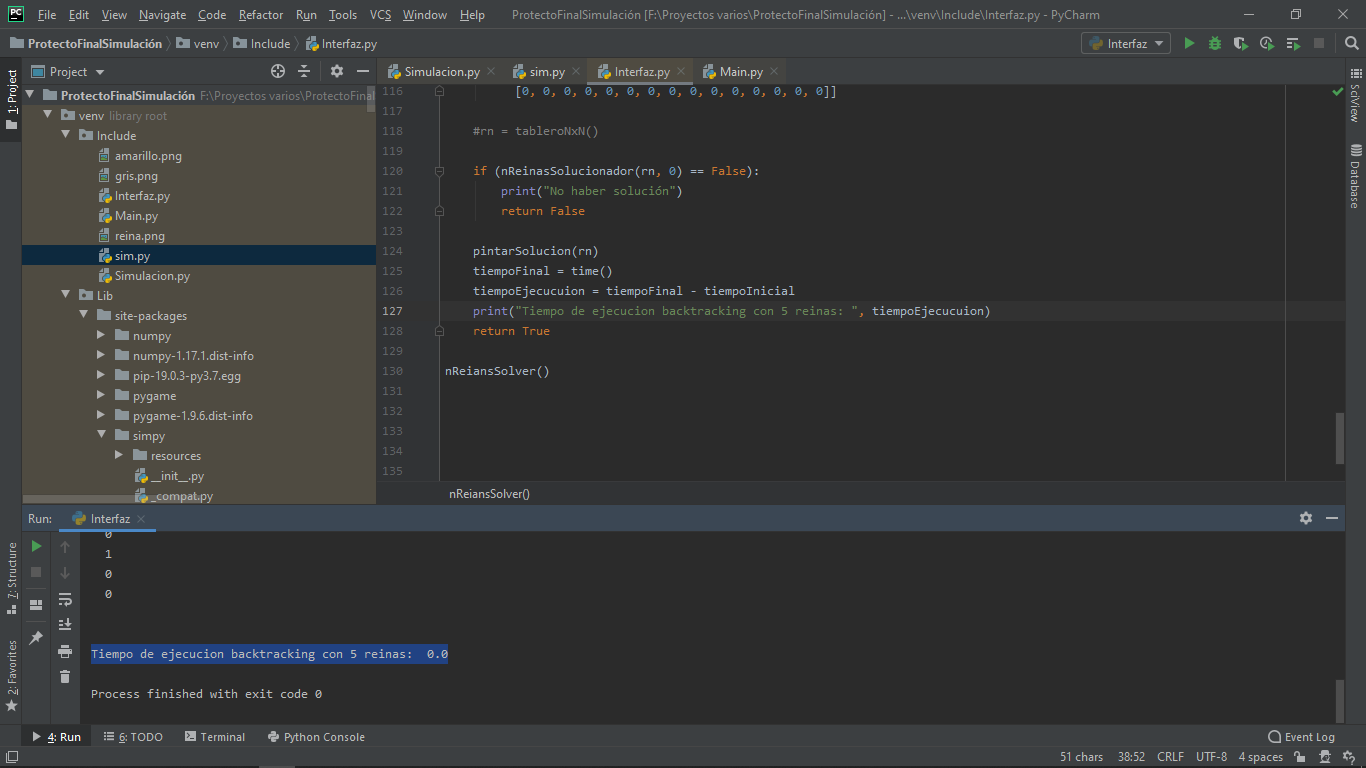
T2 = 0.0



T3 = 0.0



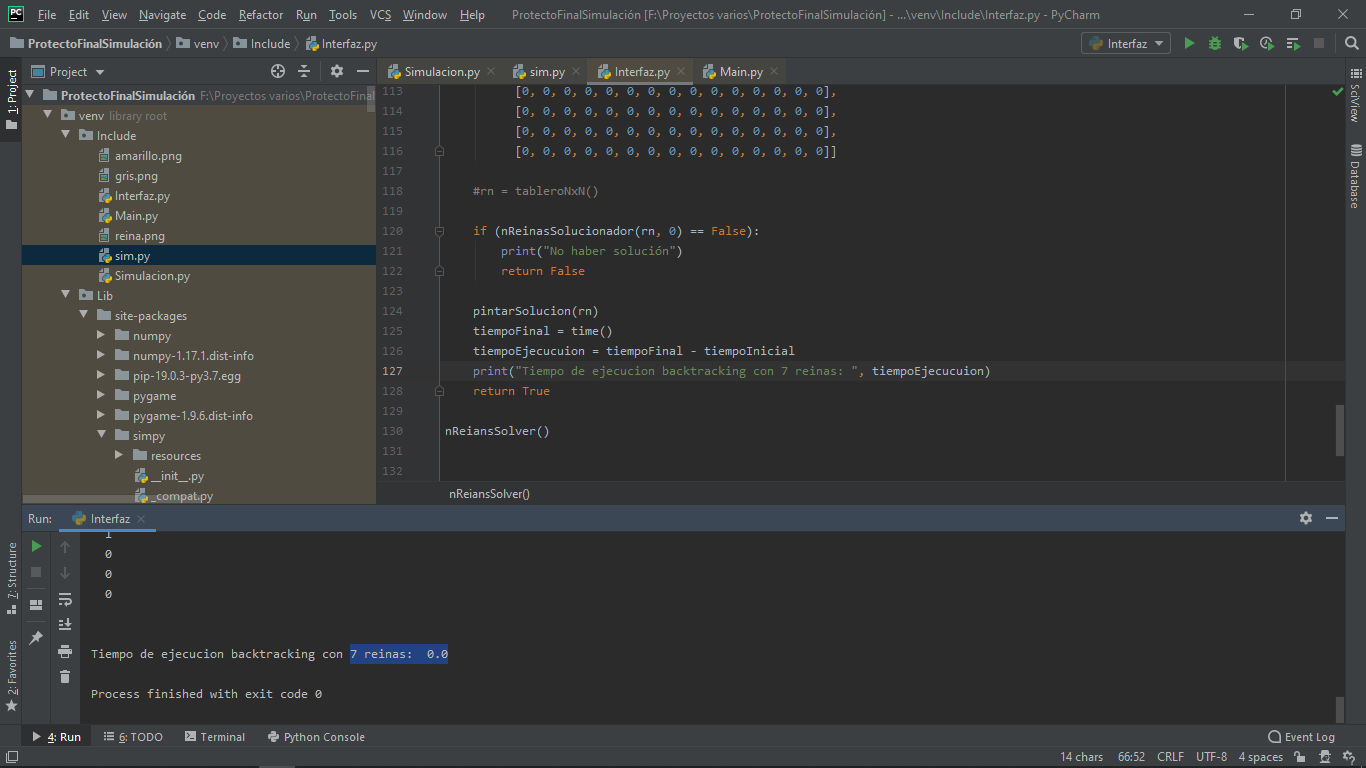
T4 = 0.0



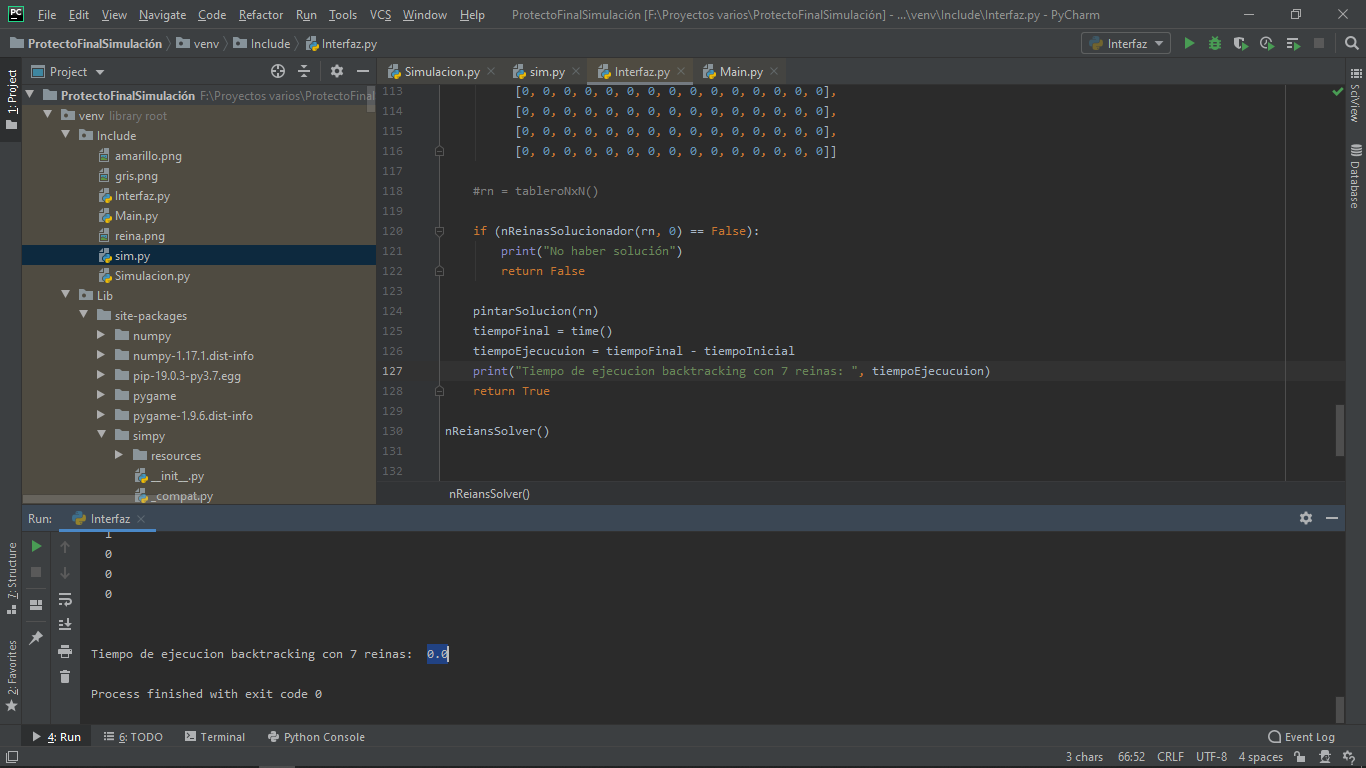
T5 = 0.0

**Promedio tiempos back tracking con N = 5:** 0.0 segundos

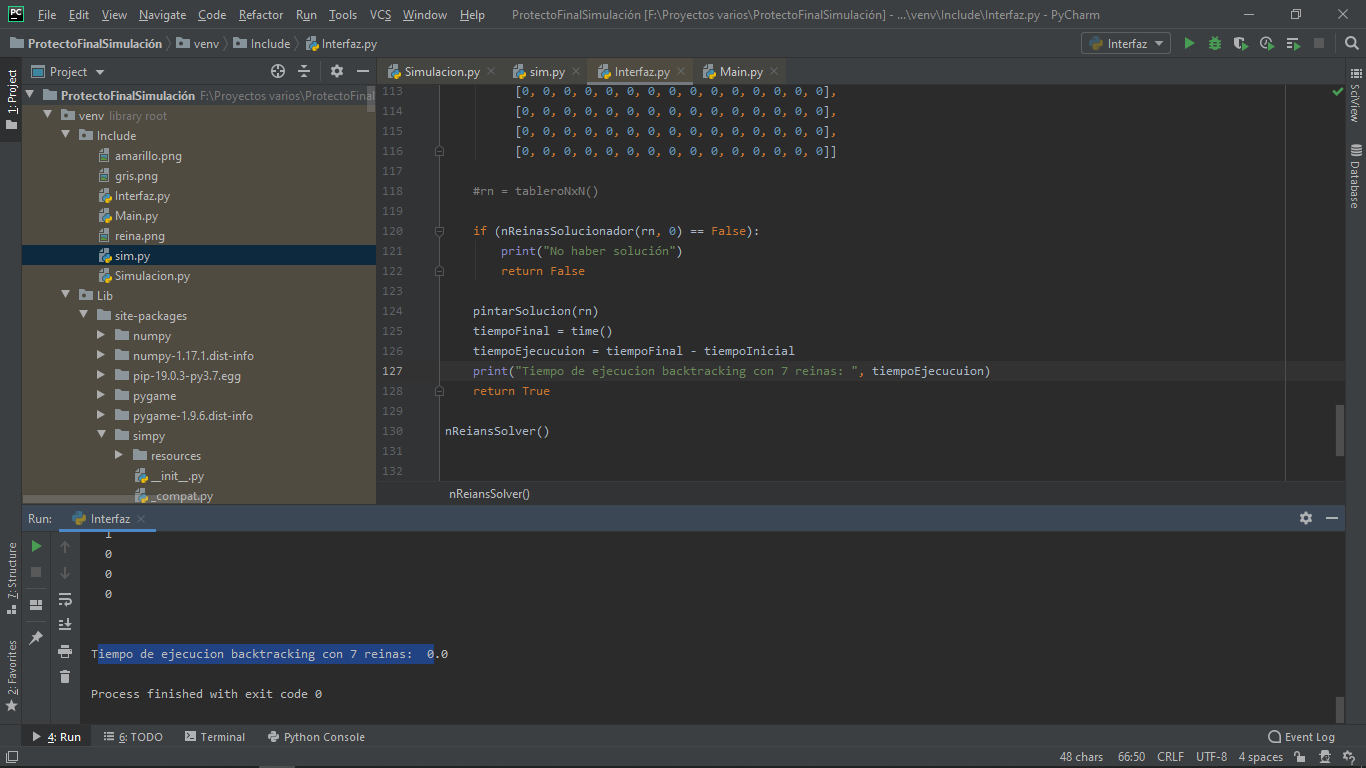
**N = 7 Determinista**



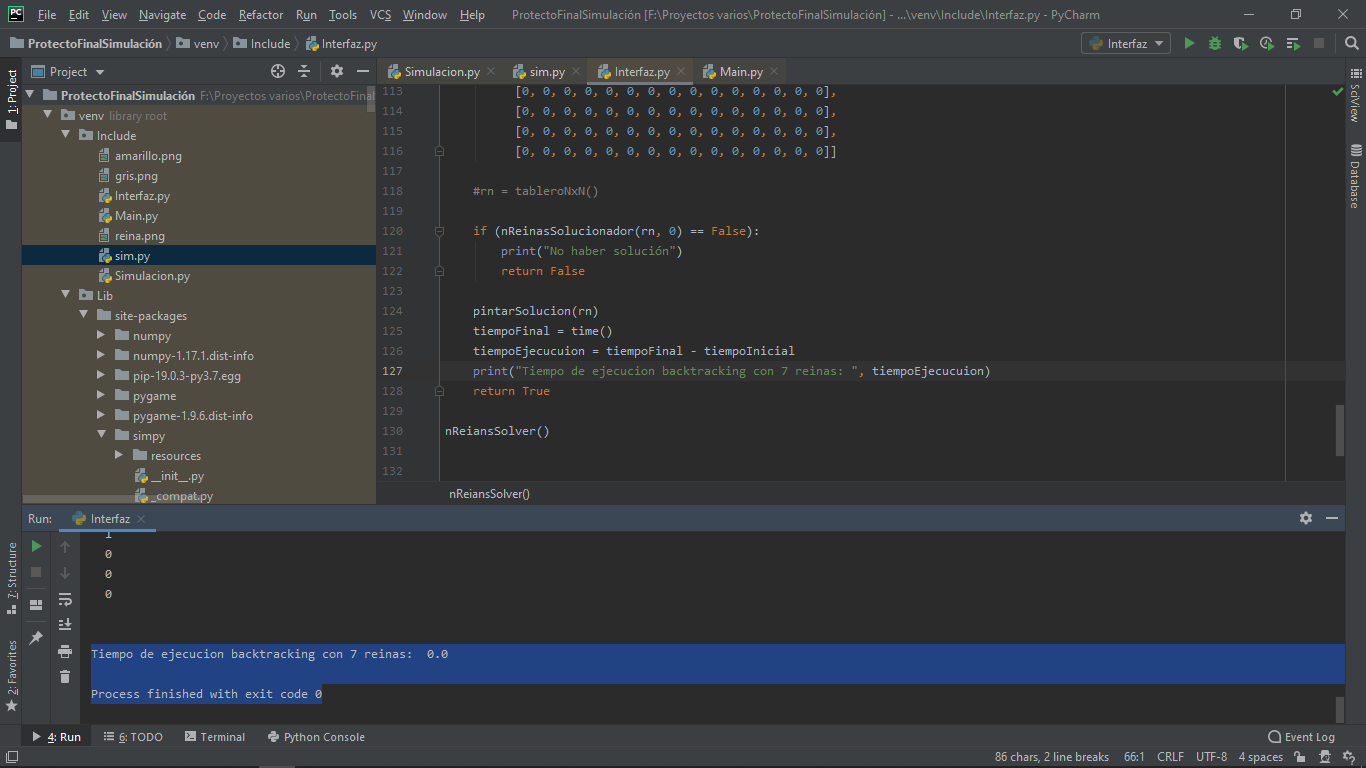
T1 = 0.0



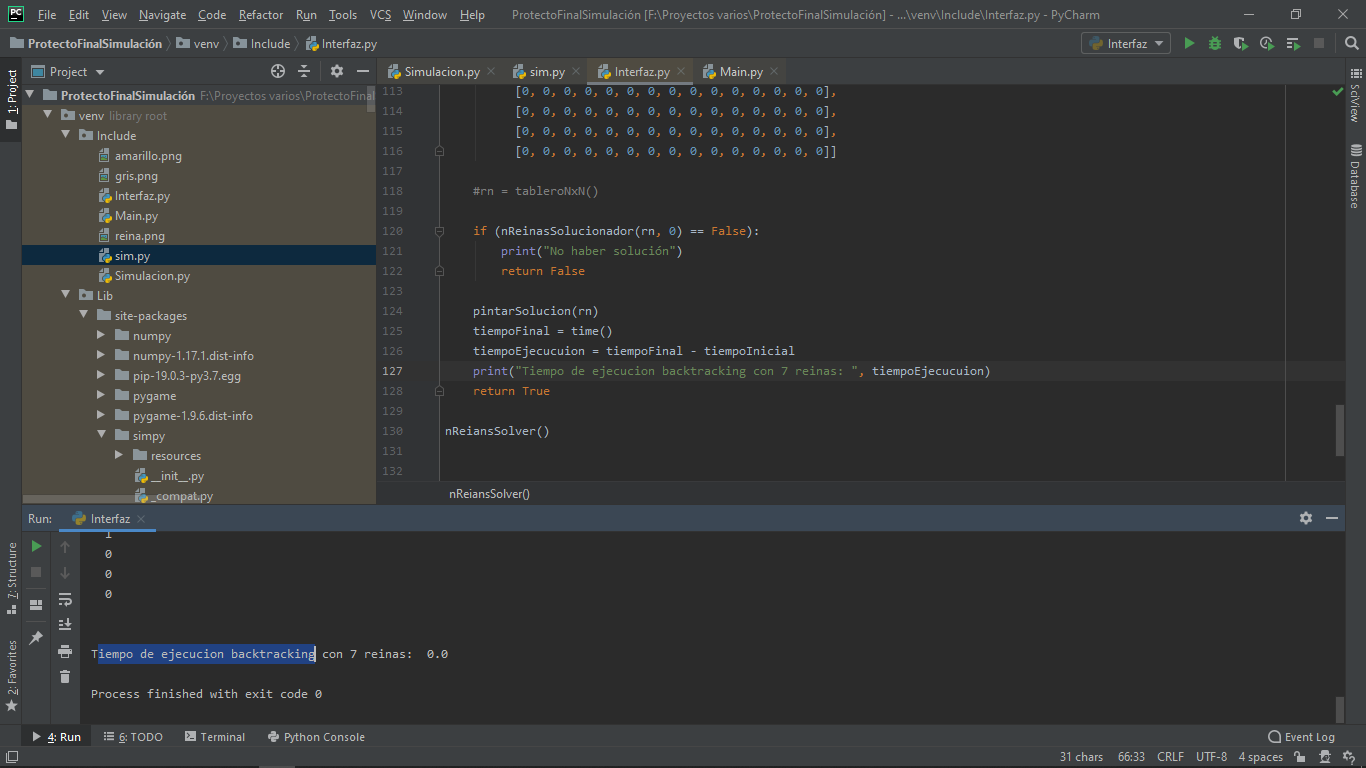
T2 = 0.0



T3 = 0.0



T4 = 0.0



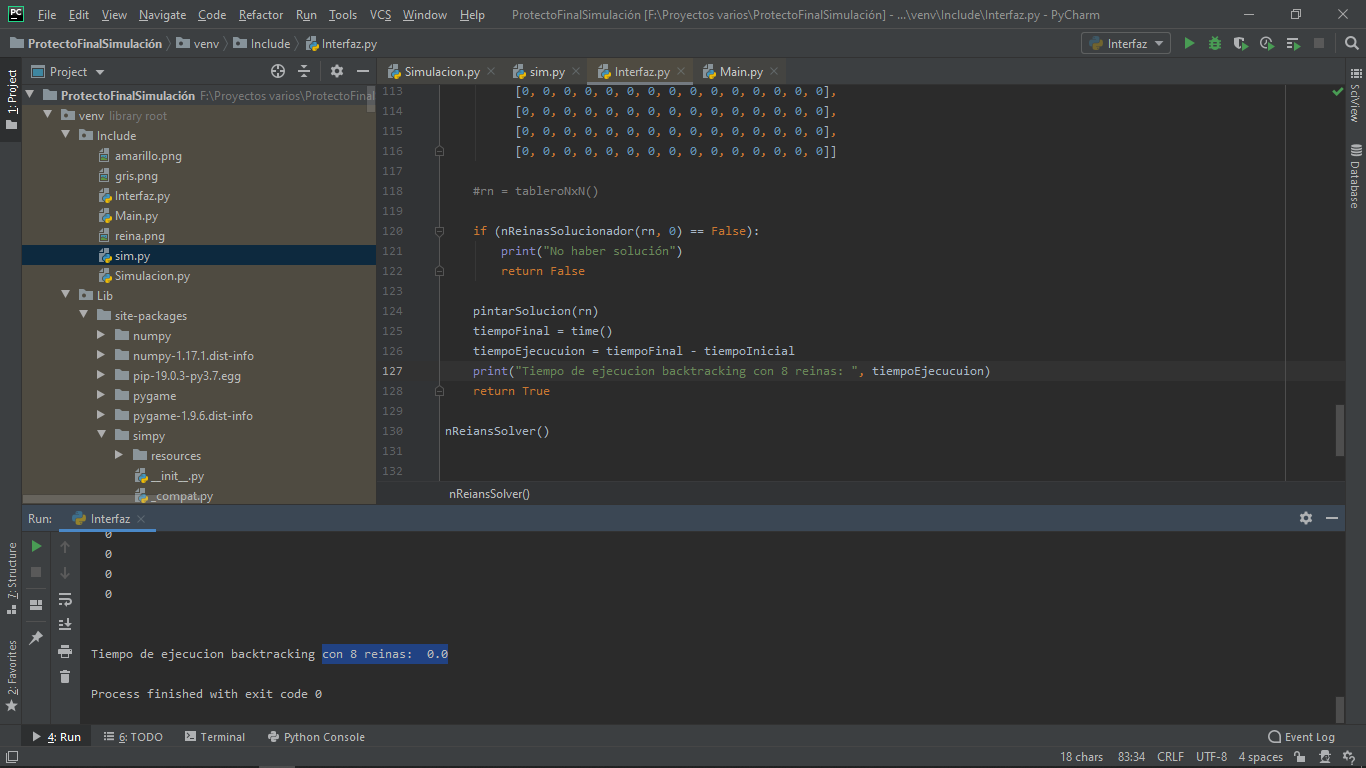
T5 = 0.0

**Promedio tiempos back tracking con N = 7:** 0.0 segundos

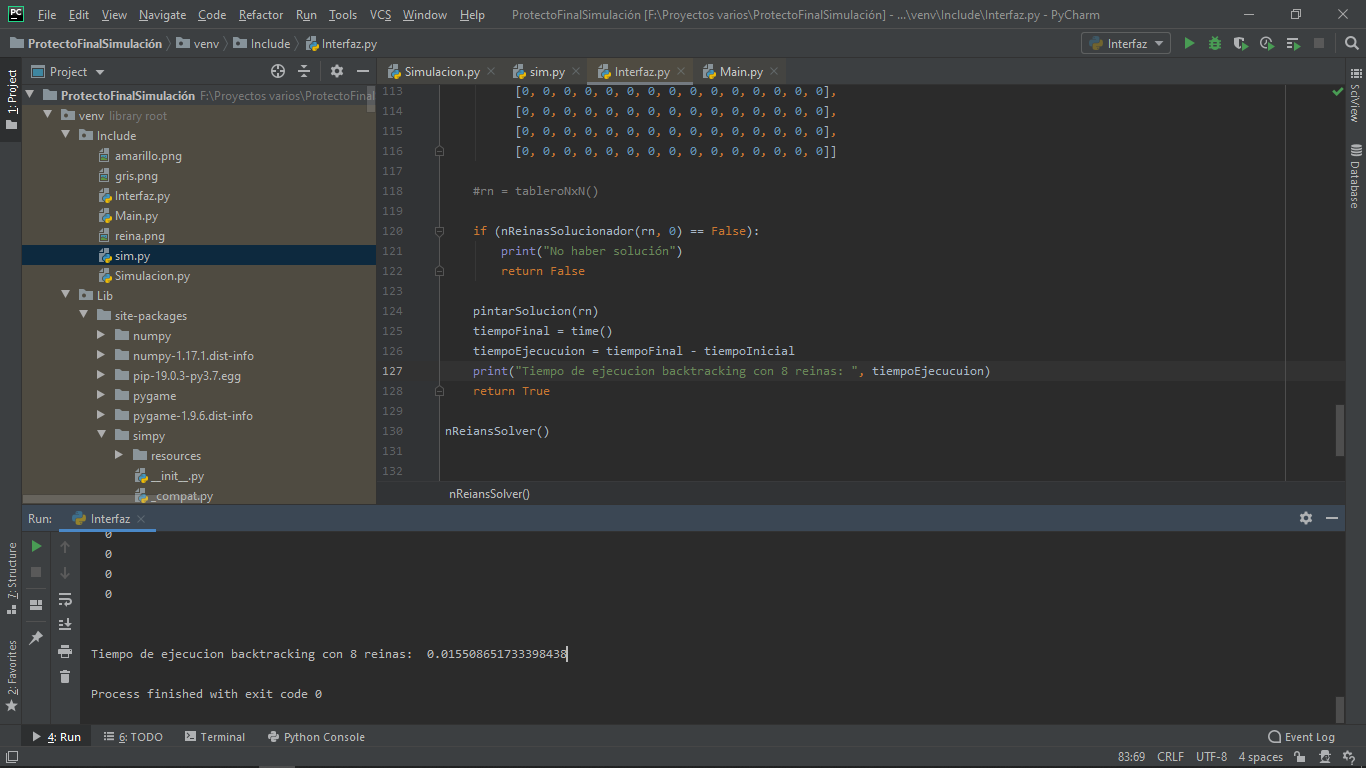
**N = 8 Determinista**



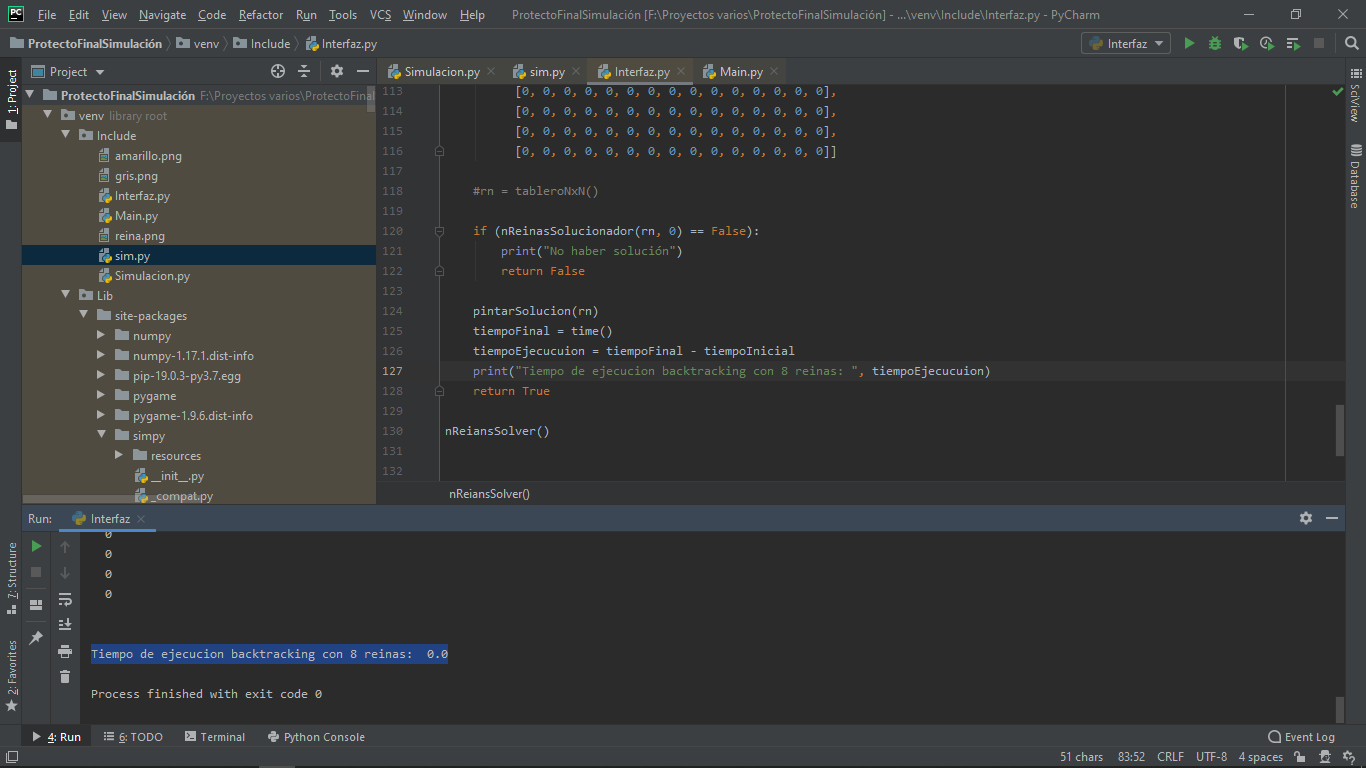
T1 = 0.015624761581420898



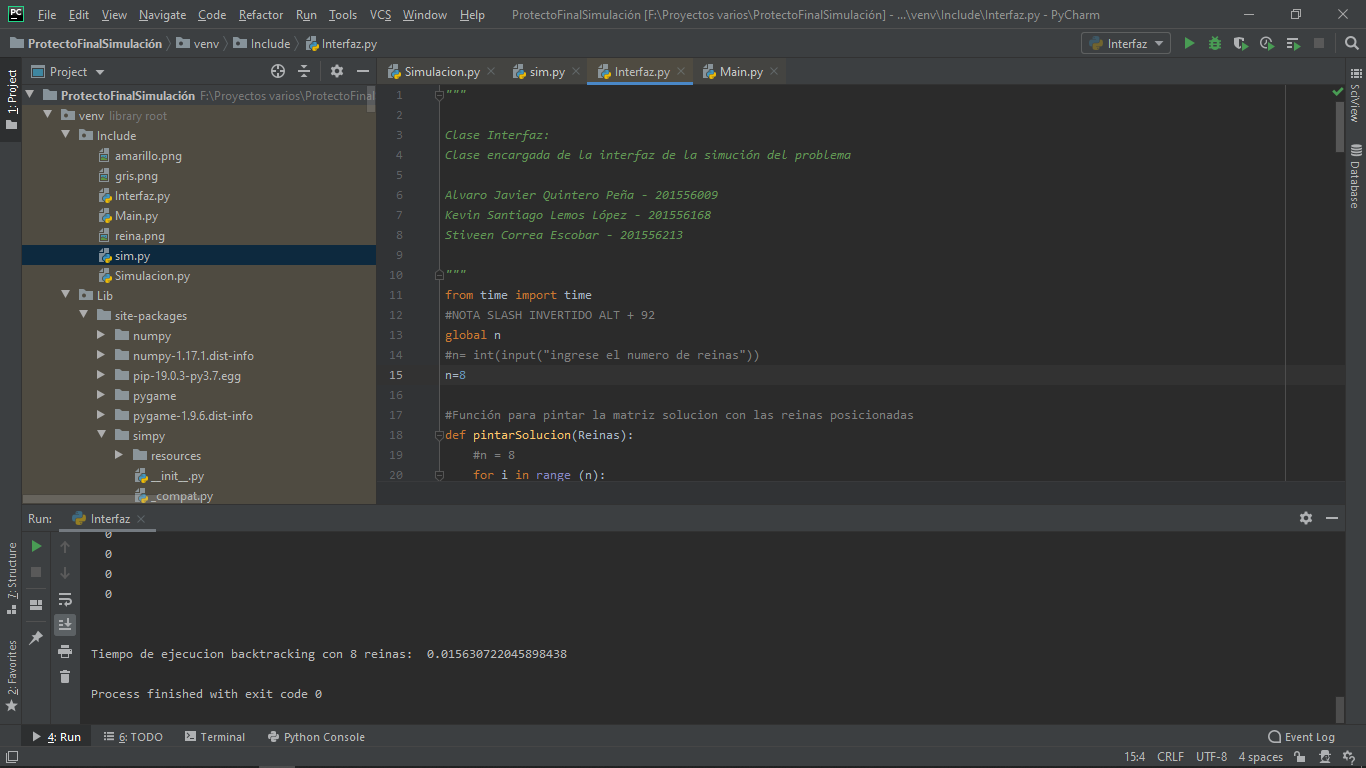
T2 = 0.0



T3 = 0.015508651733398438



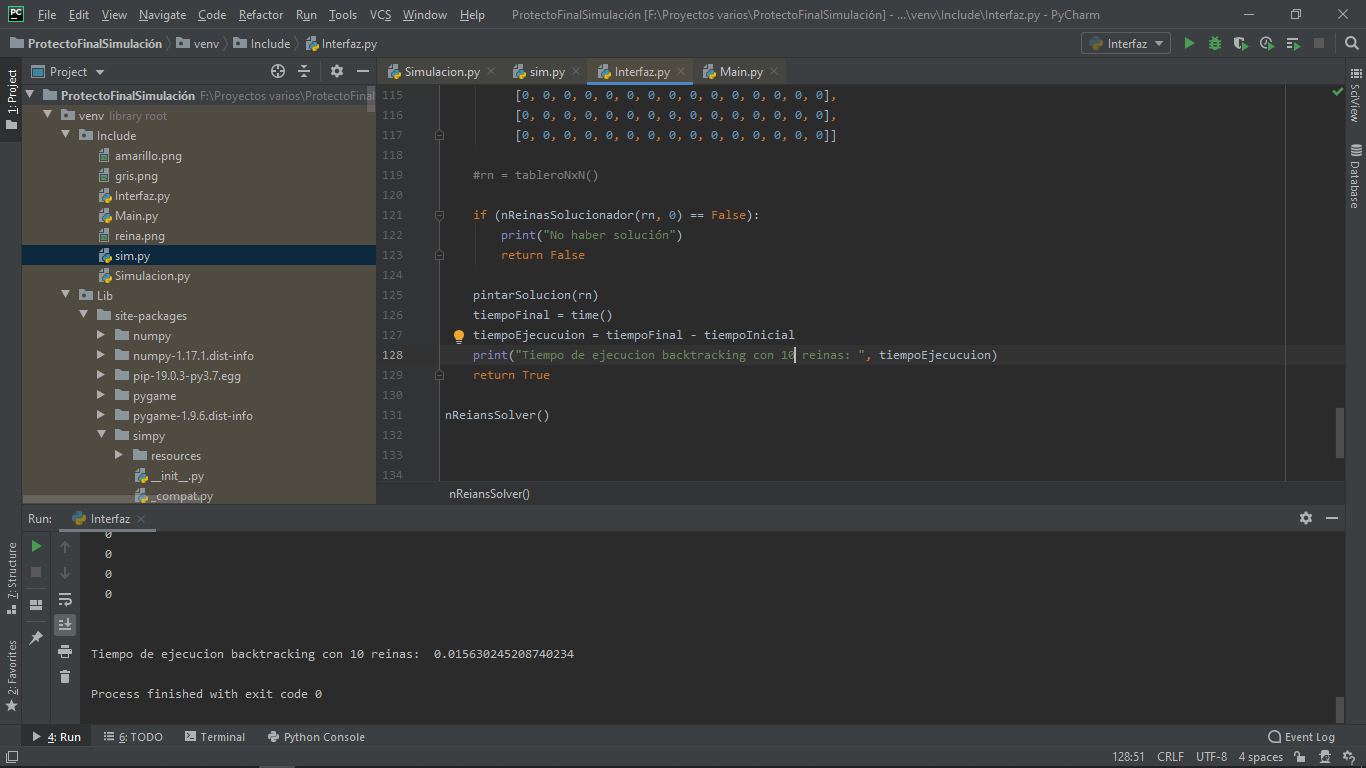
T4 = 0.0



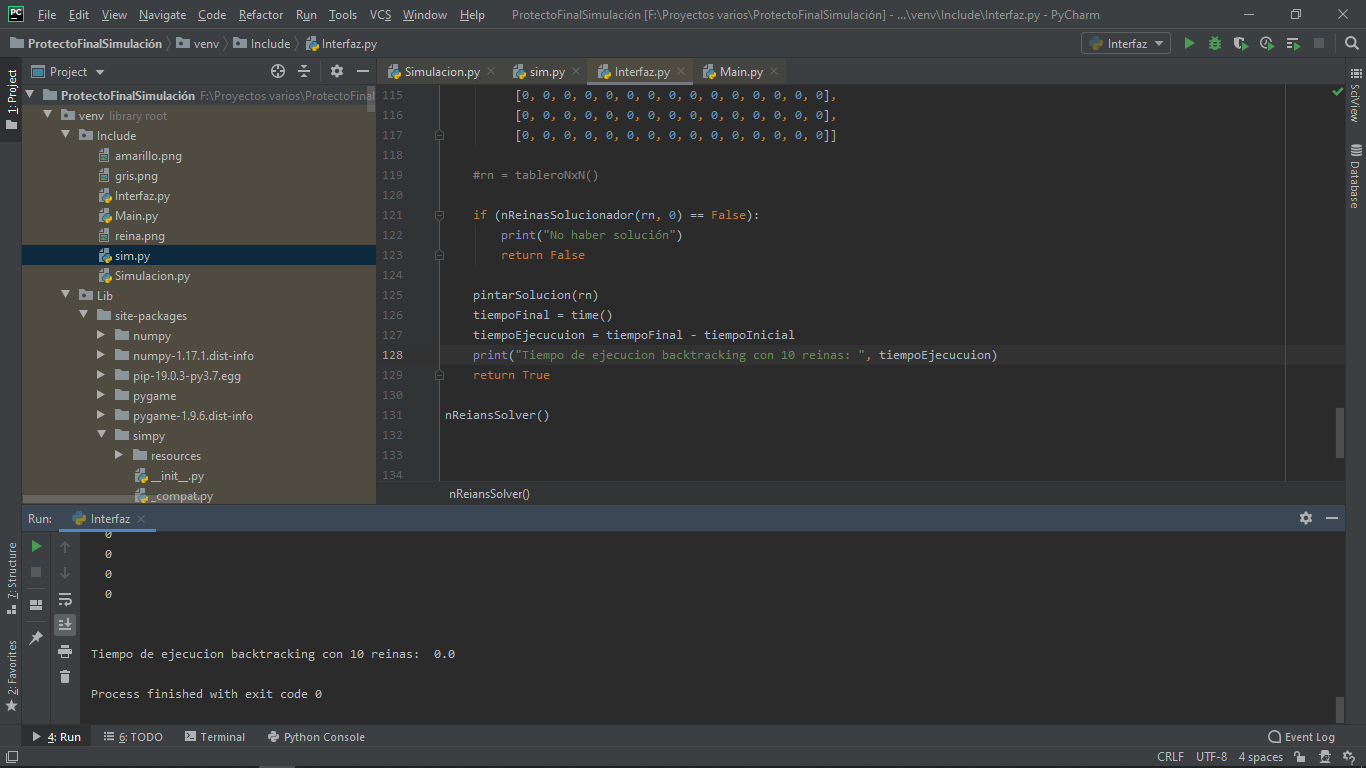
T5 = 0.015630722045898438

**Promedio tiempos back tracking con N = 8:** 0.00935282707 segundos

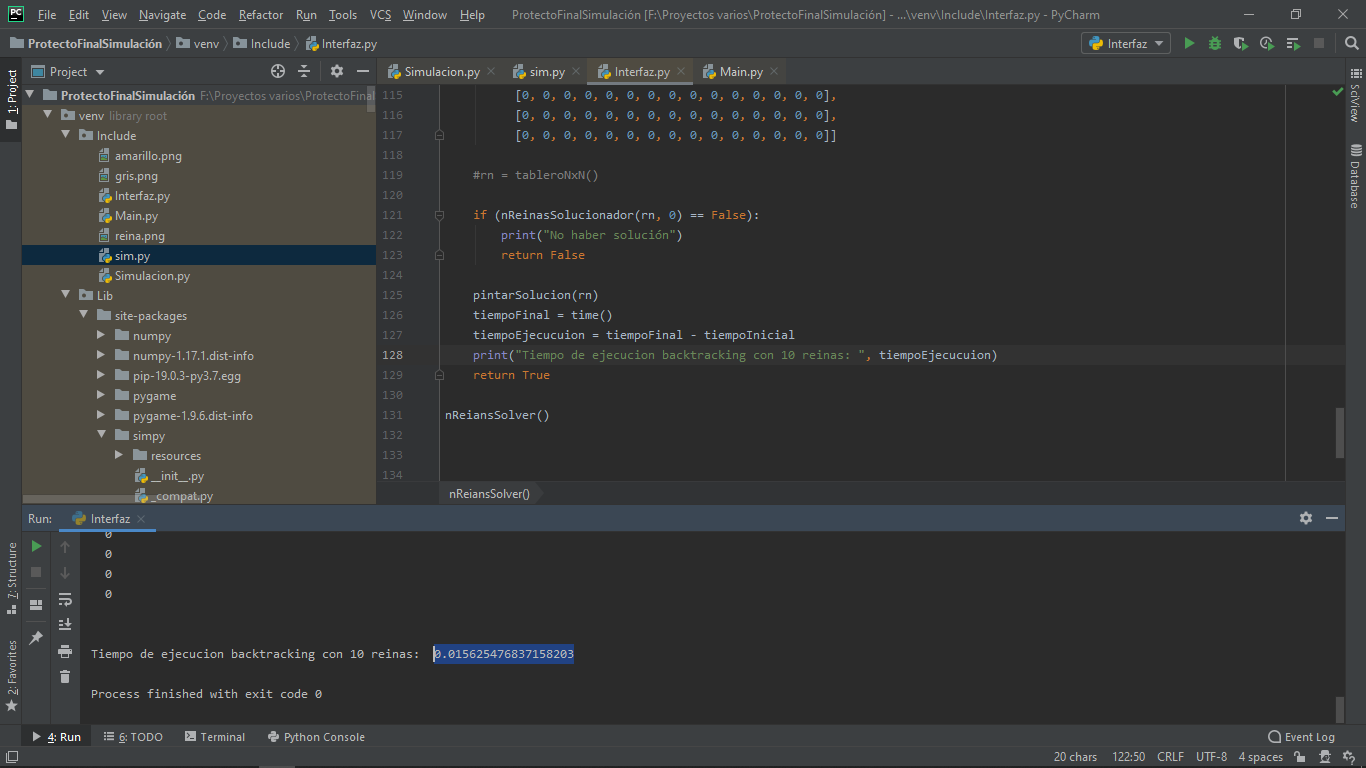
**N = 10 Determinista**



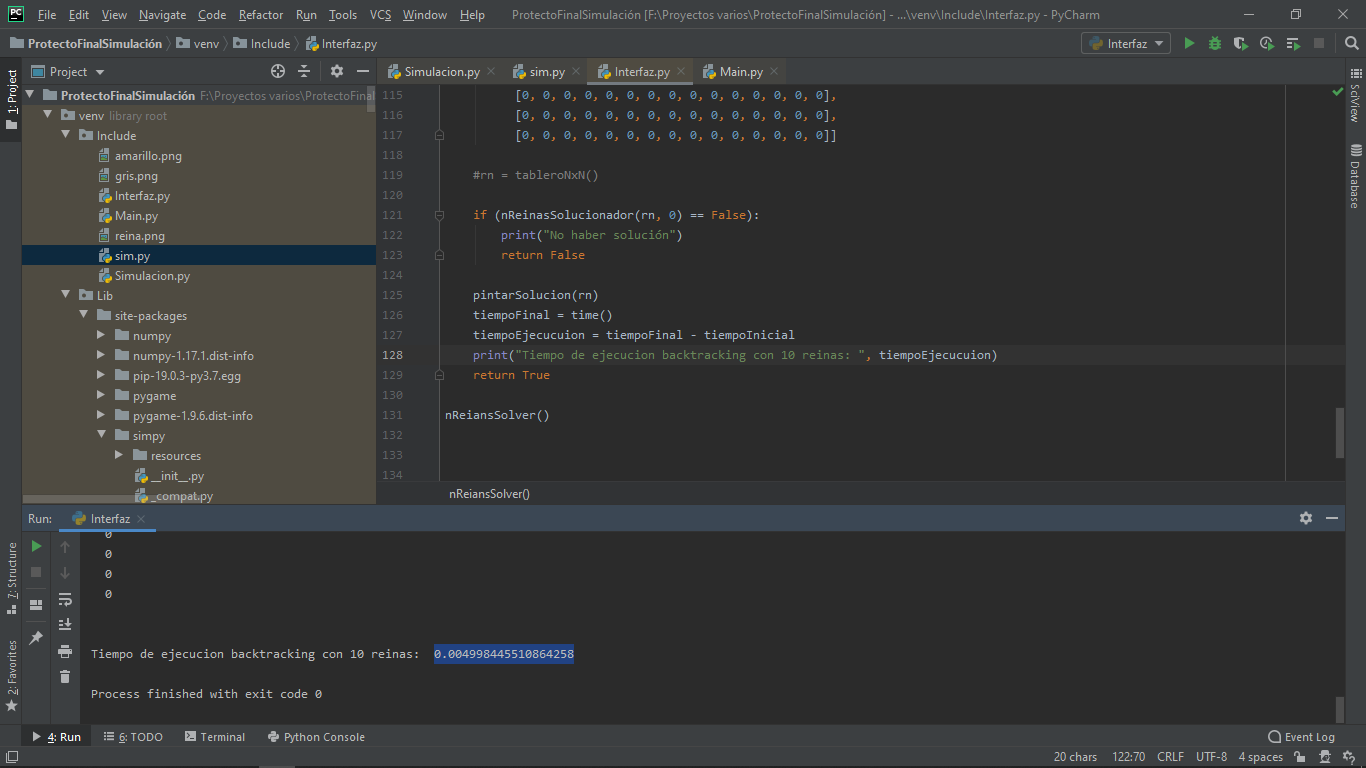
T1 = 0.015630245208740234



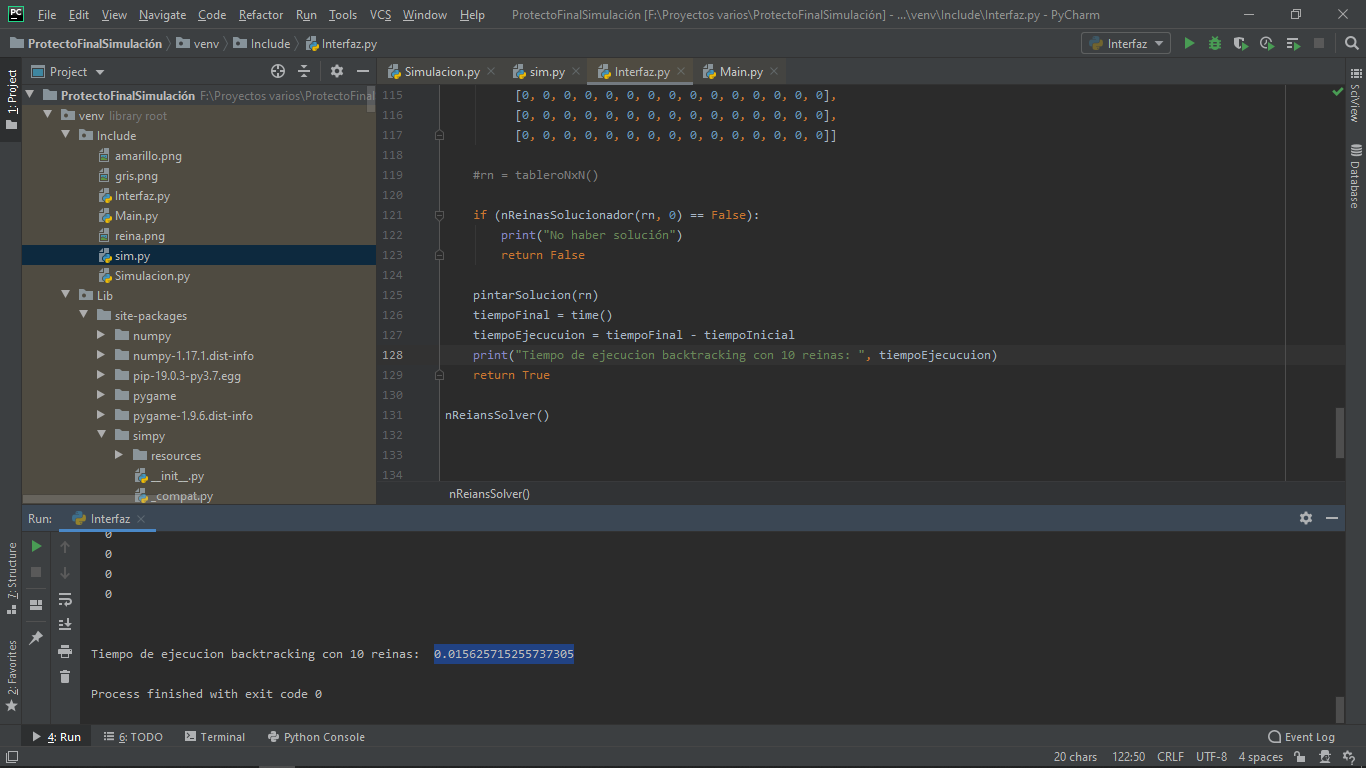
T2 = 0.0



T3 = 0.015625476837158203



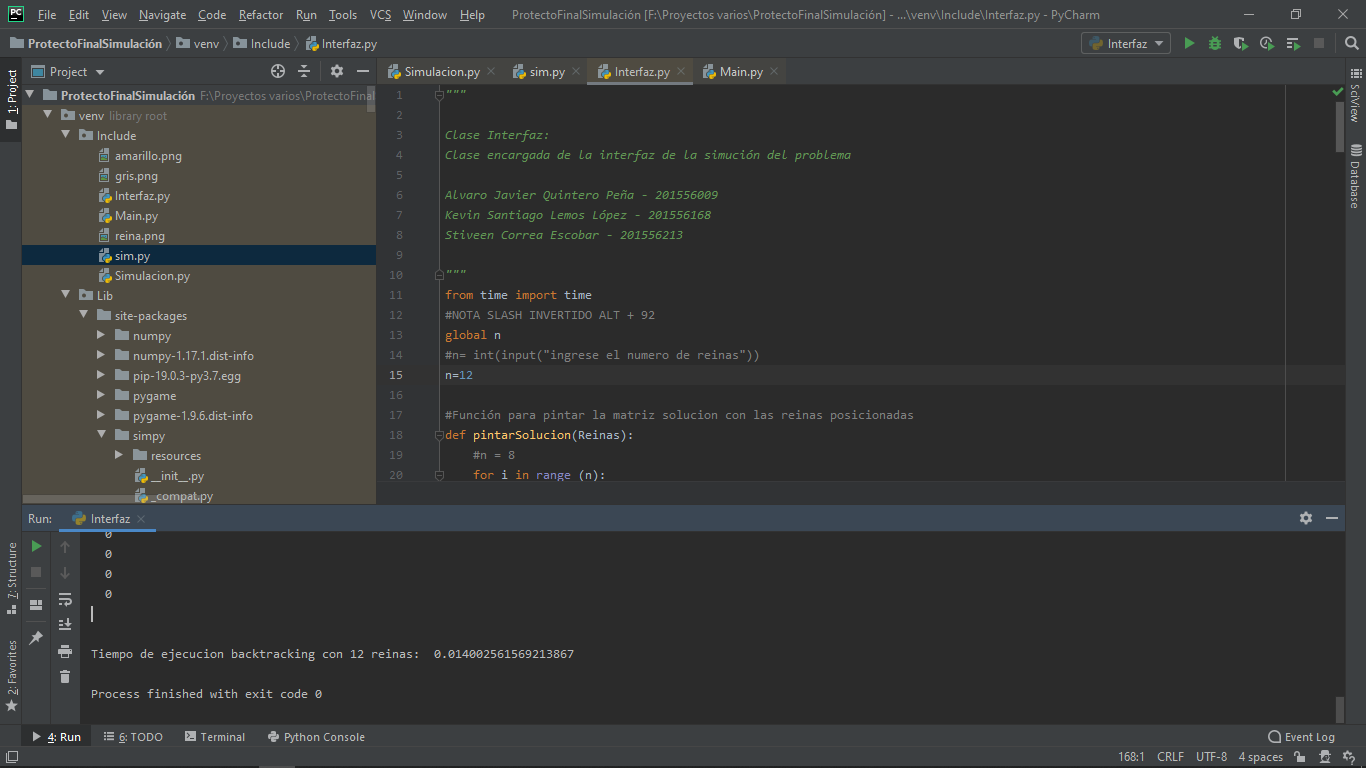
T4 = 0.004998445510864258



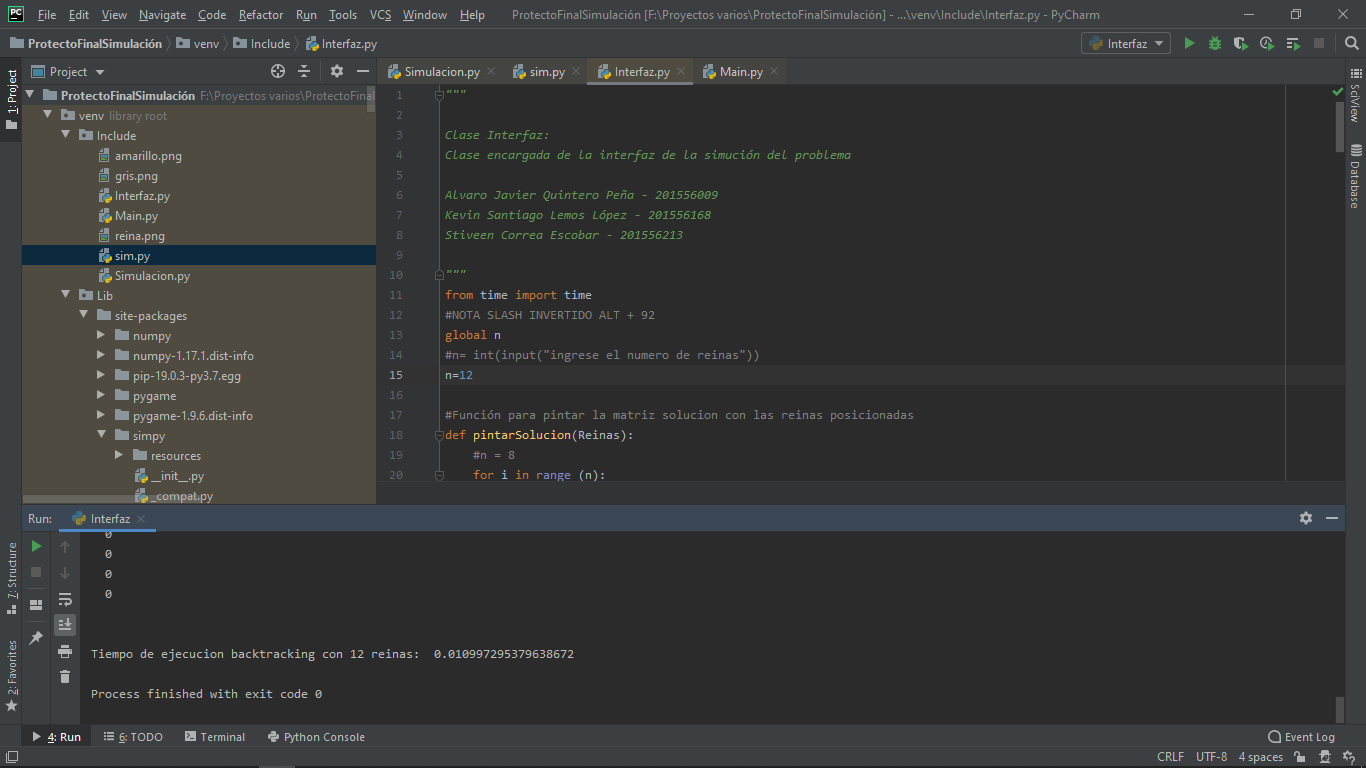
T5 = 0.015625715255737305

**Promedio tiempos back tracking con N = 10:** 0.01037597656 segundos

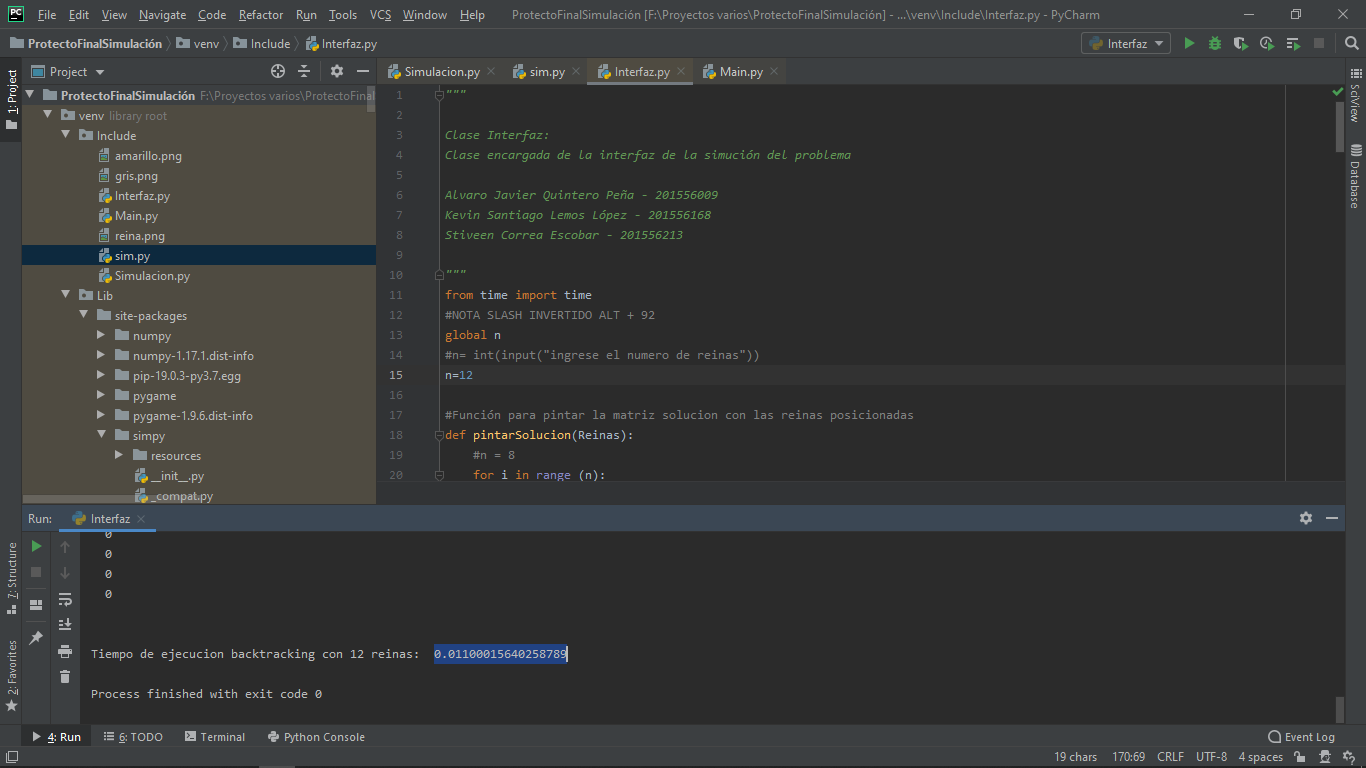
**N = 12 Determinista**



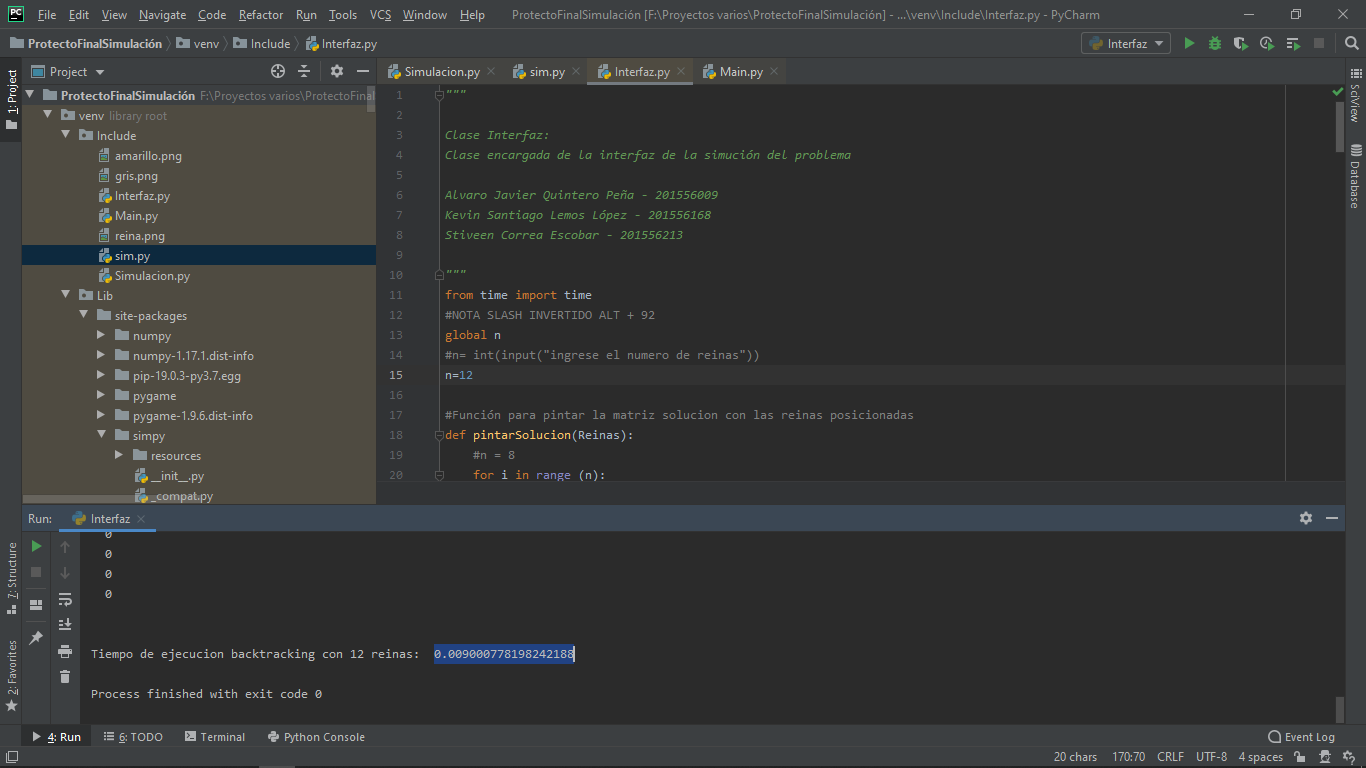
T1 = 0.014002561569213867



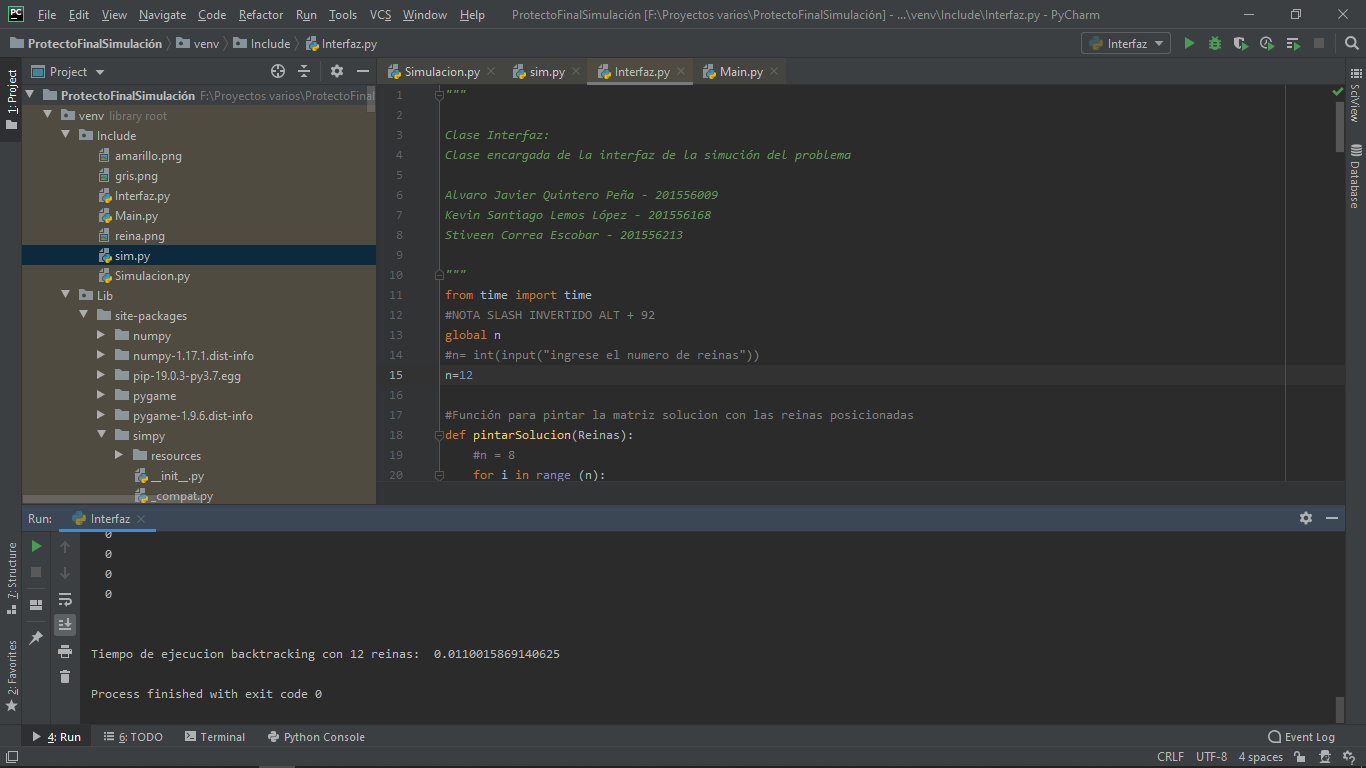
T2 = 0.010997295379638672



T3 = 0.01100015640258789



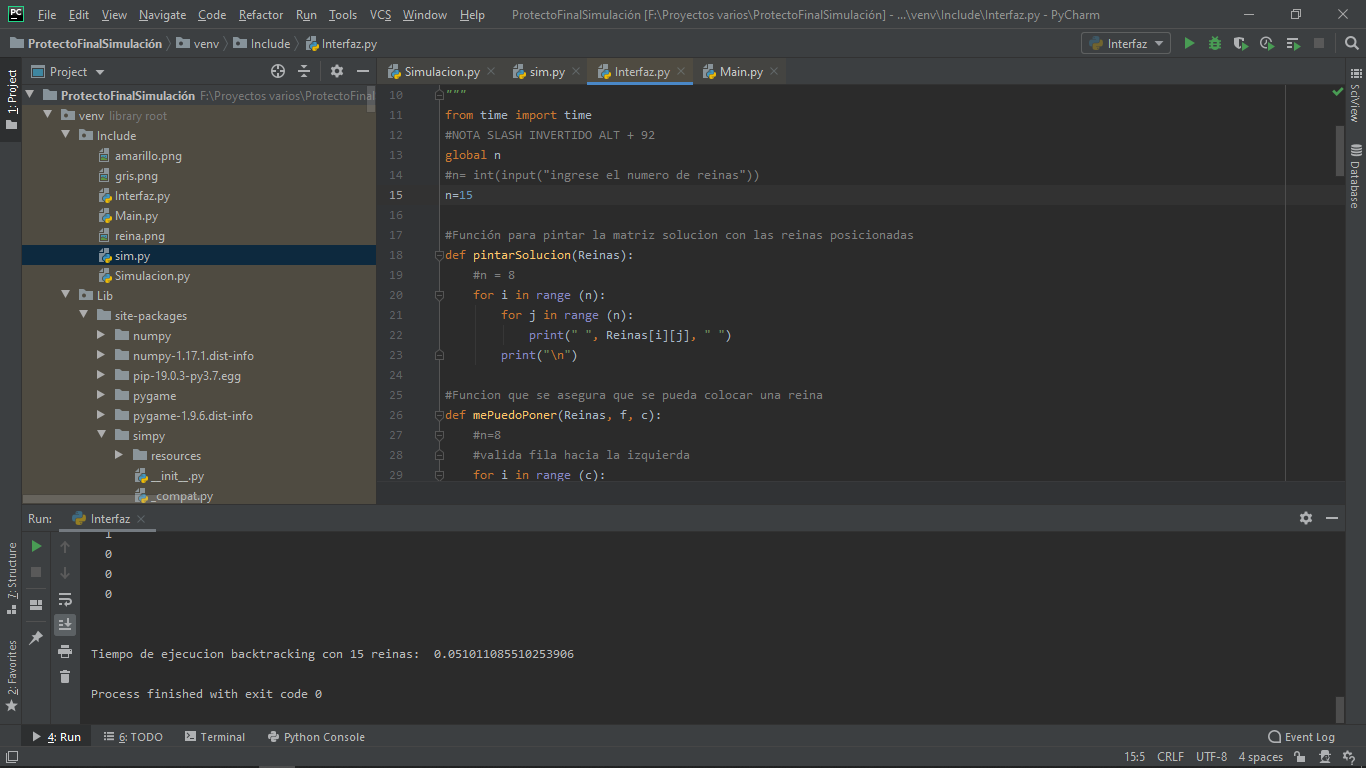
T4 = 0.009000778198242188



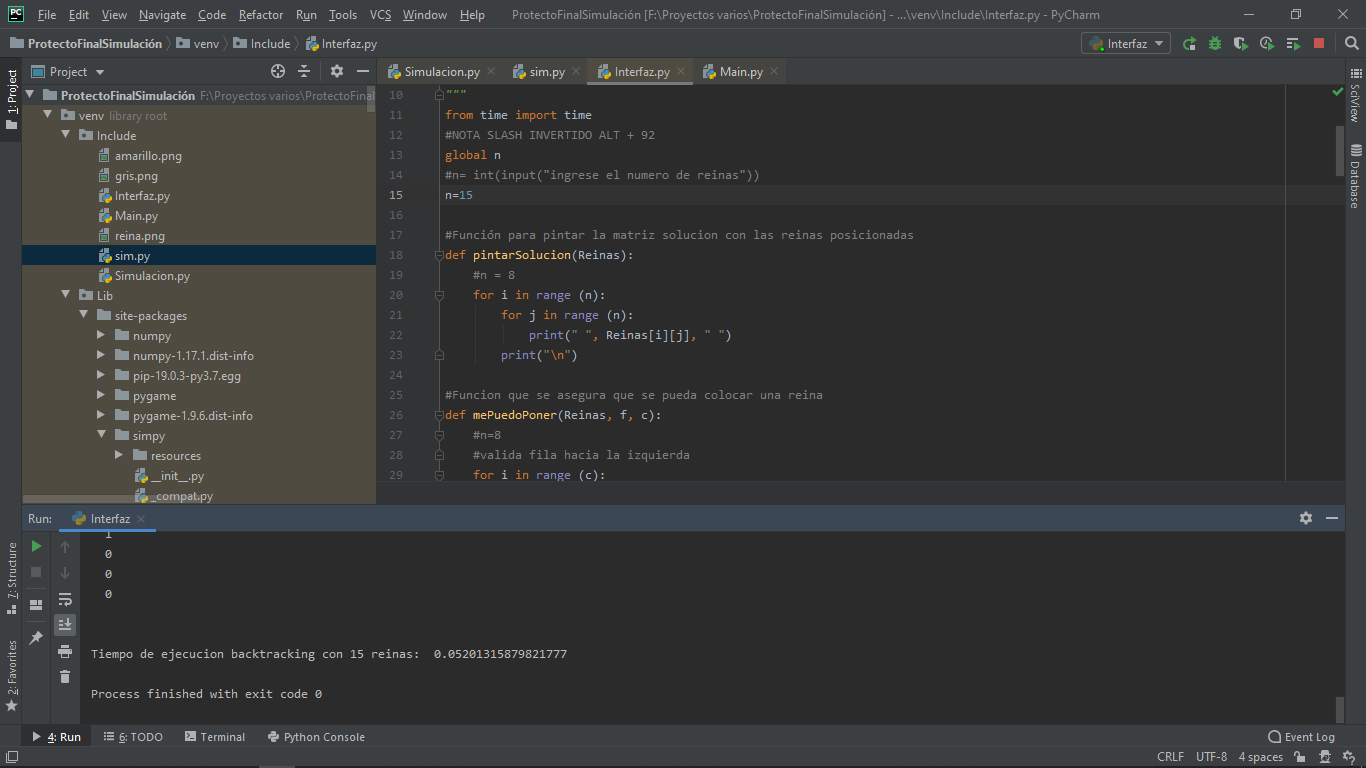
T5 = 0.0110015869140625

**Promedio tiempos back tracking con N = 12:** 0.01120047569 segundos

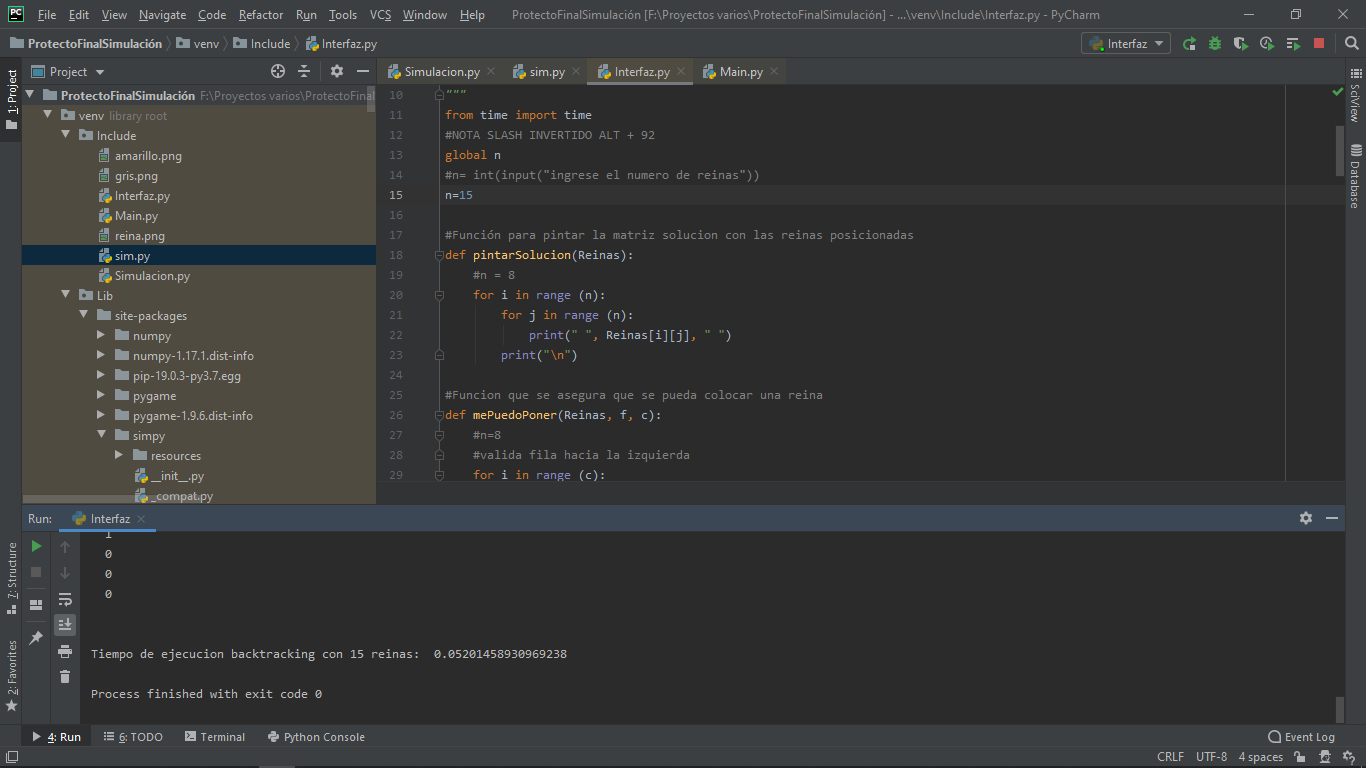
**N = 15 Determinista**



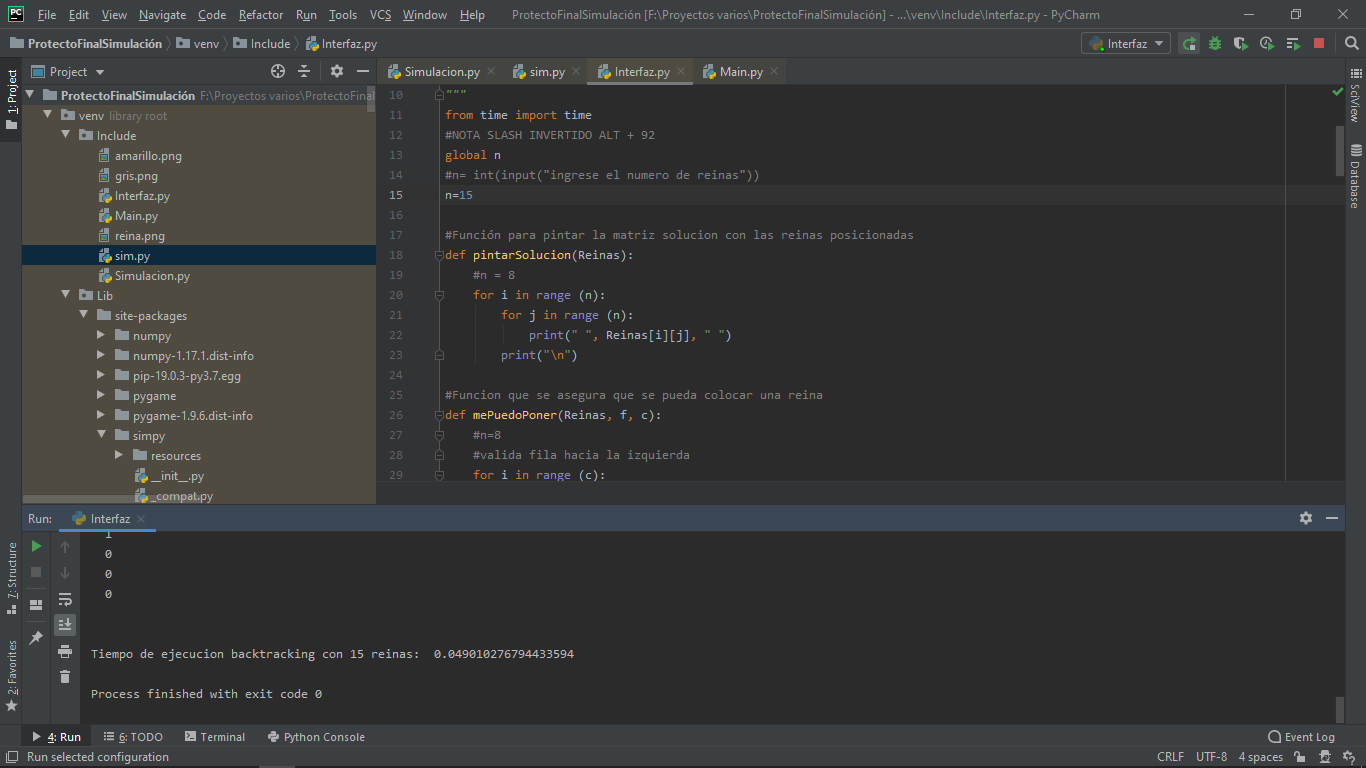
T1 = 0.051011085510253906



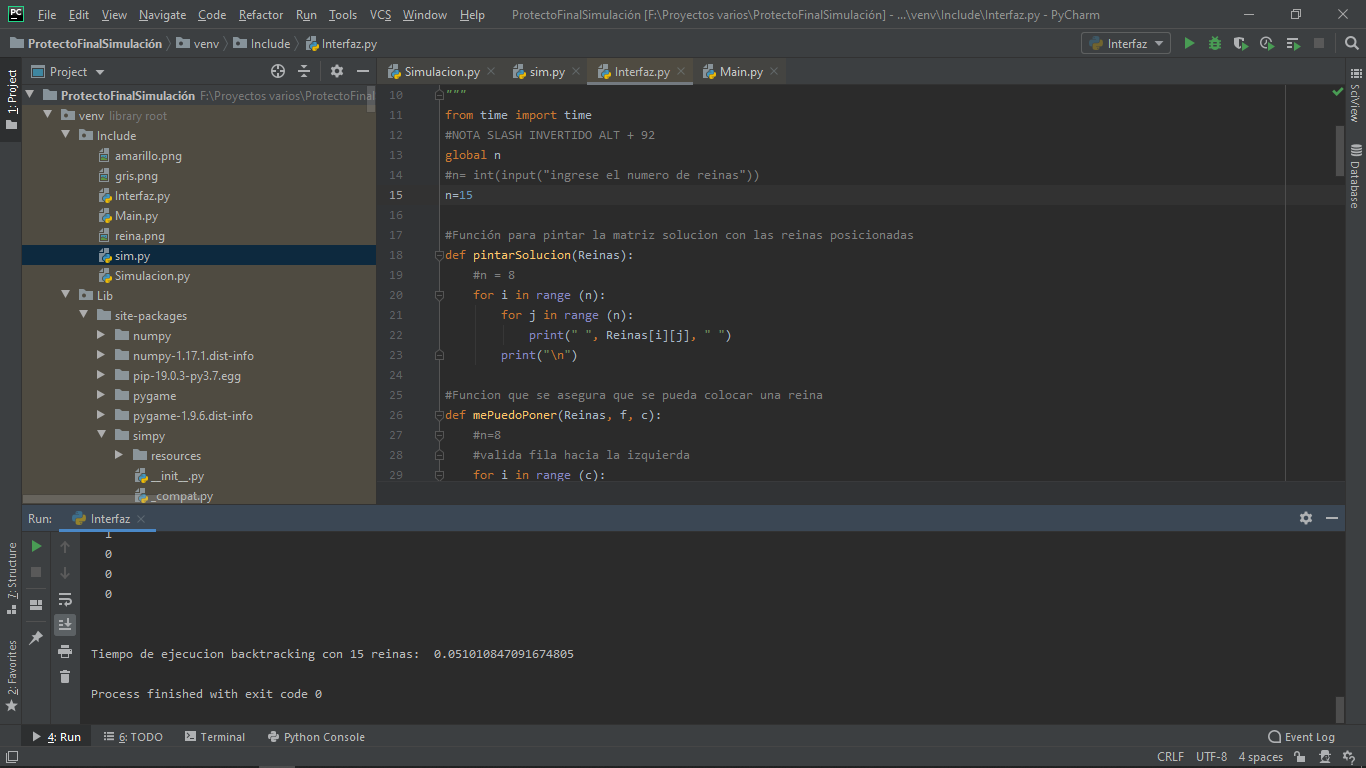
T2 = 0.05201315879821777



T3 = 0.05201458930969238



T4 = 0.049010276794433594



T5 = 0.051010847091674805

**Promedio tiempos back tracking con N = 15:** 0.0510119915 segundos

**Grafica tiempos algoritmo las vegas**

**Gráfica tiempos algoritmo determinista**

**NOTA:** Debido a los altos tiempos del algoritmo las vegas no ponemos ambos algoritmos en una sola grafica ya que no se podría apreciar una comparación.

**¿Qué recomienda de acuerdo a sus escenarios?**

**Escenario1:** Se nota que el profesor resuelve los tableros de mayor tamaño más rápido que los robots.

**Escenario2:** Se incrementa la llegada de los robots (llegan más rápido), el jugar más partidas, el profesor tiene más posibilidades de ganar, pero también de perder.

**Escenario3:** Se pone un tablero fijo para todas las partidas, el profe siempre gana (Así como nosotros :’)) ya que es un tablero grande y el profesor es más ágil para estos tableros.

**Recomendación final:** Para mejorar las ganancias del profesor (ya que está sin trabajo y dinero) y del pobre estudiante (que está intentando pasar el curso) se pueden dedicar a jugar al maestro del ajedrez con n reinas contra los robots, pero escogiendo tableros grandes porque de tanto jugar el profesor ya se sabe las posiciones secretas de las reinas que le llevarán a la victoria