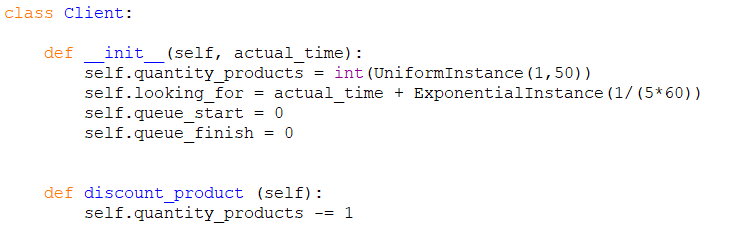
**Problema 1**

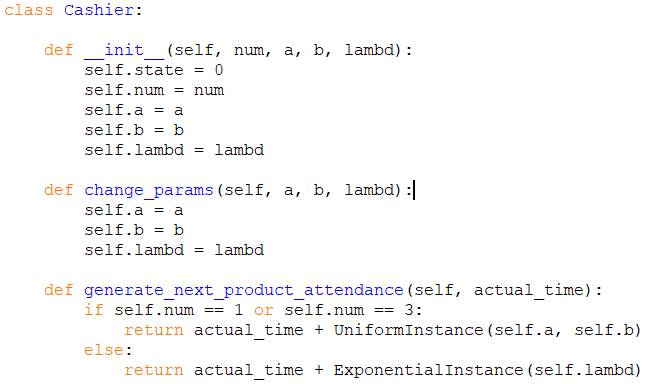
El código generado para este problema se encuentra en este [link](https://github.com/narierao/Tarea-4/blob/master/Pregunta%201.py)

Todas las tasas del problema fueron ajustadas para que estuvieran dadas por la unidad de tiempo de segundos.

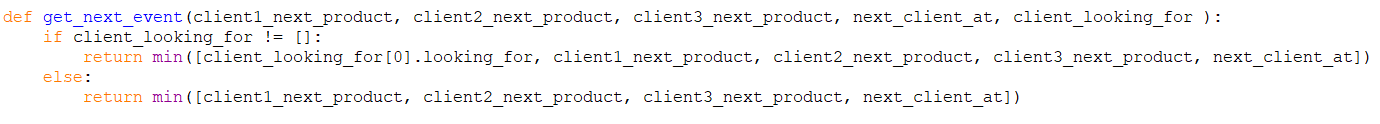
Explicación del código:



Esta sección del código es una clase que corresponde a los clientes, en donde en sus atributos encontramos la cantidad de productos dada por una Uniforme(1, 50) sacando su parte entera. Esta clase se instancia en el momento en el que llega un nuevo cliente, por lo que se encontrará buscando sus productos desde el tiempo que llega + una exponencial que distribuye 1/300. Por otro lado, se tiene que si estos entran en la cola, se guarda su tiempo de entrada y salida para poder obtener información de los tiempos medios de espera en la cola.

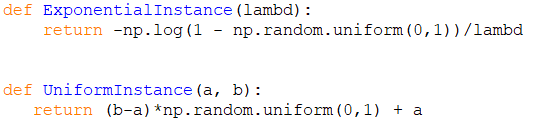


Esta clase corresponde a los cajeros, en donde se setean sus tasas al momento de instanciarse y estas se modifican en función del tiempo, además, retornan cuánto falta para poder procesar el siguiente producto del cliente que se encuentran atendiendo.

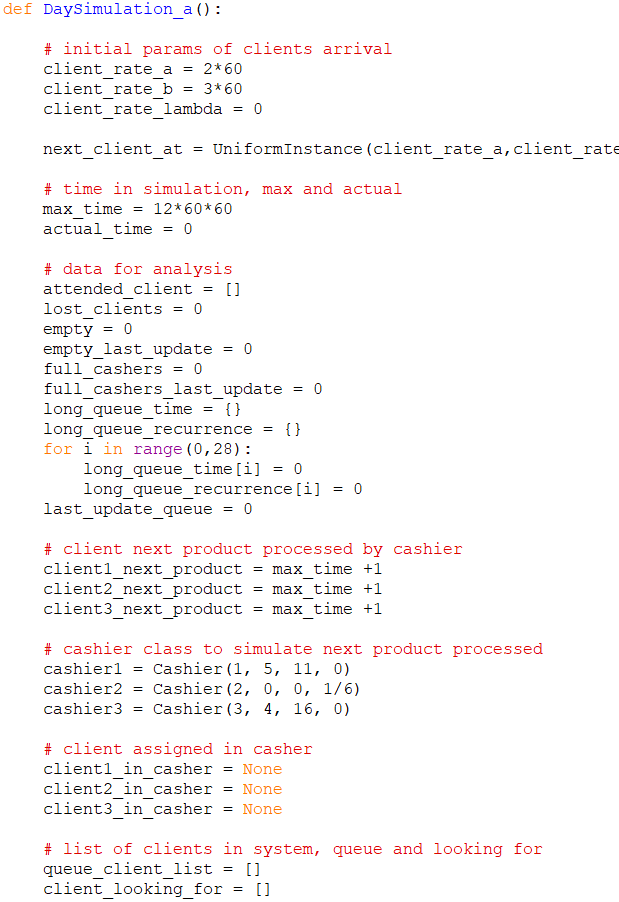


Esta funcion nos permite obtener el siguiente evento que va a ocurrir, el cual será escogido como el mínimo entre los tiempos de:

1. Procesamiento de un producto por parte del cajero 1
2. Procesamiento de un producto por parte del cajero 2
3. Procesamiento de un producto por parte del cajero 3
4. Llegada de un cliente
5. El mínimo de la lista de clientes que se encuentran viendo sus productos en el supermercado.



Estas dos funciones nos generan instancias de una exponencial y una uniforme utilizando el teorema de la transformada inversa con una distribución Uniforme(0, 1)



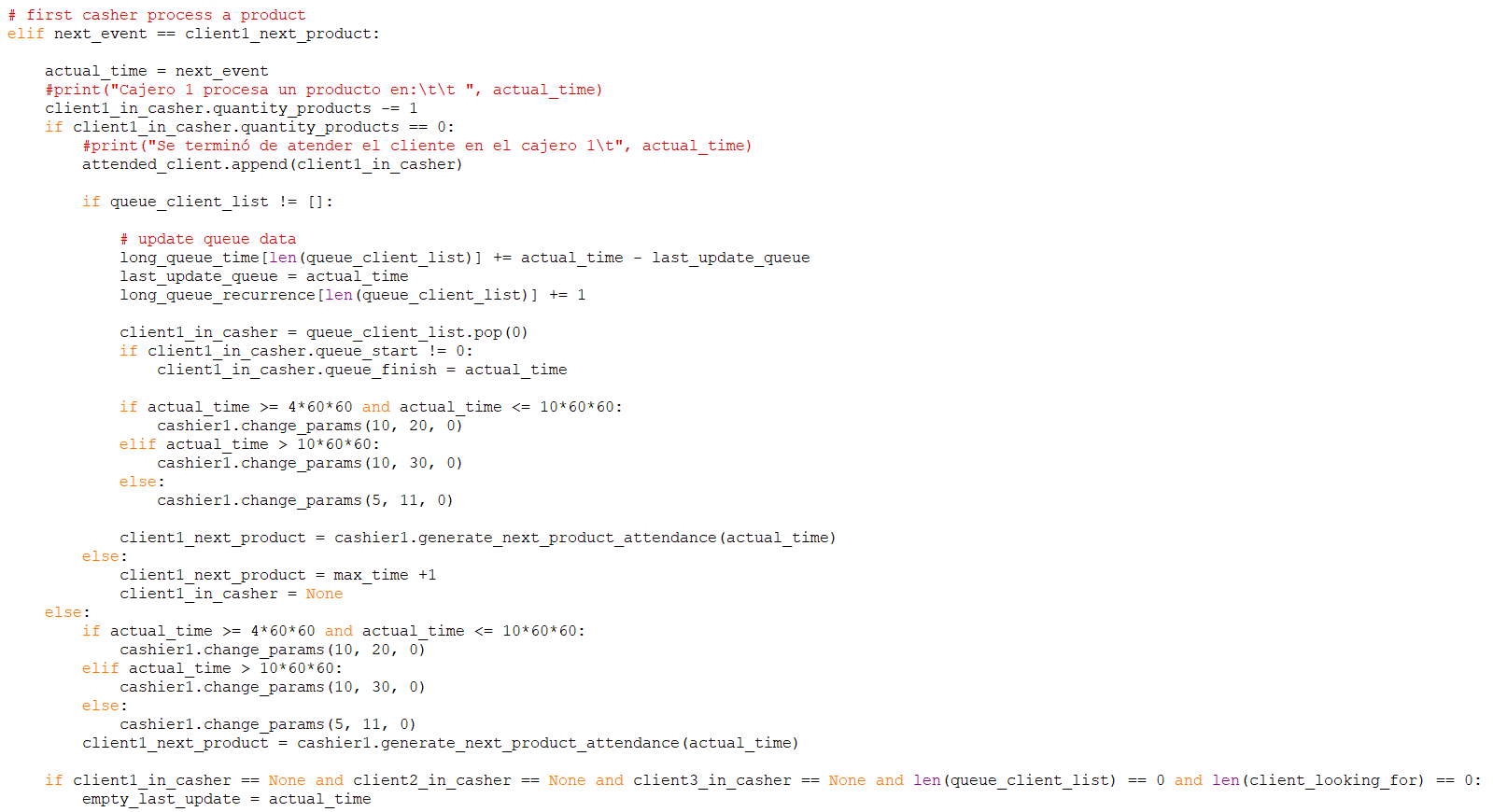
Esta función es la encargada de generar un día de simulación para la parte “a” del problema 1. Se setan parámetros iniciales de la simulación y variables necesaria para el análisis.



Comienza la simulación y se evalúa si el siguiente evento se pasa del tiempo máximo permitido para terminar la simulación. Se hacen cálculos previos para analizar los datos recolectados y se retornan para poder calcular las medias de 100 simulación en una función que explicaré más adelante.



Este segmento de código procesa si es que el evento más próximo es un cliente. Se verifica que no hayan más de 30 personas en toda la tienda entre los cajeros, y ambas colas (los que se encuentran buscando productos y los que están esperando ser atendidos). Si son menos de 30, se agrega elcliente a la lista de los que están buscando cosas, si no, se suma un cliente perdido. Se corroboran las tasas en base al tiempo y se calcula la llegada del siguiente cliente. Se avanza el tiempo cada vez que ocurre cualquier evento.

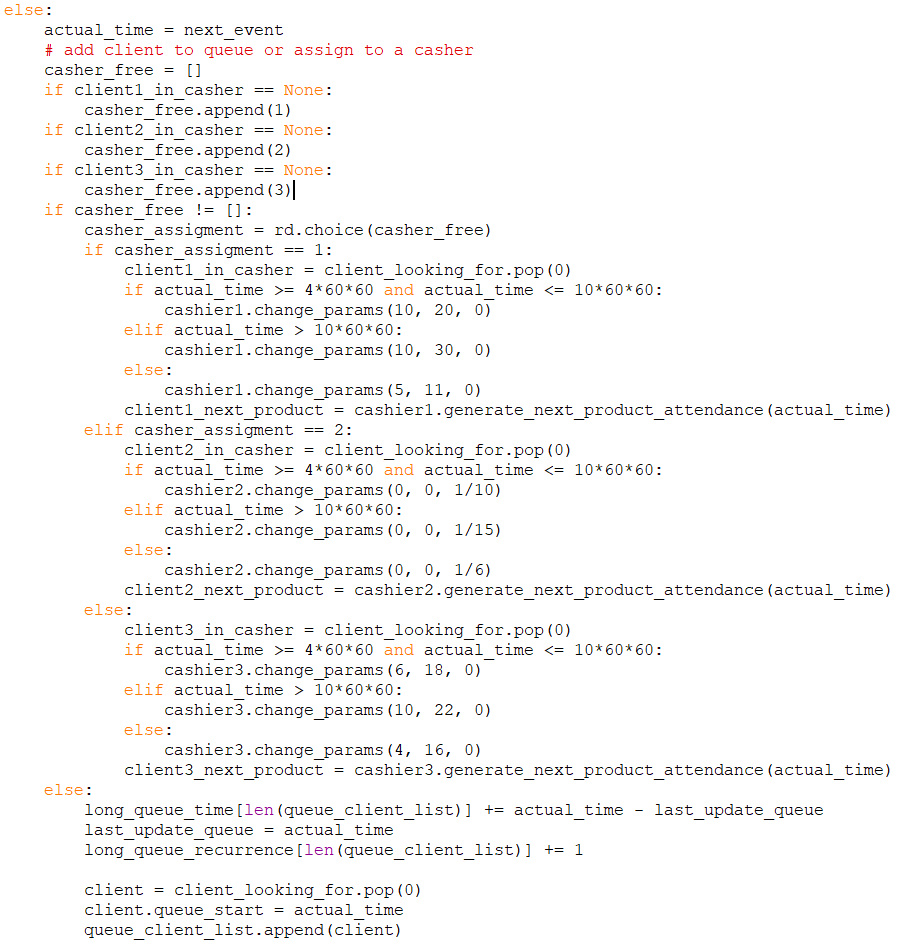


Este código se repite para el cajero 1, 2 y 3. Lo que hace es que el cajeri i procesa un producto del cliente que está atendiendo. Aquí pueden ocurrir 2 casos.

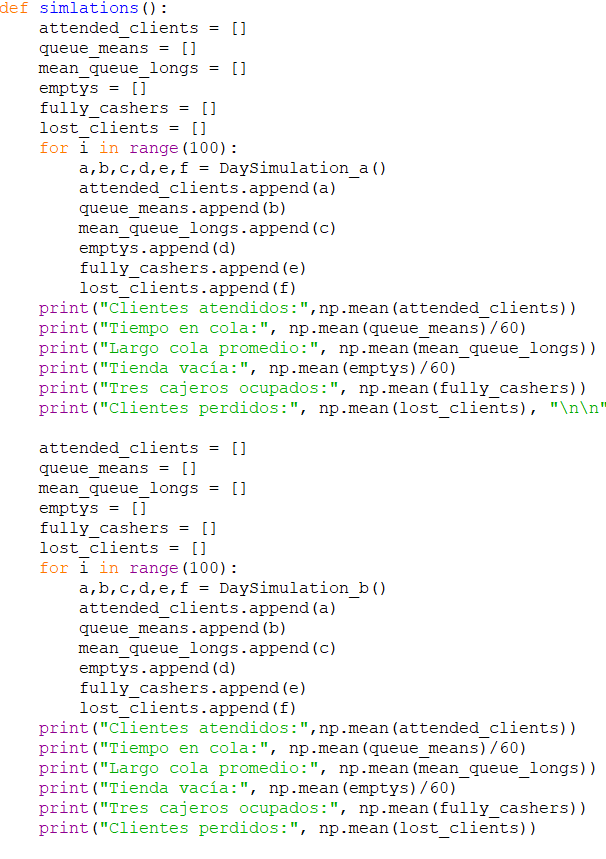
1. Si el producto no es el último del cliente, se calcula el próximo evento de procesar un producto en base al tiempo actual.
2. Si el producto corresponde al último del que estoy atendiendo, se verifica la cola de espera a pagar.
   1. Si esta se encuentra vacía, se asocia un None al cliente del cajero, ya que no tiene un cliente. Al tiempo de atención del próximo producto se le asigna el valor máximo + 1 para que no influya en los próximos cálculos.
   2. Si hay al menos un cliente, este es asociado al cajero y se calcula en proximo evento de procesamiento de un producto en base al tiempo actual y al cajero que lo está atendiendo.

Finalmente, encontramos el código que corresponde al evento de que un cliente deja de buscar sus productos en el supermercado, lo que puede llevar a 2 casos:

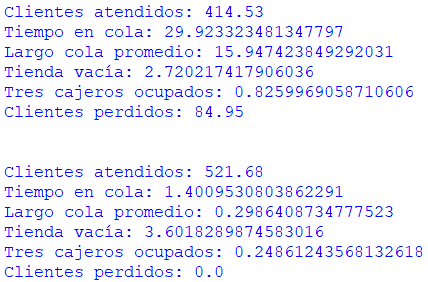
1. Si es que hay cajeros desocupados, se le asigna de manera random cualquiera de los que se encuentre disponible, es necesario calcular el procesamiento del siguiente producto.
2. Si los tres cajeros están ocupados, se agrega la persona a la cola y al cliente se le setea el tiempo en el que entra a la cola, posteriormente, cuando este sale de la cola, se le asigna ese tiempo final (con esto, calcularemos el tiempo promedio de espera en la cola).



El código que se muestra a continuación muestra el cálculo de las variables y la simulación de los 100 días del supermercado:



La simulación de la parte b es la misma que en a, pero cambia la tasa a la cual atienden los productos, para así cumplir con lo que piden. Finalmente, los resultados de la simulación arrojan los siguientes datos:



En donde los tiempos pedidos se encuentran en minutos. La primera sección corresponde a los datos simulados en a y los segundos de los datos Simulados en b.

**Problema 2**

1. Para calcular el valor de c es necesario integrar la función densidad en el recorrido e igualar el resultado a 1.

Evaluando en los extremos de la función obtenemos que:

Despejando c obtenemos:

1. Para calcular la acumulada, debemos calcular la integral entre 0 y X.

Finalmente, la función acumulada es:

1. Para utilizar el método de la transformada es necesario que calculemos la inversa de la función acumulada. En este caso, esto corresponde a:

Finalmente, tenemos que y es una U(0,1) .

Para generar esta uniforme utilizaremos el método enseñado en clases.

Sea:

Para este caso escogeremos un m grande, como 10.000, esto es para tener valores más variados de , podríamos escoger una valor de a = 1 y c = 1 pero con esto haríamos el método dejara de ser tan aleatorio, ya que estamos condicionando a priori a que las realizaciones de la variable aleatoria uniforme(0,1) aumente de 1/m en cada iteración.

Si tomamos otros valores de a y c como a = 2 y c = 5 las realizaciones de la variable aleatoria se vuelven “a la vista” más aleatorio, no tan secuencial como en el ejemplo anterior. Lo que más importa es que el valor de m sea lo más grande posible, ya que nos entrega un “conjunto” de posibles Xi de mayor cardinalidad, lo que haría la muestra más “dispersa” con más realizaciones de la variable Ui.

1. Para desarrollar un algoritmo de aceptación y rechazo explicado en clases, debemos seguir los siguientes pasos:
   1. Generamos una instancia de una instancia U ~ Uniforme(0,1)
   2. Hacemos t(x) = max(f(x)) en donde t(x) es una función constante e igual a 1.

Tenemos que t(x) = 1

* 1. Integramos t(x) en el recorrido de la función [0,c] obteniendo v = c.
  2. Generamos la función h(x) = ~ Uniforme(0,c)
  3. Hacemos g(x) =
  4. Finalmente, tenemos que si U ≤ g(Y) tal que Y ~ h(x) Aceptamos el valor, si no, nos devolvemos al paso en a.

1. Tenemos que la cantidad de iteraciones para poder obtener una instancia distribuye Geomética(), por lo que el número esperado de rechazos es c-1, ya que la geométrica me da el primer éxito.

Resolviendo, obtenemos que el número medio de rechazos es: 1.38 -1 = 0.38.