



Universidad de Concepción
Facultad de Ingeniería
Departamento de Ingeniería Industrial



PROGRAMACIÓN APLICADA A LA INGENIERÍA INDUSTRIAL

TAREA 1

Profesor: Carlos Contreras Bolton

Fecha: 09 de Octubre 2020

1. Entrega

Martes 20 de Octubre del 2020 hasta las 23.59 en Canvas.

2. Objetivo de la tarea

- Desarrollar su capacidad para llevar a cabo un programa de pequeña escala en el lenguaje de programación Python.
- Generar tipos de datos consistentes y comprender los estilos de implementación disponibles en Python, particularmente el uso de estructura de datos lineales.

3. Enunciado

En un banco se tienen múltiples filas, las cuales se encuentran divididas en varios tipos de filas: para público general, titulares de cuenta y empresas. Las dos primeras tienen la característica que si llega una persona de la tercera edad, movilidad reducida o embarazada serán atendidas primero. Si llegan varias de ellas al mismo tiempo (asumiendo que todavía no ha terminado la(s) persona(s) que está(n) atendiéndose) se ubican atrás de ellas (antes que el resto de las personas). Hay que tener en cuenta que las personas con preferencia se ubican en la fila que les corresponde.

El banco cuenta con varias cajas para atender a su público, las cuales son especializadas en un tipo de cliente. Se atiende al cliente con preferencia, dependiendo de su tipo. Cuando una caja no tiene más clientes de su tipo, puede perfectamente atender a otro cliente de alguna de las otras filas, a la fila que le quede más clientes.

Las personas al pasar por caja, indistintamente qué operación realice, ya sea pagar, depositar, girar, etc. tiene un tiempo para realizar la operación (tiempo discreto). Además, cada persona debe identificarse si corresponde a una persona que tiene preferencia o no y qué tipo de cliente es.

Por pantalla se debe mostrar en todo momento las cajas (enumeradas y de qué tipo es), a qué persona se está atendiendo (mostrando sus características), los clientes que se encuentren en las distintas filas, tiempo de llegada, tiempo de demora en el trámite de cada cliente. El usuario debe ser capaz de avanzar paso a paso la ejecución del programa, de tal manera de ir viendo el movimiento de los clientes dentro del banco. Mostrar además el tiempo actual que se encuentra.

Además, el programa debe considerar una pila con los clientes que vayan saliendo de la caja. Deberá ser mostrada en todo momento e indicar en qué caja estuvo. El programa debe permitir salir en cualquier momento, guardar la información procesada.

4. Entrada

El formato de entrada es el siguiente: un archivo de texto plano, en donde la primera línea indica la cantidad de cajas, cantidad de cajas de público general, cajas de clientes y cajas para atender empresas. Todos separados por un espacio. Las siguientes líneas serán los clientes que ingresan al banco. Cada línea contendrá la siguiente información:

1. Identificador de cliente, un número correlativo.
2. Tipo de cliente. En donde puede ser:
 - g** Público general.
 - e** Empresa.
 - t** Titular de una cuenta.
3. Preferencia:
 - 0** Cliente sin preferencia, este valor será también para clientes empresas.
 - 1** Cliente con preferencia.
4. Tiempo de llegada.
5. Tiempo de demora en el trámite.

El archivo de entrada se llama “**banco.in**”. El archivo de entrada no necesariamente se encuentra ordenado por orden de llegada.

5. Salida

Como se mencionó anteriormente, las salida es por pantalla. Se sugiere utilizar una ventana de 80×60 caracteres. No se debe limpiar la ventana, de forma que la información vaya apareciendo, de manera que lo más antiguo suba. Se sugiere marcar con línea los movimientos que se irán haciendo.

En un archivo de texto plano, llamado “**banco.out**” son guardadas las cajas, junto a la persona que se encuentre atendiendo actualmente, separado por un espacio (si no se encuentra atendiendo a ninguna persona, colocar un guión ‘-’). Cada caja corresponde a una fila, separado por un espacio en blanco, la fila de personas clientes titulares, la fila de empresas, la fila de público general y finalmente, la pila de clientes ya atendidos. Todas las filas y la pila se encuentran separadas por un espacio. Cada cliente corresponde a una fila. Si una fila no tiene más clientes, se coloca un guión ‘-’.

Tabla 1: Ejemplo de entrada y salida de archivos. El programa se detiene en el instante 2.

Entrada	Salida
3 1 1 1	1 2
1 e 0 1 3	2 3
2 g 0 1 2	3 1
3 t 0 2 1	4
4 t 1 4 1	-
	-
	-

En la Tabla 1 se muestra un ejemplo de entrada y del archivo de salida. Se detiene el programa en el instante 2. En donde quedan todas las cajas ocupadas con personas. La persona con id 4 queda en la fila de titulares y ninguna más en las otras dos filas. Tampoco en ese instante hay personas que han terminado su atención.

6. Condiciones de la tarea

- La tarea es individual.
- Las dudas respecto al trabajo, se realizan de manera personal (vía Teams) o vía correo electrónico.
- Está estrictamente prohibido utilizar matrices, arreglos y listas de Python para realizar las operaciones que se necesitan. Utilice otra estructura de dato.
- El lenguaje a utilizar es Python 3.x de manera estándar. No se aceptarán Bibliotecas/Módulos no vistos en clases.
- Cree las bibliotecas o utilice las bibliotecas dadas por el profesor.
- El enunciado podría sufrir modificaciones que serán publicadas en Canvas.
- Formato de entrega:
 - El código debe estar en un archivo de extensión **.py**.
 - El sistema debe ser robusto, se penalizarán las caídas de cualquier tipo.
 - Debe estar bien documentado.
 - La entrega se hace mediante la plataforma Canvas, en el link disponible para subir la tarea.
 - En caso de detectarse copia, los estudiantes involucrados tendrán la nota mínima sin apelación.
 - El archivo final subido a la plataforma debe ser un archivo comprimido con el siguiente formato:
`letraInicialNombre.ApellidoPaterno.letraInicialApellidoMaterno.extensión`
Ejemplo: Juan Pérez Valdivia j_Pérez_V.zip
 - Si considera necesario, se puede agregar un archivo de texto plano, llamado **LEEME.txt**, donde se pueden agregar instrucciones para que el profesor pueda ejecutar de manera correcta su tarea. Sin embargo, no se permiten instrucciones en la cuales se deba intervenir el código para el buen funcionamiento.
 - En caso de no seguir alguna de las condiciones o instrucciones será penalizado con 1.0 punto por cada infracción.