|  |  |  |
| --- | --- | --- |
| TEC | Estructuras de datos | |
|  | | Proyecto #1 |
|  | | Semestre II 2017 |

## ESPECIFICACIÓN DEL PRIMER PROYECTO PROGRAMADO

# Justificación

El desarrollo de programas de simulaciones es una de las áreas de aplicación más interesantes de la programación, pues provee estimaciones eficaces y eficientes a problemas que de manera manual son complicados.

El primer proyecto programado consiste en desarrollar una simulación de este tipo que fortalecerá su formación como programador.

# Descripción

El proyecto consiste en desarrollar un programa en C++ o C que implemente una simulación del proceso productivo de una fábrica de galletas de chocolate.

A continuación se presenta un diagrama del proceso productivo que se debe representar:



Es importante destacar que la simulación se debe representar gráficamente como se detalla en el diagrama anterior (el diagrama es una sugerencia, no necesariamente se debe representar con las mismas imágenes). La simulación tiene los siguientes requerimientos:

En la fábrica hay varios componentes:

1. **Galletas de chocolate**
   1. Hay **muchos (n)** tipos de galletas, en realidad es la misma galleta pero empacada en diferentes presentaciones (paquete de 4, paquete de 10 de 4, tubo de 16, etc.), cada uno tiene una cantidad de galletas. Según su cantidad por ejemplo: tubo (16 galletas), paquetito (4 galletas). Los tipos de galletas se definen en una LISTA DOBLEMENTE ENLAZADA CIRCULAR, esto antes de iniciar una simulación. También se puede deterner una simulación y crear más tipos.
   2. Las galletas son los objetos del proceso productivo, que consiste en:
      1. **Una persona planifica** la producción, el Planificador. Este indica cuántas unidades de cada uno de los tipos de productos se van a fabricar en una ejecución de la simulación. Por ejemplo, dirá: 100 unidades de paquetitos, 250 unidades de tubos. En este caso se requieren fabricar 4400 galletas, para lograr el plan establecido.
      2. Receta: la galleta debe tener una receta de cantidad de masa y cantidad de chocolate. Ambas cantidades en gramos. Puede modificarse la receta mientras la simulación esté detenida (pausa o antes de iniciar).
      3. Almacén de materia prima: en este almacén existe ilimitadas cantidades de mezcla para masa y chocolate. Solamente tiene un carrito que reparte mezcla o chocolate en las máquinas para este fin. El carrito debe configurarse en capacidad de transporte en gramos de mezcla y de chocolate que entrega a las máquinas en x segundos. Además, si está encendido o apagado.
      4. Máquinas de mezcla y chocolate: son dos máquinas de mezcla y una de chocolate, que tardan x tiempo en procesar n cantidad de gramos (configurar estos valores en cada máquina). La máquina además, debe tener un mínimo y un máximo de gramos de mezcla o chocolate (configurar en cada máquina), cada vez que llega a ese mínimo, según lo que procesa, debe pedir al almacén más mezcla para completar el máximo. Estas peticiones deben ir a una COLA, que el almacén procesa conforme llegan peticiones en orden. DEBE PODER VERSE LAS PETICIONES PENDIENTES DEL ALMACEN Y LAS QUE HA PROCESADO; ADEMÁS LA CANTIDAD DE MEZCLA O CHOCOLATE QUE YA HA PROCESADO CADA MAQUINA Y LA QUE TIENE PROCESANDO EN EL MOMENTO. (tipo, cantidad solcitada, cantidad entregada).
      5. Máquina ensambladora: es la máquina que combina la masa con el chocolate para hacer las galletas, que serán horneadas posteriormente. Esta máquina tiene antes dos bandas transportadoras, con una capacidad máxima de mezcla y chocolate (una banda es para las dos máquinas de mezcla, otra sola para el chocolate). Si las máquinas anteriores llenan una banda transportadora, automáticamente se paran las mezcladoras o la de chocolate, según la que se llene. La máquina hace n galletas en x segundos y disminuye la cantidad correspondiente de mezcla o chocolate que tiene la banda correspondiente. DEBE VERSE EN TODO MOMENTO LA CANTIDAD DE MEZCLA Y CHOCOLATE QUE TIENE LA BANDA. DEBE VERSE EN TODO MOMENTO LA CANTIDAD DE GALLETAS QUE HA HECHO LA MAQUINA.
      6. Las galletas pasan al Horno, el cual se llena con una banda donde la emsambladora coloca las galletas, esta banda con un máximo y actual (DEBE VERSE ESOS DATOS SIEMPRE). El horno tiene una capacidad de galletas para hornear, y solo si se completa esa cantidad, inicia el horneado. Sin embargo, el horno tiene de 2 a 6 bandejas, cada una separada en capacidad de horneado. Debe configurarse cada bandeja en cantidad de galletas y tiempo de horneado. Además, las bandejas deben utilizarse de la primera a la sexta, en ese orden, si la primera está ocupada, se empieza a llenar la segunda. Como son de 2 a 6 bandejas, debe darse la opción de encender o apagar las 4 últimas, 2 siempre estarán activas. Si las bandejas activas están ocupadas y la máquina ensambladora sigue enviando galletas, esta última debe detenerse automáticamente. DEBE VERSE SIEMPRE LA CANTIDAD DE GALLETAS EN ESPERA Y HORNEADAS POR CADA BANDEJA.
      7. La máquina empacadora se llena después del horno, el cual, cada bandeja pasa la cantidad de galletas horneadas por la banda transportadora hacia la mÁquina empacadora (la banda tiene máximo y actual). La diferencia con esta banda es que tiene dos inspectores, los cuales son personas que revisan las galletas y sacan y desechan algunas galletas. Para esto, a cada uno se le configura una probabilidad de desecho, y debe generar un random que determine si desecha galletas o no, y cuántas desecha. DEBE VERSE LA CANTIDAD DE GALLETAS QUE APROBÓ Y LAS QUE RECHAZÓ CADA INSPECTOR.

La máquina empacadora tiene una capacidad de empaque de n paquetes del mismo tipo en x segundos (mejor si se configura esto por tipo, es decir, cada tipo lleva una cantidad por unidad de tiempo, en cuanto a empaque). Para determinar cual tipo va a empacar, se debe crear una probabilidad para cada uno de los tipos de empaques (debe sumar 100 las probabilidades de todos los tipos), se genera un random para determinar cuál es el paquete que se empacará. DEBE VERSE EN TODO MOMENTO LA CANTIDAD EMPACADA POR PAQUETE.

* + 1. Al terminar de empacar, existe n cantidad de tipos de transportadores, uno por cada tipo de paquete. Cada transporte, tiene una cantidad máxima del tipo de paquete que puede transportar en x tiempo. Cuando se llena hasta el maximo, va a entregar al almacén final en ese tiempo. Si la máquina empacadora hizo un tipo de paquete cuyo transportador no está porque anda en el almacén, debe esperar a que vuelva el transporte. DEBE VERSE EN TODO MOMENTO EL MÁXIMO Y EL ACTUAL DE CADA TRANSPORTE.
    2. Almacén de producto terminado, se debe ver la cantidad de cada paquete y el total de galletas que han llegado.
  1. **Máquinas y personas:** Todas las máquinas y personas tiene dos posibles estados:
     1. **OFF**: no está trabajando, no hace su tarea. Si es máquina no hay producción.
     2. **ON**: Está trabajando.

1. **Bandas transportadoras**
   1. Las bandas transportadoras son colas donde se van colocando galletas o mezclas para pasar de una etapa del proceso productivo a otra. Son alimentadas por una máquina y alimentan a otra máquina. (implementada mediante UNA COLA)
   2. Cada cola tiene un límite de galletas o mezcla que puede contener, este límite debe ser modificable en todo momento por el usuario (parámetro).
   3. En la pantalla, en todo momento se debe mostrar la cantidad de galletas o mezcla que hay en cada cola y el máximo seteado.
   4. Cuando una cola ha alcazando su capacidad máxima no puede recibir más galletas o mezcla y colocará la máquina que la alimenta en pausa hasta que la cola tenga espacio para alimentarle mezcla o galletas.
2. **Máquinas**
   1. Tienen una velocidad de procesamiento como parámetro, esta velocidad indica cada cuántos segundos (**con decimales**) hacen n cantidad de galletas o procesan m cantidad de mezcla. **Estos parámetros pueden ser modificables en cualquier momento por el usuario.**
   2. Una máquina puede estar en 2 diferentes Status:
      1. **OFF**: la máquina está apagada, por tanto no procesan.
      2. **ON**: la máquina está encendida, por tanto procesa cada x segundos n cantidad, según lo indique su parámetro.
   3. Las máquinas deben indicar en todo momento cuántas galletas o mezcla de cada tipo han procesado y están procesando (esto donde aplique, como el horno)

# SIMULACIÓN

1. **Controles**
   1. La simulación tendrá los controles:
      1. Play: inicia la simulación
      2. Pausa: pausa la simulación
      3. Stop: detiene totalmente la simulación
      4. Resume: reanuda la simulación después de estar en pausa (puede ser el mismo de pause)
   2. Adicionalmente cada máquina o persona podrá ser puesta en ON o en OFF según el usuario lo desee
   3. Al inicial la simulación TODAS las máquinas y personas están en status OFF
2. **Parámetros**
   1. Todos los valores que se indican como parámetros se pueden modifcar por el usuario, apenas un usuario modifique uno de los parámetros la simulación cambiará para adaptarse a los nuevos parámetros.
3. **Ejecución**
   1. El usuario podrá iniciar ejecutando una nueva simulación.

# Detalles de Implementación

El programa se desarrollará completamente en modo gráfico.

Se calificará la presentación y creatividad usada para representar el juego.

# Aspectos Administrativos

El proyecto será realizado en parejas o individual.

Se debe realizar documentación escrita que contenga las siguientes partes: Portada, Análisis de Resultados (qué sirve, qué no sirve, principales problemas encontrados), y conclusiones. La documentación escrita NO debe incluir el código fuente como parte del documento.

Además los archivos fuente deben estar **documentados internamente**. Se deben documentar las funciones (descripción de cada una, qué hacen y qué hace cada entrada) y las variables globales.

El proyecto será entregado el día sábado 7 de octubre con citas de revisión.

Al entregar el proyecto los estudiantes deben garantizar que el programa funciona correctamente, si este no puede ser leído el programa se calificará con nota cero.

Implementarlo en C o C++. No usar Visual Studio ni orientación a objetos (caso C++).

“Si usted quiere tener resultados sobresalientes, debe tener objetivos sobresalientes”. ***Mitchel Leibitniz***