

Pipeline sacerdote Playmobil

La empresa Brandstater desea fabricar y vender un nuevo muñeco trending en la comunidad, para ello, indicará a su línea de producción cada uno de los procesos a realizar, con el fin de controlar los ingresos de los materiales y egresos de sus productos finales correspondientes a las ventas.



Se comienza con un archivo que llega con los paquetes de materia prima que se necesita para diseñar el producto, la cantidad de materia es aleatoria.

Cada uno de los procesos recorrerá el archivo recibido correspondiente exactamente una sola vez.

1º Proceso Pipeline (Fabricante)

Registro archivo "Paquetes.bin"

```
typedef struct {  
    int cantMateria; //gramos  
    char calidad; // "a", "b", "c"  
    float costo; //cada bloque posee distintos costos en  
    base a los gramos  
}PaqueteMateriaPrima;
```

Llega la materia prima en paquetes de distintos tamaños. Se utilizará la cantidad de materia necesaria para generar una sucesión de 3 componentes distintas y se irá descontando del paquete cada una, una vez que el paquete posee material insuficiente para fabricar la siguiente componente, ese material restante se descartará y se procederá al siguiente paquete para tratar de fabricar esa misma componente.

A medida que se genera una componente se efectúa la salida sobre el archivo de componentes.

- Se deben generar componentes con fallas aleatorias, entre 0 y 2.
- Los costos de cada componente son: costo materia prima + costos de proceso:

Componente	Materia prima(gr)	Costo proceso(\$)
Muñeco	25	6
Copa de vino	3	0.75
Biblia	2	0.50

TP Final Laboratorio 1 Comisión 4 y 6 - 2021

UTN FRMDP

Efectuar un archivo de salida con la correspondiente información por componente

Registro archivo "Componentes.bin"

```
typedef struct{
    char nombre[20]; //"Biblia", "Copa de vino", "Doll"
    int falla; //puede tener 0...2 fallas
    float costo;
    char calidad; //según la calidad de la materia, será la calidad
del producto
}Componente;
```

Ejemplo de cómo realizar el proceso de fabricación: [Video](#)

2° Proceso Pipeline (Ensamblador)

Se recibe el archivo de componentes listos y ensamblará las partes para generar el producto final.

Se trabajará con 3 arreglos de tipo Componente, cada uno contendrá un tipo de componente ("Doll", "Copa de Vino", "Biblia").

-Si el producto posee 2 fallas, se descartará. Cada costo será almacenado en **su propio** registro del archivo de "Perdidas.bin".

-El proceso de ensamblado tiene un costo definido por la calidad de producto:

Calidad componente	Costo de ensamble(\$)
a	14
b	12
c	10

- Para generar el producto final se necesita: 1 biblia, 1 copa de vino, 1 muñeco de la **misma calidad**.
- El costo final será: costo componente + costo ensamble
- Se generará la cantidad de productos que pueda y se hará la salida de los productos listos.
- El archivo "Perdidas.bin" posee un dato de tipo float almacenado por cada falla.

Registro Archivo "ProductosFinales.bin"	Registro Archivo "Perdidas.bin"
<pre>typedef struct{ float costo; char calidad; }PlayMobil;</pre>	<pre>float costoPerdida</pre>

Ejemplo de cómo realizar el proceso de ensamble: [Video](#)

3° Proceso Pipeline (Venta de Lote)

Recibe el archivo de productos finales y el archivo de pérdidas y realiza la salida del Lote.

- El precio del producto parte de una base que se calcula a través del promedio efectuado de la cantidad de productos recibidos con la suma del campo "costo" del PlayMobil y se le sumará el 120%.
- A este precio se le adicionará un pequeño porcentaje según la calidad:

Calidad	"a"	"b"	"c"
Precio	5%	3%	-

Archivo "Contaduria.bin"

- Se generará un solo registro sobre el archivo.
- costoLote: suma del campo costo del archivo "ProductosFinales.bin"
- costoPerdidas: suma campo costo archivo "Perdidas.bin".

Registro archivo "Contaduria.bin"	Registro "DetalleLote.bin"
<pre>typedef struct { ///planilla final float gananciaBruta; float costoLote; //de todos los prod float costoPerdidas; int cantTotal; }Contaduria;</pre>	<pre>typedef struct { float precioFinal; int cant; char calidad; }DetalleLote;</pre>

El lote se compone de los detalles de las 3 calidades

- Se deberá informar si se se logró superar los costos totales (costoLote + costoPerdidas) de todos los productos.

Pasos para procesamiento de datos:

1. Al recorrer el archivo "ProductosFinales.bin" se procederá a cargar parte de los campos del arreglo de DetalleLote y se devuelve la suma de los costos.
2. El retorno se debe almacenar en el campo costoLote de Contaduria.
3. Al campo cantTotal se asigna la suma de las cantidades de producto final de las 3 calidades. Con ambos campos se puede sacar el promedio de costo base necesario para el precioFinal de DetalleLote.

Ejemplo de cómo realizar el proceso de Venta: [Video](#)

Condiciones de entrega:

- **3 integrantes estricto, o 4 (por excepción)**
- **No se admite ningún proyecto con algún proceso del pipeline sin realizar permaneciendo la misma cantidad de integrantes en el grupo (sin excepciones).**

TP Final Laboratorio 1 Comisión 4 y 6 - 2021
UTN FRMDP

Puntaje:

	Puntaje	Obtenido
1º Proceso	30pts	
2º Proceso	30pts	
3º Proceso	30pts	
Informe y Presentación	10pts	

Régimen de aprobación:

10	20	30	40	50	60	70	80	90	100
Desaprobado					Aprobado				