

## Miniproyecto Ruby – Cervecería Glaciar

Usted deberá **modelar e implementar** el sistema de creación de la cerveza glaciar brebaje Stolz. El sistema se describe de la siguiente manera:

- El sistema está compuesto por una serie de máquinas.
- Cada máquina tomará diferentes tipos de insumos según la tabla en el anexo 2.
- El sistema correrá en ciclos de trabajo.
- Cada ciclo de trabajo se pasará por cada maquinaria y se le pasarán insumos o se procesará según su estado. (Los estados se describen más adelante).
- Sólo se tomarán en cuenta los siguientes insumos básicos: Cebada, Mezcla Arroz/Maíz, Lúpulo y Levadura. Así como los productos de cada máquina. El agua, vapor, monóxido de carbón y aire serán ignorados por fines prácticos.
- Se consideran también como insumos los productos de cada máquina.
- La cantidad máxima de ciclos al igual que las disposiciones de insumos básicos se definirán en los argumentos de su aplicación. (Datos de la implementación más adelante).
- Tanto los insumos y los productos derivados se calcularán en kilogramos (Kg)
- Al llegar a la última máquina (llenadora y tapadora), si esta está en espera, se imprimirán el número de cervezas que se procesaron y ésta pasará a inactiva.
- Hay **4 cervezas por cada Kg de producto final**. (250 gramos por cerveza)
- Una vez el sistema terminé de correr se mostrarán cuantas cervezas se produjeron y cuantos insumos sobraron (sólo los básicos).

Deberá modelar cada una de las máquinas planteadas como clases que cumplen los siguientes requerimientos:

- Toda máquina recibe cierto insumo/s específico/s según la tabla en anexo 1.
- Por razones de simplicidad, **sólo habrá una instancia de cada máquina.**
- Cada máquina puede tener hasta **4 estados posibles**: *inactiva*, *procesando*, *en espera* y *llena*.
- Una máquina está inactiva cuando está cargando insumos.
- Cuando una maquina alcanza el máximo de su capacidad, esta pasa al estado llena.
- Una máquina que se encuentra llena pasa al estado procesando en el siguiente ciclo de trabajo (se explican más adelante).
- Una vez una maquina termina de procesar pasa al estado en espera, donde se quedará así hasta pasar los insumos a la siguiente máquina.
- **Cada ciclo de trabajo** la máquina **deberá imprimir su estado actual**, diciendo **cuantos insumos posee**.
- Los porcentajes de mezcla dicen cuantos insumos se requieren con respecto a la cantidad máxima.
- Una vez se termina el producto, se le resta el desecho. La cantidad de productos que tienes es igual a:

$$\text{Cantidad Producto} = \text{Capacidad Máxima} * (1 - \text{Desecho})$$

### Respecto a la implementación:

Su programa principal será main.rb. Tomara como argumentos:

```
./main.rb <numero de ciclos> <cantidad cevada> <cantidad  
mezcla arroz/maiz> <cantidad de levadura> <cantidad de  
lúpulo>
```

Ejemplo:

```
./main.rb 100 4000 1000 30 20
```

### Su Salida:

Serán todas las impresiones de las maquinas, las cervezas procesadas. Y los inicios y fines de cada ciclo bajo el siguiente formato:

#### Por cada ciclo:

```
Inicio Ciclo <numero>  
<Impresión por cada máquina en la cadena>  
<Impresión cervezas procesados; opcional, solo si hay>  
Fin Ciclo <numero>
```

#### Por cada máquina:

```
Maquina <Nombre>  
Estado: <estado>  
Insumos (en case de estar inactiva o llena):  
<Insumo 1> <Cantidad>  
<Insumo 2> <Cantidad>
```

Solo se imprimen los insumos si la maquina está inactiva o llena.

#### Para las cervezas:

```
Cervezas salientes: <cantidad>
```

Sólo si la última máquina de la cadena está en el estado espera.

### Esqueleto general de Salida:

```
Inicio Planta  
<Ciclos>  
Cerveza Total: <cantidad>  
Cebada Sobrante: <cantidad>  
Lúpulo Sobrante: <cantidad>  
Levadura Sobrante: <cantidad>  
Mezcla Arroz Maíz Sobrante: <cantidad>
```

Terminado el programa se muestran cuantas cervezas se produjeron (en total) y cuanto sobro de de los insumos.

### Detalles adicionales:

- Para el diseño en UML, se recomiendan los siguientes software: *Enterprise Architect, Rational, Modelio, StarUML, DIA* o el de su preferencia. También se permite escaneado siempre y cuando sea con letra clara y visible.

### Necesario para la entrega:

- Breve Informe con: Integrantes y **Diagrama de clases** en UML. En formato PDF.
- Implementación en Ruby, levemente documentada (cabeceras de clases y clase o código controlador de los ciclos de trabajo). Dentro de un **tar.gz**.
- Equipos **hasta 4 personas** (máximo).

### Entrega:

La entrega se realizará el día 1 de febrero (semana 12) más tardar las 11:59am (**mediodía**) Enviar correo con asunto “Miniproyecto Ruby – Cervecería Glaciar” a fsahmkow27@gmail, el correo **debe incluir nombres y carnets** de los integrantes. Tanto el informe como el código deben estar en el mismo correo.

**Anexos:**

Maquina	Insumos	Cantidad Máxima(Kg)	Porcentaje Mezcla	Desecho	Ciclos Procesamiento
Silos de Cebada	C	400	100%C	0%	0
Molino	PA	100	100%PA	2%	1
Paila de Mezcla	M, PA	150	40%M 60%PA	0%	2
Cuba de filtración	PA	135	100%PA	35%	2
Paila de cocción	PA, L	70	97.5%PA 2.5%L	10%	3
Tanque pre-clarificador	PA	35	100%PA	1%	1
Enfriador	PA	60	100%PA	100%	2
TCC	PA, V	200	99%PA 1%V	10%	10
Filtro de Cerveza	PA	100	100%PA	0%	1
Tanques Para Cerveza Filtrada	PA	100	100%PA	0%	0
Llenadora y Tapadora	PA	50	100%PA	0%	2

**Leyenda:**

C: Cebada    M: Mezcla de Arroz/Maíz    L: Lúpulo    V: Levadura

PA: Producto Máquina Anterior

# ELABORACIÓN DE LA CERVEZA

