

# Métodos/Técnicas de Ingeniería de Software

## -- Evaluación 1 --

### 1. Descripción del trabajo

Los alumnos, en forma **personal**, deben desarrollar y desplegar una aplicación web diseñada en base a una arquitectura por capas (monolítica). El proyecto debe ser gestionado usando SCRUM.

### 2. Evaluación

- La nota final se calcula de la siguiente manera:

$$PE = \text{PROMEDIO}(\text{CalidadProyectoProducto}, \text{CalidadRespuestas})$$

- El detalle de cada uno de los parámetros de evaluación es el siguiente:
  - **CalidadProyectoProducto:** El/La alumno(a) deberá mostrar resultados en vivo (desde su computador) tanto de la gestión del proyecto como del desarrollo y despliegue del producto de software. El profesor es quien solicita que es lo que debe mostrar.
  - **CalidadRespuestas:** Nota que refiere a la calidad de las respuestas que el/la alumno(a) entregue cuando el profesor haga las preguntas dirigidas.

### 3. Lineamientos generales

- La evaluación se realizará en forma "**personal**".
- Para la evaluación no se debe entregar ningún informe escrito.
- Cada alumno debe presentarse en forma puntual en la fecha/hora programada. En caso contrario se le calificará con la nota mínima 1.0
- A la evaluación solamente deben presentarse aquellos alumnos que fueron planificados para la fecha. No se permitirá el ingreso de otros alumnos.

## 4. Acerca del proyecto de software

### 4.1 Contexto del problema

La empresa **MilkStgo** es una empresa que se dedica a la producción de yogurt, mantequilla y quesos. Tiene más de 40 años produciendo estos productos para el mercado nacional.

Para producir el yogurt, la mantequilla y los quesos, la empresa necesita comprar la materia prima (leche) a sus proveedores ubicados en diferentes zonas del país.

Diariamente (mañana y tarde) la empresa recibe la leche de cada uno de sus proveedores en sus diferentes centros de acopio ubicados a lo largo del país.

Al final de cada quincena, la empresa **MilkStgo** realiza el pago a sus diferentes proveedores según la cantidad y calidad de leche que enviaron durante la quincena.

Actualmente todo el proceso de registro de acopio de leche y el pago respectivo se realiza en forma manual usando hojas Excel. Todo este proceso manual ha venido funcionando sin mayores problemas por mucho tiempo. Sin embargo, en los últimos meses se han venido registrando reiteradas quejas de parte de los proveedores indicando que los montos de sus pagos no son correctos.

Para enfrentar este problema, la empresa **MilkStgo** ha tomado la decisión de desarrollar una aplicación web que le permita automatizar todo este proceso de registro y pago de leche. El propósito de esta decisión es evitar los errores que se vienen cometiendo en el actual proceso manual de acopio y pago de leche y de esta manera reducir al mínimo las quejas de los proveedores.

### 4.2 Información respecto del acopio de leche y su respectivo pago

La empresa tiene un total de 120 proveedores de leche. Los datos de estos proveedores tienen la siguiente estructura.

- Código (numérico de 5 dígitos. Dos primeros dígitos indican la región del país donde está ubicado el proveedor)
- Nombre Proveedor
- Categoría (A, B, C, D)
- Afecto a Retención (Sí, No)

Al final de cada quincena, cada centro de acopio debe enviar a la oficina central de pagos (ubicada en Santiago) un archivo Excel con todo el acopio de leche de la quincena. Cada día se realizan dos acopios de leche, uno en la mañana y otro en la tarde. La información del acopio de leche de la quincena se registra en archivos Excel que tienen la siguiente estructura.

	A	B	C	D
1	Fecha	Turno	Proveedor	KLS Leche
2	2023/03/17	M	01003	50
3	2023/03/17	T	01003	45
4	2023/03/18	M	01011	30
5	2023/03/18	M	01025	35
6	2023/03/18	T	01078	25
7				
8				

El cálculo del pago a los proveedores (por la leche que enviaron) se realiza en función de la cantidad de leche enviada (Kls enviados), el porcentaje de grasa de la leche y el porcentaje de sólidos totales (ST) de la leche. Las siguientes tablas muestran los valores que se deben considerar para el pago de estos parámetros.

Categoría del proveedor	Pago x Kilo leche (en pesos)
A	700
B	550
C	400
D	250

% Grasa	Pago x Kilo leche (en pesos)
0 - 20%	30
21% - 45%	80
46% - más	120

% Sólidos Totales (ST)	Pago x Kilo leche (en pesos)
0 - 7%	-130
8% - 18%	-90
19% - 35%	95
36% - más	150

Los valores del porcentaje de Grasa y Sólidos Totales (ST) es enviado desde cada uno de los centros de acopio. El laboratorio de cada centro de acopio es quien calcula estos valores. El personal de laboratorio analiza diariamente estos dos parámetros para el acopio de leche de cada proveedor. Al final de la quincena se envía un archivo Excel con estos dos valores.

	A	B	C
1	Proveedor	% Grasa	% Solido Total
2	01003	25	14
3	01011	8	11
4	01025	30	50
5	01078	40	27
6			
7			

La empresa aplica una bonificación especial (frecuencia) a aquellos proveedores que hayan enviado leche con más frecuencia en la quincena.

Más de 10 días en la quincena	Bonificación x Frecuencia
Mañana y Tarde	20% del total pago por KLS Leche
Solo mañana	12% del total pago por KLS Leche
Solo tarde	8% del total pago por KLS Leche

La empresa aplica un descuento cuando hay un porcentaje de variación negativa de KLS de leche, %Grasa y %ST respecto de la quincena anterior. Las siguientes tablas resumen esos porcentajes:

% Variación negativa KLS Leche	% Descuento sobre el Pago Acopio Leche
0 – 8%	0%
9% - 25%	7%
26% - 45%	15%
46% - más	30%

% Variación negativa Grasa	% Descuento sobre el Pago Acopio Leche
0 – 15%	0%
16% - 25%	12%
26% - 40%	20%
41% - más	30%

% Variación negativa ST	% Descuento sobre el Pago Acopio Leche
0 – 6%	0%
7% - 12%	18%
13% - 35%	27%
36% - más	45%

Según las leyes del gobierno se debe aplicar un impuesto (retención) de 13% para todos aquellos montos de pago mayores a 950.000 pesos. Por lo tanto, aquellos proveedores que estén afectos a retención se les debe aplicar este impuesto según corresponda sobre el monto del *Pago TOTAL*.

Finalmente, el cálculo del pago final por la leche se obtiene según las siguientes operaciones:

- **Pago Acopio Leche** = Pago por leche + Pago grasa + Pago por ST + Bonificación por frecuencia
- **Descuentos** = Dcto. por Variación Leche + Dcto. por Variación Grasa + Dcto. por Variación ST
- **Pago TOTAL** = Pago Acopio Leche – Descuentos
- **Pago FINAL** = Pago TOTAL - Retención

### 4.3 Historias de Usuario

Las Historias de Usuario que se deben implementar son las siguientes:

- HU1: Ingreso de datos de los proveedores desde pantalla.
- HU2: Listar datos de los proveedores.
- HU3: Importar acopio de leche desde archivo Excel.
- HU4: Importar valores de Grasa y Solidos Totales desde archivo Excel.

- HU5: Calcular planilla de pagos de leche.
- HU6: Mostrar reporte por pantalla de la planilla de pagos con los siguientes datos:
  - Quincena (AAAA/MM/Q)
  - Código proveedor
  - Nombre proveedor
  - TOTAL KLS leche
  - Nro. días que envió leche
  - Promedio diario KLS leche
  - %Variación Leche
  - %Grasa
  - %Variación Grasa
  - %Solidos Totales
  - %Variación ST
  - Pago por Leche
  - Pago por Grasa
  - Pago por Solidos Totales
  - Bonificación por Frecuencia
  - Dcto. Variación Leche
  - Dcto. Variación Grasa
  - Dcto. Variación ST
  - Pago TOTAL
  - Monto Retención
  - Monto FINAL

## 5. Aspectos del desarrollo del producto

### 5.1 Gestión del proyecto

- El proyecto debe ser gestionado usando los artefactos y ceremonias de *SCRUM*.
- La gestión del proyecto debe realizarse usando la herramienta *JIRA*.

### 5.2 Restricciones de desarrollo

- La aplicación de software debe ser desarrollada usando *Spring Boot* (dependencias a usar: Spring Web, MySQL Driver, Spring Data JPA, Thymeleaf, Lombok).
- Debe ser implementada en el lenguaje de programación Java usando programación orientada a objetos.
- Debe ser una aplicación web monolítica basada en el patrón arquitectural por capas (layers).
- Debe usar una base de datos relacional (Por ejemplo, *MySQL*).
- La aplicación debe ser desarrollada en *IntelliJ*.

### 5.3 Despliegue del producto en producción

- La aplicación web debe quedar desplegada y totalmente funcionando en un servidor de la nube (*AWS, Digital Ocean, etc.*).
- Se debe usar un script de *Terraform* para provisionar el servidor en la nube. El servidor debe tener Linux como sistema operativo.
- El despliegue de la aplicación en el servidor se debe realizar usando *Docker Compose*. Los componentes para desplegar son: Base de Datos (*MySQL*), Aplicación Web (3 réplicas), y *Nginx* como balanceador de carga (load balancer). Todos estos componentes deben ser desplegados desde sus imágenes Docker respectivas.
- La aplicación web debe poder ser accedida desde cualquier navegador web local.

### 5.4 Desarrollo y testing del producto

- El proyecto debe tener un repositorio en *GitHub* para el código fuente.

- La gestión del versionamiento del código fuente de la aplicación web debe ser realizado usando *Git*.
- Se debe configurar y usar un pipeline de entrega continua (en *Jenkins*) para automatizar el proceso de desarrollo de la aplicación web. El pipeline debe automatizar las siguientes tareas:
  - o Obtener automáticamente el código fuente desde *Github*.
  - o Ejecutar las pruebas unitarias (*Junit*) asociadas al proyecto.
  - o Realizar análisis estático del código fuente con la herramienta *Sonarqube*.
  - o Encapsular la aplicación en un contenedor *Docker* y subirla a *DockerHub*.
- La aplicación debe tener una cobertura de testing mayor o igual a 90%.
- El análisis estático de código fuente de *Sonarqube* debe mostrar resultados aceptables para Confiabilidad (*Reliability*), Mantenibilidad (*Maintainability*) y Cobertura (*Coverage*).