

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 academia preuniversitaria **TopEducation** es un preuniversitario ubicado en la ciudad de Santiago. Tiene más de 15 años preparando jóvenes con una educación de calidad para que tengan rendimientos académicos óptimos de tal manera que puedan lograr rendir satisfactoriamente las pruebas de admisión a las universidades del país.

Los estudiantes que ingresan a este preuniversitario deben pagar una matrícula y un arancel de estudios. El pago del arancel se puede realizar al contado o en cuotas pactadas entre el estudiante y el preuniversitario.

Actualmente, toda la gestión de pagos de aranceles ya sea al contado o en cuotas, se realiza en forma manual usando hojas Excel como apoyo. Todo este proceso manual ha venido funcionando sin mayores problemas. Sin embargo, en los últimos dos años se han venido registrando reiteradas quejas de parte de los estudiantes indicando que el estado de sus pagos no es correcto.

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

4.2 Información respecto al proceso de pagos de aranceles

El preuniversitario tiene un total de 350 estudiantes de los cuales se tienen registrados los siguientes datos:

- RUT
- Apellidos
- Nombres
- Fecha Nacimiento
- Tipo Colegio Procedencia (Municipal/Subvencionado/Privado)
- Nombre Colegio
- Año egreso del colegio

Cuando un estudiante decide estudiar en **TopEducation**, debe pagar la matrícula y el arancel de estudio. El monto de la matrícula tiene un valor de \$ 70,000. El arancel de estudio tiene un valor de \$ 1.500,000. Para el caso de pagos al contado, se aplica un descuento directo del 50% (respecto del arancel total). En el caso de cuotas, se aplican los siguientes descuentos acumulables según corresponda. Los pagos al contado o acuerdo de cuotas se realizan 5 días antes del inicio de clases.

Tipo de Colegio de procedencia	% Descuento respecto del arancel total
Municipal	20%
Subvencionado	10%
Privado	0%

Años desde que egresó del colegio	% Descuento respecto del arancel total
< 1 años	15%
1 – 2 años	8%
3 – 4 años	4%
5 – más	0%

Para el caso de los pagos en cuotas, el número máximo de cuotas se define según la siguiente tabla. Las cuotas se pagan entre el 05 y 10 de cada mes. Por el momento los pagos no se pueden realizar via web. Solamente se realizan pagos en forma presencial en las oficinas del preuniversitario.

Tipo de Colegio de procedencia	# Máx. de cuotas
Municipal	10
Subvencionado	7
Privado	4

Como parte de la formación del preuniversitario, los estudiantes deben rendir pruebas que simulan un examen de admisión. Estas pruebas se rinden el último viernes de cada mes. Según el promedio de estas pruebas los estudiantes son beneficiados con descuentos en el arancel del mes (ver tabla). Para el caso de los estudiantes que pagaron al contado, al término de los estudios se les reembolsará según el promedio final que hayan obtenido.

Puntaje promedio pruebas	% Dcto. (se aplica a todas las cuotas pendientes de pago)
950 – 1000	10%
900 – 949	5%
850 – 899	2%
< 850	0%

Los puntajes de cada prueba vienen almacenados en una hoja Excel donde se tiene registrados todos los estudiantes que rindieron el examen. En dicha hoja Excel se tienen almacenados los siguientes datos: RUT del estudiante, fecha del examen y el puntaje obtenido. Se tiene una hoja Excel por cada examen. Esta hoja Excel está disponible el lunes inmediato después del viernes en que se rindió la prueba.

En el caso de que las cuotas no se paguen, se procederá a aplicar un interés sobre cada una las siguientes cuotas pendientes de pago según lo especificado a continuación:

# meses de atraso	% interés (se aplica a todas las cuotas pendientes de pago)
0	0%
1	3%
2	6%
3	9%
> 3	15%

4.3 Historias de Usuario

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

- HU1: Ingreso de datos de los estudiantes desde pantalla.
- HU2: Generar cuotas de pago.
- HU3: Listar cuotas de pago de un estudiante y el estado de pago de cada cuota.
- HU4: Registrar pagos de cuotas de arancel.
- HU5: Importar notas de exámenes desde archivo Excel.
- HU6: Calcular planilla de pagos de arancel
- HU7: Calcular reporte resumen de estado de pagos de los estudiantes.
 - RUT estudiante
 - Nombre del estudiante
 - Nro. exámenes rendidos
 - Promedio puntaje exámenes
 - Monto total arancel a pagar
 - Tipo Pago (Contado/Cuotas)
 - Nro. total de cuotas pactadas
 - Nro. cuotas pagadas
 - Monto total pagado
 - Fecha último pago
 - Saldo por pagar
 - Nro. Cuotas con retraso

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/PostgreSQL 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*, *PostgreSQL*, etc.).
- La aplicación debe ser desarrollada usando un IDE (por ejemplo, *IntelliJ*, *VS Code*, etc.).

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/PostgreSQL), 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*).