

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**  
**DISEÑO DE SOFTWARE**  
**PROYECTO SEGUNDO PARCIAL - II TÉRMINO 2019**  
**FECHA Y HORA DE ENTREGA: 23H59 ENERO 23, 2019**  
**DR. CARLOS MERA GÓMEZ**

---

**Objetivos Específicos del Proyecto**

1. Diseñar un producto de software en el que se apliquen principios de diseño, para que sea robusto, fácil de mantener y modificar
2. Construir modelos apropiados que satisfagan las especificaciones de requerimientos, para la creación de un producto de software

**Resultados de Aprendizaje**

1. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa
2. Habilidad para aplicar teoría de ciencias computacionales y fundamentos de desarrollo de software para producir soluciones basadas en computación.

**Nombre del Proyecto**

Veterinaria Dr. Pelos

**Descripción**

La empresa Dr. Pelos brinda servicios (atención veterinaria, peluquería etc.) y productos veterinarios (medicinas, juguetes, ropas, alimentos, etc.) en algunos puntos de la ciudad a través de una agencia principal y sucursales. Dr. Pelos importa productos veterinarios para la venta en sus locales al por menor y al por mayor con precios diferenciados. Usted y su equipo de desarrollo deben diseñar el sistema con diagramas en UML y modelos entidad-relación (ER).

Se desea que realice el diseño completo de la aplicación (todos los requerimientos), pero únicamente debe implementar lo que está señalado con **[I]**. Los requerimientos que no son necesarios de implementar se deben mostrar como “Funcionalidad en construcción”.

**Especificaciones**

Realice el diseño de una aplicación utilizando la arquitectura MVC y los paradigmas de programación que considere necesarios para el manejo de inventario, compras y ventas de productos y servicios. Utilice patrones de diseño y cree paquetes, según el nombre de cada patrón utilizado, para organizar las clases correspondientes a dicho patrón. El sistema debe cumplir con las siguientes características:

1. **[I]** Se debe trabajar con 2 contenedores (Docker), uno para la base de datos y otro para ejecutar el prototipo del sistema diseñado. De forma que solo se trabajará con una base de datos, pero 2 contenedores independientes. Adicionalmente, se pide que se demuestre la calidad del código fuente con un reporte de SonarQube.
2. **[I]** Las sucursales trabajarán contra una base de datos central; pero en caso de pérdida de conectividad con la base de datos central, la venta de productos al por menor estará restringida al stock existente en la agencia/sucursal. Una base de datos local contendrá el stock disponible en la agencia/sucursal.
3. En caso pérdida de conectividad, las demás transacciones se almacenarán localmente en un archivo para poder replicar la información cuando la conectividad se restablezca.
4. **[I]** En algunas sucursales no se brindará servicios veterinarios, solo se venderá productos.
5. **[I]** Se brindará servicio a domicilio de traslado de mascotas pequeñas y entrega de productos veterinarios.
6. La definición de precios de productos es manejada por un jefe de bodega en la agencia principal. La entrega de productos se hará por medio de repartidores que utilizan únicamente documentación impresa y que concluida su entrega regresan al jefe de bodega con la correspondiente conformidad del usuario en la recepción.

7. Los directivos de la empresa deben poder consultar toda la información que maneja el sistema. Debe también existir un administrador con todos los permisos CRUD (Create, Read, Update and Delete), excepto eliminar (manejar eliminado lógico).
8. **[I]** El personal de caja debe poder agregar clientes, realizar cotizaciones, ventas de productos y servicios, consultar entregas de productos y traslados de mascotas.
9. **[I]** El sistema debe permitir la búsqueda de productos y servicios veterinarios por nombre, categoría o descripción. Esta búsqueda debe estar disponible al momento de facturar o cotizar.
10. **[I]** El sistema debe almacenar la información personal y credenciales de los usuarios que acceden al sistema.
11. El sistema debe aceptar formas de pago en efectivo o tarjeta de crédito VISA. Sin embargo, debe ser extensible a otras tarjetas de crédito y formas de pago.
12. **[I]** El sistema debe emitir facturas, notas de crédito y cotizaciones en forma electrónica. Estos documentos poseerán un identificador único, con secuencias independientes para cada caso. El sistema debe, además, registrar información relacionada tal como fecha de emisión, usuario emisor.
13. **[I]** El inventario de Dr. Pelos debe ser manejado por local. Los stocks son actualizados por el administrador de la sucursal/agencia. Desde cada sucursal se puede solicitar el suministro de un producto en particular a la agencia principal. Los despachos toman menos de 24 horas laborables y los stocks se actualizan según corresponda por medio de una importación de datos realizada por el administrador de sucursal/agencia.
14. **[I]** Crear un conjunto de datos de pruebas (en la BD) para probar todas las funcionalidades del sistema.

**Nota:** Puede agregar los datos que considere pertinentes para este tipo de sistemas.

#### Entregables (en un archivo ZIP o RAR)

1. Un **documento (en formato docx)** que contenga las imágenes correspondientes a cada diagrama solicitado. Además, el documento debe listar los integrantes del equipo en la carátula y nombrar la herramienta utilizada para modelar los diagramas (por ejemplo, Visual Paradigm). Cada diagrama debe ser legible en una página y la imagen debe tener buena resolución. El documento debe contar, al menos, con lo siguiente:
  - 1.1. Carátula, descripción de su propuesta.
  - 1.2. Diagrama de casos de uso con su documentación.
    - 1.2.1. Identificación de casos de uso correctos
    - 1.2.2. Identificación de actores correctos
    - 1.2.3. Uso correcto de estereotipos
    - 1.2.4. Uso correcto de notación y coherencia del contenido del diagrama
  - 1.3. Diagrama de clases.
    - 1.3.1. Uso de paquetes y patrones
    - 1.3.2. Relaciones coherentes entre clases
  - 1.4. Diagramas de secuencia (Cinco).
    - 1.4.1. Uso correcto y coherente de mensajes síncronos, asíncronos y retorno
    - 1.4.2. Identificación de objetos
    - 1.4.3. Coherencia del diseño y uso correcto de la notación
  - 1.5. Diagrama entidad-relación del modelo lógico de la base de datos.
2. Diagramas UML y ER con alta resolución en formato PNG y PDF. Los archivos deberán estar en una carpeta llamada **diagramas**, en donde cada archivo tendrá la siguiente identificación: <secuencia de dos dígitos><nombre del archivo><tipo de archivo>.<extensión>. Por ejemplo: 01diagramaxyz-pdf.pdf
3. Implementación del sistema en repositorio Git
  - 3.1. Uso de Docker separando el sistema de la base de datos.
  - 3.2. Datos de prueba en base de datos.
  - 3.3. Reporte de SonarQube.
  - 3.4. Pruebas unitarias (Diez).
4. Cada estudiante deberá evaluar a los miembros de su equipo de trabajo usando la rúbrica que será puesta a disposición como tarea individual en SidWeb.

## Rúbrica de Calificación

Descripción	Valor
<b>Documentación</b>	
Diagrama de casos de uso con su documentación	05
Diagrama de clases	20
Diagramas de secuencia	10
Diagrama entidad-relación del modelo lógico de la base de datos	05
<b>Implementación</b>	
Prototipo del sistema en repositorio Git	30
Uso de Docker en prototipo completo	10
Reporte SonarQube en prototipo completo	10
Pruebas unitarias en prototipo completo*	10
<b>Total</b>	<b>100</b>
Penalidad por cada día o fracción de día	-50
No subir a SidWeb los entregables requeridos con sus respectivas especificaciones	-50
No subir la rúbrica de coevaluación en tarea individual (afectación individual)	-20
Cada error de notación UML	-02