

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA**

LENGUAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS

Examen 1

(Segundo semestre 2020)

Indicaciones generales:

- Duración: 3 horas.
- Materiales o equipos para utilizar: Material del PAIDEIA y documentación oficial de Microsoft.
- No está permitido el uso de ningún material desarrollado por otra persona durante el examen, de detectarse que la otra persona es alumno del horario, todos los implicados tendrán la nota inapelable de cero.

Puntaje total: 20 puntos

Sistema de gestión de ventas de una empresa comercializadora

La empresa LPOO se dedica al negocio de la comercialización de productos, iniciando sus operaciones con la comercialización de solo dos tipos de productos: libros y equipos electrónicos. Posee varias tiendas físicas en diferentes lugares del país. LPOO desea administrar las ventas de su negocio mediante un software y por ello contrata a la empresa "06M1 S.A." para que desarrolle esta aplicación.

Como parte de la fase de análisis del sistema, se han obtenido los siguientes requisitos de software:

- El sistema debe permitir administrar todos los productos, clientes, vendedores, gerentes, tiendas, departamentos, asignar productos a una tienda y soportar la operación de la venta de sus productos.
- Un usuario es aquel que tendrá acceso al sistema, por lo que todo usuario tiene un id, nombre de usuario, contraseña, nombre (s), apellido (s) y género. Hay dos tipos de usuarios, los empleados y los clientes.
- El sistema debe permitir gestionar los datos de los empleados. Los empleados pueden ser gerentes o vendedores. Todo empleado tiene, además de los datos de usuario, un salario y un estado. Los empleados de tipo gerente tiene un número de empleados bajo su dirección. Los empleados de tipo vendedor tienen una meta de ventas que cumplir (cuota de ventas) y el departamento de una tienda en el que trabajan.
- El sistema debe permitir gestionar los datos de los productos. Los productos pueden ser de dos tipos, libros o productos electrónicos. Todos los productos tienen atributos comunes como un código o id, nombre, descripción, precio, estado, stock total (suma del stock en todas las tiendas), foto y una categoría. Adicionalmente, los productos que son libros tienen atributos como autor, título del libro, edición, editorial y el año de publicación; y los productos que son electrónicos tienen atributos como marca, familia, modelo y meses de garantía. Por un tema tributario, los productos electrónicos están obligados a pagar impuesto, mientras que los libros, no.
- El sistema debe permitir gestionar los datos de las tiendas. Las tiendas tienen atributos como código o id, nombre, dirección y código postal.
- El sistema debe permitir gestionar los datos de los departamentos que pueden existir en cada una de sus tiendas. Los nombres de los departamentos coinciden con los nombres de las categorías de los productos. Un departamento siempre tiene un id, nombre y un porcentaje de comisión que se le da al vendedor por cada venta del producto de dicho departamento. Algunos ejemplos de departamentos sobre electrónica en las tiendas son TV y audio, cómputo y tecnología, línea blanca, electrodomésticos, videojuegos, cuidado personal, relojes, salud y combos. Algunos ejemplos de departamentos sobre libros en las tiendas son infantil, juvenil, cómics, literatura, biografía y crítica literaria, cultura y sociedad, naturaleza y tiempo libre, conocimiento, libros de bolsillo, libros en otros idiomas y especiales.

- En cada tienda puede haber varios departamentos, en el que cada departamento tiene un área física en metros cuadrados y un promotor del departamento, que siempre es un vendedor destacado en dicho departamento.
- El sistema debe permitir gestionar los datos de los clientes. Los clientes pueden ser de dos tipos, personas naturales o personas jurídicas (empresas). Todos los clientes tienen atributos comunes como un código o id, dirección, teléfono, correo electrónico y un estado. Adicionalmente, los clientes que son personas naturales tienen atributos como DNI y fecha de nacimiento; y los clientes que son personas jurídicas (empresas) tienen atributos como razón social, RUC, representante legal y página web. Como el sistema se está proyectando para que en un futuro pueda permitir las ventas por Internet, se debe considerar que los clientes son también usuarios del sistema.
- El sistema debe permitir gestionar los productos por tienda, en el que habría un stock por tienda, un valor mínimo de stock que representa la cantidad mínima de stock que debe haber en esa tienda, de estar por debajo del mínimo el sistema reportaría una alerta para posteriormente solicitar una renovación de stock, y también habría un descuento que representa el porcentaje de descuento sobre el precio base que tendría el producto en esa tienda.
- El sistema debe permitir registrar las ventas. Una venta es realizada por un vendedor en una fecha a un solo cliente, la fecha no puede ser anterior a la fecha actual. Si la venta es OK tiene el estado de Activo (A) y si es anulada tiene el estado (N). En la información de la venta debe estar también el total de toda la venta.
- En una venta se pueden vender varias unidades de una misma familia de productos disponibles en una tienda, es por ello que se puede tener un subtotal por cada ítem de producto. La venta puede tener varios ítems de diferentes productos. Considere el comprobante de pago de la Figura 1 como un ejemplo de algunos de los datos que se trabajan en una venta.

Empresa Comercializadora

LPOO

Tienda San Miguel, Av. Universitaria 1801, San Miguel
Vendedor: John Baldomero

RUC: 69696969696
BOLETA DE PAGO
000-00000001

Cliente: **Freddy Krugger**
DNI: **66666666**

Fecha de emisión: **2020/05/30**

Cantidad	Descripción	Precio Unitario	Valor de Venta
2	La Pequeña Princesa	65.00	130.00
3	La Magia del Orden	50.00	150.00

CANCELADO

Total: **280.00**

Desarrollado por LPOO 06M1 S.A.

Figura 1: Ejemplo de Comprobante de Pago

El diagrama de clases que se muestra en la Figura 2 modela el caso descrito que permite soportar todo el desarrollo de la aplicación de ventas

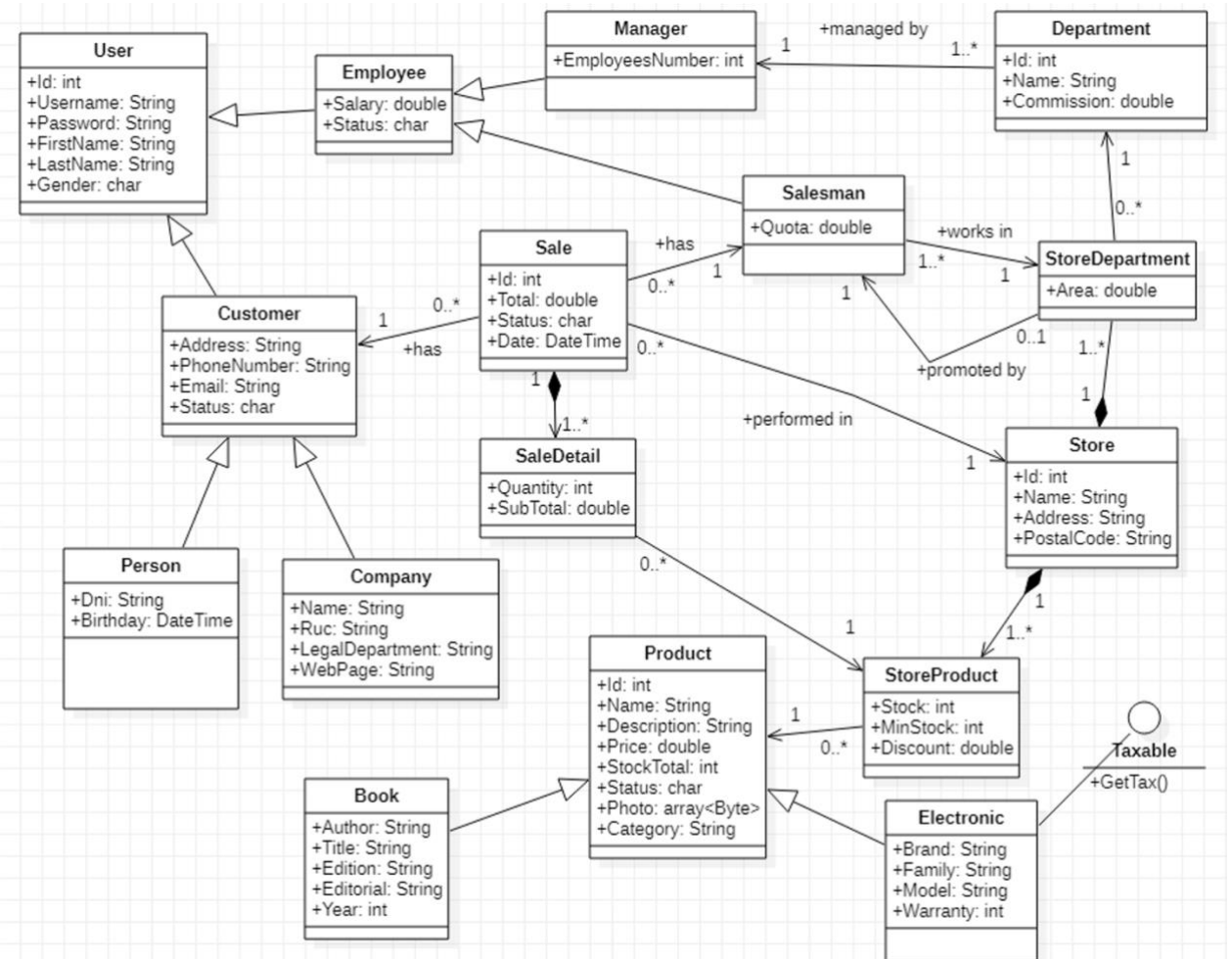


Figura 2: Diagrama de Clases del Sistema de Ventas

Para esta etapa de desarrollo, se pide implementar todo el mantenimiento de Gerentes utilizando la siguiente GUI:

Sistema de Ventas

Archivo Operaciones Tendencias Mantenimiento Ayuda

Gerentes

Id:

Usuario:

Password:

Nombre (s):

Apellido (s):

Género: ☒ Masculino ☐ Femenino

Número de empleados:

Salario (S/.):

Agregar Modificar Eliminar

	Id	Nombre (s)	Apellido (s)	Usuario
*				

El mantenimiento de Gerentes debe funcionar de la siguiente manera:

1. En la ventana principal, al seleccionar la opción de Mantenimiento / Gerentes, se debe mostrar la ventana interna *ManagerForm* cuyo título es "Gerentes".
2. Al mostrar la ventana interna *ManagerForm*, debe mostrarse en el grid los gerentes que ya se encuentren registrados en el archivo de texto en formato XML **managers.xml** o en el archivo binario **managers.bin**. De no existir el archivo o estar vacío, no mostraría ningún gerente. El grid solo muestra los campos Id, nombre (s), apellido (s) y el usuario del gerente.
3. En la ventana interna *ManagerForm*, cuando se seleccione una fila del grid, lo que hace es **Buscar** el gerente por el código o id del gerente seleccionado, para luego mostrar todos los datos en el formulario de dicho gerente. Debe validar que solo se seleccione una fila que corresponde a un gerente.
4. En la ventana interna *ManagerForm*, al presionar la opción **Nuevo** del menú, debe limpiar los valores de los campos del formulario y sólo habilitar el botón de *Agregar* que permita ingresar un nuevo gerente. Los demás botones estarán deshabilitados y debe validar los datos para que el registro del nuevo gerente sea correcto.
5. En la ventana interna *ManagerForm*, el usuario debe ingresar todos los datos referentes al nuevo gerente y presionar el botón *Agregar*. Luego de registrar un nuevo gerente, se debe limpiar los valores del formulario y se debe visualizar en el grid una nueva fila con los datos del nuevo gerente. **Al presionar el botón Agregar debe validar que cuando se ingrese un nuevo gerente no se repita el código y el usuario del gerente. En caso de que esto suceda no debe permitir el registro y debe mostrar un mensaje de error.** Recuerde que los datos del nuevo gerente deben estar registrados en el archivo **managers.xml** o **managers.bin**.
6. En la ventana interna *ManagerForm*, si desea modificar un gerente, se debe seleccionar la opción **Editar** del menú de la ventana, se habilitan los campos del formulario, los botones de *Actualizar* y *Eliminar*, y se deshabilita el botón *Agregar*.

7. En la ventana interna *ManagerForm*, luego que el usuario seleccione un gerente del grid, se visualizan sus datos en el formulario; a partir de ahí, el usuario puede modificar cualquier dato del gerente, a excepción del código del gerente y luego presionar el botón *Actualizar*. Al presionar *Actualizar*, se debe guardar la modificación realizada, se ve reflejado en el grid de Gerentes y también se actualiza el archivo *managers.xml* o *managers.bin*.
8. En la ventana interna *ManagerForm*, si se desea eliminar un gerente, se debe seleccionar la fila del gerente del grid y presionar el botón *Eliminar*. Al realizarse esta acción, el grid debe actualizarse y eliminar el gerente seleccionado. El cambio también debe verse reflejado en el archivo *managers.xml* o *managers.bin*.

Adicionalmente, se está solicitando el mantenimiento de los departamentos. Para ello se tendrán las siguientes pantallas:

The screenshot shows a window titled 'Sistema de Ventas' with a menu bar containing 'Archivo', 'Operaciones', 'Tendencias', 'Mantenimiento', and 'Ayuda'. Inside, there is a sub-window titled 'Departamentos'. This sub-window contains a form with the following fields: 'Id' (text input), 'Nombre' (text input), 'Gerente' (dropdown menu), and 'Comisión (%)' (text input). Below these fields are three buttons: 'Agregar', 'Modificar', and 'Eliminar'. At the bottom of the sub-window is a table with two columns, 'Id' and 'Nombre'. The table contains one row with an asterisk (*) in the 'Id' column and an empty 'Nombre' column.

La funcionalidad de mantenimiento de departamentos debe funcionar de la siguiente manera:

1. En la ventana principal, al seleccionar la opción de *Mantenimiento / Departamentos*, se debe mostrar la ventana interna *DepartmentForm* con el título "Departamentos".
2. En la ventana interna *DepartmentForm* debe mostrarse en el grid los departamentos que ya se encuentren registrados en el archivo de texto en formato XML *departments.xml* o el archivo binario *departments.bin*, considerando el gerente que tiene asignado. De no existir el archivo o estar vacío, no mostraría ningún departamento. El grid solo muestra los campos Id y nombre del departamento.
3. En la ventana interna *DepartmentForm*, cuando se seleccione una fila del grid, lo que hace es **Buscar** el departamento por el id del departamento seleccionado, para luego mostrar todos los datos en el formulario de dicho departamento. Debe validar que solo se seleccione una fila que corresponde a un departamento.
4. En la ventana interna *DepartmentForm*, al presionar la opción **Nuevo** del menú, debe limpiar los valores de los campos del formulario y sólo habilitar el botón de *Agregar* que permita ingresar un nuevo departamento. Los demás botones estarán deshabilitados y debe validar los datos para que el registro del nuevo departamento sea correcto.
5. En la ventana interna *DepartmentForm*, el usuario debe ingresar todos los datos referentes al nuevo departamento y presionar el botón *Agregar*. Para asignar un gerente, debe llenar el combo de Gerentes para que se pueda elegir un gerente que se asignará a dicho

departamento. Luego de registrar un nuevo gerente, se debe limpiar los valores del formulario y se debe visualizar en el grid una nueva fila con los datos del nuevo departamento. **Al presionar el botón Agregar debe validar que cuando se ingrese un nuevo departamento no se repita el id del departamento. En caso de que esto suceda no debe permitir el registro y debe mostrar un mensaje de error.** Recuerde que los datos del nuevo departamento deben estar registrados en el archivo **departments.xml** o **departments.bin**.

De acuerdo con todo lo descrito anteriormente, se le pide en C++/CLI:

Pregunta 1: Sobre Mantenimiento de Gerentes (9 puntos)

- a. (0.5 ptos.) Implemente los cambios de ser necesarios en el proyecto SalesModel que permita soportar lo descrito en esta fase para mantenimiento de gerentes.
- b. (3.0 ptos.) Implemente los cambios necesarios en el proyecto SalesController que permita soportar lo descrito en esta fase para el mantenimiento de gerentes.
- c. (5.5 ptos.) Implemente los cambios necesarios en el proyecto SalesView que permita soportar lo descrito en esta fase.
 - (1.0 pto.) Implementación del mostrar los gerentes en el grid de ManagerForm.
 - (1.5 pto.) Implementación del registro de un gerente utilizando el ManagerForm.
 - (2.5 ptos.) Implementación del visualizar y actualizar un gerente usando el ManagerForm.
 - (0.5 ptos.) Implementación del eliminar un gerente usando el ManagerForm.

Pregunta 2: Sobre Mantenimiento de Departamentos (11 puntos)

- a. (4.5 ptos.) Implemente los cambios necesarios en el proyecto SalesController que permita soportar lo descrito en esta fase para la funcionalidad de mantenimiento de departamentos.
- b. (6.5 ptos.) Implemente los cambios necesarios en el proyecto SalesView que permita soportar lo descrito en esta fase para la funcionalidad de mantenimiento de departamentos.
 - (1.5 pto.) Implementación del presentar los gerentes en el combo de DepartmentForm.
 - (1.5 ptos.) Implementación del mostrar los departamentos en el grid de DepartmentForm.
 - (3.5 ptos.) Implementación del registro de un departamento utilizando el DepartmentForm.

Nota. - Recuerde que debe seguir el modelo MVC y definir, como propiedades, los atributos de las clases que pertenecen al modelo (Model). Los proyectos deben compilar correctamente y **por ningún motivo se corregirá código colocado como comentario.** Adicional a ello, la estructura de los archivos no puede ser modificada.

Profesor del curso: Johan Baldeón.

San Miguel, 26 de octubre de 2020