PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS
Examen 1
(Primer 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 FNAC se dedica al negocio de la comercialización de productos, y ha abierto una sede en Perú, 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. FNAC desea administrar las ventas de su negocio mediante un software y por ello contrata a la empresa "LPOO 06M2 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 todas sus tiendas, vendedores, clientes, productos, productos por tienda, y por supuesto, soportar las operaciones de la venta en sí de sus productos.
- El sistema debe permitir administrar los datos de las tiendas. Las tiendas tienen atributos como código o id, nombre, dirección y código postal.
- El sistema debe permitir administrar los datos de sus vendedores. Los vendedores tienen atributos como código o id, nombre, apellido, género, salario, meta de ventas en soles por mes y la tienda a la que pertenece.
- El sistema debe permitir administrar 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, teléfono y dirección. Adicionalmente, los clientes que son personas naturales tienen atributos como nombre, apellido paterno, DNI, correo electrónico y fecha de nacimiento; y los clientes que son personas jurídicas (empresas) tienen atributos como razón social, RUC, representante legal y página web.
- El sistema debe permitir administrar 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 y stock total (suma del stock en todas las tiendas). Adicionalmente, los productos que son libros tienen atributos como autor, título del libro, edición y editorial; 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 administrar 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

FNAC

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

Total: 280.00

CANCELADO

Desarrollado por LPOO 06M2 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

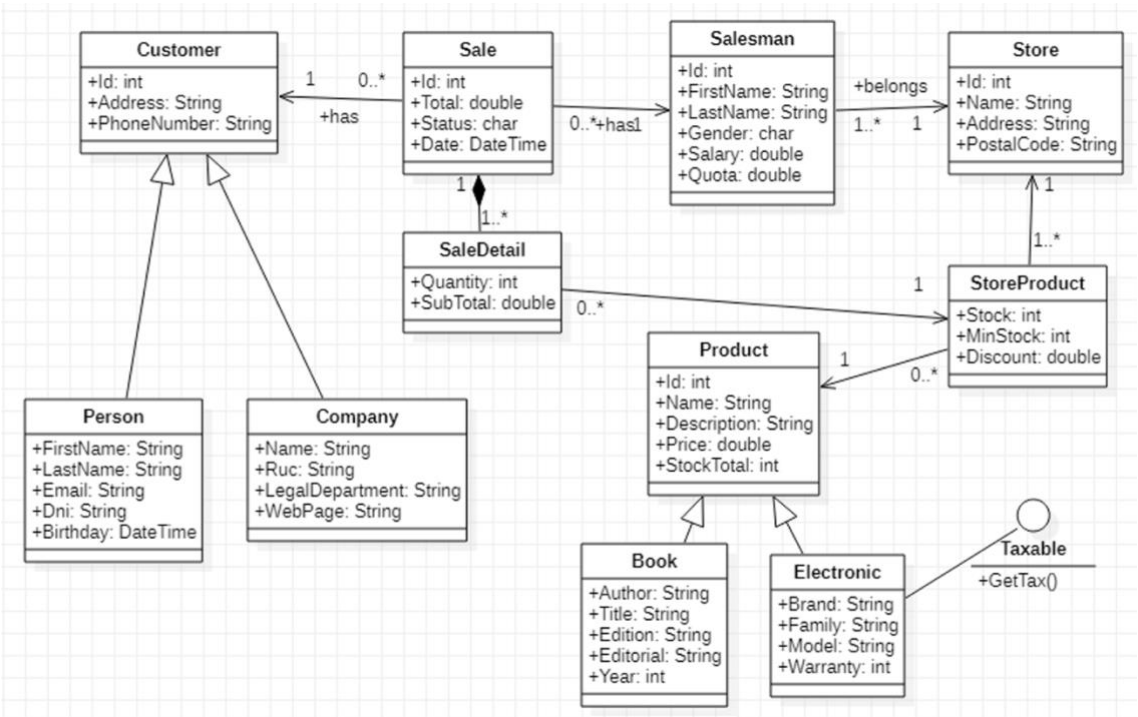
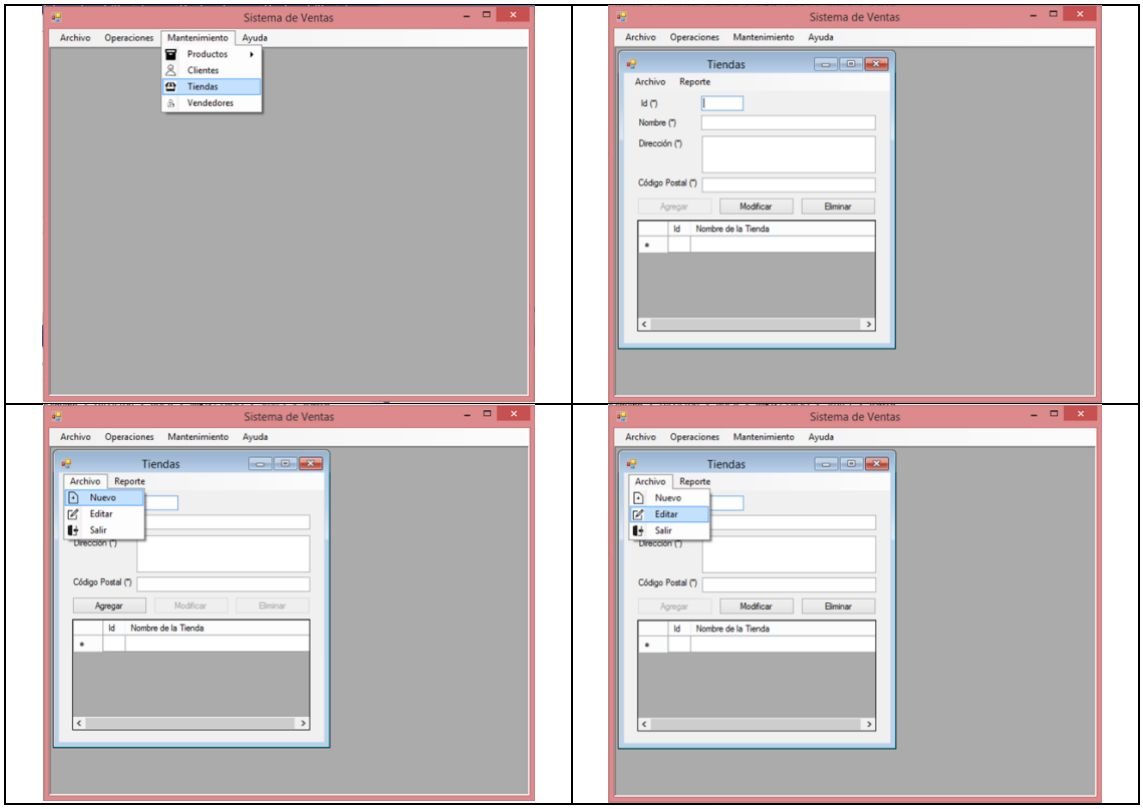


Figura 2: Diagrama de Clases del Sistema de Ventas

Para esta etapa de desarrollo, se pide implementar todo el mantenimiento de tiendas utilizando las siguientes pantallas:

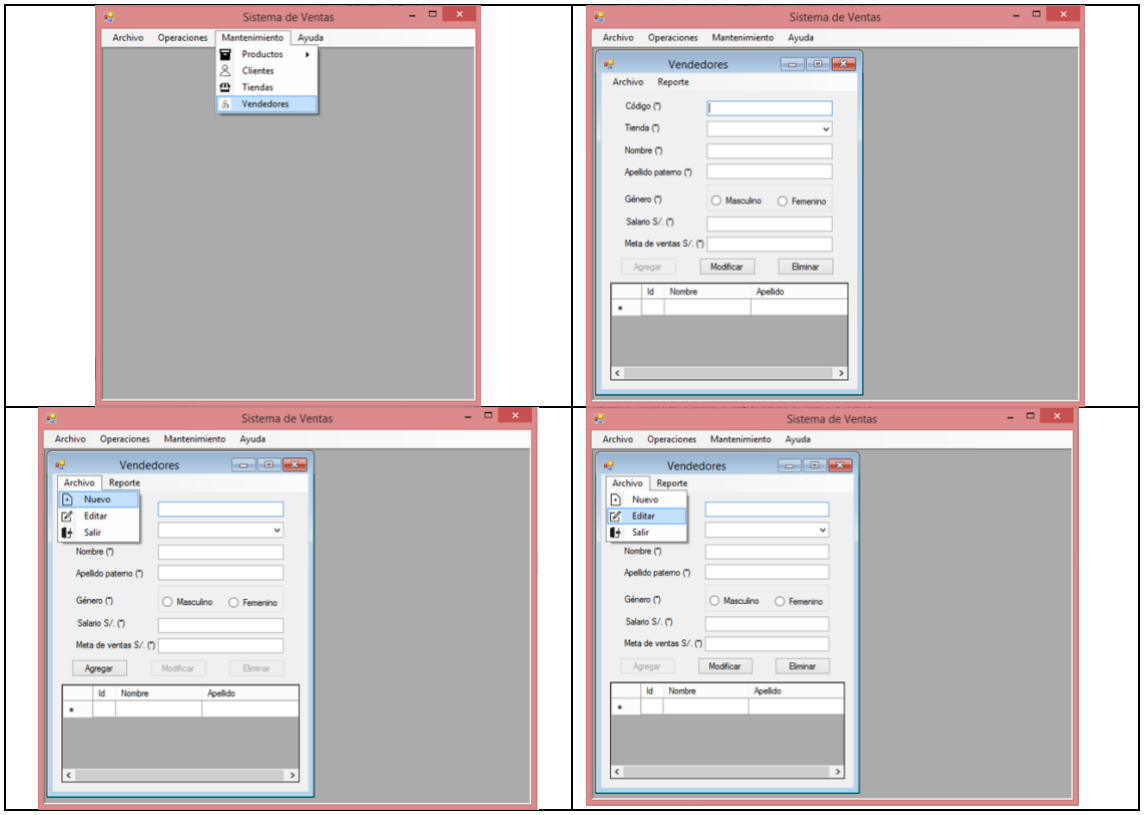


El mantenimiento de Tiendas debe funcionar de la siguiente manera:

1. En la ventana principal, al seleccionar la opción de Mantenimiento / Tiendas, se debe mostrar la ventana interna *StoreForm* cuyo título es “Tiendas”.
2. Al mostrar la ventana interna *StoreForm*, debe mostrarse en el grid las tiendas que ya se encuentren registrados en el archivo de texto en formato XML **stores.xml**. De no existir el archivo o estar vacío, no mostraría ninguna tienda. El grid solo muestra los campos Id y nombre de la tienda.
3. En la ventana interna *StoreForm*, cuando se seleccione una fila del grid, lo que hace es **Buscar** la tienda por el código o id de la tienda seleccionada, para luego mostrar todos los datos en el formulario de dicha tienda. Debe validar que solo se seleccione una fila que corresponde a una tienda.
4. En la ventana interna *StoreForm*, 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 una nueva tienda. Los demás botones estarán deshabilitados y debe validar los datos para que el registro de la nueva tienda sea correcto.
5. En la ventana interna *StoreForm*, el usuario debe ingresar todos los datos referentes a la nueva tienda y presionar el botón *Agregar*. Luego de registrar una nueva tienda, se debe limpiar los valores del formulario y se debe visualizar en el grid una nueva fila con los datos de la nueva tienda. **Al presionar el botón *Agregar* debe validar que cuando se ingrese una nueva tienda no se repita el código de la tienda. En caso que esto suceda no debe permitir el registro y debe mostrar un mensaje de error**. Recuerde que los datos de la nueva tienda deben estar registrados en el archivo ***stores.xml***
6. En la ventana interna *StoreForm*, si desea modificar una tienda, 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 *StoreForm*, luego que el usuario seleccione una tienda del grid, se visualizan sus datos en el formulario; a partir de ahí, el usuario puede modificar cualquier dato de la tienda, a excepción del código de la tienda y luego presionar el botón *Actualizar*. Al presionar *Actualizar*, se debe guarda la modificación realizada, se ve reflejado en el grid de Tiendas y también se actualiza el archivo ***stores.xml***.

8. En la ventana interna *StoreForm*, si se desea eliminar una tienda, se debe seleccionar la fila de la tienda del grid y presionar el botón *Eliminar*. Al realizarse esta acción, el grid debe actualizarse y eliminar la tienda seleccionada. El cambio también debe verse reflejado en el archivo **stores.xml**.

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



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

1. En la ventana principal, al seleccionar la opción de *Mantenimiento / Vendedores*, se debe mostrar la ventana interna *SalesmanForm* con el título “Vendedores”.
2. En la ventana interna *SalesmanForm* debe mostrarse en el grid los vendedores que ya se encuentren registrados en el archivo de texto en formato XML **salesmen.xml,** considerando la tienda a la que cada vendedor está asignado. De no existir el archivo o estar vacío, no mostraría ningún vendedor. El grid solo muestra los campos Id, nombre y apellido del vendedor.
3. En la ventana interna *SalesmanForm*, cuando se seleccione una fila del grid, lo que hace es **Buscar** el vendedor por el código o id del vendedor seleccionado, para luego mostrar todos los datos en el formulario de dicho vendedor. Debe validar que solo se seleccione una fila que corresponde a un vendedor.
4. En la ventana interna *SalesmanForm*, 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 vendedor. Los demás botones estarán deshabilitados y debe validar los datos para que el registro del nuevo vendedor sea correcto.
5. En la ventana interna *SalesmanForm*, el usuario debe ingresar todos los datos referentes al nuevo vendedor y presionar el botón *Agregar*. Para el caso del vendedor, debe llenar el combo de Tiendas para que se pueda elegir una tienda a la que se asignará el vendedor. Luego de registrar un nuevo vendedor, se debe limpiar los valores del formulario y se debe visualizar en el grid una nueva fila con los datos del nuevo vendedor. **Al presionar el botón *Agregar* debe validar que cuando se ingrese un nuevo vendedor no se repita el código del vendedor. En caso que esto suceda no debe permitir el registro y debe mostrar un mensaje de error**. Recuerde que los datos del nuevo vendedor deben estar registrados en el archivo ***salesmen.xml***

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

Pregunta 1: Sobre Mantenimiento de Tiendas (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 ubicaciones.
b. (3.0 ptos.) Implemente los cambios necesarios en el proyecto SalesController que permita soportar lo descrito en esta fase para el mantenimiento de tiendas.
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 las tiendas en el grid de StoreForm.
    - (1.5 pto.) Implementación del registro de una tienda utilizando el StoreForm.
    - (2.5 ptos.) Implementación del visualizar y actualizar una tienda usando el StoreForm.
    - (0.5 ptos.) Implementación del eliminar una tienda usando el StoreForm.

Pregunta 2: Sobre Mantenimiento de Vendedores (11 puntos)

d. (5.0 ptos.) Implemente los cambios necesarios en el proyecto SalesController que permita soportar lo descrito en esta fase para la funcionalidad de mantenimiento de vendedores.
e. (6.0 ptos.) Implemente los cambios necesarios en el proyecto SalesView que permita soportar lo descrito en esta fase para la funcionalidad de mantenimiento de vendedores.
    - (1.0 pto.) Implementación del presentar las tiendas en el combo de SalesmanForm.
    - (1.5 ptos.) Implementación del mostrar los vendedores en el grid de SalesmanForm.
    - (3.5 ptos.) Implementación del registro de un vendedor utilizando el SalesmanForm.

Nota.- Recuerde que debe seguir el modelo MVC y solo los atributos de las clases que pertenecen al modelo (Model) pueden ser públicos para simular el efecto de propiedades. Los atributos de las clases de los proyectos View y Controller deben ser privados. 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, 01 de junio de 2020