

Ministerio de Producción

Secretaría de Industria y Servicios

Subsecretaría de Servicios Tecnológicos y Productivos

y

Ministerio de Educación y Deportes

A través del



Analistas del Conocimiento

Dimensión Programador

Guía de Ejercicios Prácticos para el Módulo

Programación Orientada a Objetos

```
System.out.println("Kaizen");
```

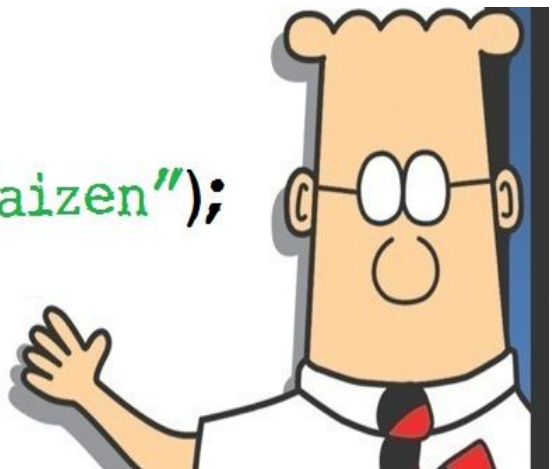


Tabla de Contenido

INTRODUCCIÓN	5
ENUNCIADOS DE LOS EJERCICIOS A DESARROLLAR EN LA GUÍA	7
EJERCICIO 1: CASO PRÁCTICO – CINE	7
PRESENTACIÓN DEL CASO DE ESTUDIO	7
MODELO DE DOMINIO.....	8
VISTA PARCIAL DEL DIAGRAMA DE CASOS DE USO.....	9
PROTOTIPO DEL CASO DE USO 14. REGISTRAR PELÍCULA.....	9
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 14. REGISTRAR PELÍCULA.....	10
DIAGRAMA DE SECUENCIA DEL ESCENARIO DEL CURSO NORMAL DEL CASO DE USO 14. REGISTRAR PELÍCULA.....	11
VISTA REGISTRAR PELÍCULA – DIAGRAMA DE CLASES DE ANÁLISIS	12
CONSIGNAS PARA EL DISEÑO E IMPLEMENTACIÓN:.....	13
EJERCICIO 2: CASO PRÁCTICO – PIZZERÍA.....	14
PRESENTACIÓN DEL CASO DE ESTUDIO	14
MODELO DE DOMINIO.....	15
DIAGRAMA DE MÁQUINA DE ESTADO DE LA CLASE PEDIDO.....	15
VISTA PARCIAL DE LOS CASOS DE USO DE LA PIZZERÍA.....	16
PROTOTIPO DEL CASO DE USO 22. GENERAR FACTURA.....	16
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 2. GENERAR FACTURA.....	17
DIAGRAMA DE SECUENCIA DEL ESCENARIO DEL CURSO NORMAL DEL CASO DE USO 22 GENERAR FACTURA.....	18
VISTA GENERAR FACTURA – DIAGRAMA DE CLASES DE ANÁLISIS	19
CONSIGNAS PARA EL DISEÑO E IMPLEMENTACIÓN:.....	20
EJERCICIO 3: CASO PRÁCTICO – ESTACIONAMIENTO DE UNIVERSIDAD	21
PRESENTACIÓN DEL CASO DE ESTUDIO	21
MODELO DE DOMINIO.....	22
VISTA PARCIAL DEL DIAGRAMA DE CASOS DE USO.....	22
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 3. REGISTRAR COBRO DE ABONO DE ESTACIONAMIENTO.....	24
DIAGRAMA DE SECUENCIA DEL ESCENARIO DEL CURSO NORMAL DEL CASO DE USO 3. REGISTRAR COBRO DE ABONO DE ESTACIONAMIENTO	25
VISTA REGISTRAR COBRO DE ABONO DE ESTACIONAMIENTO – DIAGRAMA DE CLASES DE ANÁLISIS	26
CONSIGNAS PARA EL DISEÑO E IMPLEMENTACIÓN:.....	27
EJERCICIO 4: CASO PRÁCTICO – MERCADO DE ABASTO	28
PRESENTACIÓN DEL CASO DE ESTUDIO	28
MODELO DE DOMINIO.....	29
DIAGRAMA DE MÁQUINA DE ESTADO DE LA CLASE PUESTO	29
VISTA PARCIAL DEL DIAGRAMA DE CASOS DE USO.....	30
PROTOTIPOS ASOCIADOS ALQUILER DE UN PUESTO	31
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 1. REGISTRAR ALQUILER DE PUESTO.....	32
DIAGRAMA DE SECUENCIA DEL ESCENARIO DEL FLUJO BÁSICO DEL CASO DE USO 3. REGISTRAR ALQUILER DE PUESTO.....	33
VISTA REGISTRAR ALQUILER DE PUESTO – DIAGRAMA DE CLASES DE ANÁLISIS	34
CONSIGNAS PARA EL DISEÑO E IMPLEMENTACIÓN:.....	35
EJERCICIO 5: CASO PRÁCTICO – PANADERÍA.....	36
PRESENTACIÓN DEL CASO DE ESTUDIO	36

PROTOTIPOS ASOCIADOS A LA FUNCIONALIDAD DE REGISTRAR COBRO	37
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 1. REGISTRAR VENTA	38
CONSIGNAS:.....	39
EJERCICIO 6: CASO PRÁCTICO – CONSULTORIO ODONTOLÓGICO	40
PRESENTACIÓN DEL CASO DE ESTUDIO	40
DESCRIPCIÓN CON RESUMEN ESENCIAL DEL CASO DE USO 1. GENERAR AGENDA MENSUAL DE ODONTÓLOGO	41
CONSIGNAS:.....	42
<u>FUENTES DE INFORMACIÓN</u>	<u>43</u>

Introducción

La guía práctica del Módulo Programación Orientada a Objetos, incluye los casos de estudio vinculados a los contenidos desarrollados en el apunte teórico del módulo. El objetivo de esta guía es brindar una herramienta de apoyo, que facilite el desarrollo de los temas y posibilite aplicar los conocimientos adquiridos mostrando casos prácticos y su resolución propuesta.

En el apunte teórico se trabajó alrededor de los siguientes ejes temáticos: los lenguajes de programación y los paradigmas de programación, presentados de forma introductoria. Luego una parte desarrolla el paradigma de objetos con todo su fundamento teórico, a continuación, se presenta el UML como lenguaje de modelado, para sistemas orientados a objeto y por último, se introduce la programación orientada a objetos en el lenguaje JAVA.

Este enfoque se utiliza para la importancia de transmitir que la programación de un software se desarrolla en forma evolutiva (en espiral), esa evolución se da desde el inicio, teniendo en cuenta los resultados obtenidos previamente, en otras palabras, antes de llegar a la programación se ha modelado lo que el usuario necesita, el diseño, y en ese momento inicia la programación con todas herramientas necesaria para que satisfaga las necesidades para las que fue creado.

El desarrollo de software se realiza en forma de evolución continua y espiralada del conocimiento, para que ese proceso sea correcto, los pasos previos que se deben realizar se basan en varias formas de entendimiento del problema, utilizando modelos, los cuales son de gran utilidad.

Es importante que el estudiante aprenda a programar conociendo los conceptos que fundamentan el paradigma, en este caso nos referimos la forma de representar e interpretar el problema.

El material práctico está organizado partiendo de un caso de estudio basado en un *Complejo de Cines*, este caso ya se ha trabajado en el apunte teórico. Se utilizan los modelos ya realizados y su solución en lenguaje java. Luego se presentan tres casos de estudio más que van evolucionando desde los requerimientos (necesidades del usuario), los cuales ya han sido esquematizados con modelos construidos en el lenguaje UML, también se ha desarrollado las formas en que el software le será presentado al usuario (prototipos de interfaz de usuario).

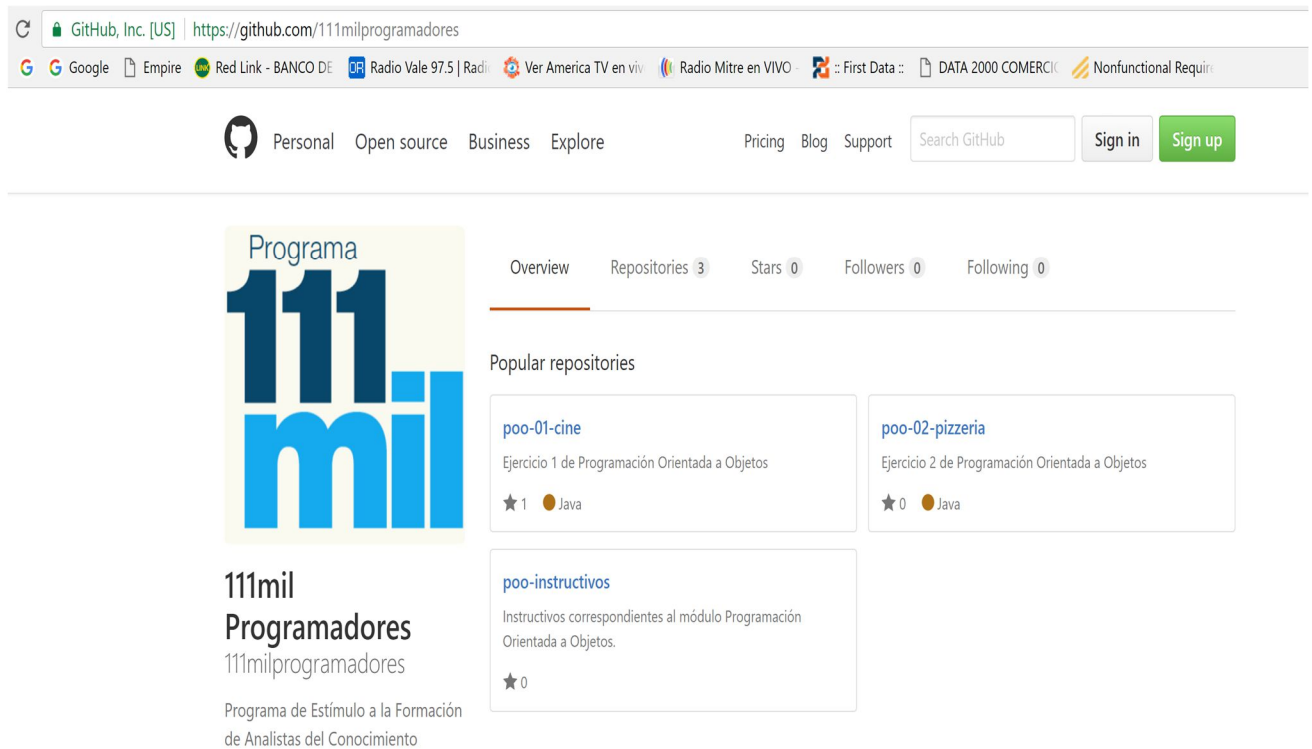
Finalmente, los últimos dos casos no tienen modelos de UML, solo tienen la descripción de la función que se espera obtener, puesto que también hay que preparar a los estudiantes para la situación que tengan que programar basados en pedidos escritos, no modelizados y no formales, debido a que hay organizaciones en las que se trabaja de esta forma, sin modelos previos.

Para la instalación de las aplicaciones necesarias para trabajar con la programación en Java (denominado usualmente implementación en JAVA), se han creado los siguientes instructivos:

- Instructivo de instalación de Java JDK y NetBeans IDE.
- Instructivo de creación de un nuevo proyecto en NetBeans.
- Instructivo de diseño de interfaces visuales con Java Swing en NetBeans IDE.

También se ha creado un repositorio público, que contiene estos instructivos y la implementación en código Java de los casos de estudio utilizados.

El repositorio se encuentra en: <https://github.com/111milprogramadores>



The screenshot shows the GitHub profile page for '111mil Programadores'. The profile name is '111mil Programadores' with the username '111milprogramadores'. The bio states 'Programa de Estímulo a la Formación de Analistas del Conocimiento'. The profile has 3 repositories, 0 stars, 0 followers, and 0 following. The 'Popular repositories' section lists three repositories: 'poo-01-cine' (Ejercicio 1 de Programación Orientada a Objetos, 1 star, Java), 'poo-02-pizzeria' (Ejercicio 2 de Programación Orientada a Objetos, 0 stars, Java), and 'poo-instructivos' (Instructivos correspondientes al módulo Programación Orientada a Objetos, 0 stars).

Finalmente, en el repositorio también hay un instructivo para la descarga del proyecto desde el repositorio público de 111mil creado en la herramienta GitHub.

Enunciados de los Ejercicios a Desarrollar en la Guía

Ejercicio 1: Caso Práctico – Cine

Presentación del Caso de Estudio

Este caso se encuentra mencionado en el material teórico del módulo, se trata de un Sistema de Gestión de Ventas y Reservas de Entradas para un Complejo de Cines. El funcionamiento del negocio se describe a continuación:

Un complejo de cines que está integrado por varios cines ubicados principalmente en los centros comerciales de la ciudad. Cada cine cuenta con una cantidad de salas, que son las que exhiben las distintas funciones cinematográficas. La programación de las salas se renueva en forma semanal, existiendo la posibilidad de que algunas salas queden sin uso. Cabe mencionar que no todas las salas tienen la misma capacidad (cantidad de butacas).

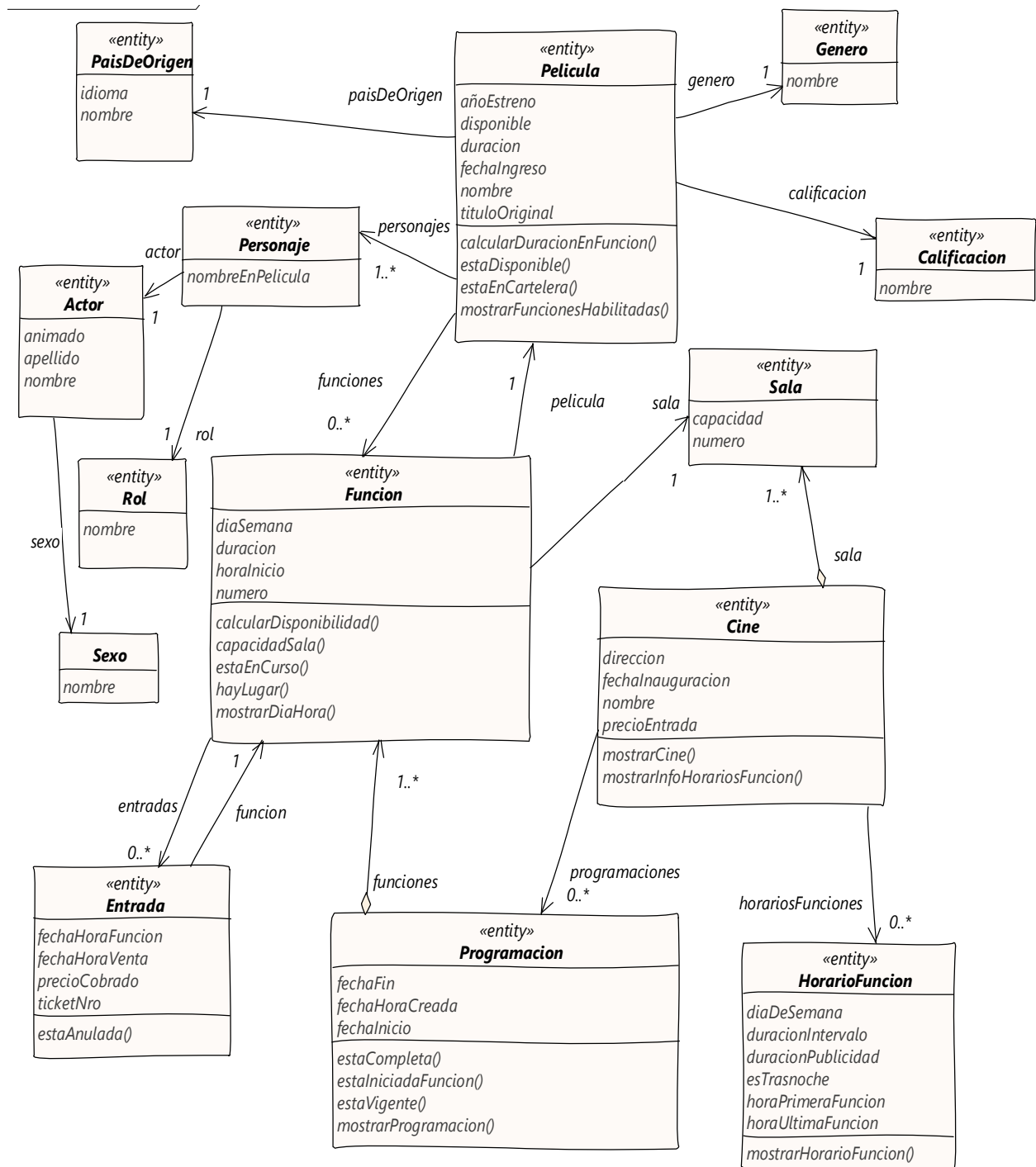
La programación es la que determina qué películas van a proyectarse y los horarios para cada función de cada sala, para todos los cines. Esta programación se realiza en forma centralizada, desde la administración del Complejo, tomándose como base la información de las películas próximas a estrenar, que envía el INCAA (Instituto Nacional de Cines y Artes Audiovisuales). La programación implica el diseño de las funciones y sus horarios en forma anticipada, debiendo el responsable de la misma, habilitar cada función en el momento que desee permitir la reserva y/o venta de entradas para la misma.

La entrada que se le entrega al cliente representa el comprobante de venta y como tal debe cumplir con lo reglamentado en la Ley de Facturación vigente, debiendo contener como datos: nro. de venta, fecha de venta, número de función, sala en la que se proyecta la película, el nombre de la película, fecha y hora de la función, el precio, el tipo de entrada (si es mayor, menor, jubilado) y la calificación de la película, que según especificaciones de la Ley de Cine Nro. 17.741, debe ser informada tanto en la entrada como al inicio de la película. Es importante destacar que la entrada es válida únicamente para la fecha, hora y función indicadas en la misma.

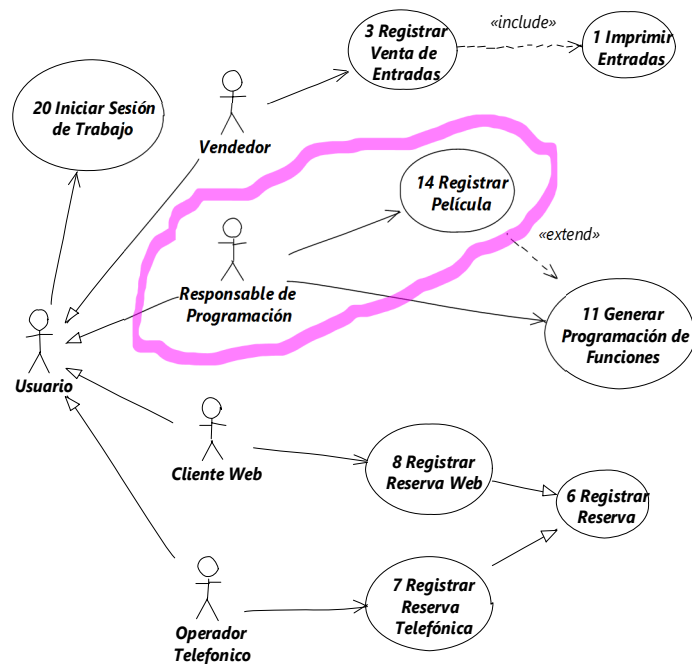
Los tipos de entradas y los días y horarios de proyección son los que determinan el precio de la entrada, que también pueden variar en cada cine del complejo. Las funciones admiten ciertos tipos de entradas y otros no, dependiendo de factores como: horarios, calificación de las películas, etc. Por ejemplo: si una película está calificada como para mayores de 16 años, para esa función no se pueden vender entradas de TIPO = MENOR. Cada función tiene asociado un tipo de función, que determina si la función es un pre-estreno, un estreno, una función normal.

A continuación, se muestra un modelo de programación, que servirá como ejemplo:

Modelo de Dominio




Vista parcial del Diagrama de Casos de Uso



Prototipo del Caso de Uso 14. Registrar Película

Registrar Película

 Ingrese Nombre:

Seleccione los siguientes datos

Género: <input type="text" value="Comedia"/>	Calificación: <input type="text" value="Excelente"/>	País de Origen: <input type="text" value="Argentina"/>
Drama Suspenso Terror Acción	Muy Buena Buena Regular Mala	Australia Estados Unidos Francia

Ingrese los datos restantes

Duración en Min:

Título Original:

Año de Estreno:

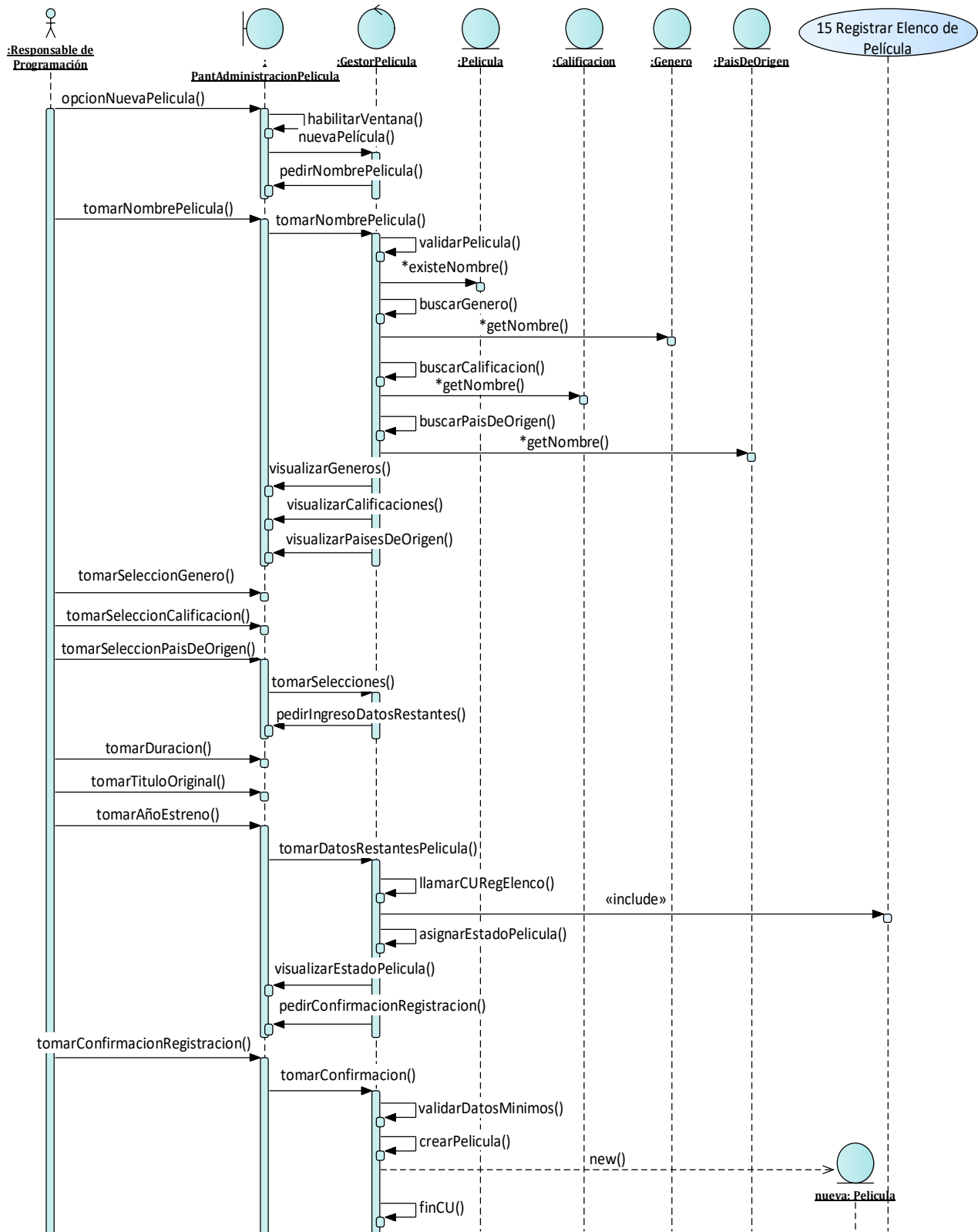
Registrar Elenco:

Disponible: ☐ Si ☒ No

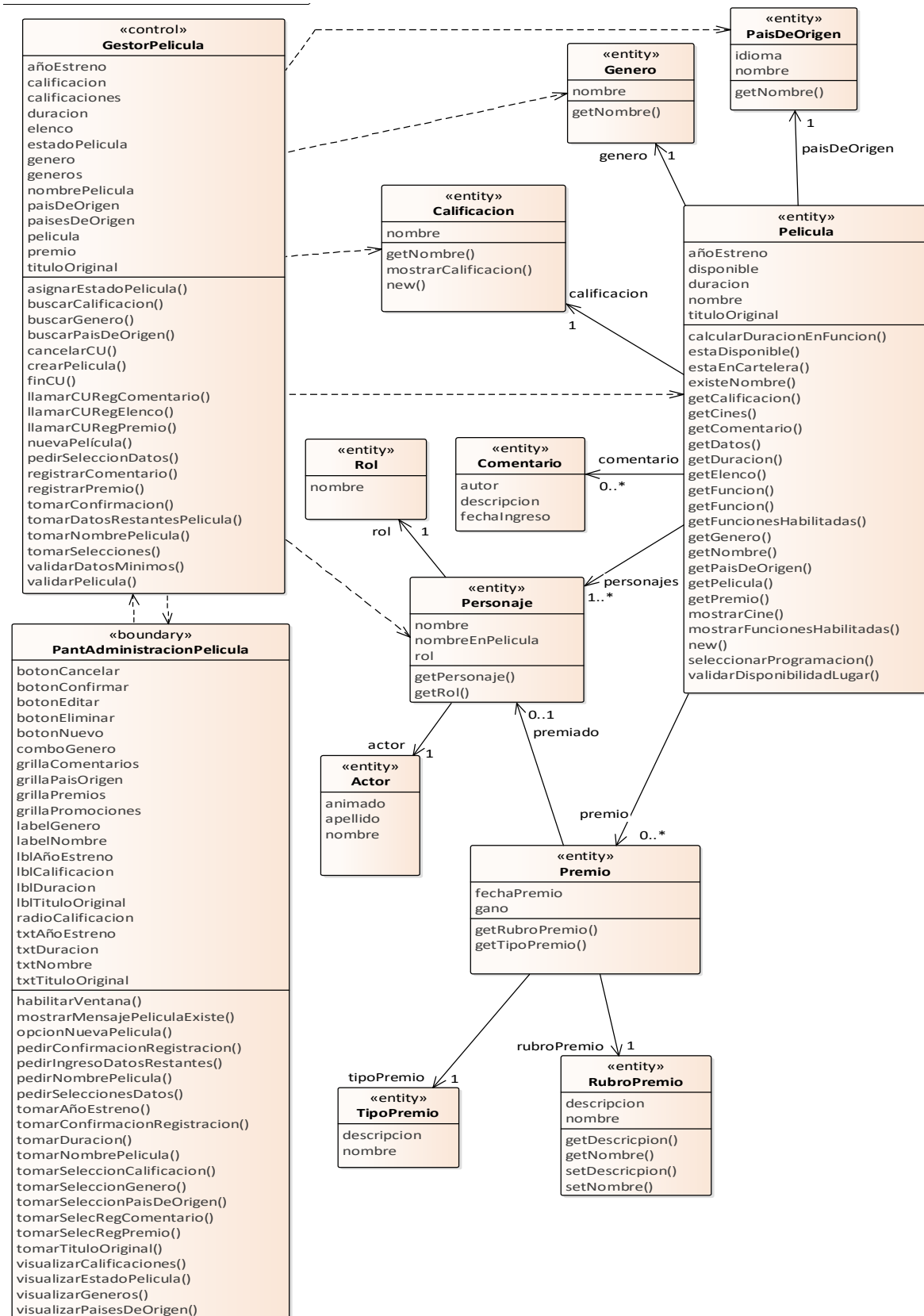
Descripción con resumen esencial del Caso de Uso 14. Registrar Película

Paquete: Administración de Películas	
Nombre del Caso de uso: Registrar Película	Nro. de Orden: 14
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Actor Principal: Responsable de Programación (RP)	Actor Secundario: no aplica
Tipo de Caso de uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Objetivo: registrar cada una de las películas que integrarán las distintas funciones que se programarán para los cines del complejo.	
Flujo Básico	
1. RP: selecciona la opción de ingresar nueva película .	
2. Sistema: solicita se ingrese nombre de película.	
3. RP: ingresa el nombre de la película que desea registrar.	
4. Sistema valida que no exista una película registrada con el mismo nombre y no hay.	
5. Sistema: busca y muestra los Géneros, Calificaciones y Países de Origen de la película y solicita se seleccionen.	
6. RP: selecciona los datos correspondientes a la película.	
7. Sistema: solicita se ingresen los demás datos para la película.	
8. RP ingresa: duración, título original, año de estreno.	
9. Sistema: Para ingresar elenco de película llamar al caso de uso Registrar Elenco de Película .	
10. Sistema: El elenco se registró correctamente.	
11. Sistema: asigna el estado de la película como "NO DISPONIBLE" y el RP no desea modificarlo.	
12. Sistema: solicita confirmación para registrar película ingresada.	
13. RP confirma.	
14. Sistema: valida que se hayan especificado los datos mínimos (nombre, duración, año de estreno, calificación, género, país de origen, elenco), requeridos para realizar la registración de la película y han sido especificados.	
15. Sistema registra la película.	
Flujos Alternativos	
A1: Existe una película registrada con el mismo nombre.	
A2: Elenco no pudo registrarse.	
A3: El RP desea ingresar premios de películas	
A4: El RP desea ingresar comentarios de películas	
A5: El sistema asigna el estado de la película como "NO DISPONIBLE" y el actor <u>desea</u> modificarlo.	
A6: No se han ingresado los datos mínimos requeridos para efectuar la registración de la película.	
A7: El RP no confirma registración de la película.	
Observaciones:	
1. El RP puede cancelar la ejecución del caso de uso en cualquier momento.	

Diagrama de Secuencia del escenario del curso normal del Caso de Uso 14. Registrar Película



Vista Registrar Película – Diagrama de Clases de Análisis



Consignas para el Diseño e Implementación:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *“Instructivo de instalación de Java JDK y NetBeans IDE.”*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *“Instructivo de creación de un nuevo proyecto en NetBeans IDE”*.
- Crear cada una de las clases según lo especificado en el Modelo de Dominio planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método `get()` y método `set()`], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción y el diagrama de secuencia que modela el escenario descrito, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (`public` o `private`), el valor de retorno y los parámetros necesarios.
- Implementar las clases de *fabricación pura* que se consideren necesarias, para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *“Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE.”*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el *“Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub”*.

Ejercicio 2: Caso Práctico – Pizzería

Presentación del Caso de Estudio

Una pizzería de la ciudad ofrece a sus clientes una amplia variedad de pizzas de fabricación propia, de varios tamaños (8, 10 y 12 porciones). Los clientes tienen a disposición un menú que describe para cada una de las variedades, el nombre, los ingredientes y el precio según el tamaño de la pizza. Los clientes pueden realizar sus pedidos personalmente en el mostrador o telefónicamente, en este último caso, se le pide al cliente el domicilio completo, el número de teléfono y luego de informado el monto se le consulta con cuanto va a pagar, para llevar preparado el vuelto, en caso que sea necesario, para el momento en el cual se le entrega el pedido en su domicilio.

El pedido debe contener además de los datos del cliente (que se requieren únicamente si el pedido es a domicilio), la cantidad de pizzas, el tamaño, la variedad, es posible que una pizza sea de una o dos variedades en el caso que se pida media pizza de cada variedad diferente, la fecha del pedido, la hora en la que el pedido debe entregarse y la demora estimada informada al cliente.

El pedido va a la cocina y cuando está preparado se informa al que lo tomó para que se genere la factura correspondiente y se le entregue el pedido al cliente.

Una vez entregado el pedido el repartidor debe rendir cuenta del dinero correspondiente a los pedidos que entregó como así también del comprobante en el cual el cliente firma la recepción del pedido consignando la hora de recepción del mismo. Dicha información es registrada posteriormente en el pedido para calcular demoras promedio de cumplimentación de pedidos.

Además del reporte antes mencionado el dueño de la pizzería ha manifestado la necesidad de acceder al menos a la siguiente información:

- Variedades de pizzas más pedidas por los clientes.
- Ingresos (recaudaciones) por turno de trabajo y por períodos de tiempo.
- Pedidos (cantidad y monto) entregados por repartidor.

Modelo de Dominio

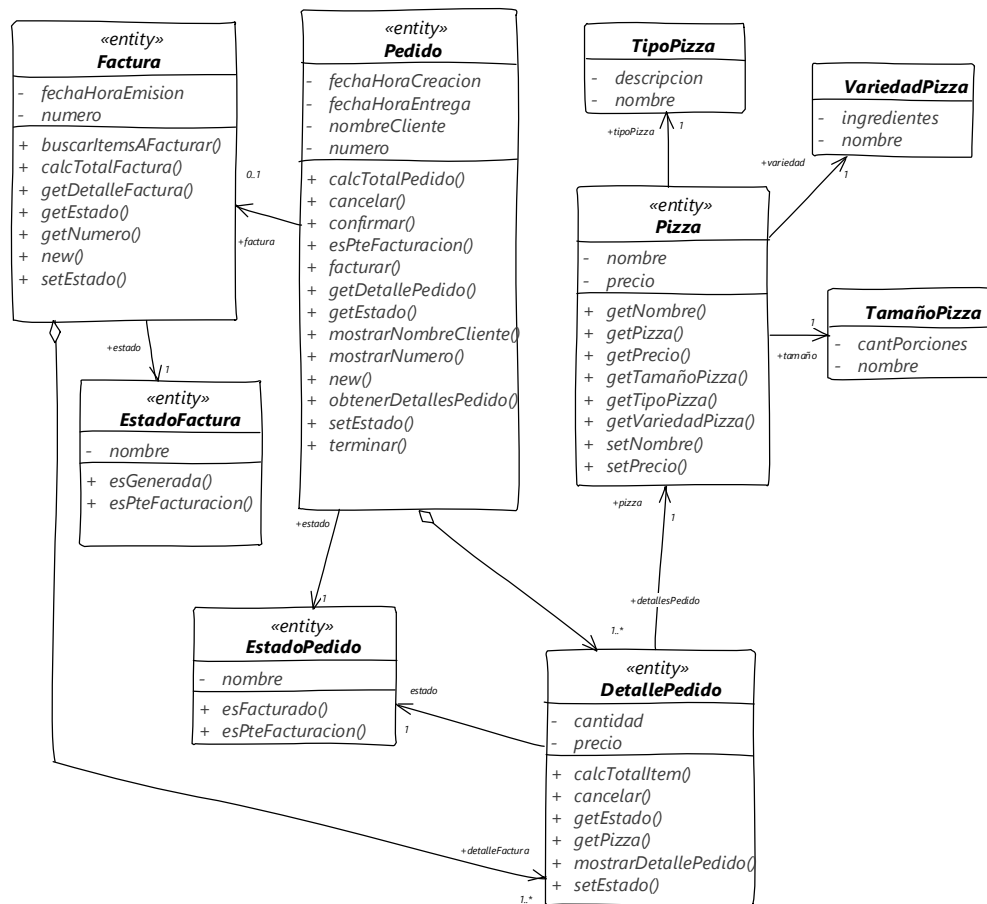
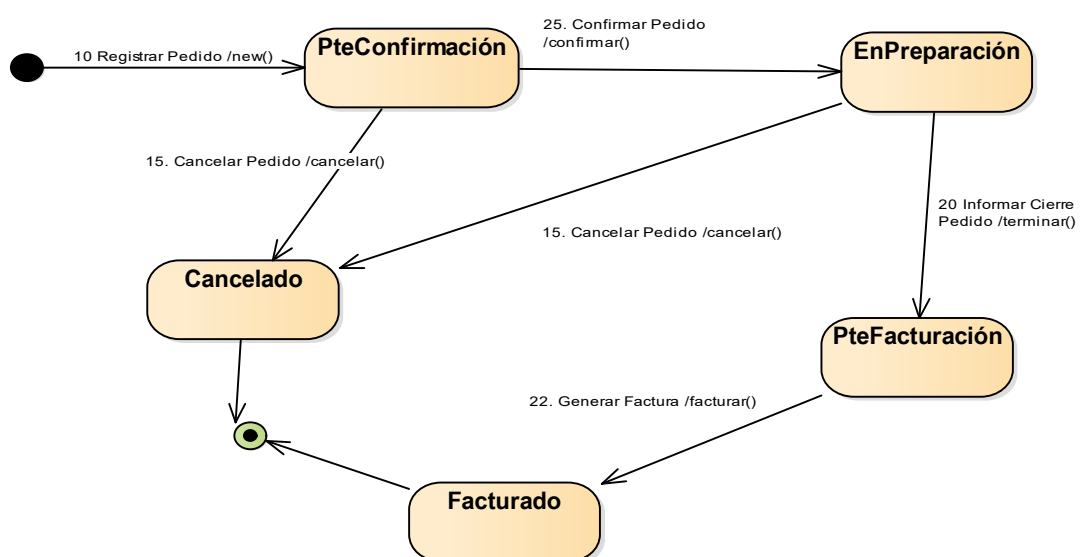
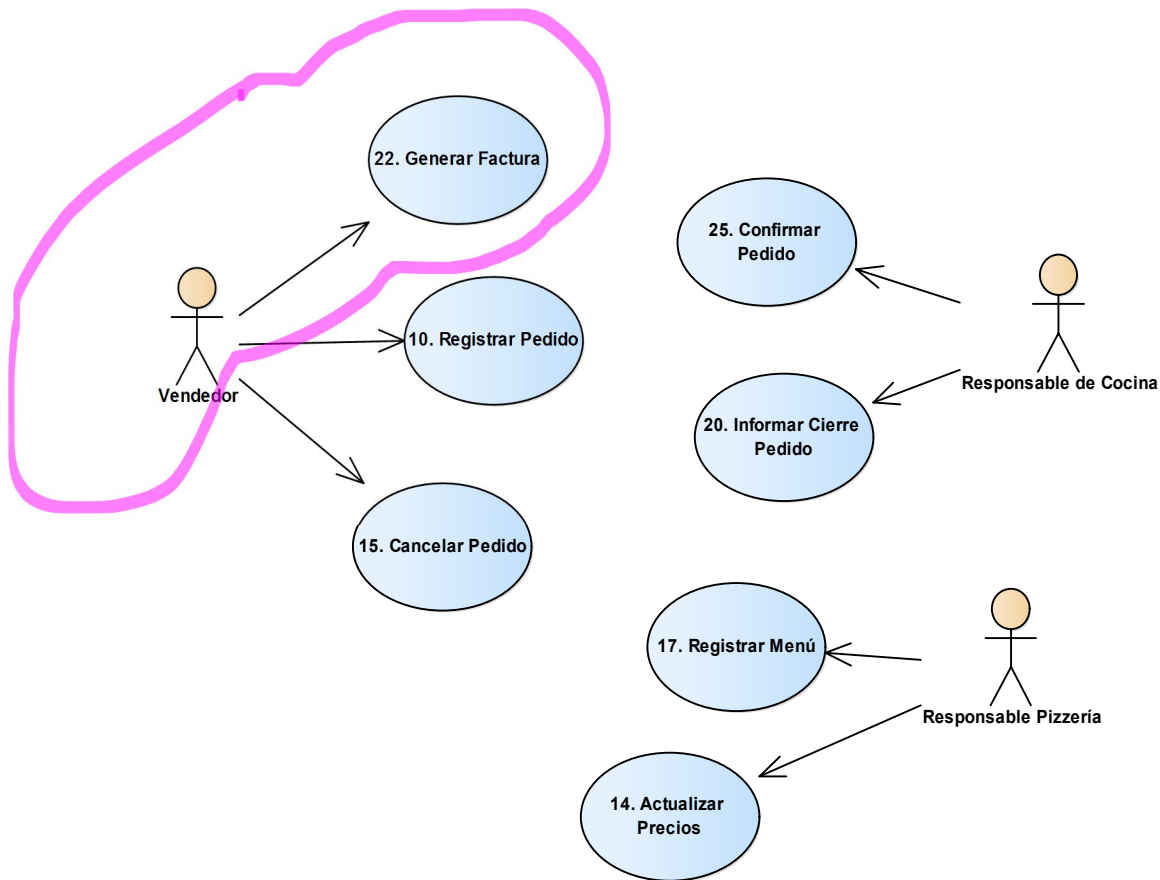


Diagrama de Máquina de Estado de la Clase Pedido



Vista parcial de los casos de uso de la Pizzería



Prototipo del Caso de Uso 22. Generar Factura

Generar Factura

Seleccione un pedido

Número	Fecha	Hora	Cliente
1	11/10/2016	21:30	María Cortez
2	11/10/2016	21:45	Carmen Lopez
3	11/10/2016	21:50	Juan Ferreyra
4	11/10/2016	21:58	Martin Sanchez

Datos del Pedido Seleccionado

Pedido N°: 1 - Fecha y Hora: 11/10/2016 - Cliente: María Cortez

Cantidad	Pizza	Tipo	Variedad	Tamaño	Precio	SubTotal
3	Napolitana Normal	Parrilla	Napolitana	8	\$125	\$375
1	Especial Grande	Horno	Especial	12	\$165	\$165
2	Mozzarella Mediana	Parrilla	Mozzarella	10	\$120	\$240

Monto Total del Pedido: \$780

Confirmación

¿Desea generar la factura?

Factura Generada

Número de Factura: 0006451

Factura N° 0006451

Fecha de Emisión: 11/10/2016

Cliente: María Cortez

Hora de Emisión: 22:15

Detalle Factura

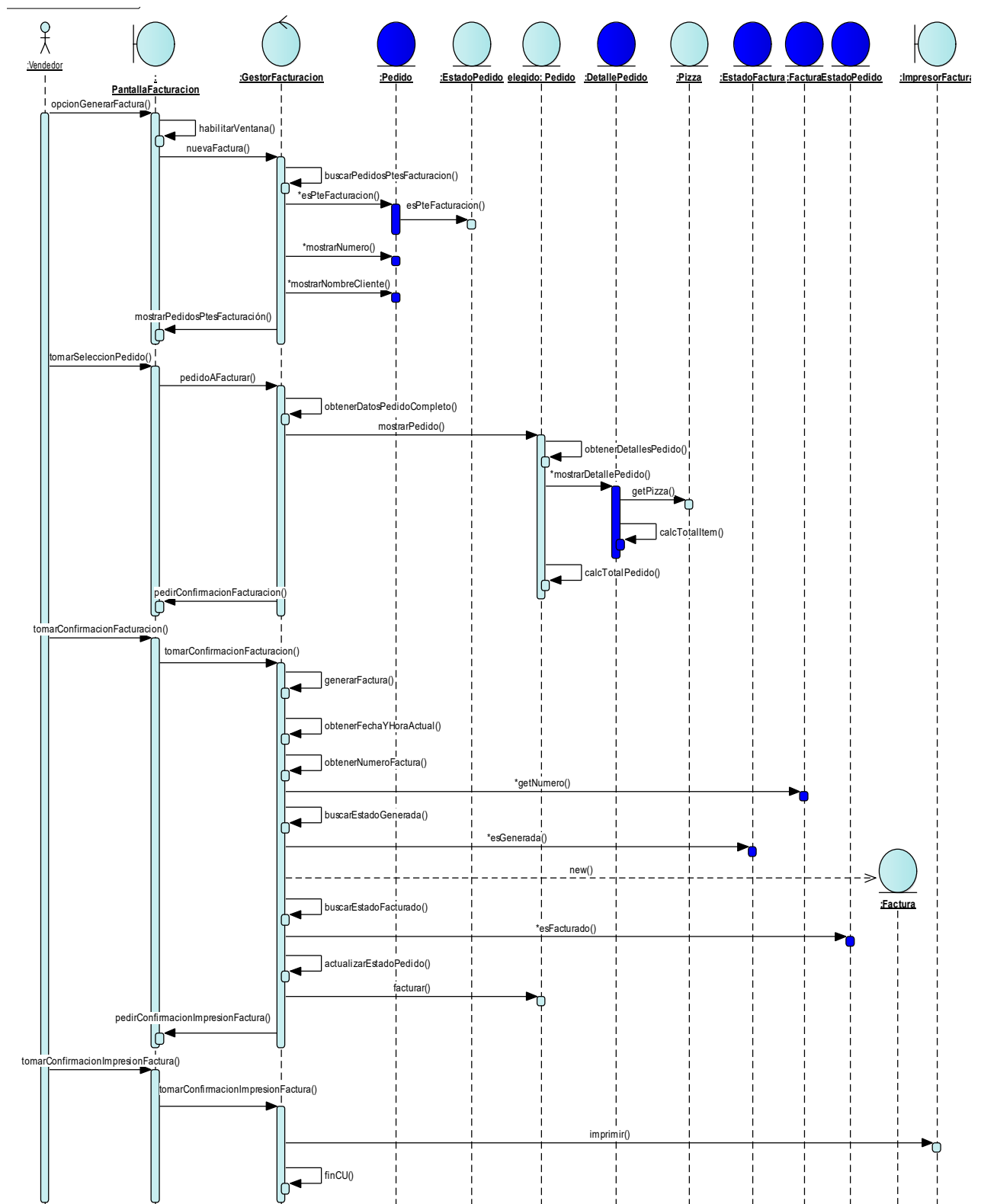
Cantidad	Pizza	Tipo	Variedad	Tamaño	Precio	SubTotal
3	Napolitana Normal	Parrilla	Napolitana	8	\$125	\$375
1	Especial Grande	Horno	Especial	12	\$165	\$165
2	Mozzarella Mediana	Parrilla	Mozzarella	10	\$120	\$240

Monto Total: \$780

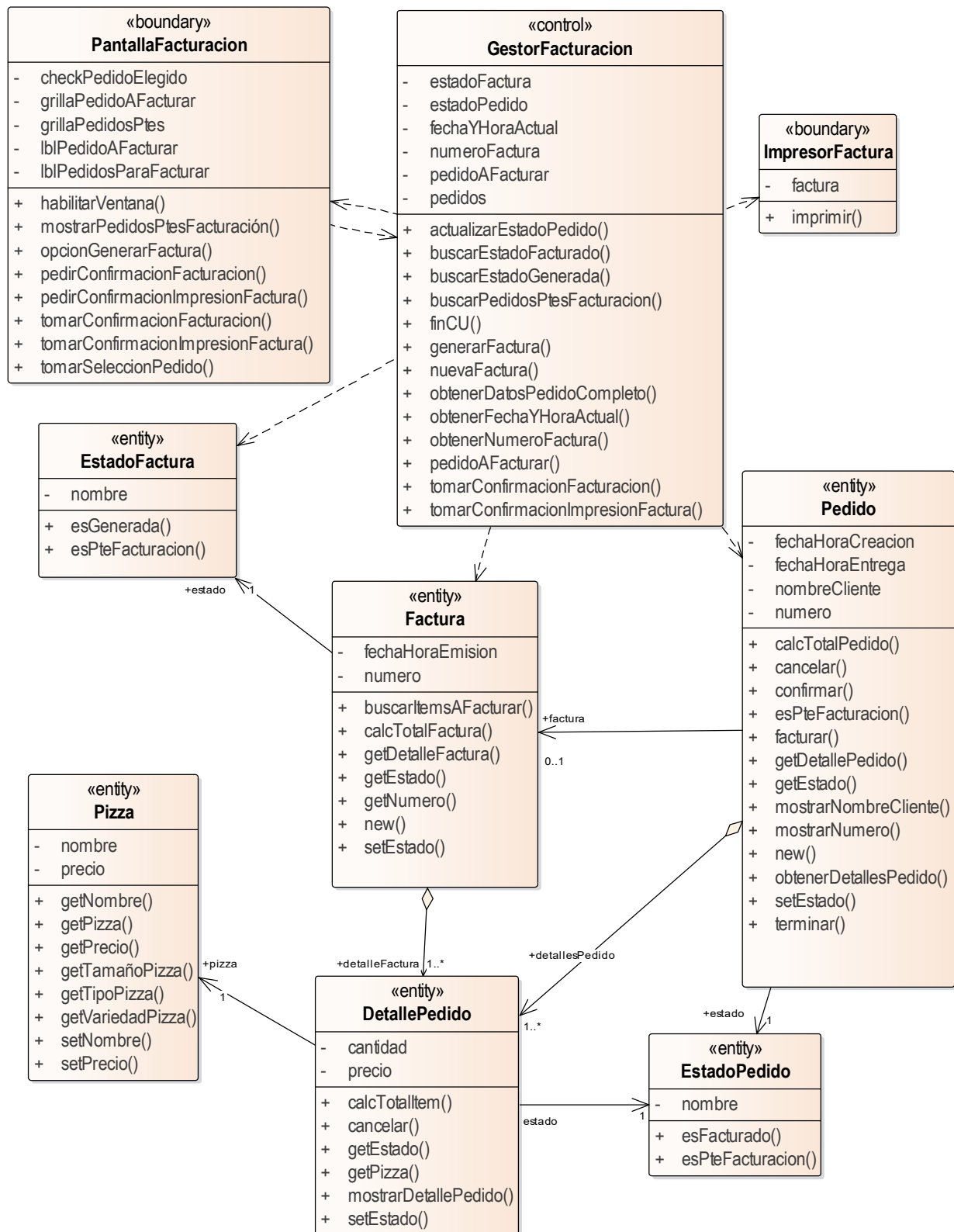
Descripción con resumen esencial del Caso de Uso 2. Generar Factura

Nombre del Caso de uso: Generar Factura		Nro. de Orden: 2
Actor Principal: Vendedor		Actor Secundario: no aplica
Objetivo: generar la factura asociada a un pedido determinado.		
Flujo Básico		
1. Vendedor: selecciona la opción “Generar Factura”.		
2. Sistema: busca y muestra la lista de pedidos que estén listos y no hayan sido facturados aún y solicita se seleccione uno.		
3. Vendedor: selecciona el pedido que desea facturar.		
4. Sistema: muestra los datos completos del pedido y calcula el monto total a cobrar.		
5. Sistema: solicita la confirmación de la generación de la factura.		
6. Vendedor: confirma la generación de la factura.		
7. Sistema: genera la factura en estado “Generada” y actualiza el estado del pedido a “Facturado”.		
8. Sistema: solicita confirmación de la impresión de la factura.		
9. Vendedor: confirma la impresión.		
10. Sistema: imprime la factura. Fin del caso de uso.		
Flujos Alternativos		
A1: No hay pedidos pendientes de facturación.		
A2: El vendedor no confirma la generación de la factura.		
A3: El vendedor NO confirma la impresión.		
Observaciones:		
1. El Vendedor puede cancelar la ejecución del caso de uso en cualquier momento.		

Diagrama de Secuencia del escenario del curso normal del Caso de Uso 22 Generar Factura.



Vista Generar Factura – Diagrama de Clases de Análisis



Consignas para el Diseño e Implementación:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *“Instructivo de instalación de Java JDK y NetBeans IDE.”*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *“Instructivo de creación de un nuevo proyecto en NetBeans IDE.”*
- Crear cada una de las clases según lo especificado en el Modelo de Dominio planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método `get()` y método `set()`], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción, la máquina de estado que modela el ciclo de vida de los objetos de la clase con sus estados y transiciones permitidas y el diagrama de secuencia que modela el escenario descrito, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (`public` o `private`), el valor de retorno y los parámetros necesarios.
- Implementar las clases de *fabricación pura* que se consideren necesarias, para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *“Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE.”*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el *“Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub”*.

Ejercicio 3: Caso Práctico – Estacionamiento de Universidad

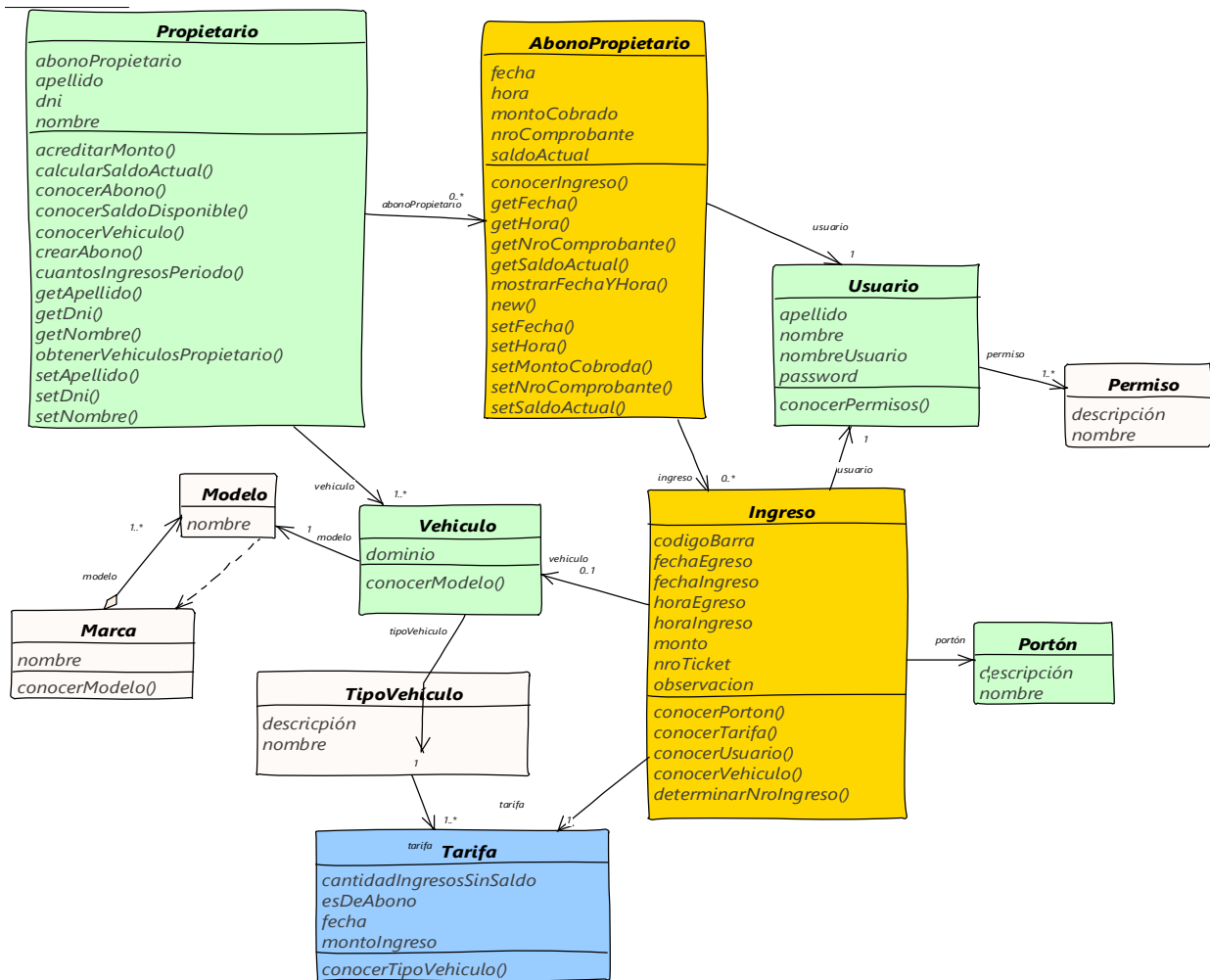
Presentación del Caso de Estudio

Se describe a continuación el funcionamiento de la playa de estacionamiento de la Universidad Tecnológica y del sistema de información que le da soporte.

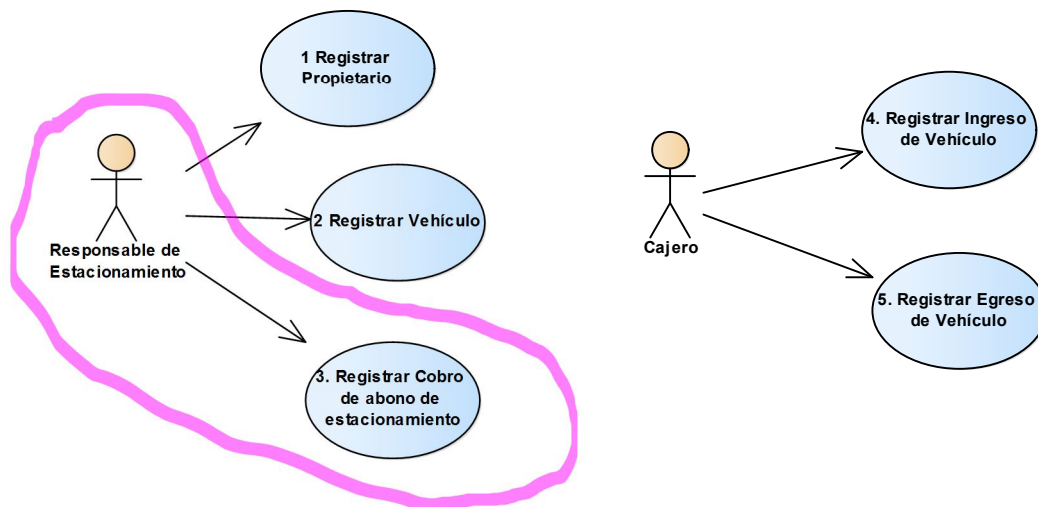
- ⇒ Pueden estacionar distintos tipos de vehículos (motos/automóviles), cada uno con una tarifa de ingreso diferente. Si tiene abono el precio es menor.
- ⇒ Se puede ingresar a la playa de estacionamiento por varios portones de ingreso diferentes
- ⇒ No se asignan lugares específicos para los vehículos, las personas que ingresan al estacionamiento deberán ubicar su vehículo en el lugar que se encuentre disponible.
- ⇒ Los interesados pueden comprar un abono de estacionamiento, de pago anticipado que hace que el valor de cada estacionamiento sea más económico que si paga cada vez que ingresa a la playa. Debe informar su DNI y la cantidad de dinero que desea acreditar.
- ⇒ Si es la primera vez, debe registrar sus datos personales (apellido, nombre, dni), y los datos del o los vehículos (marca, modelo, dominio), con los cuales desea ingresar a la playa de estacionamiento.
- ⇒ Una vez registrado el propietario, cada vez que necesite acreditar dinero informa su DNI y la cantidad de dinero y se le cobra entregándole un comprobante donde consta: apellido y nombre, dni, fecha de la transacción, monto acreditado y monto disponible en su cuenta.
- ⇒ El comprobante (ticket) que se entrega como constancia del cobro tiene los siguientes datos: apellido y nombre del propietario, dni, fecha y hora de la transacción, monto acreditado y monto disponible en su cuenta, los números de dominio de todos los vehículos registrados de ese propietario y un número único de identificación del comprobante.
- ⇒ Puede tener hasta dos ingresos sin crédito, es decir saldo negativo, que se descontarán de la siguiente vez que acredite dinero en su cuenta.
- ⇒ La persona mientras tenga crédito puede ingresar a la playa con cualquiera de los vehículos registrados.
- ⇒ La persona puede en cualquier momento agregar y/o cambiar los vehículos con los que ingresará a la playa de estacionamiento.
- ⇒ El valor del estacionamiento es por el día completo, sin límite de tiempo ni inferior ni superior, es decir se paga un ingreso diario, que es válido independientemente de la cantidad de ingresos que haga durante el mismo día y del tiempo que permanezca en la playa.
- ⇒ Al ingresar se le entrega a la persona un comprobante que contiene: dominio del vehículo, apellido y nombre del dueño del vehículo, el valor del ingreso, la fecha de ingreso y el saldo disponible. También se informa el número de ingreso del día. El portón por el que ingresa y el usuario logueado.
- ⇒ Si el vehículo no está registrado se guarda en el ingreso el número de dominio del vehículo y se informa como observación que no está registrado.
- ⇒ El primer ingreso del día se cobra, descontando del saldo disponible. A partir del segundo ingreso del día en adelante, el monto debe figurar en cero y se debe informar que número de ingreso, por ejemplo: "Segundo ingreso del día".
- ⇒ A las personas que desean ingresar a la playa de estacionamiento sin tener el abono de pago anticipado, se les cobra al momento del ingreso, registrando como observación el número de dominio del vehículo, entregándoles un comprobante con el monto cobrado, los datos del comprobante en ese caso son: dominio del vehículo, monto, fecha de ingreso, número de vez que ingresa a la playa de estacionamiento, usuario logueado, fecha y hora y portón por el que ingresa.

- ⇒ Si la persona tiene abono, puede tener hasta el valor de dos estacionamientos como saldo negativo, que se descontarán de la siguiente vez que acredite dinero en su cuenta.

Modelo de Dominio



Vista parcial del Diagrama de Casos de Uso



Prototipo del Caso de Uso 3. Registrar Cobro de abono de estacionamiento


Cobro de abono de estacionamiento


UNIVERSIDAD TECNOLÓGICA Fecha: 03/04/2013
usuario: HLOPEZ

Fecha Actual:

Hora Actual:

Buscar Persona

DNI: 

Propietario 

Apellido


Nombre

Estado de Cuenta

Saldo Actual: \$

Ingresar importe: \$

Información

 El saldo actualizado es:

ASOCIACION TECNOLOGICA CORDOBA

15458669 Toledo Ana Paula

Ticket N°	Descripción	Importe
01112905	CRÉDITO PLAYA	\$ 50.00

Importe Total: \$ 50.00

Patente: GRA256, BQL566, EBG558

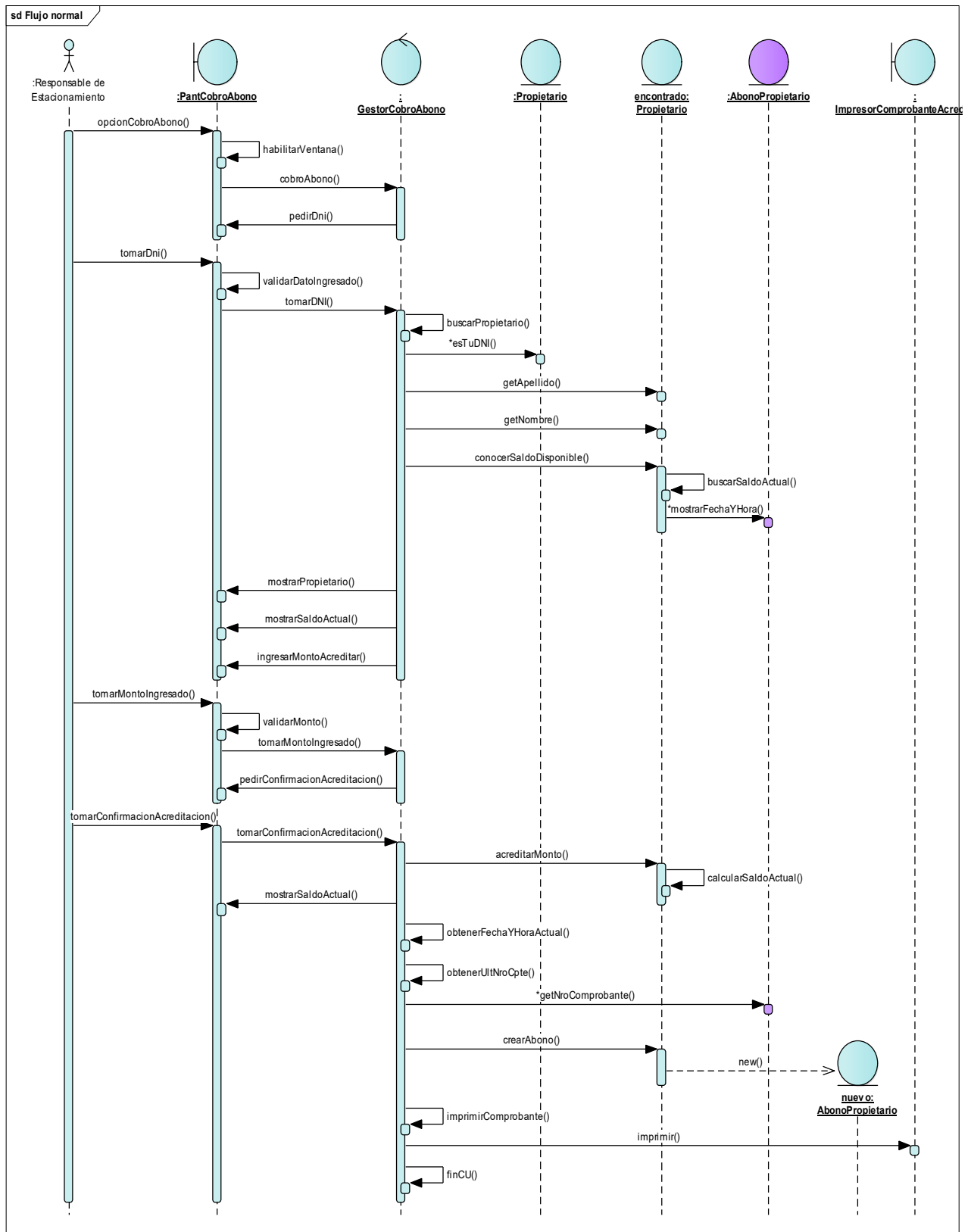
Crédito Actual Playa: \$53

06/09/2012 10:15:04

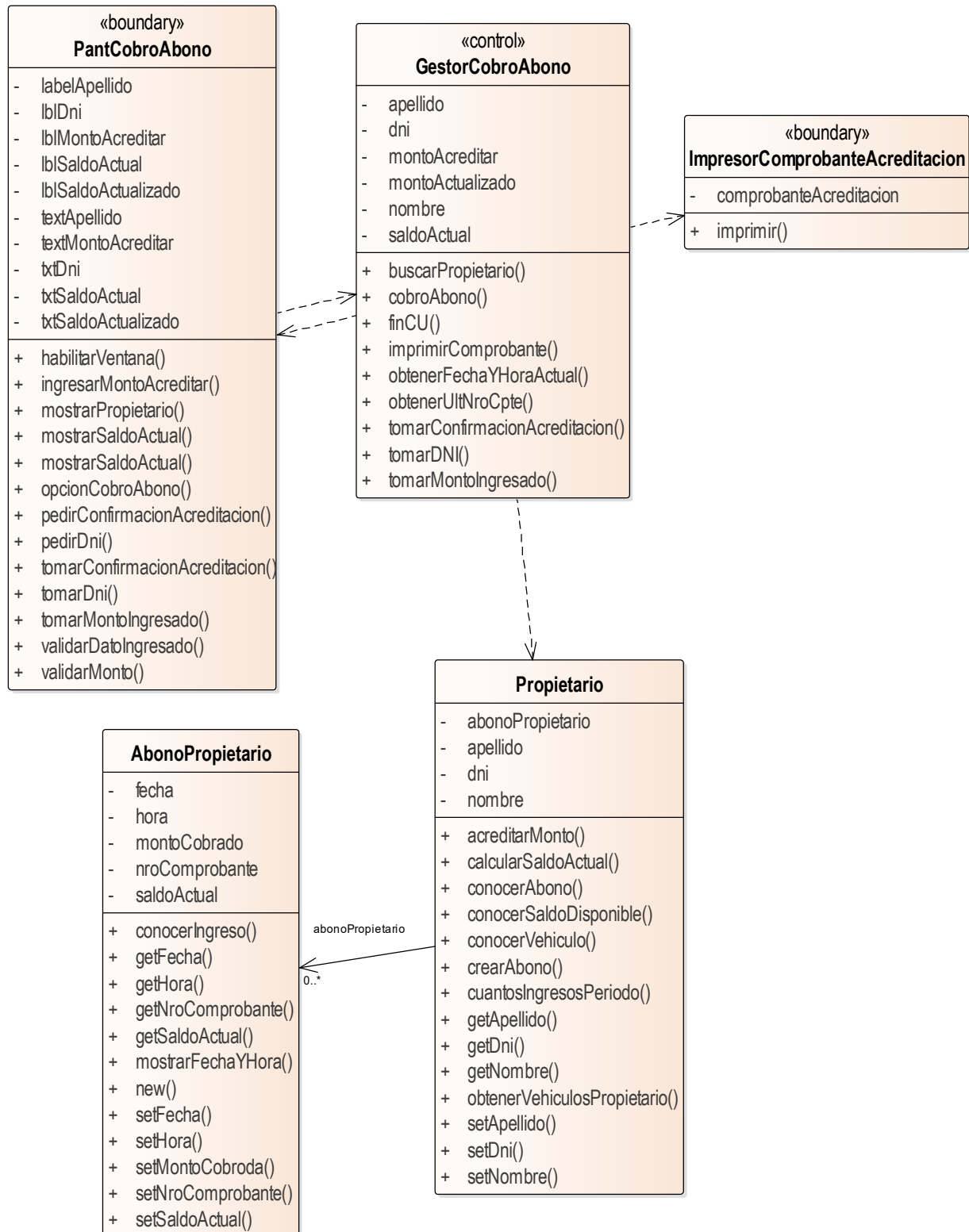
Descripción con resumen esencial del Caso de Uso 3. Registrar Cobro de abono de estacionamiento

Nombre del Caso de uso: Registrar Cobro de abono de estacionamiento		Nro. de Orden: 3
Actor Principal: Responsable de Estacionamiento (RE)		Actor Secundario: no aplica
Objetivo: acreditar saldo en la cuenta de un propietario		
Flujo Básico		
1. RE: selecciona la opción de registrar cobro de abono de estacionamiento .		
2. Sistema: solicita se ingrese DNI del propietario al que se desea acreditar saldo.		
3. RE: ingresa el número de DNI.		
4. Sistema controla que el número de DNI sea válido y que corresponda a número de DNI registrado y es así. Muestra el apellido y nombres y el saldo actual del propietario registrado con ese número de DNI.		
5. Sistema: solicita se ingrese el monto a acreditar.		
6. RE: ingresa el monto a acreditar.		
7. Sistema: valida que el monto ingresado sea un número válido y lo es. Solicita confirmación para la acreditación del monto a la cuenta del propietario.		
8. RE: Confirma la acreditación del monto en la cuenta.		
9. Sistema: muestra el saldo actualizado de la cuenta (saldo anterior + monto acreditado).		
10. Sistema: genera e imprime el comprobante de acreditación de saldo en la cuenta.		
11. Sistema acredita el monto, con la fecha y hora actual, número de comprobante y saldo actual. Fin del caso de uso.		
Flujos Alternativos		
A1: El DNI ingresado no es un número válido.		
A2: El DNI ingresado no corresponde a un propietario registrado.		
A3: El monto ingresado no es un monto válido.		
A4: El RE no confirma la acreditación del monto a la cuenta.		
Observaciones:		
1. El RP puede cancelar la ejecución del caso de uso en cualquier momento.		
2. El saldo actual de una cuenta puede ser negativo, en ese caso debería mostrarse en color rojo y con un signo negativo delante.		

Diagrama de Secuencia del escenario del curso normal del Caso de Uso 3. Registrar Cobro de abono de estacionamiento



Vista Registrar Cobro de abono de estacionamiento – Diagrama de Clases de Análisis



Consignas para el Diseño e Implementación:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *“Instructivo de instalación de Java JDK y NetBeans IDE.”*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *“Instructivo de creación de un nuevo proyecto en NetBeans IDE”*.
- Crear cada una de las clases según lo especificado en el Modelo de Dominio planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método `get()` y método `set()`], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción y el diagrama de secuencia que modela el escenario descrito, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (`public` o `private`), el valor de retorno y los parámetros necesarios.
- Implementar las clases de *fabricación pura* que se consideren necesarias, para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *“Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE.”*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el *“Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub”*.

Ejercicio 4: Caso Práctico – Mercado de Abasto

Presentación del Caso de Estudio

El Mercado de Abasto de Frutas y Verduras de una ciudad de la región necesita un Sistema de Información que brinde soporte a las actividades que allí se realizan.

El mercado está organizado en sectores. Cada sector contiene puestos, los cuales son alquilados a empresas y quinteros (genéricamente clientes) para que allí realicen sus ventas. Existen distintos tipos de puestos (con techo, sin techo, con cámara refrigerante, etc.) y distintas dimensiones para cada puesto (10m², 15m², etc.), para poder ajustarse mejor a las necesidades de cada cliente.

El precio del alquiler depende del sector en el que se encuentre el puesto, el tipo de puesto y sus dimensiones, y está predefinido.

Cuando un cliente desea alquilar uno o más puestos, se verifica la disponibilidad del tipo de puesto que requiere. Si existe disponibilidad y el cliente está de acuerdo con el precio, se realiza un contrato de alquiler por cada puesto que se alquile. En el contrato se especifica la fecha de inicio y fin del alquiler, el monto mensual del alquiler y tiene además un número que identifica el contrato que es único y el nombre del responsable por parte del Mercado que intervino en la firma del contrato y el responsable de la registración del mismo. Además, cada puesto cuenta con un medidor para el consumo de energía eléctrica. Mensualmente se registran las lecturas de cada medidor, ya que el consumo de cada puesto es facturado al cliente que está alquilando ese puesto. En el momento de efectuar el alquiler, se registra en el contrato la última lectura del medidor del puesto que se está alquilando. Los aspectos vinculados a la facturación quedan excluidos del alcance del sistema, como así también la gestión de cobro de los alquileres.

Modelo de Dominio

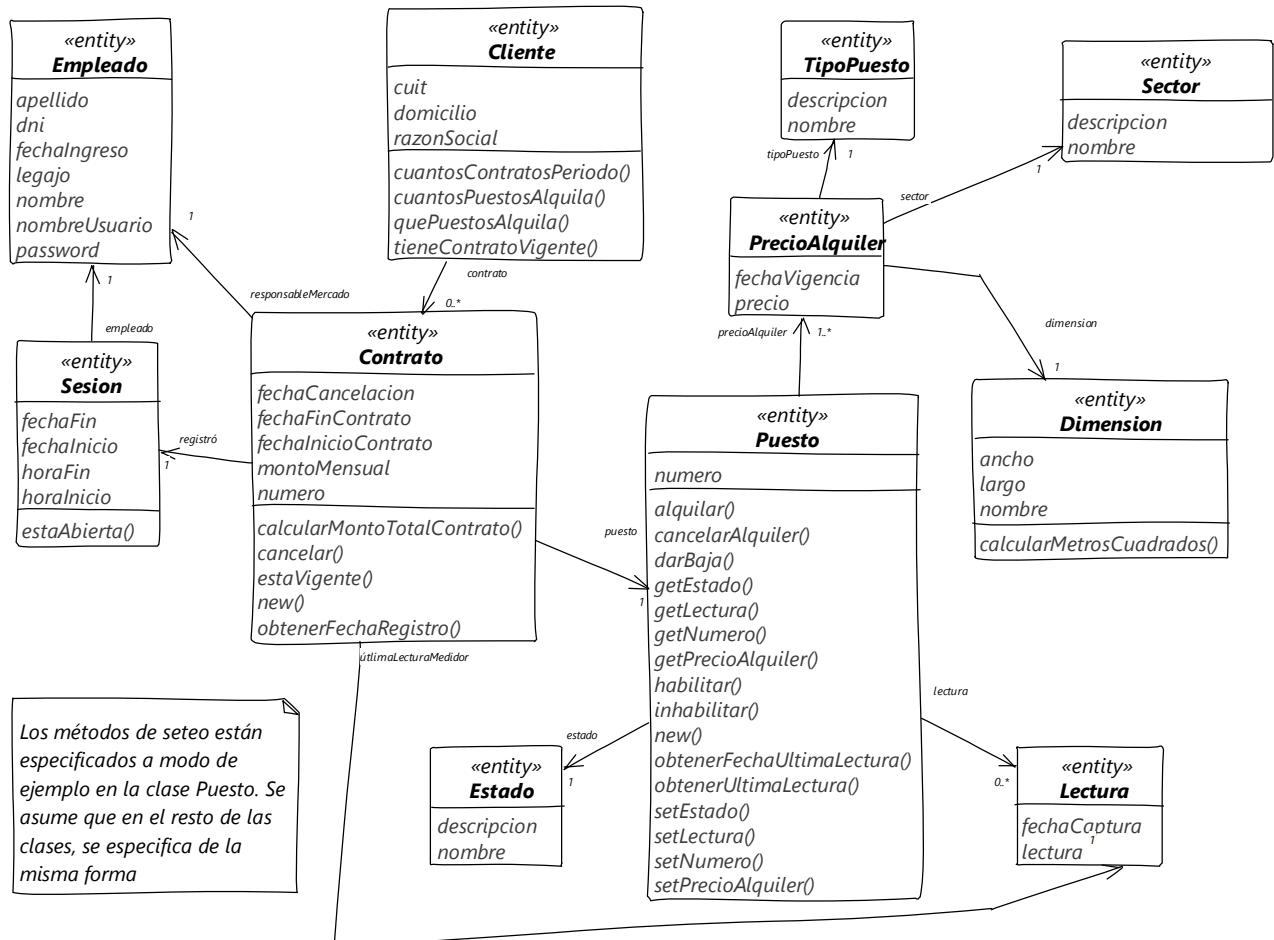
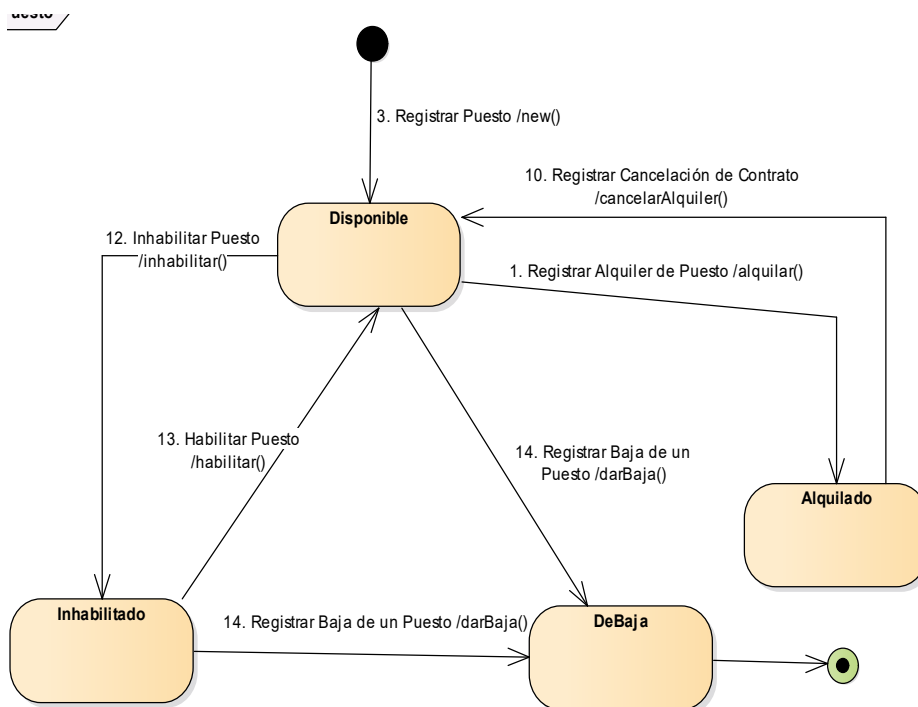
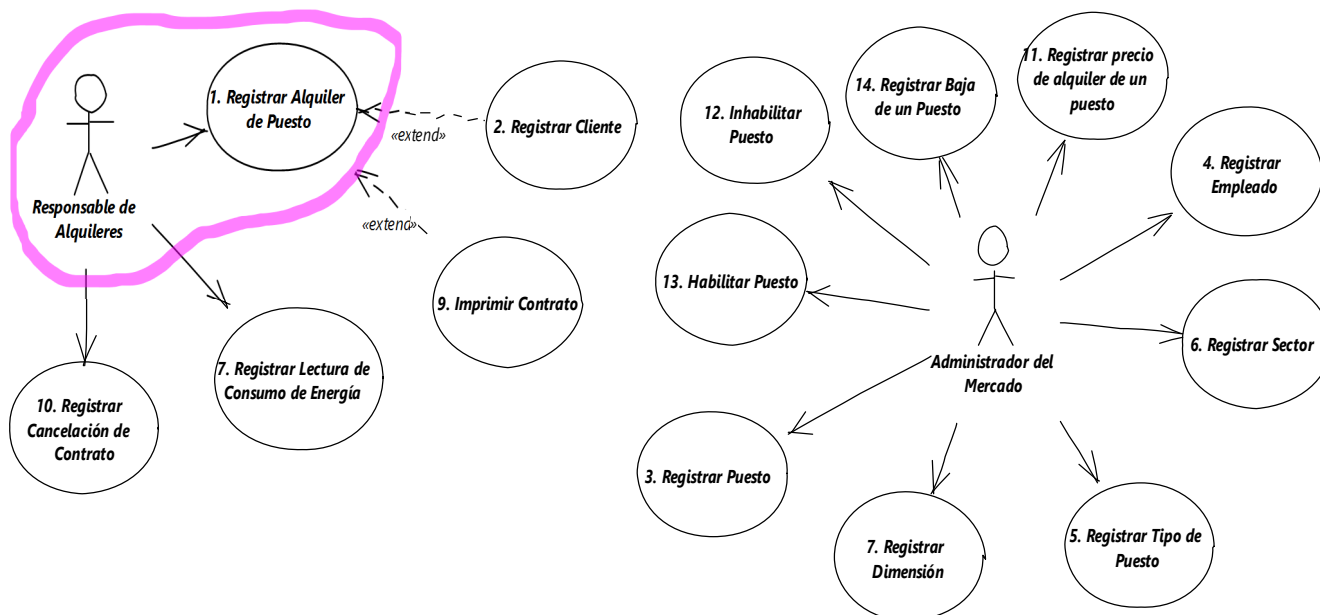


Diagrama de Máquina de Estado de la Clase Puesto



Vista parcial del Diagrama de Casos de Uso



Nro.	Nombre del Caso de Uso	Breve Descripción
1.	Registrar Alquiler de Puesto	Registrar el alquiler de un puesto a un cliente, generando un contrato por un período de tiempo definido y actualizando disponibilidad del puesto.
2.	Registrar Cliente	Registrar datos identificatorios del cliente que va a alquilar un puesto del mercado
3.	Registrar Puesto	Registrar los datos que definen cada uno de los puestos del mercado de abasto.
4.	Registrar Empleado	Registrar datos identificatorios de los empleados que trabajan en el mercado
5.	Registrar Tipo de Puesto	Registrar las características de un tipo de puesto del mercado.
6.	Registrar Sector	Registrar datos de un sector del mercado.
7.	Registrar Dimensión	Registrar las medidas tipo para un puesto.
8.	Registrar Lectura de Consumo de Energía	Registrar la lectura de un consumo mensual de energía de un puesto del mercado.
9.	Imprimir Contrato	Imprimir un contrato generado a papel, a un archivo o a formato PDF.
10.	Registrar Cancelación de Contrato	Registrar la fecha de cancelación anticipada de un contrato de alquiler de un puesto del mercado.
11.	Registrar precio de alquiler de un puesto	Registrar la definición del precio de alquiler, que depende del sector en el que se encuentra el puesto, el tipo de puesto y sus dimensiones.
12.	Inhabilitar Puesto	Registrar la inhabilitación de un puesto, para evitar que el mismo pueda ser alquilado.
13.	Habilitar Puesto	Registrar la habilitación de un puesto para que el mismo esté disponible para ser alquilado
14.	Registrar Baja de un Puesto	Registrar la fecha de baja de un puesto del mercado, que ya no podrá ser utilizado.

Prototipos asociados Alquiler de un Puesto

Registrar Alquiler de Puesto

MERCADO Fecha Actual: 29/03/14
usuario: RGOMEZ

Seleccione Periodo

Fecha Inicio: / / Fecha Vencimiento: / /

Seleccione características del puesto

Tipo de Puesto: **Techado** Dimensión: **10** m²

Seleccione un puesto disponible

Sector	Número de	Dimensión	Alquiler
<input checked="" type="radio"/> Ala Sur	15	10 m ²	\$ 450
<input type="radio"/> Ala Sur	8	10 m ²	\$ 330
<input type="radio"/> Ala	3	10 m ²	\$ 240

Aceptar Cancelar

Registrar Contrato Alquiler de Puesto

MERCADO Fecha Actual: 29/03/14
usuario: RGOMEZ

Buscar Cliente

Nombre/Razón Social de Cliente:
Pedro Varela

Dirección: Avenida Colón 2514
CUIT: 30-78665102-9 **Nuevo Cliente**

Puesto

Última Lectura Medidor: 56 Kw

Aceptar Cancelar

Contrato Registrado

Número de Contrato : **14 - 154** Imprimir

Aceptar Cancelar


MERCADO DE ABASTO 29/03/14
imprimió: RGOMEZ

CONTRATO DE ALQUILER DE PUESTO

A los **veintinueve** días del mes de **Marzo** del año **2014** se establece que el señor/a:

GONZALES, FACUNDO C.U.I.T: 30-7462518-4

con residencia en Av. Martinez 1500, se compromete a abonar la suma mensual de **\$450** durante un periodo que inicia el día: **15/04/14** y finaliza el **15/07/14** en concepto de alquiler del puesto detallado a continuación:

 Puesto numero: **15**

Puesto Superficie: **10 m²**

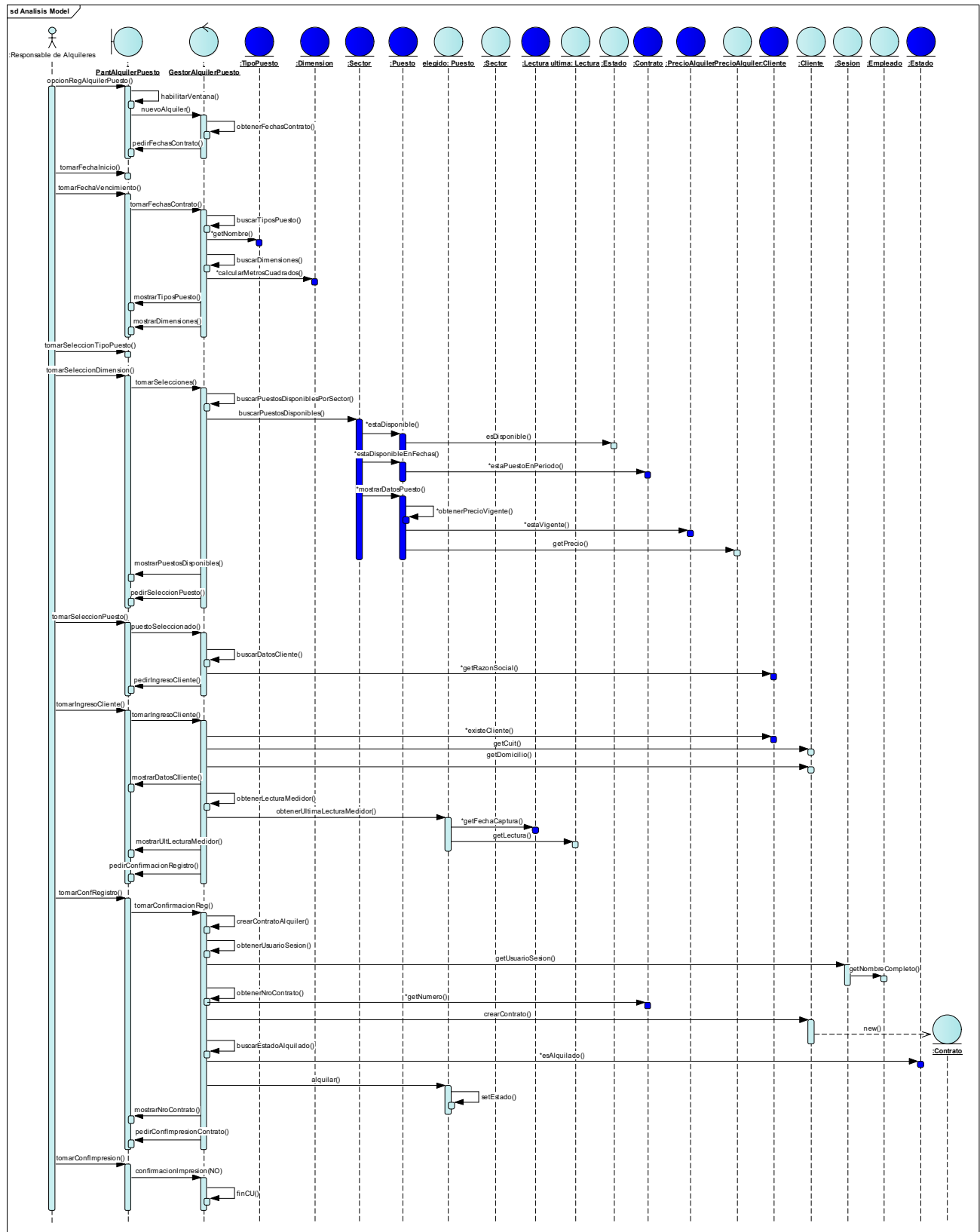
Ubicación: **Ala sur del mercado cito en Bv Quiroga 2103**

Ultima lectura medidor de Luz : **56Kw**

Descripción con resumen esencial del Caso de Uso 1. Registrar Alquiler de Puesto

Nombre del Caso de uso: Registrar Alquiler de Puesto		Nro. de Orden: 1
Actor Principal: Responsable de Alquileres (RA)		Actor Secundario: no aplica
Objetivo: Registrar el alquiler de un puesto a un cliente, generando un contrato por un período de tiempo definido y actualizando disponibilidad del puesto.		
Flujo Básico		
1. RA: selecciona la opción “ Registrar Alquiler de Puesto ”		
2. Sistema: solicita se ingrese la fecha de inicio y de vencimiento del contrato de alquiler.		
3. RA: ingresa las fechas requeridas.		
4. Sistema: valida las fechas (fecha inicio menor a fecha de vencimiento) y son correctas.		
5. Sistema: busca y muestra los distintos tipos de puesto y dimensiones (en metros cuadrados) y solicita se seleccione un tipo de puesto y una dimensión.		
6. RA: selecciona un tipo de puesto y una dimensión.		
7. Sistema: busca en cada sector del mercado los puestos disponibles para las fechas ingresadas (Ver Obs. 2) que correspondan al tipo de puesto y dimensión seleccionados, encuentra al menos uno, mostrando para cada puesto los siguientes datos: nombre del sector, número de puesto, dimensiones y precio vigente. Solicita se seleccione uno.		
8. RA: selecciona un puesto.		
9. Sistema: solicita se ingrese nombre o razón social del cliente.		
10. RA: ingresa los datos solicitados.		
11. Sistema: valida la existencia del cliente y se encuentra registrado muestra demás datos del cliente (domicilio y CUIT).		
12. Sistema: busca y muestra la última lectura del medidor del puesto a alquilar.		
13. Sistema: solicita la confirmación para registrar el alquiler.		
14. RA: confirma la registración.		
15. Sistema: registra el alquiler del puesto con el período de alquiler, referencia al cliente, al puesto alquilado, número de contrato, responsable que efectuó la registración del contrato que es el usuario actual, fecha, última lectura del medidor del puesto alquilado; actualizando el estado del puesto a Alquilado.		
16. Sistema: muestra el número de contrato generado y consulta si se desea imprimir el contrato registrado.		
17. RA: No desea imprimir el contrato. Fin del Caso de Uso		
Flujos Alternativos		
A1: Las fechas ingresadas, no son válidas.		
A2: El cliente no está registrado, para registrarlo se llama al Caso de Uso 2. Registrar Cliente.		
A3: El RA desea imprimir el contrato, llama al Caso de Uso 9. Imprimir Contrato.		
A4: El RA no confirma la registración del alquiler.		
Observaciones:		
1. El RP puede cancelar la ejecución del caso de uso en cualquier momento.		
2. La fecha de vencimiento debe ser mayor a la fecha de inicio del contrato.		
3. El período de vigencia del contrato (fecha de inicio del contrato – fecha de fin del contrato) debe validarse para que no se superponga con otros contratos para el mismo puesto.		

Diagrama de Secuencia del escenario del flujo básico del Caso de Uso 3. Registrar Alquiler de Puesto



Consignas para el Diseño e Implementación:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *“Instructivo de instalación de Java JDK y NetBeans IDE.”*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *“Instructivo de creación de un nuevo proyecto en NetBeans IDE”*.
- Crear cada una de las clases según lo especificado en el Modelo de Dominio planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método `get()` y método `set()`], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción, la máquina de estado que modela el ciclo de vida de los objetos con sus estados y transiciones permitidas y el diagrama de secuencia que modela el escenario descrito, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (`public` o `private`), el valor de retorno y los parámetros necesarios.
- Implementar las clases de *fabricación pura* que se consideren necesarias, para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *“Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE.”*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el *“Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub”*.

Ejercicio 5: Caso Práctico – Panadería

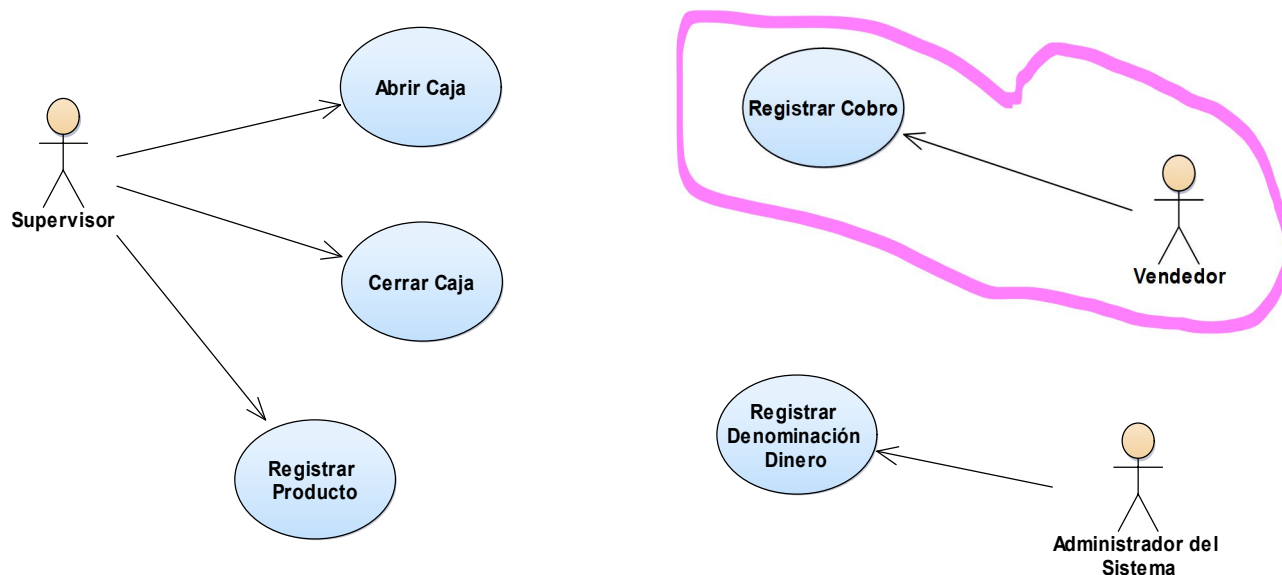
Presentación del Caso de Estudio

La Panadería que se describe en este caso de estudio, pertenece a la Fundación Brisas de Cambio, ubicada en el interior de la provincia de Córdoba. La Fundación tiene el propósito fundamental de contener laboralmente a un grupo numeroso de jóvenes y adultos con discapacidades intelectuales y físicas. Su objetivo es desarrollar proyectos productivos que les permita desempeñarse en un oficio para sentirse útiles y adquirir a diario el conocimiento necesario para desempeñarse en esta actividad dentro de un ambiente laboral sano.

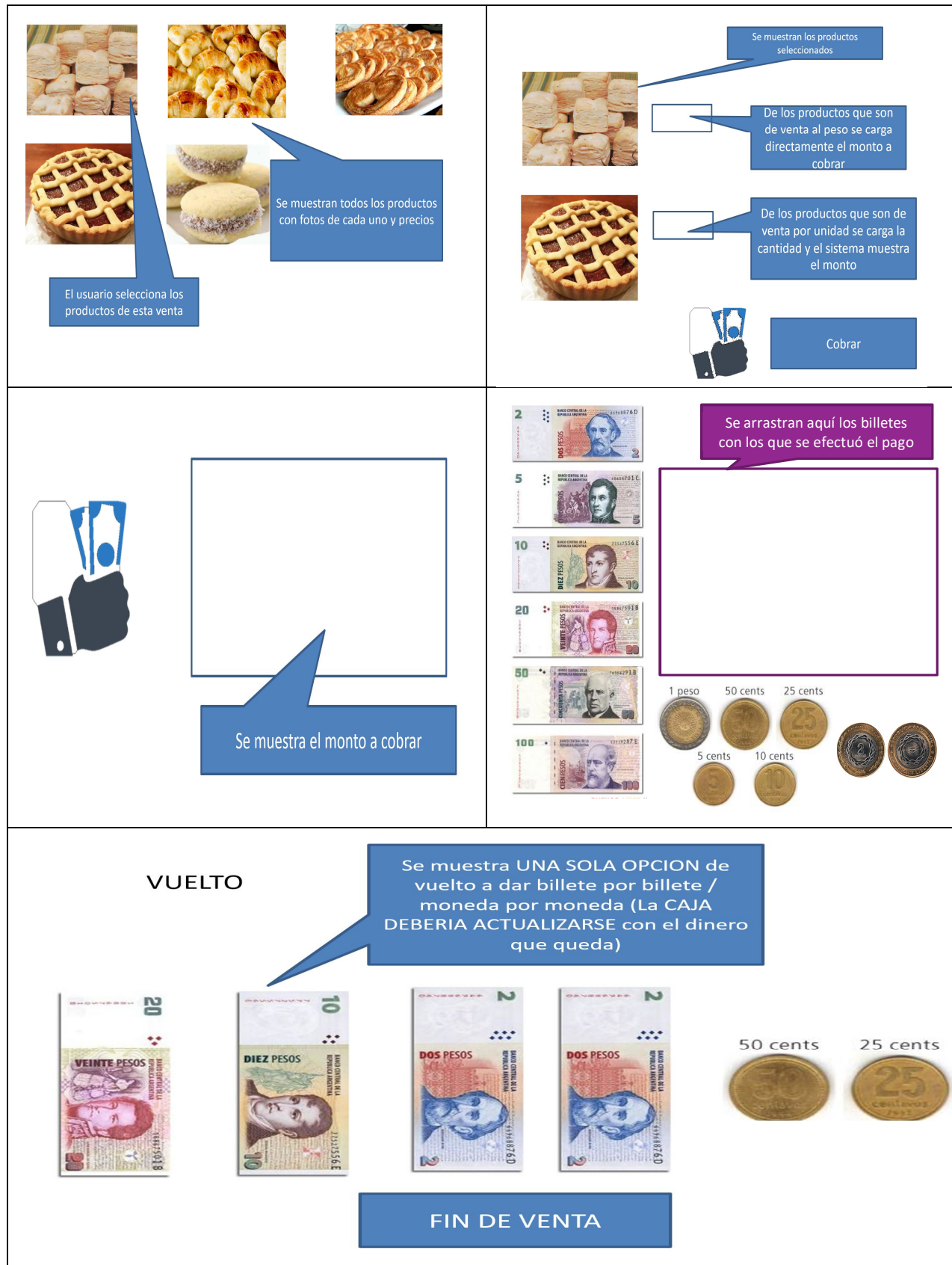
En este contexto, la panadería está atendida por este grupo de personas con capacidades especiales y la intención es desarrollar un producto de software que asista a las personas en el proceso de venta y cobro de los productos que la panadería vende.

Toda la interacción con el producto debe ser basada en imágenes, y muy simple para lo cual, se presentan a continuación una serie de prototipos que ayudarán a visualizar lo que se pretende construir.

El producto de software esencialmente realizará las siguientes funcionalidades. En este caso haremos foco en el desarrollo del caso de uso que está remarcada, el cual se presenta a continuación:



Prototipos Asociados a la Funcionalidad de Registrar Cobro



Descripción con resumen esencial del Caso de Uso 1. Registrar Venta

Nombre del Caso de uso: Registrar Cobro		Nro. de Orden: 1
Actor Principal: Vendedor		Actor Secundario: no aplica
Objetivo: registrar la venta y cobro de productos de la panadería.		
Flujo Básico		
1. Vendedor: selecciona la opción “Registrar Cobro”		
2. Sistema: muestra imágenes de todos los productos disponibles para la venta con sus precios y solicita se seleccionen los productos que se incluirán en la venta.		
3. Vendedor: selecciona cada uno de los productos.		
4. Sistema: para cada producto de venta por unidad solicita se ingrese la cantidad que se desea considerar en la venta.		
5. Vendedor: ingresa la cantidad de cada producto.		
6. Sistema: calcula para cada producto en función de la cantidad ingresada el precio (precio del producto * cantidad). Muestra el total de la venta y las imágenes de billetes y monedas para que seleccione el dinero que recibirá para el cobro.		
7. Vendedor: selecciona cada uno de los billetes y/o monedas que le entregan para pagar.		
8. Sistema: Calcula el vuelto (dinero entregado – importe total de venta) y si corresponde entregar dinero como vuelto, muestra imágenes de cada uno de los billetes y/o monedas que corresponde entregar.		
9. Sistema: Actualiza el dinero existente en la caja. Fin del caso de uso.		
Flujos Alternativos		
A1: Los productos a cobrar son por peso, debe ingresa el precio.		
A2: Algunos de los productos a cobrar son por peso, debe ingresa el precio y otros por cantidad, debe ingresar unidades y calcular el precio.		
A3: No hay en la caja billetes y/o monedas para conformar el vuelto.		
Observaciones:		
1. El Vendedor puede cancelar la ejecución del caso de uso en cualquier momento.		

Consignas:

Análisis del Caso a Implementar:

- Estudiar el caso de estudio y de la descripción del caso de uso que se debe implementar.
- Identificar las clases que se consideran necesarias para la implementación del escenario del caso de uso descrito.
- Crear un diagrama de clases, con las clases necesarias para implementar escenario del caso de uso.
- Crear el prototipo para el caso de uso, esto facilitará la comprensión del escenario del caso de uso y guiará la implementación.

Implementación y Diseño:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *"Instructivo de instalación de Java JDK y NetBeans IDE."*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *"Instructivo de creación de un nuevo proyecto en NetBeans IDE"*.
- Crear cada una de las clases según lo especificado en el diagrama de clases planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método get() y método set()], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (public o private), el valor de retorno y los parámetros necesarios.
- Implementar las clases de fabricación pura que se consideren necesarias para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *"Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE."*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el "Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub".

Ejercicio 6: Caso Práctico – Consultorio Odontológico

Presentación del Caso de Estudio

En este consultorio odontológico trabajan varios profesionales que brindan sus servicios. Cuando un paciente necesita atención, debe solicitar un turno previamente. No se atiende a pacientes que no tienen turno. El odontólogo para el que el paciente solicita el turno es el que lo va a atender. Cada Odontólogo tiene una agenda con los días y horarios en los que puede atender, que se crea mensualmente en función de la disponibilidad que el odontólogo informa; con turnos de 30 minutos de duración. Esta agenda genérica representa los días y horarios de atención que tiene disponible ese odontólogo en términos generales y la duración de su consulta. Esta información se tomará como base para crear la agenda cada mes, considerando para cada mes los días y/u horarios que en ese mes no podrá atender.

Cuando el paciente llama por teléfono, se le pregunta el motivo de la consulta y en función de eso se le asigna uno o más turnos. Por ejemplo, si lo que debe hacerse es un tratamiento de conducto, se le asignan dos turnos de media hora, consecutivos.

Descripción con resumen esencial del Caso de Uso 1. Generar Agenda Mensual de Odontólogo

Nombre del Caso de uso: Generar Agenda Mensual de Odontólogo		Nro. de Orden: 1
Actor Principal: Secretaria del Consultorio (SC)		Actor Secundario: no aplica
Objetivo: generar la agenda para un odontólogo, de un mes en particular, así podrán registrarse los turnos de los pacientes.		
Flujo Básico		
1. SC: selecciona la opción “ Generar Agenda Mensual de Odontólogo ”		
2. Sistema: busca y muestra los odontólogos registrados y solicita se seleccione el odontólogo al que le generará la agenda mensual.		
3. SC: selecciona odontólogo.		
4. Sistema: busca y muestra para el odontólogo seleccionado la definición de horarios para agenda vigente y solicita seleccione el mes/ año para el cual generará la agenda.		
5. SC: selecciona mes/año.		
6. Sistema: genera la agenda para el mes, teniendo en cuenta la cantidad de días hábiles del mes y los días y horarios de inicio y fin de atención definidos para el odontólogo, como así también la duración de cada turno y si hay horarios de intervalo en los que no se debe asignar turnos. (Ver observación 2)		
7. Sistema: muestra la agenda generada y consulta si desea realizar algún cambio en los turnos generados. Solicita confirmación para su registración.		
8. SC: confirma la registración de la agenda sin modificaciones.		
9. Sistema: registra la agenda generada con todos los turnos en estado “Disponible”. Fin del Caso de uso.		
Flujos Alternativos		
A1: No hay definición de horarios para agenda para el odontólogo seleccionado.		
A2: SC no confirma la registración de la agenda.		
A3: SC realizada cambios en la agenda (quita o agrega turnos).		
A4: ..		
Observaciones:		
1. La SC puede cancelar la ejecución del caso de uso en cualquier momento.		
2. Horario de intervalo: se refiere a períodos de tiempo en los que no se debe asignar turnos, por ejemplo, para el almuerzo.		

Consignas:

Análisis del Caso a Implementar:

- Estudiar el caso de estudio y de la descripción del caso de uso que se debe implementar.
- Identificar las clases que se consideran necesarias para la implementación del escenario del caso de uso descrito.
- Crear un diagrama de clases, con las clases necesarias para implementar escenario del caso de uso.
- Crear el prototipo para el caso de uso, esto facilitará la comprensión del escenario del caso de uso y guiará la implementación.

Implementación y Diseño:

- Crear un nuevo proyecto en NetBeans. Si aún no está instalado, las instrucciones de instalación y configuración están en el *"Instructivo de instalación de Java JDK y NetBeans IDE."*
- Si se requieren instrucciones acerca de la creación de un nuevo proyecto, consultar el *"Instructivo de creación de un nuevo proyecto en NetBeans IDE"*.
- Crear cada una de las clases según lo especificado en el diagrama de clases planteado. Para cada clase deberás:
 - Implementar sus atributos con su tipo de datos correspondiente.
 - Definir un constructor para cada clase.
 - Crear los métodos de acceso [método get() y método set()], para poder obtener y/o asignar el valor de los atributos de una clase.
- Considerando la funcionalidad del caso de uso especificada en su descripción, implementar los métodos necesarios para que los objetos de las clases puedan realizar el escenario modelado. Considerar la *signatura completa* del método, es decir: el modificador de acceso (public o private), el valor de retorno y los parámetros necesarios.
- Implementar las clases de fabricación pura que se consideren necesarias para permitir que el usuario pueda ejecutar la funcionalidad del escenario modelado del caso de uso.
- Diseñar la interfaz gráfica, utilizando la vista de diseño que brinda NetBeans. Para ayuda sobre esto, está disponible el *"Instructivo de Diseño de Interfaces Visuales con Java Swing en NetBeans IDE."*
- Ejecutar el proyecto y verificar la implementación realizada.
- Una propuesta de solución está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores>. Para obtener asistencia sobre cómo descargar los archivos de este repositorio, está disponible el "Instructivo de descarga de proyecto del repositorio público de 111mil en GitHub".

Fuentes de Información

- Todos los casos de estudio planteados, son elaboración del equipo de Formadores que preparó el material. (Meles, Judith /Robles Joaquín / Fey Candelaria).