

Consignas:

Análisis del Caso a Implementar:

- Estudiar el caso de estudio y la descripción del caso de uso con su prototipo, para el escenario que se debe implementar.
- Identificar las clases que se consideran necesarias para la implementación del escenario del caso de uso descripto.
- Crear una vista de clases, con las clases necesarias para implementar escenario del caso de uso, utilizando un diagrama de clases.
- Modelar la realización del escenario del caso de uso utilizando un diagrama de secuencia

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.
- Crear la base de datos, utilizando la herramienta MySQL Workbench. Para más información ver el instructivo: **Crear una base de datos en MySQL Workbench**, que está disponible en el repositorio, en la siguiente dirección: <https://github.com/111milprogramadores/bd-instructivos>.
- Crear los archivos de mapeo (hbm) para cada una de las clases del Modelo de Dominio, según lo explicado en el apunte teórico de Base de Datos, en la sección **El archivo de mapeo XML**.
- Configurar la conexión de la aplicación con la base de datos creada utilizando el archivo XML **hibernate.cfg.xml**, según lo explicado en el apunte teórico de Base de Datos, en la sección **El archivo de configuración**.
- Implementar la capa de acceso a datos, utilizando la herramienta Hibernate. Para más información consultar en el apunte teórico de Base de Datos en la sección **El archivo de configuración**.
- 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.
- Publicar el proyecto en un repositorio público en GitHub, según los lineamientos del especificados en el instructivo, disponible en: https://docs.google.com/document/d/16aErk2jTjZ5C_Vvt2nKWP4l-aVpi46sNWCm-EMKKCsA/edit

Pizzería

Descripción del Caso

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 y el tipo (a la piedra, a la parrilla, de molde) de la pizza. Los clientes realizan sus pedidos en el mostrador.

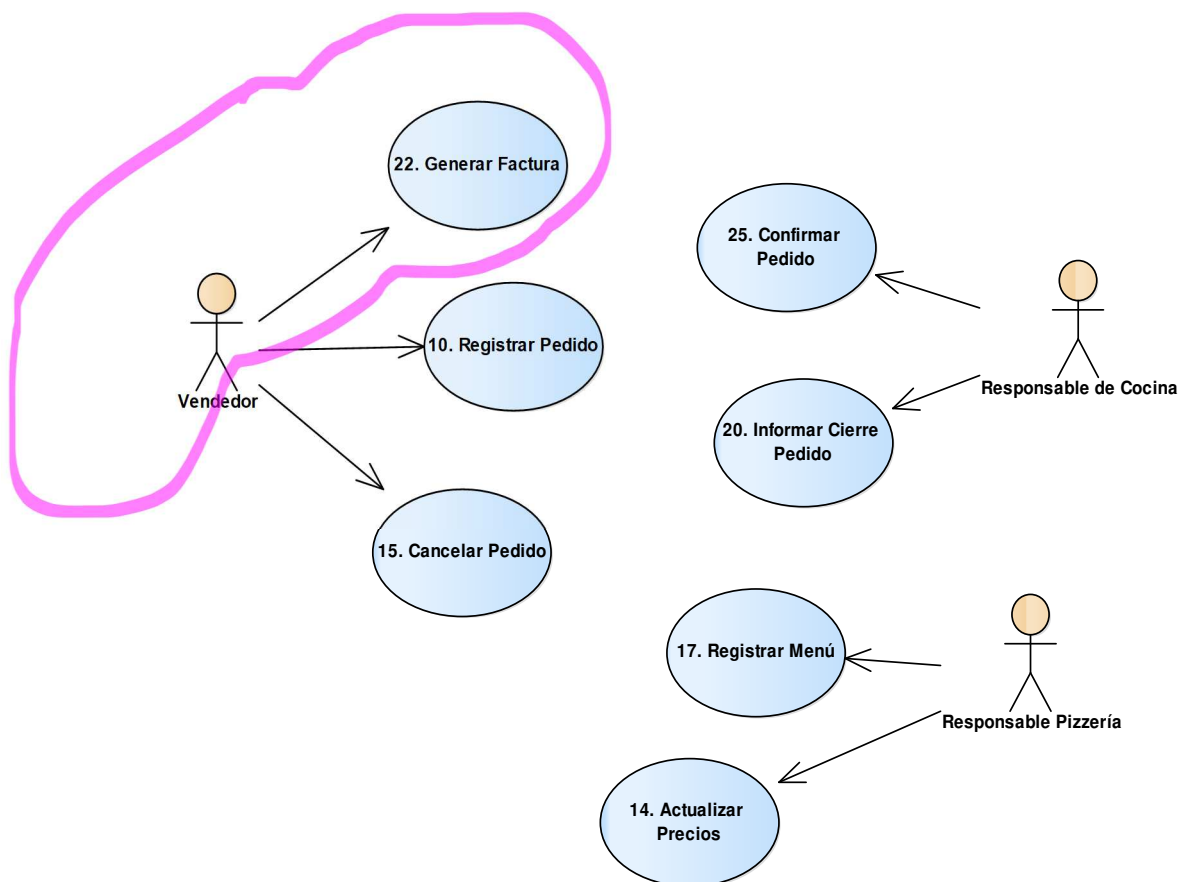
El pedido debe contener el nombre del Cliente, para llamarlo cuando su pedido está listo; la cantidad de pizzas, el tamaño, la variedad, la fecha del pedido, la hora en la que el pedido debe entregarse y la demora estimada informada al cliente.

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.

El dueño de la pizzería ha manifestado la necesidad de acceder al menos a la siguiente información:

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

Vista de casos de uso esencial



Prototipo de la Interfaz de Usuario asociados a la Funcionalidad de Registrar Cobro

Generar Factura

Seleccione un pedido

Número	Fecha	Hora	Cliente
1	11/10/2016	21:30	Maria Cortez
2	11/10/2016	21:45	Carmen Lopez
3	11/10/2016	21:50	Juan Ferreyra


Datos del Pedido Seleccionado

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

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

Monto Total del Pedido: \$780

Confirmar Cancelar



Confirmación

¿Desea generar la factura?

Si No

Factura Generada

Número de Factura : 0006451

Imprimir

Aceptar Cancelar

Factura N° 0006451

Fecha de Emisión: 11/10/2016

Cliente: Maria Cortez

Hora de Emisión: 22:15

Detalle Factura

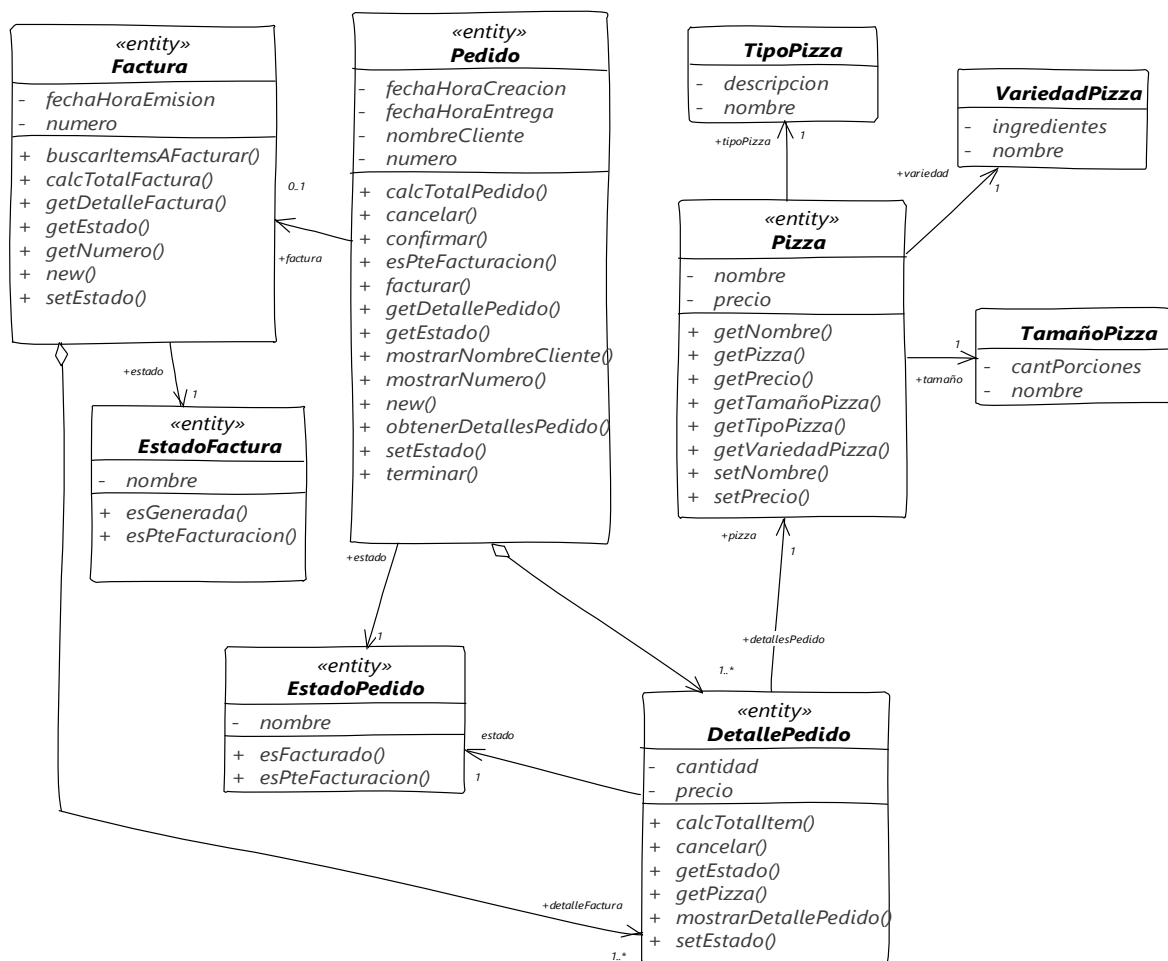
Cantidad	Pizza	Tipo	Variedad	Tamaño	Precio	SubTotal
3	Napolitana Normal	Parrilla	Napolitana	8	\$125	\$375
1	Especial Grande	Piedra	Especial	12	\$165	\$165
2	Mozzarella Mediana	Parrilla	Mozzarella	10	\$120	\$240

Monto Total : \$780

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

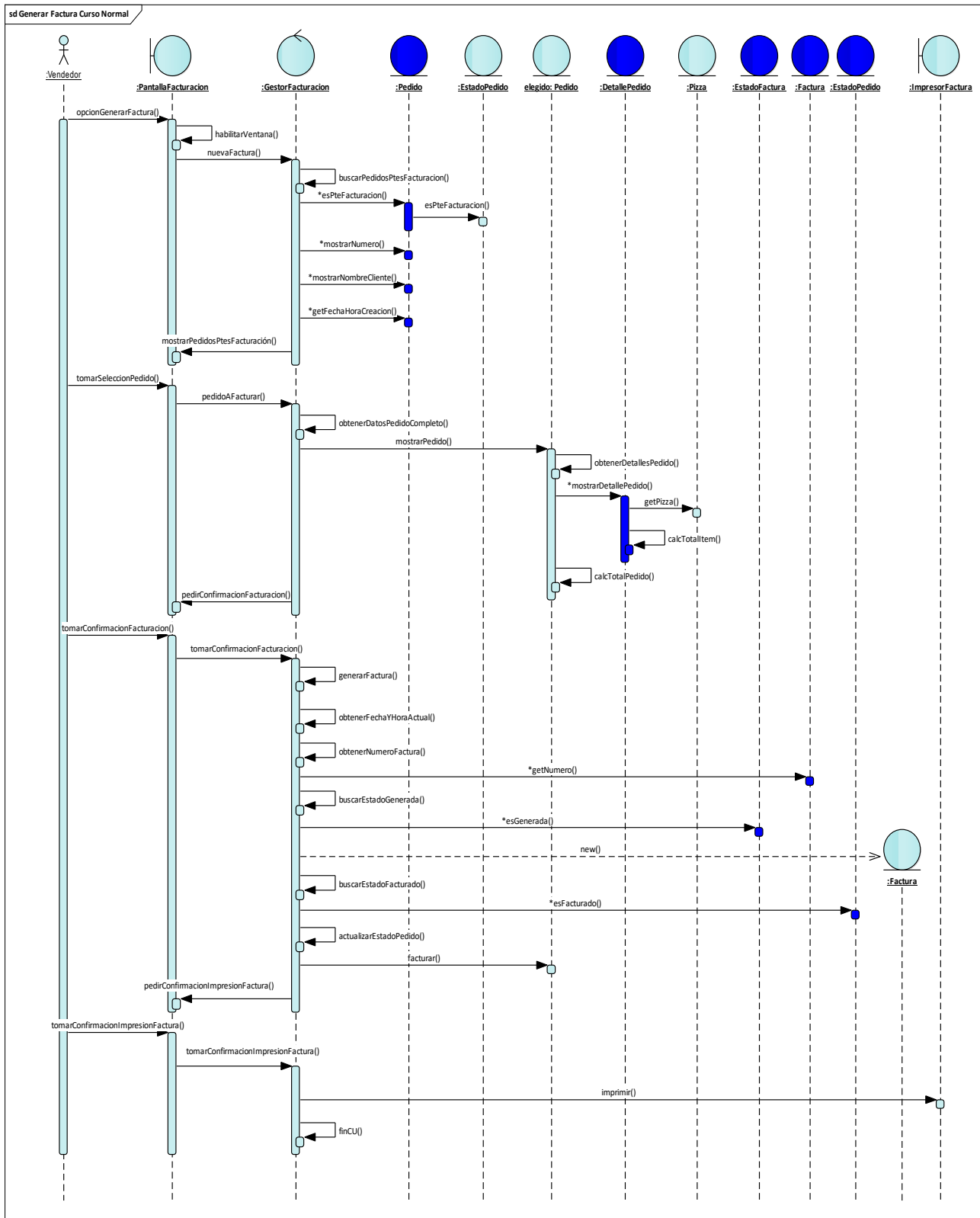
Nombre del Caso de uso: Generar Factura		Nro. de Orden: 2
Actor Principal: Vendedor		Actor Secundario: no aplica
Objetivo: generar la factura asociada a un pedido determinado.		
Flujo Básico		
1. Vendedor: selecciona la opción “Generar Factura”.		
2. Sistema: busca y muestra la lista de pedidos que estén 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.		

Modelo de Dominio

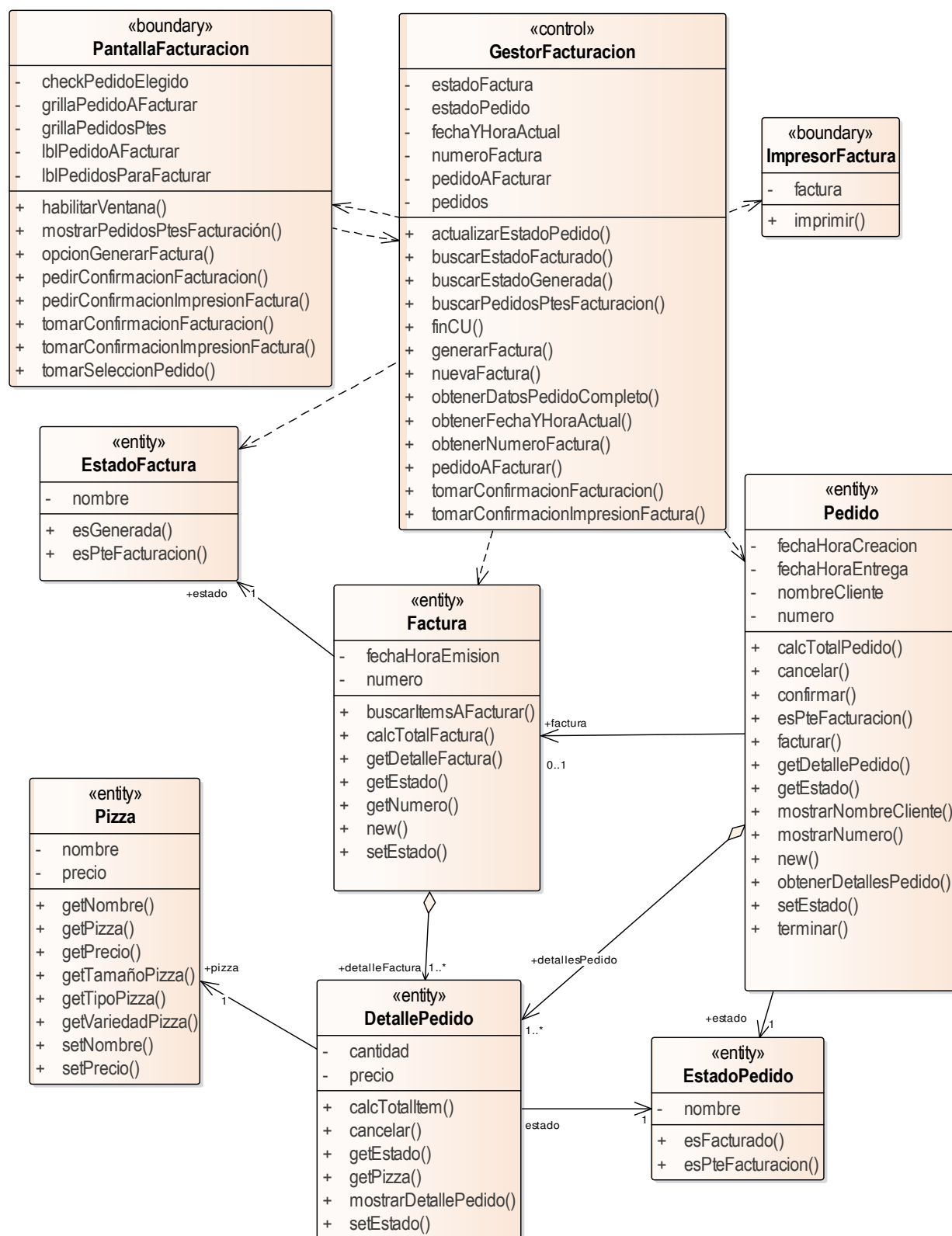


Realización del escenario del flujo básico del caso de uso Generar Factura

Vista Dinámica



Vista de Estructura de Clases de Análisis para el caso de uso Generar Factura



Modelo de Entidad Relación que modela las tablas para persistir en la base de datos

