



UNIVERSITAT OBERTA DE CATALUNYA  
Ingeniería informática

PROYECTO FIN DE CARRERA

SISTEMA DE INFORMATIZACIÓN  
INTEGRAL DE LA GESTIÓN  
FARMACÉUTICA

Autor: **Aimen Ben Attahellah**  
Consultor: **Juan Carlos González Martín**

Junio/2013

## Resumen

El presente proyecto fin de carrera tiene como objetivo informatizar todo el sistema de gestión de farmacias con el fin de dar un servicio a los farmacéuticos, más independiente, más eficaz y más sencillo.

Independiente, porque la aplicación podrá ser adaptada y personalizada para permitir su utilización en cualquier tipo de farmacia de cualquier especialidad.

Eficaz, porque el almacenamiento electrónico permite localizar datos específicos al momento sin necesidad de las tradicionales carpetas que puedan llevar a confundir o traslapar los datos manipulados.

Sencillo, porque el sistema ofrece una interfaz amigable y una navegación sencilla, guiando en todo momento al usuario y ofreciendo la posibilidad de explicar las funcionalidades integradas en el sistema.

El sistema permite al farmacéutico de realizar todas las tareas diarias que se realizan en todas las farmacias, como la adquisición y venta de medicamentos o artículos en general, gestión de stock, de clientes y proveedores, siendo éstas las tareas más frecuentes y demandadas en este sector.

Cabe destacar, que la toma de requisitos de este proyecto está basada sobre peticiones de un cliente, farmacéutico de profesión, con lo cual se trata de un caso de estudio real. Debido a esto, se ha tenido que seguir un modelo y una gestión de riesgos adaptados por su puesto a este caso real.

**Palabras clave:** Artículo, stock, venta, clientes, proveedores, caja, modelo clásico con prototipado, principios de metodología ágil, interfaz amigable de uso fácil y eficaz, C#.NET, ADO.NET.

## Índice general

1.	Introducción .....	1
1.1.	Justificación.....	1
1.2.	Objetivo .....	1
2.	Gestión de proyecto .....	2
2.1.	Modelo y ciclo de vida .....	2
2.2.	Metodología.....	3
2.3.	Análisis de riesgos .....	3
2.4.	Planificación y seguimiento .....	4
3.	Análisis del sistema .....	7
3.1.	Análisis de requisitos .....	7
3.1.1.	No funcionales.....	7
3.1.1.	Funcionales .....	8
3.2.	Casos de uso .....	9
3.2.1.	Plantillas de casos de uso.....	10
3.2.2.	Diagramas de casos de uso .....	17
3.3.	Diagrama de clases .....	22
4.	Diseño del sistema .....	22
4.1.	Arquitectura del sistema .....	22
4.1.1.	Arquitectura física del sistema .....	23
4.1.1.	Arquitectura software del sistema .....	23
4.2.	Diagramas de secuencia.....	24
4.2.1.	Autenticación .....	24
4.2.2.	Gestión de clientes .....	25
4.2.3.	Gestión de pedidos .....	27
4.2.4.	Inventario .....	29
4.2.5.	Gestión de ventas.....	30
4.2.6.	Gestión de Caja.....	33
4.3.	Diseño de base de datos.....	37
4.3.1.	Diagrama de entidad relación .....	37
4.3.2.	Definición de tablas.....	37
4.4.	Diseño de interfaz de usuario .....	40

5.	Implementación.....	43
5.1.	Jerarquía del proyecto.....	44
5.2.	Algunos aspectos de implementación.....	45
5.2.1.	Auto compleción .....	45
5.2.2.	Llamadas a procedimientos almacenados.....	46
5.2.3.	Mensajes de error y afirmación .....	46
5.2.4.	Generación de informes .....	46
5.1.	Algunos módulos de la aplicación .....	47
5.1.1.	Módulo de administración - Gestión de usuarios .....	48
5.1.2.	Módulo de ventas – Nueva venta .....	49
5.1.3.	Módulo de ventas – Histórico de ventas .....	50
5.1.4.	Módulo de stock – Modificar pedido .....	51
5.1.5.	Módulo de caja – Caja registradora.....	52
6.	Conclusiones .....	53
7.	Líneas de futuro .....	53
8.	Bibliografía.....	54

## Índice de ilustraciones

Ilustración 1-Ciclo de vida clásico con prototipado.....	2
Ilustración 2 - Planificación inicial – Diagrama Gantt .....	5
Ilustración 3 - Consumido real - Diagrama Gantt .....	6
Ilustración 4 - Diagrama UC – Sistema .....	17
Ilustración 5 - Diagrama UC – Subsistemas.....	17
Ilustración 6 - Diagrama UC – gestión administrativa .....	18
Ilustración 7 - Diagrama UC – Autenticación .....	18
Ilustración 8 - Diagrama UC - Gestión de ventas .....	19
Ilustración 9 - Diagrama UC - Caja .....	19
Ilustración 10 - Diagrama UC - Gestión de socios .....	20
Ilustración 11 - Diagrama UC - Gestión de stock .....	21
Ilustración 12 - Diagrama de clases .....	22
Ilustración 13 – Arquitectura del sistema – cliente servidor .....	23
Ilustración 14 – Diagrama de secuencia - autenticación .....	24
Ilustración 15 – Diagrama de secuencia – alta cliente.....	25
Ilustración 16 – Diagrama de secuencia – consultar cliente.....	25
Ilustración 17 – Diagrama de secuencia – modificar cliente.....	26
Ilustración 18 – Diagrama de secuencia – suprimir cliente.....	26
Ilustración 19 – Diagrama de secuencia – alta pedido.....	27
Ilustración 20 – Diagrama de secuencia – consultar pedido .....	27
Ilustración 21 – Diagrama de secuencia – modificar pedido.....	28
Ilustración 22 – Diagrama de secuencia – suprimir pedido.....	28
Ilustración 23 – Diagrama de secuencia – consultar inventario .....	29
Ilustración 24 – Diagrama de secuencia – modificar inventario stock .....	29
Ilustración 25 – Diagrama de secuencia – nueva venta .....	30
Ilustración 26 – Diagrama de secuencia – histórico de ventas.....	31
Ilustración 27 – Diagrama de secuencia – presupuesto venta .....	32
Ilustración 28 – Diagrama de secuencia – imprimir presupuesto venta .....	32
Ilustración 29 – Diagrama de secuencia – pago diferido.....	33
Ilustración 30 – Diagrama de secuencia – Consultar caja .....	34
Ilustración 31 – Diagrama de secuencia – Finalizar venta .....	35
Ilustración 32 – Diagrama de secuencia – jornada.....	35
Ilustración 33 – Diagrama de secuencia – detalle de jornada .....	36
Ilustración 34 – Diagrama de entidad relación.....	37
Ilustración 35 – prototipo – nueva venta .....	40
Ilustración 36 – prototipo – recordatorio de ventas .....	41
Ilustración 37 – prototipo – cliente .....	41
Ilustración 38 – prototipo – proveedor.....	42
Ilustración 39 – prototipo – articulo.....	42

Ilustración 40 - Implementación - Esquema de compilación .NET .....	43
Ilustración 41 - Implementación - Esquema ADO.NET.....	44
Ilustración 42- Implementación - Menú .....	44
Ilustración 43 – Implementación - Proyectos C# .....	45
Ilustración 44 – Implementación - Capas .....	45
Ilustración 45 – Pantalla de gestión administrativa.....	48
Ilustración 46 - Pantalla de nueva venta .....	49
Ilustración 47 – Pantalla de histórico de ventas .....	50
Ilustración 48 – Pantalla de modificar pedido.....	51
Ilustración 49 – Página caja registradora .....	52

## Índice de tablas

Tabla 1 - Plantilla UC - Interacción usuario/sistema .....	10
Tabla 2 - Plantilla UC - Crear usuarios.....	10
Tabla 3 - Plantilla UC - Nueva venta.....	11
Tabla 4 - Plantilla UC – Histórico de venta.....	12
Tabla 5 - Plantilla UC - Caja registradora .....	13
Tabla 6 - Plantilla UC – Finalizar venta en caja registradora.....	13
Tabla 7 - Plantilla UC – Consultar jornada de ventas.....	14
Tabla 8 - Plantilla UC – Detalle de jornada de ventas.....	15
Tabla 9 - Plantilla UC – Consulta de pedidos .....	15
Tabla 10 - Plantilla UC –Modificar pedido .....	16
Tabla 11 - Definición tabla - Venta.....	38
Tabla 12 - Definición tabla - Cliente .....	38
Tabla 13 - Definición tabla - Proveedor .....	39
Tabla 14 - Definición tabla - Artículo.....	39
Tabla 15 - Definición tabla – Pedido .....	39
Tabla 16 - Definición tabla - Operador.....	40

## 1. Introducción

En este apartado, se pretende hacer una introducción a este documento, en cuanto a justificación del proyecto, objetivos, método seguido y riesgos principales del proyecto.

### 1.1. Justificación

La realización de este proyecto final de carrera responde a la necesidad de realizar un sistema eficaz y simple que integra todas las funcionalidades básicas y comunes de cualquier farmacia y que al mismo, esté utilizado por cualquier usuario sin que tenga previos conocimientos de informática ni esté familiarizado con los sistemas de información.

La mayoría de los sistemas de información de gestión farmacéutica, ofrecen un importante abanico de funcionalidades donde la mayoría no son de uso común y nunca son usadas en el ámbito de la gestión farmacéutica, con lo cual, se reduce la eficacia del sistema. Otra cuestión muy importante, estos sistemas suelen ser complicados hasta para usuarios familiarizados con aplicaciones informáticas dadas sus navegaciones que suelen ser verticales y profundas. Por último la interfaz gráfica, en la mayoría de los casos, suele tener varios botones y componentes gráficos en general, que lo único que transmiten al usuario es la sensación de no dominar el sistema, lo que puede generar un cierto rechazo a la aplicación.

### 1.2. Objetivo

Vista la justificación, este proyecto da una solución a la mayoría de los sistemas que encontramos actualmente en el mercado, enfocándose en ofrecer un sistema de gestión farmacéutico eficaz, completo y sobre todo amigable y fácil de usar.

En cuanto al aspecto técnico, la elección de la plataforma Microsoft .Net viene dada por ser una plataforma potente que permite construir interfaces con componentes gráficos con mayor rapidez, eficacia y sencillez, cuando se pretende explotar al máxima las propiedades y eventos de dichos componentes y obtener así interfaces amigables siguiendo por supuesto los principios básicos del diseño de interfaces gráficas.

## 2. Gestión de proyecto

### 2.1. Modelo y ciclo de vida

Se ha usado en la elaboración de este sistema, el modelo clásico con prototipado debido al formato de entregas de la asignatura de PFC, a pesar de tener claro que en otra situación, hubiera sido mejor, los modelos evolutivos o incrementales que son más adaptables a requisitos cambiantes como veremos a continuación en el apartado de los riesgos principales del proyecto.

El modelo planteado permitió presentar un prototipo a nuestro cliente, en este caso horizontal desarrollando en parte todas las facetas de la aplicación y basándose sobre los prerequisites que se han anotado con antelación.

Este modelo está compuesto por las siguientes fases:

- Fase de análisis
  - Identificar los requisitos básicos del usuario conjuntamente con el cliente
  - Desarrollar el prototipo horizontal inicial con presentación al cliente
  - Revisar y mejorar el prototipo aplicando los cambios sugeridos por el usuario
- Fase de diseño
- Fase de implementación
- Fase de pruebas de integración

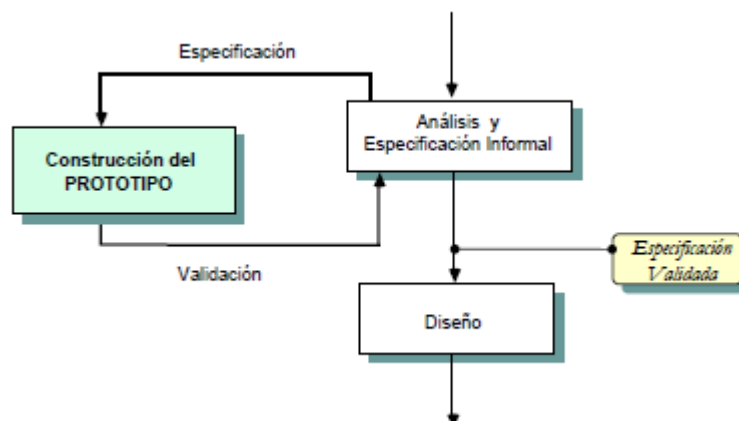


Ilustración 1-Ciclo de vida clásico con prototipado



## 2.2. Metodología

En la metodología se ha intentado basarse sobre algunos principios de las metodologías ágiles gracias a la simplicidad, tolerancia a cambios, cercanía con el cliente y el enfoque más vertical al desarrollo de un software y siendo minimalista hasta cierto punto respecto a la documentación del sistema, lo que se adapta al final a nuestro caso de estudio.

Qué nos ha aportado realmente seguir esta metodología:

- Reflexionar más a cómo ser más efectivo respetando al mismo tiempo la calidad del software
- Construir un software de una manera rápida y eficaz trabajando directamente con el usuario final y percibir rápidamente los cambios para poder aceptarlos o evitarlos eficazmente.
- Finalmente tener una documentación completa y precisa, pero anteponiendo un desarrollo de software que funciona.

## 2.3. Análisis de riesgos

Los riesgos principales que se han en esta fase inicial del proyecto, se presentan de la siguiente forma:

El autor que va a desarrollar el sistema, no tiene conocimientos avanzados de la tecnología que se va a utilizar, y se limitan en lo que se ha aprendido a lo largo de la carrera, y para el desarrollo de un sistema completo desde cero, se necesita un importante conocimiento de la arquitectura .NET como de los lenguajes C#.NET y ADO.NET.

Destaco también, la naturaleza de mi actual trabajo que requiere varios desplazamientos en toda Europa, lo cual nos llevaría al segundo riesgo de no poder respetar una dedicación constante y rigurosa a las tareas y al desarrollo normal del sistema.

Para estos dos riesgos, se pone como plan de acción, realizar un sobreesfuerzo durante la fase de arranque del proyecto, para poder avanzar el máximo posible y poder controlar más los eventuales sobresaltos, y prever vacaciones en el trabajo en caso de estar al borde de una alerta.

El tercer riesgo, es de carácter puramente técnico, y se refiere al uso del modo de conexión desconectado a la base datos, esto puede tener un impacto de rendimiento significativo. Como plan de acción se usarán los procedimientos almacenados para ajustar la balanza y poder mejorar los tiempos de respuesta.

El cuarto riesgo, es a la hora de implantar el sistema, los usuarios tienen conocimientos muy limitados de informática y pueden ser reacios al cambio y con lo cual no aceptar el sistema. El plan de acción es convencer a los farmacéuticos responsables de que el sistema va a mejorar la gestión de su farmacia e incitarles a convencer a todos los usuarios, de que el sistema les va a facilitar el trabajo.

El quinto y último riesgo, es que los farmacéuticos responsables tienen tendencia a cambiar las funcionalidades del sistema y no se ponen de acuerdo en todas las funcionalidades, debido a que no podemos suspender temporalmente el proyecto, como plan de acción paliativo, sería cerrar una primera versión, y prever si es necesario, una nueva versión del sistema una vez lo tengan más claro.

## **2.4. Planificación y seguimiento**

Se muestra abajo, en formato de diagrama de Gantt, la planificación inicial detallando así las actividades. Se ha previsto una tarea de previsión de riesgos, debido a los riesgos detectados y la cual se intentará de no producir salvo si hay sobresaltos, la PPR supone aproximadamente el 7% del propuesto global:

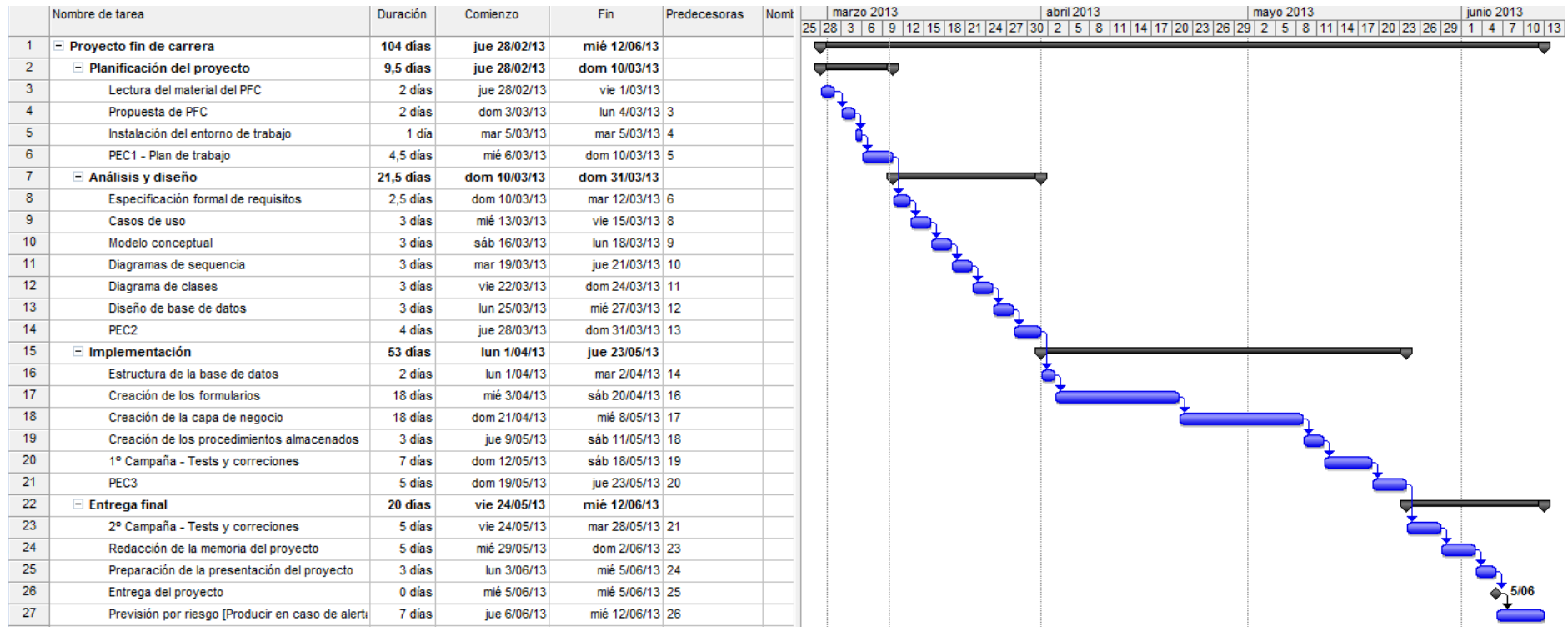


Ilustración 2 - Planificación inicial – Diagrama Gantt

Como se muestra en el consumo real de la ilustración de abajo, la fase de análisis ha sido muy costosa en cuanto a su elaboración y se ha excedido con creces lo que estaba planificado inicialmente sobre todo en lo referente a la toma de requisitos y prototipado. La fase de diseño también ha revelado un consumo mayor que lo planificado por culpa del gran volumen de diagramas de secuencia.

Gracias a una planificación, gestión de riesgos y planes de acción eficaces y a la buena producción en la fase de implementación, se ha conseguido resolver la alerta detectada en la fase de análisis, eso sí, se ha debido producir algunos días guardados para previsión por riesgos, como se puede mostrar en el siguiente diagrama de Gantt:

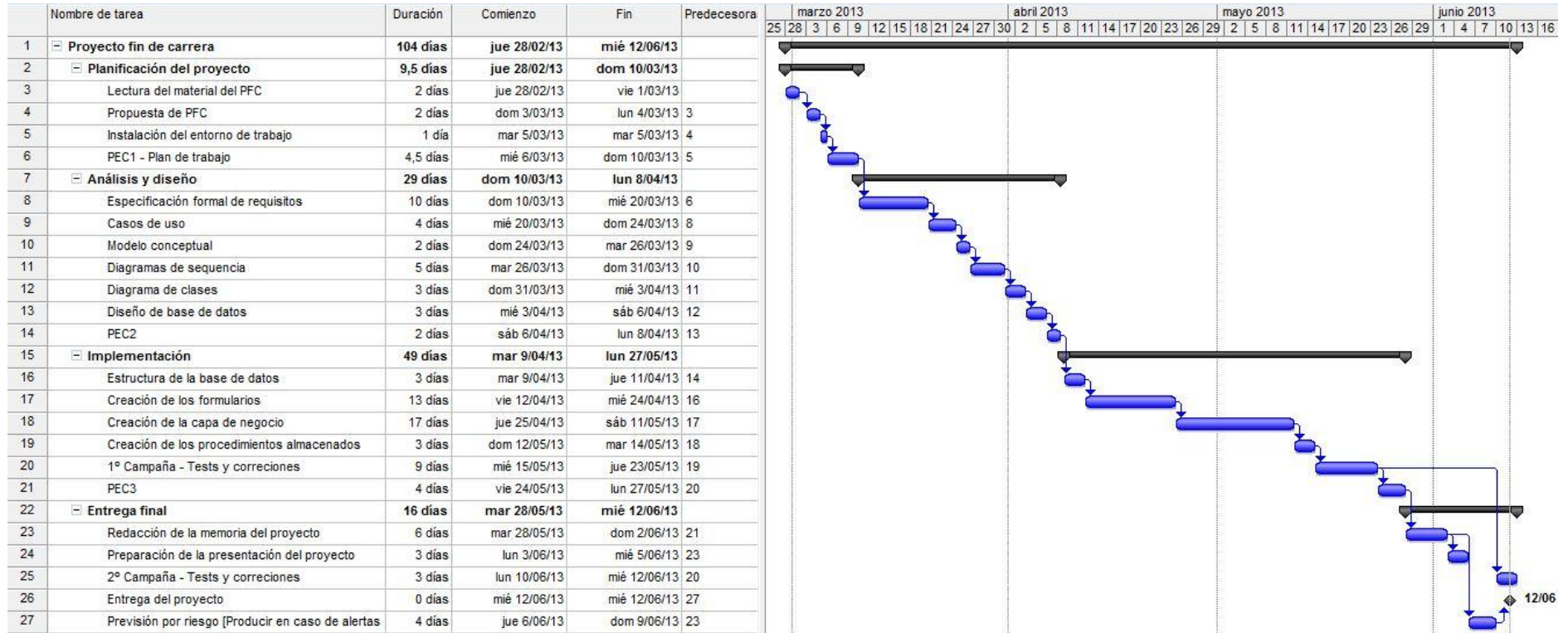


Ilustración 3 - Consumido real - Diagrama Gantt

### 3. Análisis del sistema

En este apartado se pretende explicar las tareas que se han realizado en la fase de análisis inicial del proyecto.

#### 3.1. Análisis de requisitos

Existen dos tipos de requisitos, funcionales y no funcionales:

##### 3.1.1. No funcionales

Definen tanto propiedades como restricciones del sistema, tales como el tiempo de respuesta o fiabilidad...

- **Sistema operativo:** Cualquier sistema operativo de la familia de Microsoft Windows a partir del XP.
- **Procesador / RAM cliente / RAM servidor :** Mínimo 1GB / 512MB / 1GB
- **Almacenamiento cliente / servidor:** 512MB / 5GB como mínimo dependiendo del volumen de negocio de la farmacia
- **Monitor / Tarjeta gráfica:** Resolución mínima de pantalla de 1024 x 768 píxeles
- **Tiempo de respuesta:** El tiempo de respuesta a la hora de guardar los datos no debe de superar los cinco segundos.
- **Uso del sistema:** El deseo general del usuario es la facilidad de uso de la herramienta.

### 3.1.1. Funcionales

Definen los servicios y las funcionalidades que debe proporcionar el sistema.

Los requisitos funcionales son:

- **Autenticación.** Cualquier usuario ya registrado, puede autenticarse y salir del sistema cuando quiera.
  - Iniciar la sesión
  - Cambiar de sesión
  - Salir del sistema
  
- **Administración del sistema - Gestión de Usuarios.** Todos los usuarios deben ser correctamente identificados en el sistema. Para ello, el usuario proporciona al administrador su nombre de usuario, contraseña y tipo de usuario del que se trata (Farmacéutico responsable, Técnico o Administrador/a), tras lo cual el sistema deberá validar estos datos.
  - Alta/consulta/modificación/supresión de usuarios
  
- **Gestión de ventas.** La gestión de ventas es el núcleo del sistema, ofrece la creación de una nueva venta, presupuesto de la venta para poder imprimirlo y finalmente consultar el histórico de ventas sea por fechas, por cliente o ambos.
  - Nueva venta
  - Presupuesto con generación de documentos
  - Ventas realizadas por un cliente y/o en un intervalo de tiempo
  
- **Gestión de caja.** Permite finalizar las ventas creadas en el apartado de “Nueva venta”. Ofrece la realización de los pagos diferidos para que un cliente pueda saldar parte o la totalidad de su crédito. Finalmente visualizar la jornada de ventas realizadas por tipo de pagos: efectivo, crédito, cheque, haber, descuento y diferido y también el detalle de dichas ventas.
  - Caja registradora
  - Pago diferido
  - Jornada de ventas – detalle de ventas

- **Gestión de stock.** Esta macro-funcionalidad, ofrece la posibilidad de crear o manipular los artículos de la farmacia. Inventario y actualizar el stock de artículos ya registrados asociados a un albarán de entrega. Crear, consultar y manipular los pedidos que se han realizado a un proveedor dado.  
Con lo cual clasificamos las funcionalidades de la siguiente forma:
  - Crear/Consultar/Modificar/Suprimir artículo
  - Crear/Consultar/Modificar/Suprimir pedido y sus artículos correspondientes
  - Inventario: Consulta y modificación del stock de artículos
  
- **Socios.** Se refiere con socios a los proveedores y clientes de la farmacia.
  - Proveedores. Esta funcionalidad del sistema puede ser usada por cualquier usuario. Permite agregar un nuevo proveedor o modificar uno existente proporcionando sus datos, y también permite borrarlo del origen de los datos, mediante un borrado lógico.
    - Añadir/Consultar/Modificar/Eliminar
  
  - Clientes. Sigue la misma lógica de gestión de proveedores, pero en este caso para clientes. Un cliente sólo se puede borrar, si ha saldado todos sus créditos.
    - Añadir/Consultar/Modificar/Eliminar

### 3.2. Casos de uso

Los casos de uso representan requisitos funcionales del sistema y definen un conjunto de interacción entre los elementos adyacentes al sistema y el propio sistema desde el punto de vista del usuario.

En este apartado presentaremos los casos de uso más importantes pero no todos, para no exceder el número permitido de páginas. Estos casos de uso serán representados en forma de plantillas y en forma de diagramas.

Para la realización de la especificación de requisitos, se ha adoptado el modelo estándar de ANSI/IEEE Std 830-1998 “IEEE Guide to Software requirements Specifications”

### 3.2.1. Plantillas de casos de uso

Las plantillas permiten describir los casos de uso y la forma en que se interactuar el usuario con el sistema. Esta interacción son acciones realizadas por el usuario y por el sistema tras un suceso provocado por los mismos, y que representaremos como:

	Acciones del usuario
	Acciones del sistema

Tabla 1 - Plantilla UC - Interacción usuario/sistema

#### 3.2.1.1. Crear usuarios

Identificación	
Caso de Uso:	Crear Usuario
Descripción:	El Administrador registra un nuevo usuario en el Sistema
Roles Soportados:	Administrador
Relación	
Hereda:	
Extiende:	
Incluye:	
Proceso / Descripción	
Actor Primario:	Administrador
Actores Secundarios:	
Precondiciones:	Autenticarse y a acceder a la pantalla de gestión administrativa
Post condiciones:	El nuevo usuario es registrado en el sistema
Flujo de eventos	
1. El Administrador introduce los datos del usuario	
2. El Administrador presiona el botón "Validar"	
3. El sistema comprueba la introducción de los datos obligatorios	
4. El sistema guarda el nuevo usuario en el sistema y muestra un mensaje de confirmación por pantalla	
Extensiones síncronas	
Si en el punto 3, los datos obligatorios no están introducidos el sistema muestra un mensaje de error y no avanza al punto 4.	
Extensiones asíncronas	
En cualquier momento el Administrador puede detener la ejecución del proceso pulsando el botón de "Cancelar".	

Tabla 2 - Plantilla UC - Crear usuarios



La creación o alta de clientes, proveedores y artículos es prácticamente similar a la creación de usuarios, salvo que el actor en este caso es el Técnico.

### 3.2.1.2. Nueva venta

<b>Identificación</b>	
Caso de Uso:	Nueva venta
Descripción:	El Técnico de farmacia crea una nueva venta
Roles Soportados:	Técnico de farmacia, farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Técnico de farmacia
Actores Secundarios:	
Precondiciones:	Autenticarse. Los clientes ya están registrados en el sistema
Post condiciones:	La venta está registrada en el sistema
<b>Flujo de eventos</b>	
1. El actor accede a la pantalla de nueva venta pulsando el botón de “Nueva”	
2. El sistema devuelve la pantalla de ventas, y recupera todos los clientes y artículos del sistema	
3. El actor va introduciendo y filtrando los artículos solicitados por el cliente	
4. Mientras que el actor vaya filtrando los artículos, el sistema los muestra por pantalla con su detalle	
5. El Técnico escribe la cantidad de los artículos y luego valida pulsando el botón “Enter” del teclado	
6. El sistema muestra el artículo introducido en la tabla de recapitulación de artículos	
7. El Técnico de farmacia puede introducir opcionalmente el nombre del cliente o dejar el campo vacío, para ello va filtrando mientras vaya escribiendo	
8. El actor presiona el botón “Validar”	
9. El sistema comprueba la existencia de artículos y pide confirmación de los datos introducidos	
7. El actor de farmacia acepta	
8. El sistema guarda la venta en el sistema y envía un mensaje por pantalla de confirmación	
<b>Extensiones síncronas</b>	
<b>Extensiones asíncronas</b>	
En cualquier momento el usuario puede detener la ejecución del proceso pulsando el botón de “Cancelar”.	

Tabla 3 - Plantilla UC - Nueva venta

### 3.2.1.3. Histórico de ventas

<b>Identificación</b>	
Caso de Uso:	Histórico de ventas
Descripción:	El actor consulta todas las ventas de un cliente y/o entre dos fechas
Roles Soportados:	Técnico de farmacia, Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Técnico de farmacia
Actores Secundarios:	
Precondiciones:	Autenticación. El cliente ya está registrado en el sistema y tiene ventas asociadas.
Post condiciones:	Las ventas aparecen en la pantalla
<b>Flujo de eventos</b>	
1. El actor presiona el botón “Histórico”	
2. El sistema devuelve la pantalla pedida cargando el listado de todos los clientes	
3. El actor selecciona un cliente de la lista y/o selecciona las fechas	
4. El actor presiona el botón “Buscar”	
5. El sistema recupera todas las ventas correspondientes al cliente y/o fechas seleccionadas	
<b>Extensiones síncronas</b>	
Si en el punto 4, ninguna venta es devuelta o que el usuario no haya seleccionado ninguna opción de búsqueda, el sistema muestra un mensaje de error y vuelve al punto 2.	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cancelar”.	

Tabla 4 - Plantilla UC – Histórico de venta

### 3.2.1.4. Caja registradora

<b>Identificación</b>	
Caso de Uso:	Consultar caja
Descripción:	Consultar las ventas transmitidas a la caja
Roles Soportados:	Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Farmacéutico
Actores Secundarios:	

Precondiciones:	Autenticación. La venta ha sido creada y está guardada en el sistema
Post condiciones:	
<b>Flujo de eventos</b>	
1. El sistema actualiza cada 5 segundos el listado de todas la ventas realizadas en los últimos 48 horas que aun no están finalizadas	
2. El farmacéutico selecciona una venta de la lista	
3. El sistema muestra el detalle y los artículos de la venta así como los datos del cliente en caso que sea una venta a cliente	
4. El farmacéutico selecciona el o los tipos de pago	
5. El farmacéutico introduce el importe asociado a cada tipo de pago	
<b>Extensiones síncronas</b>	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cancelar”.	

Tabla 5 - Plantilla UC - Caja registradora

### 3.2.1.5. Finalizar venta en caja registradora

<b>Identificación</b>	
Caso de Uso:	Finalizar venta
Descripción:	El farmacéutico finaliza la venta
Roles Soportados:	Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	Consultar caja
<b>Proceso / Descripción</b>	
Actor Primario:	Farmacéutico
Actores Secundarios:	
Precondiciones:	La venta ya ha sido informada por el técnico de farmacia
Post condiciones:	La venta es modificada en el sistema en estado finalizado
<b>Flujo de eventos</b>	
1. El farmacéutico valida la venta	
2. El sistema comprueba que la venta es correcta y muestra por pantalla un mensaje con los datos de la venta	
<b>Extensiones síncronas</b>	
Si en el punto 2, el sistema verifica que los datos de la venta son incorrectos, mostrara un mensaje de error	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cancelar”.	

Tabla 6 - Plantilla UC – Finalizar venta en caja registradora

### 3.2.1.6. Consultar jornada de ventas

<b>Identificación</b>	
Caso de Uso:	Consultar jornada de ventas
Descripción:	El farmacéutico consulta los pagos realizados en un periodo de fechas preseleccionado
Roles Soportados:	Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Farmacéutico
Actores Secundarios:	
Precondiciones:	Autenticación
Post condiciones:	
<b>Flujo de eventos</b>	
1. El farmacéutico presiona el botón “Jornada”	
2. El sistema muestra la pantalla deseada	
3. El farmacéutico selecciona un periodo de fechas de la venta y presiona el botón “Buscar”	
4. El sistema recupera todas las ventas finalizadas comprendidas entre las fechas seleccionadas	
5. El sistema calcula y muestra por pantalla la suma de ventas por tipo de pago	
<b>Extensiones síncronas</b>	
Si en el punto 3, el sistema verifica no hay datos para este periodo, mostrara un mensaje de error explicándolo y se queda en el punto 2.	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cerrar”.	

Tabla 7 - Plantilla UC – Consultar jornada de ventas

### 3.2.1.7. Detalle de jornada de ventas

<b>Identificación</b>	
Caso de Uso:	Detalle jornada de ventas
Descripción:	El farmacéutico desea consultar el detalle de jornada de ventas mostradas en el caso de uso de “Consultar jornada de ventas”
Roles Soportados:	Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	Consultar jornada de ventas
Incluye:	

<b>Proceso / Descripción</b>	
Actor Primario:	Farmacéutico
Actores Secundarios:	
Precondiciones:	Haber ejecutado el caso de uso “Consultar jornada de ventas”
Post condiciones:	
<b>Flujo de eventos</b>	
1. El actor presiona el botón “Detalle de ventas”	
2. El sistema recupera los artículos y clientes asociados a las ventas recuperadas con antelación	
3. El sistema muestra la pantalla con el detalle de ventas y artículos asociados	
<b>Extensiones síncronas</b>	
Si en el punto 2, el sistema verifica que los datos de la venta son incorrectos, mostrara un mensaje de error en la pantalla nueva	
<b>Extensiones asíncronas</b>	

Tabla 8 - Plantilla UC – Detalle de jornada de ventas

### 3.2.1.8. Consultar pedidos

<b>Identificación</b>	
Caso de Uso:	Consultar pedidos
Descripción:	El actor consulta la lista de pedidos por proveedor y/o en un intervalo de tiempo
Roles Soportados:	Técnico de farmacia, Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Técnico de farmacia
Actores Secundarios:	
Precondiciones:	El Pedido ha sido registrado en el sistema
Post condiciones:	
<b>Flujo de eventos</b>	
1. El actor presiona el botón “Modificar pedido”	
2. El sistema recupera la lista de proveedores y artículos y muestra la pantalla deseada	
3.El actor introduce el nombre del proveedor y/o las fechas y pulsa el botón “Buscar”	
4.El sistema muestra la lista de pedidos con sus artículos asociados	
<b>Extensiones síncronas</b>	
Si en el punto 3, ningún pedido es devuelto por el sistema o si ninguna opción de búsqueda está seleccionada, el sistema muestra un mensaje de error	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cerrar”.	

Tabla 9 - Plantilla UC – Consulta de pedidos

### 3.2.1.9. Modificar pedido

<b>Identificación</b>	
Caso de Uso:	Modificar Pedido
Descripción:	El actor modifica un Pedido previamente registrado en el sistema
Roles Soportados:	Técnico de farmacia, Farmacéutico
<b>Relación</b>	
Hereda:	
Extiende:	
Incluye:	
<b>Proceso / Descripción</b>	
Actor Primario:	Técnico de farmacia
Actores Secundarios:	
Precondiciones:	Consultar pedidos. El Pedido ha sido previamente registrado en el sistema
Post condiciones:	El Pedido está modificado en el sistema y el stock actualizado
<b>Flujo de eventos</b>	
1. El sistema muestra el listado de pedidos con sus artículos correspondientes	
2. El actor solicita modificar el pedido mediante supresión de un artículo 3.1, modificación de un artículo 4.1 o añadir un nuevo artículo al albarán 5.1	
3.1. Para ello el actor solicita suprimir un artículo presionando la “X” de la lista de artículos	
3.2.El sistema borra el artículo de la lista de artículos sin llegar a registrarlo en el sistema	
4.1. El actor solicita modificar un artículo presionando el icono de “bolígrafo” de la lista de artículos	
4.2. El sistema muestra por pantalla el detalle del artículo a modificar	
4.3. El actor modifica la fecha de caducidad o la cantidad de artículos asociados al albarán y presiona el botón Modificar	
4.4. El sistema modifica el artículo y actualiza la lista de pedidos con sus artículos sin llegar a registrar en el sistema	
5.1. El actor va filtrando los artículos mientras vaya introduciendo el nombre y seleccionada el artículo deseado	
5.2 El sistema actualiza el precio del artículo mientras se vaya filtrando	
5.3.El actor introduce la cantidad y la fecha de caducidad y presiona el botón “Añadir artículo”	
5.4.El sistema añade el nuevo artículo al albarán sin llegar a registrar en el sistema	
6. El actor valida sus modificaciones pulsando el botón “Validar”	
7. El sistema registra los cambios del albarán	
<b>Extensiones síncronas</b>	
En el punto 6, si se produce un error técnico el sistema muestra un mensaje de error por pantalla	
<b>Extensiones asíncronas</b>	
En cualquier momento el actor puede detener la ejecución del proceso pulsando el botón de “Cerrar”.	

Tabla 10 - Plantilla UC –Modificar pedido

### 3.2.2. Diagramas de casos de uso

El sistema bajo consideración tiene tres tipos de actores: Administrador, farmacéutico y técnico farmacéutico.

En éste diagrama se presenta la vista más externa del sistema estando incluidos simplemente los actores del sistema y el sistema como un todo.

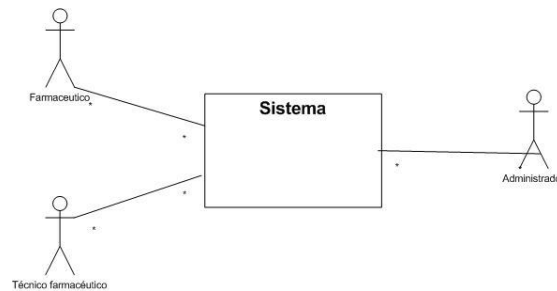


Ilustración 4 - Diagrama UC – Sistema

Este sistema se encuentra formado por cinco subsistemas, el actor técnico farmacéutico es capaz de acceder a tres de estos subsistemas mientras que el actor farmacéutico tiene acceso a los cuatro, así que el actor Administrador accede a un solo subsistema como se muestra en el siguiente diagrama:

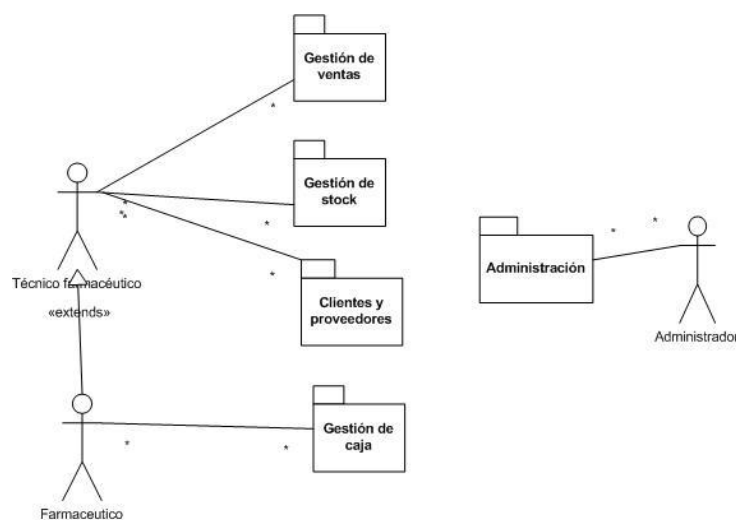


Ilustración 5 - Diagrama UC – Subsistemas

### 3.2.2.1. Subsistema de gestión administrativa

En el subsistema siguiente, se presentan los casos de uso que permiten gestionar un usuario, y darle un rol que será clave para la restricción de accesos.

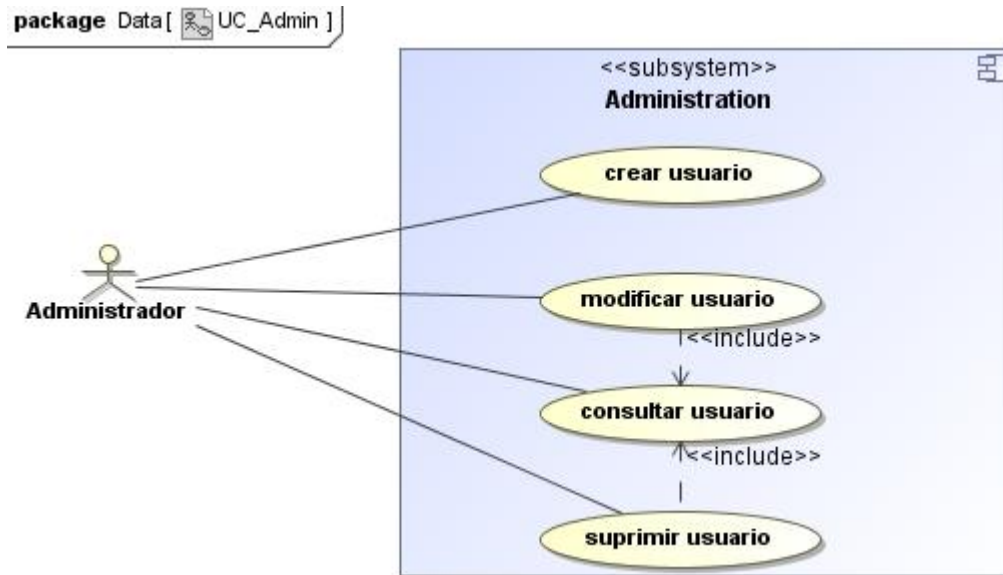


Ilustración 6 - Diagrama UC – gestión administrativa

### 3.2.2.2. Subsistema de autenticación

En el subsistema siguiente, se presentan los casos de uso que permiten a un usuario dado autenticarse/cambiar de usuario o, desconectarse del sistema.

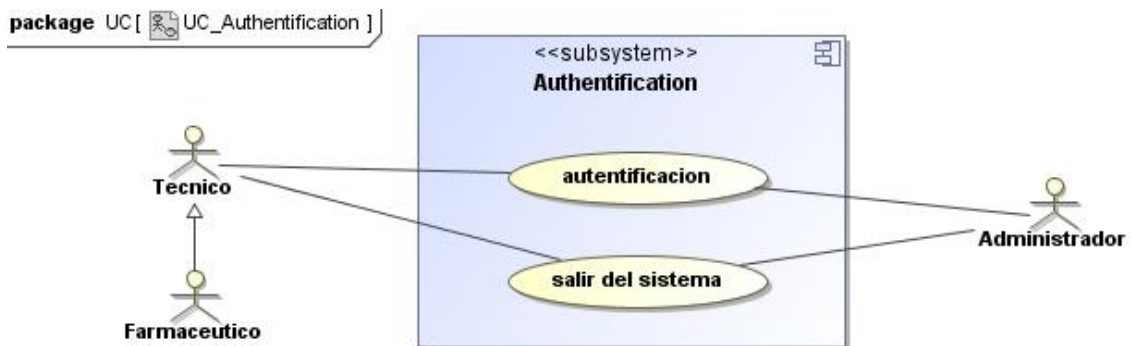


Ilustración 7 - Diagrama UC – Autenticación



### 3.2.2.3. Subsistema de gestión de ventas

Realizados por un técnico o farmacéutico, son los casos de uso más usados por los usuarios y permiten crear nuevas ventas, presupuestos y recordatorio de ventas.

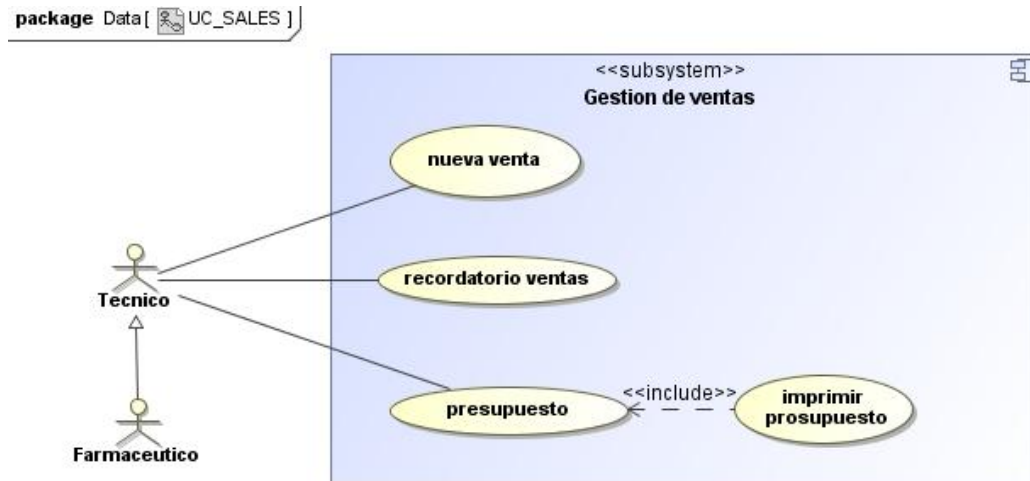


Ilustración 8 - Diagrama UC - Gestión de ventas

### 3.2.2.1. Subsistema de gestión de caja

Finalmente, para la gestión de caja tenemos los casos de uso siguientes:

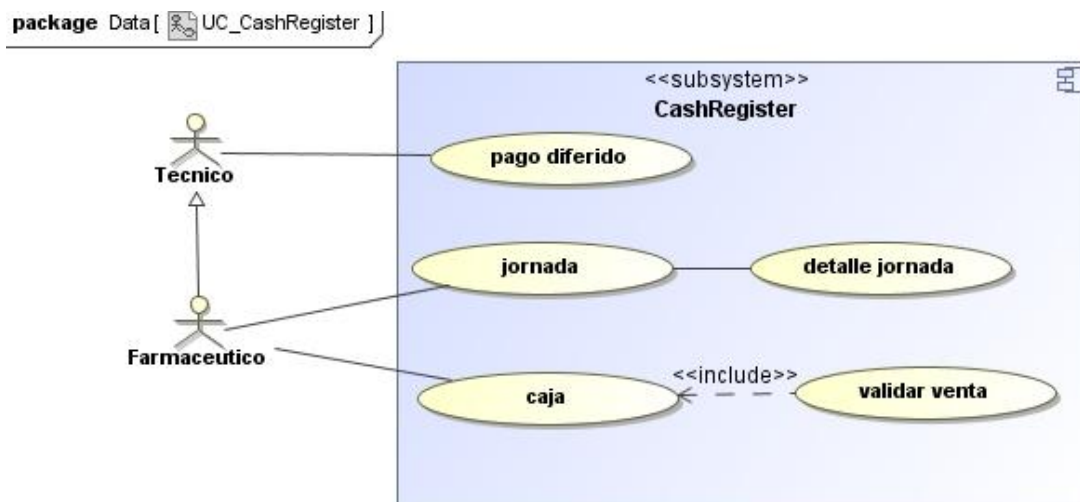


Ilustración 9 - Diagrama UC - Caja

### 3.2.2.2. Subsistema de gestión de socios

Permite crear las personas físicas y morales que interactuarán con la farmacia, y son los clientes consumidores de los artículos, y los proveedores que ofrecen dichos artículos.

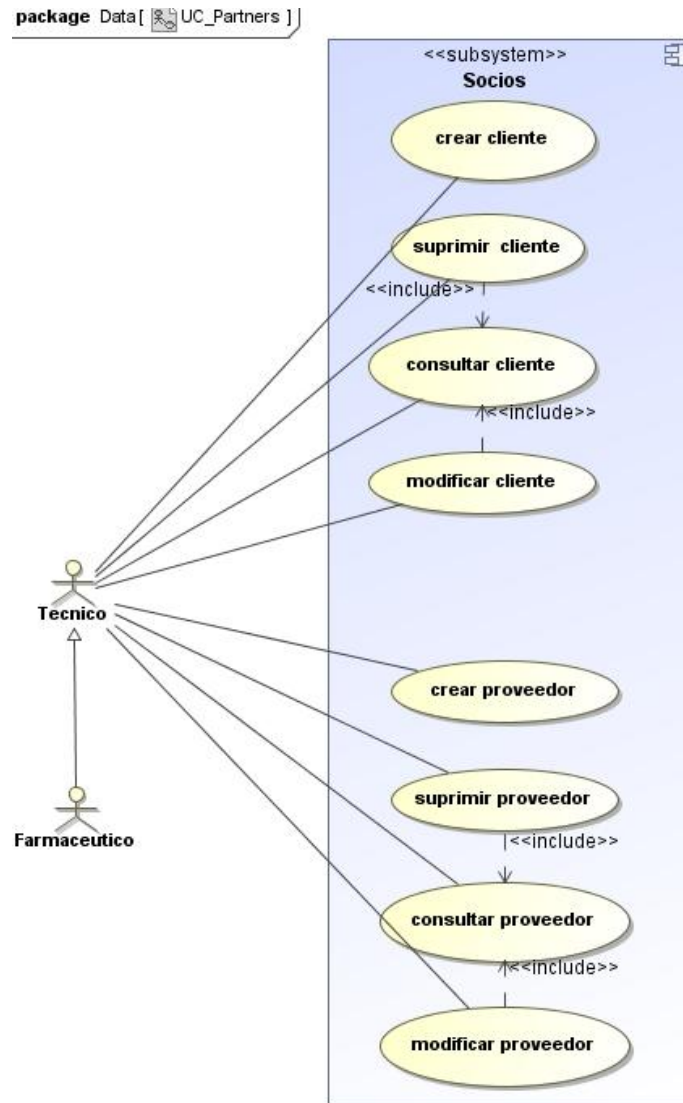


Ilustración 10 - Diagrama UC - Gestión de socios

### 3.2.2.3. Subsistema de gestión de stock

Casos de uso referentes a la gestión de los artículos, pedidos y el stock para el inventario.

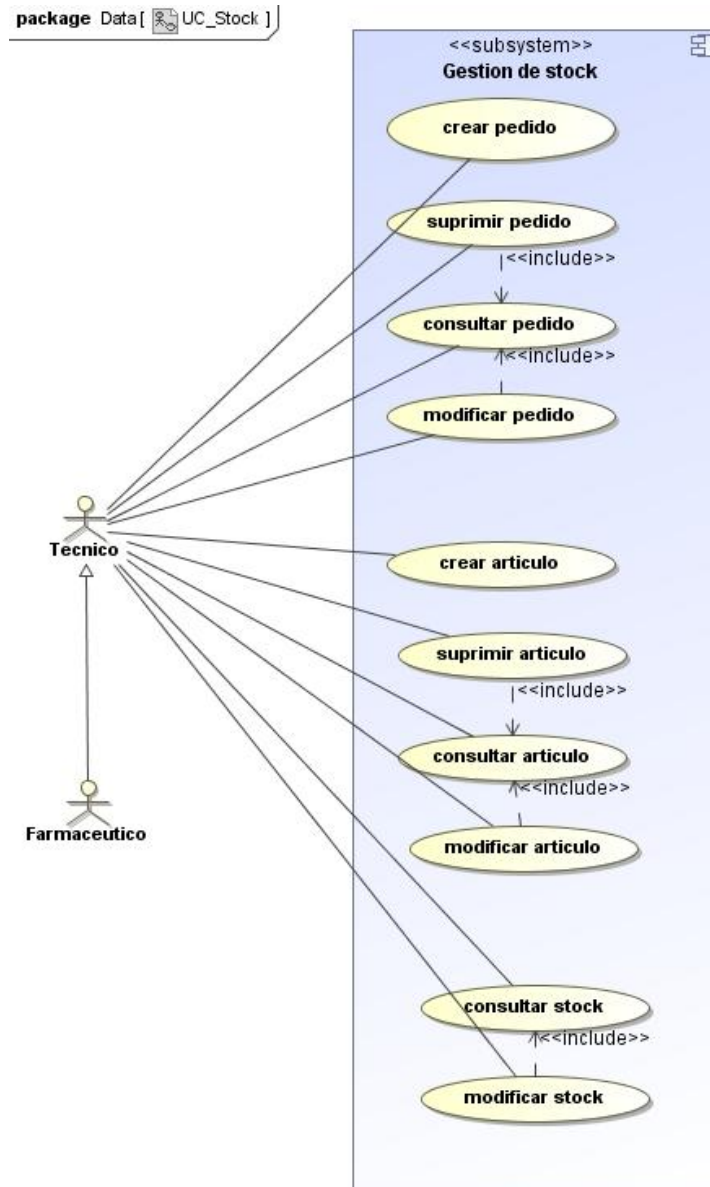


Ilustración 11 - Diagrama UC - Gestión de stock

### 3.3. Diagrama de clases

A continuación se muestra el modelo de objetos por el que se rige el sistema con todas sus relaciones entre clases:



Ilustración 12 - Diagrama de clases

## 4. Diseño del sistema

Concluida la fase del análisis del sistema que responde a la pregunta de “**qué**” trata el problema, nos introduciremos en la fase de diseño que responde a “**cómo**” se resuelve el problema para obtener la solución óptima.

### 4.1. Arquitectura del sistema

Distinguiremos entre una arquitectura física hardware y una arquitectura software:

### 4.1.1. Arquitectura física del sistema

La arquitectura del sistema usada es la de comúnmente conocida cliente / servidor conectados en una red de área local. Esto permite repartir la carga de computación entre clientes y servidor sin que alguno tenga que hacer todo el procesamiento de datos.

La arquitectura física de cliente / servidor adoptada es de dos capas constituida por tres componentes: varios clientes, un servidor y un protocolo de comunicación entre ambos conocido como *middleware*.



Ilustración 13 – Arquitectura del sistema – cliente servidor

### 4.1.1. Arquitectura software del sistema

La arquitectura propuesta está basada en la plataforma .NET de Microsoft y está diseñada en tres capas:

- Capa de presentación: representa los formularios de la aplicación, y permite visualizar por pantalla la interfaz gráfica que permite interactuar con el usuario. Para la construcción del menú, se ha usado un proyecto como componente externo. Esta capa esta interconectada con la capa de negocio.
- Capa de negocio: integrada también en la aplicación, permite realizar los tratamientos de datos y acceder a la capa de persistencia mediante llamadas a procedimientos almacenados.
- Capa de persistencia: representada por un conjunto de procedimientos almacenados previamente definidos en el sistema gestor de base de datos.

## 4.2. Diagramas de secuencia

Los diagramas de secuencia muestran las interacciones entre los objetos en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Dado el elevado número de diagramas de secuencia del sistema, se va a limitar en presentar en esta memoria, únicamente los diagramas de secuencia más relevantes.

### 4.2.1. Autenticación

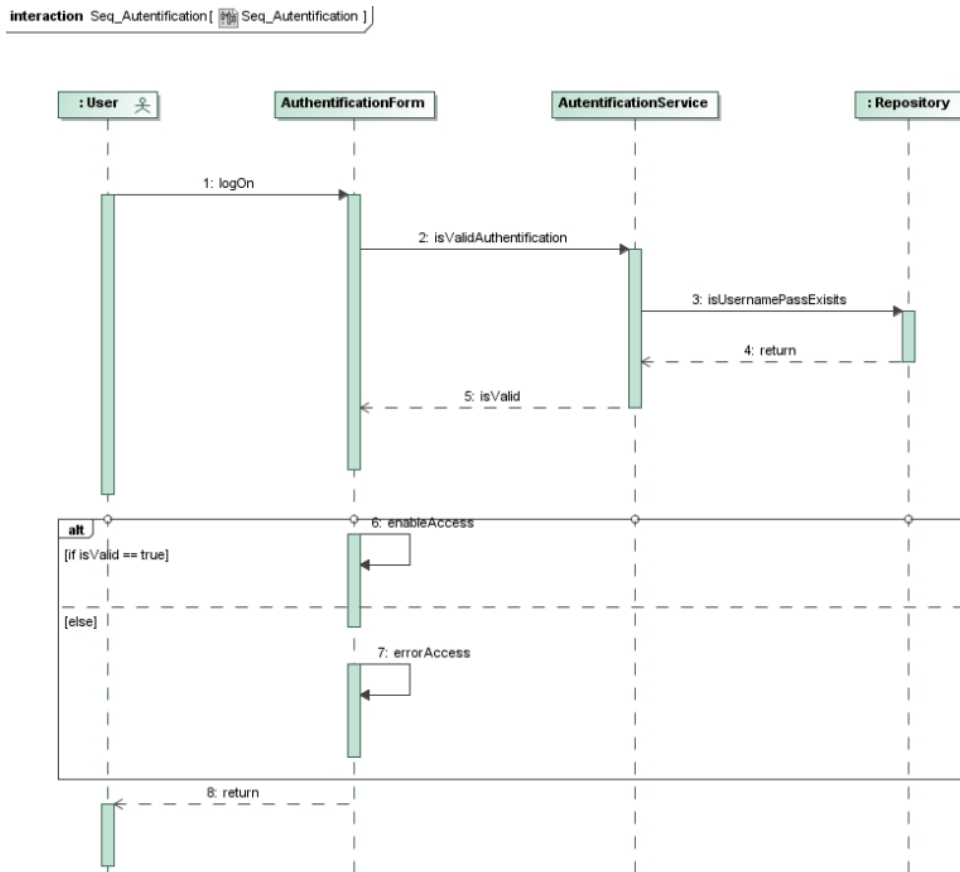


Ilustración 14 – Diagrama de secuencia - autenticación

### 4.2.2. Gestión de clientes

La gestión de clientes, usuarios, proveedores y artículos es bastante similar, por lo cual nos limitaremos en mostrar únicamente la gestión de clientes

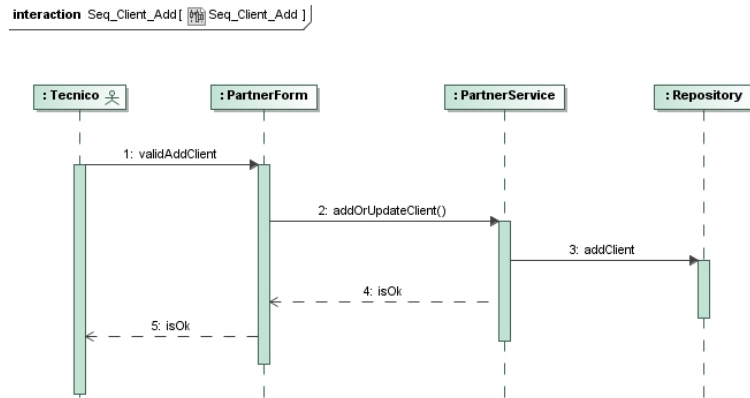


Ilustración 15 – Diagrama de secuencia – alta cliente

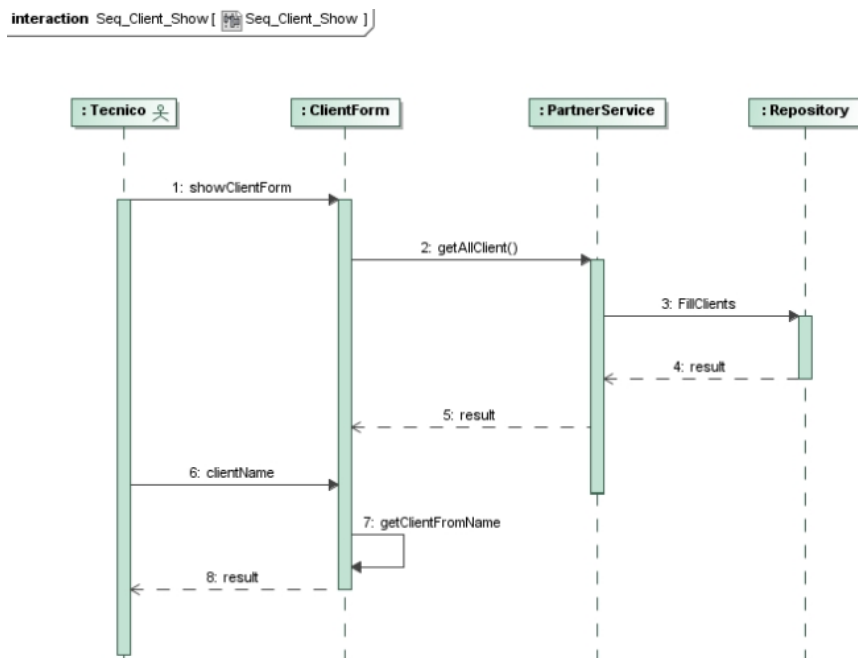


Ilustración 16 – Diagrama de secuencia – consultar cliente

interaction Seq\_Client\_Update [ Seq\_Client\_Update ]

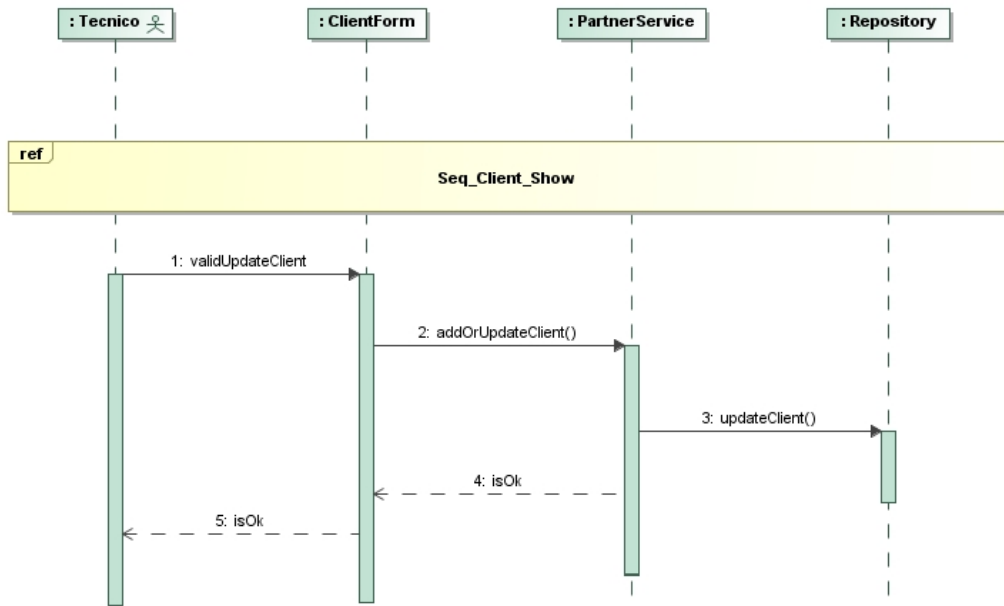


Ilustración 17 – Diagrama de secuencia – modificar cliente

interaction Seq\_Client\_Delete [ Seq\_Client\_Delete ]

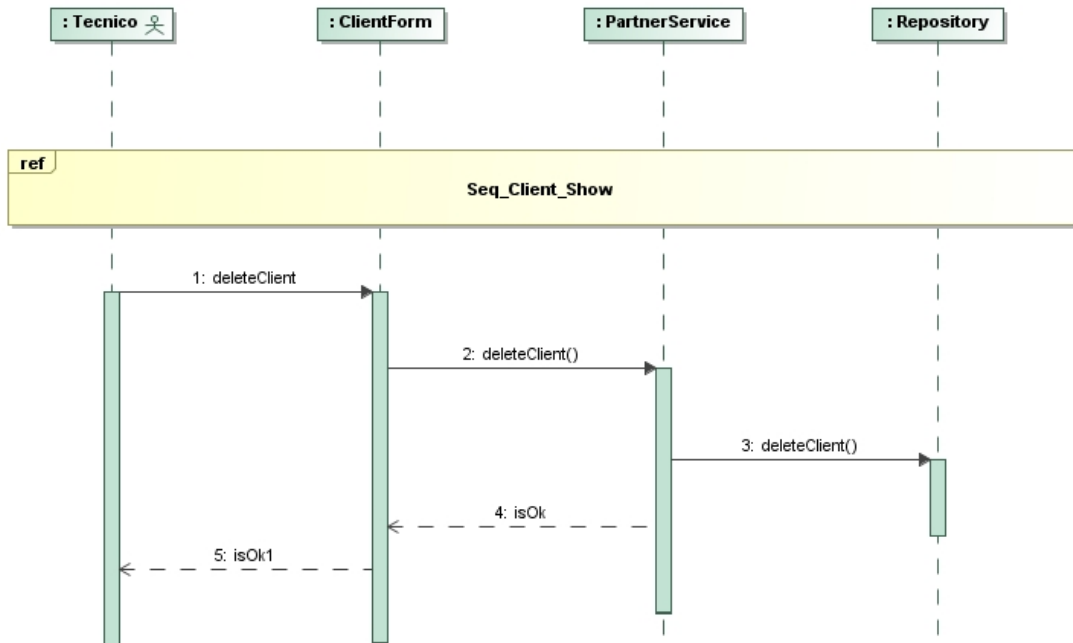


Ilustración 18 – Diagrama de secuencia – suprimir cliente



### 4.2.3. Gestión de pedidos

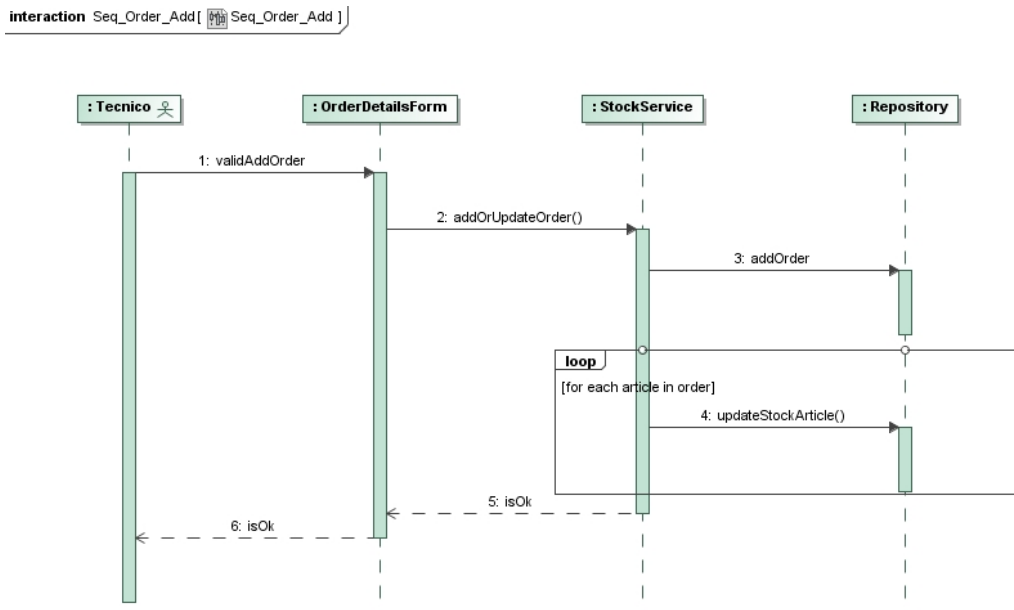


Ilustración 19 – Diagrama de secuencia – alta pedido

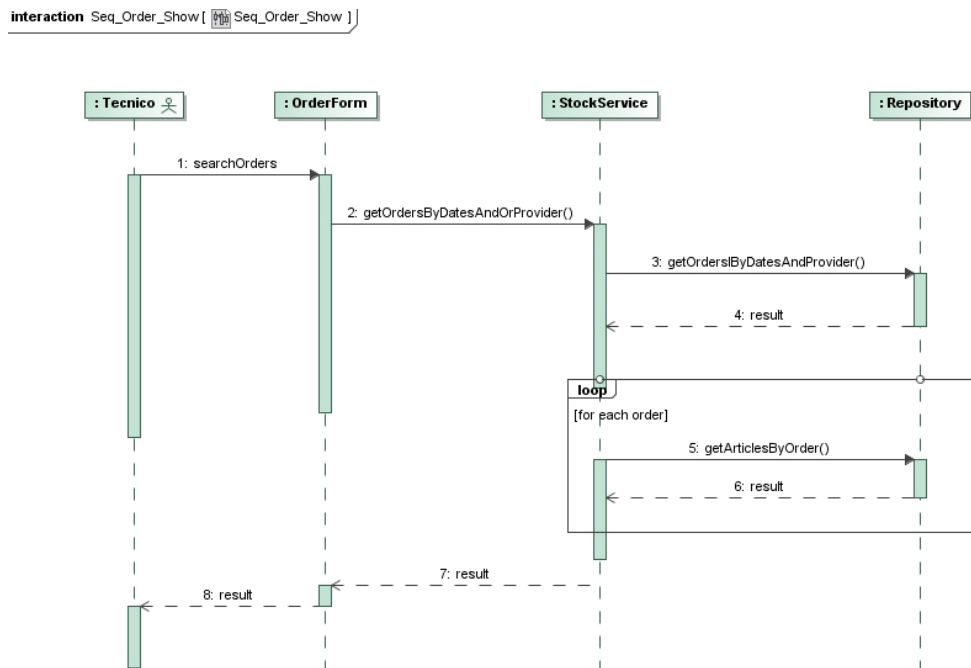


Ilustración 20 – Diagrama de secuencia – consultar pedido

interaction Seq\_Order\_Update [  Seq\_Order\_Update ]

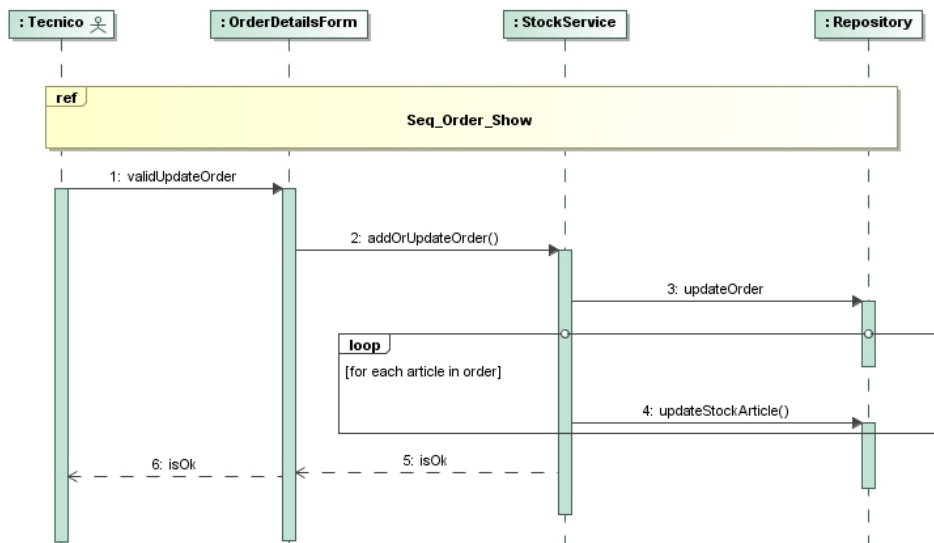


Ilustración 21 – Diagrama de secuencia – modificar pedido

interaction Seq\_Order\_Delete [  Seq\_Order\_Delete ]

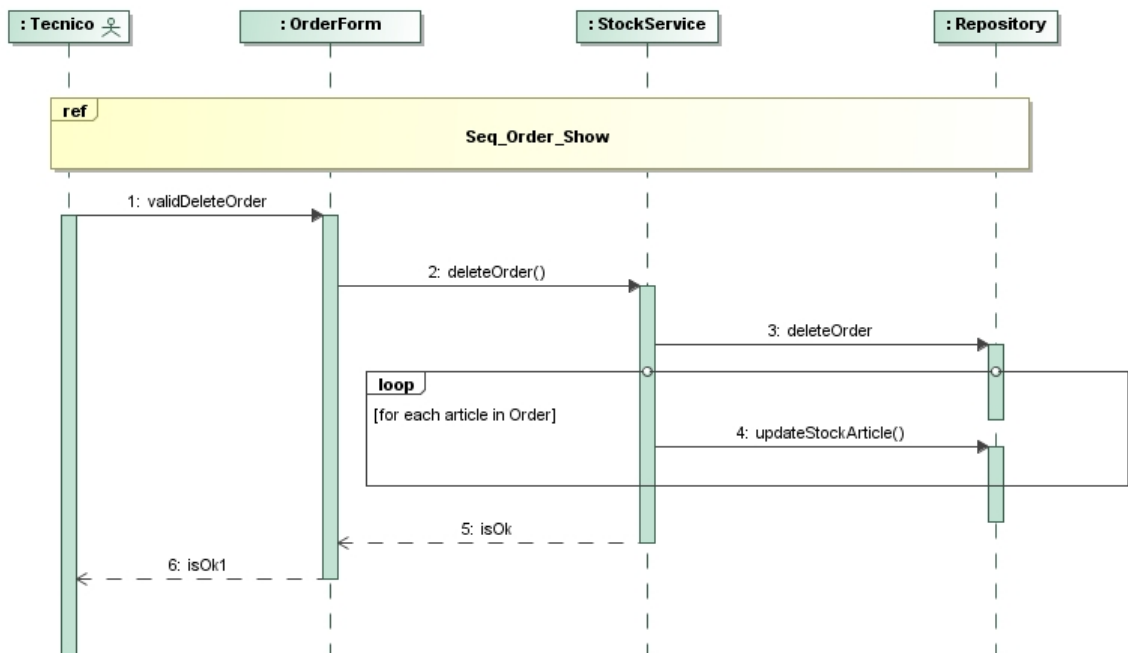

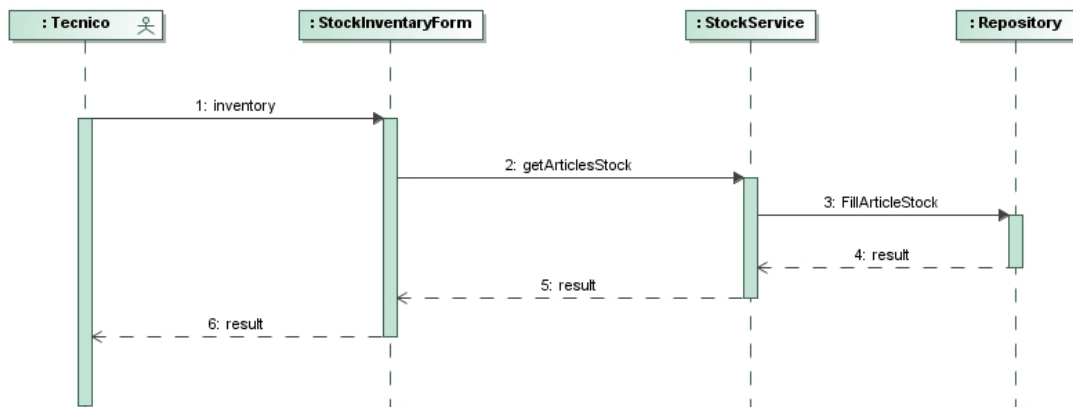


Ilustración 22 – Diagrama de secuencia – suprimir pedido

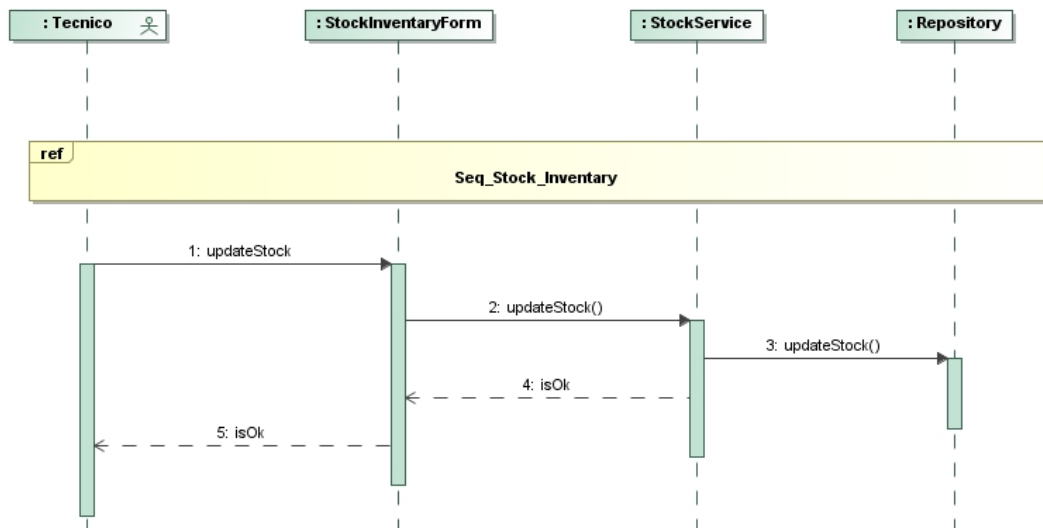
### 4.2.4. Inventario

**interaction** Seq\_Stock\_Inventory [  Seq\_Stock\_Inventory ]



**Ilustración 23 – Diagrama de secuencia – consultar inventario**

**interaction** Seq\_Stock\_Inventory\_Update [  Seq\_Stock\_Inventory\_Update ]



**Ilustración 24 – Diagrama de secuencia – modificar inventario stock**

### 4.2.5. Gestión de ventas

interaction Seq\_Sale\_New [ Seq\_Sale\_New ]

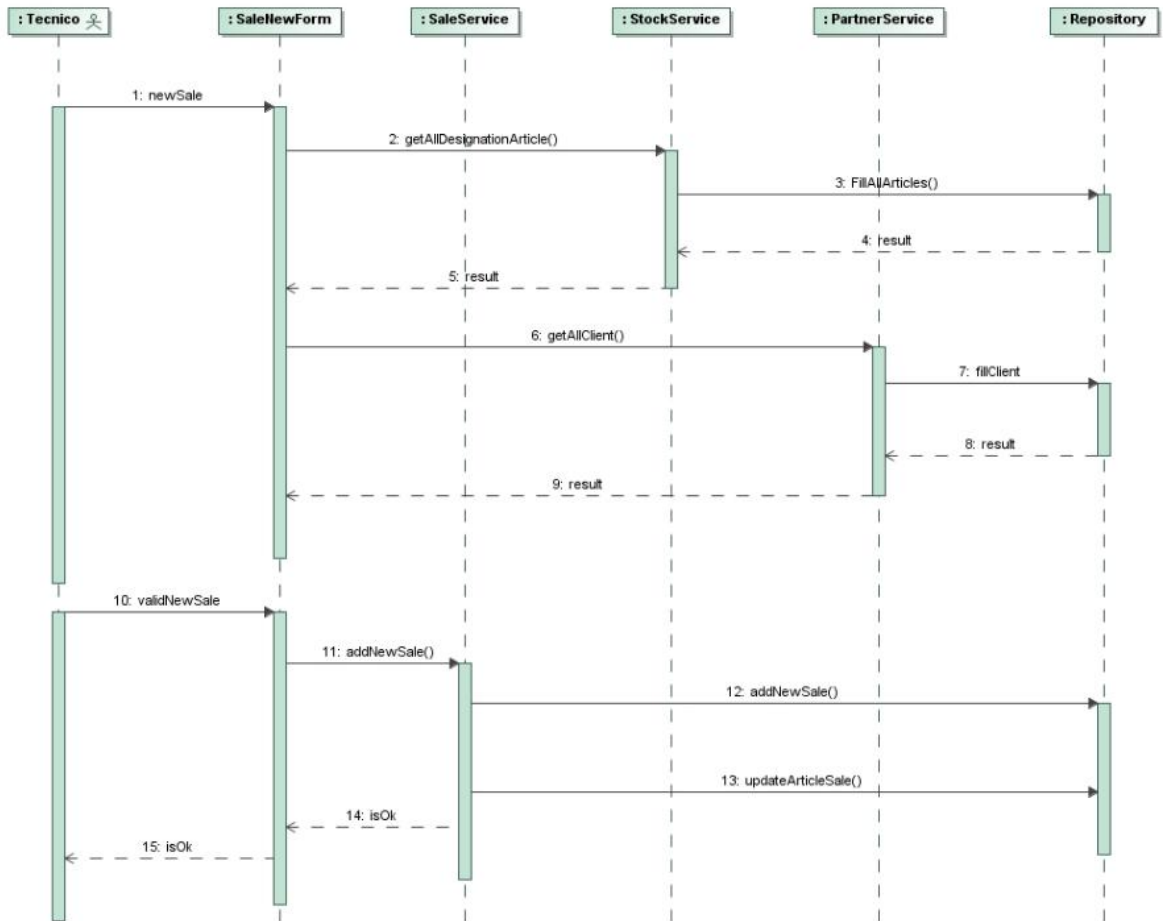


Ilustración 25 – Diagrama de secuencia – nueva venta

interaction Seq\_Sale\_Reminder [ Seq\_Sale\_Reminder ]

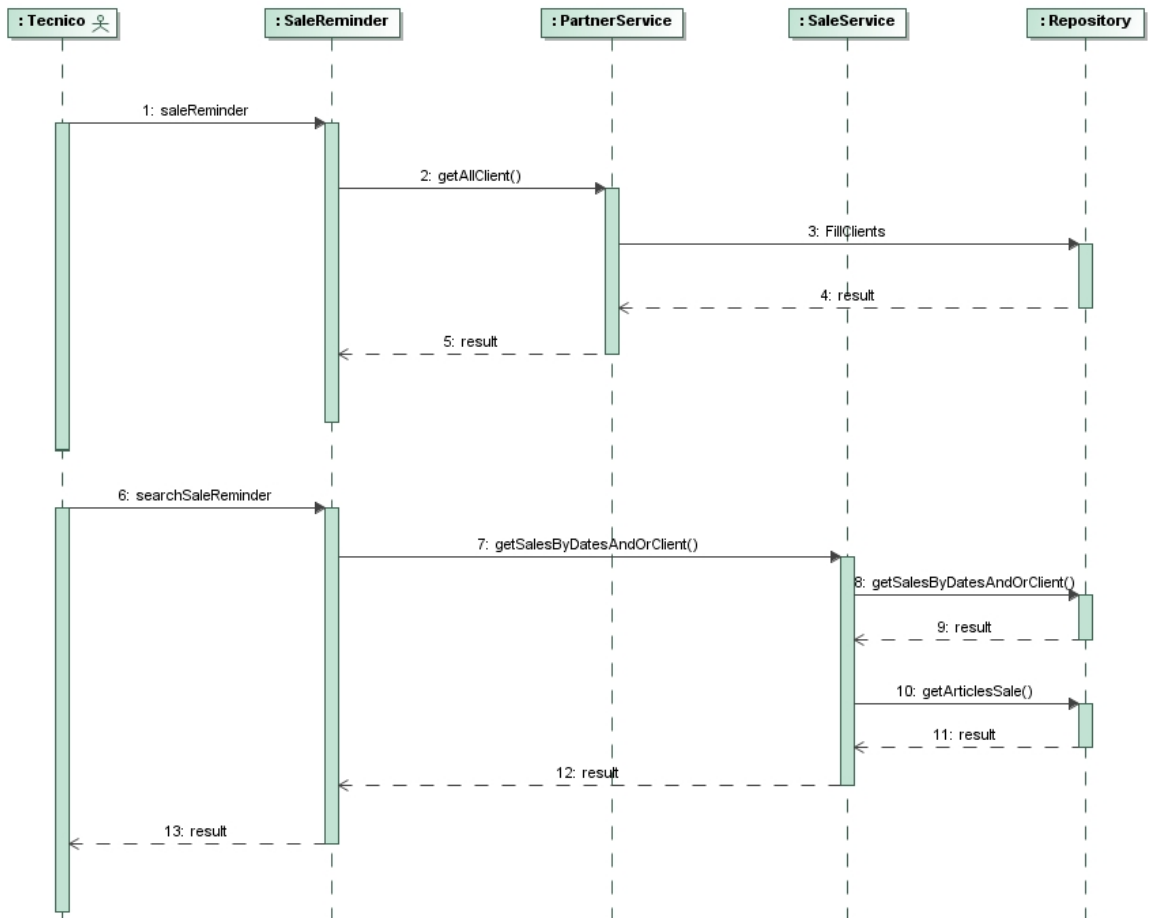



Ilustración 26 – Diagrama de secuencia – histórico de ventas

interaction Seq\_Sale\_Budget [  Seq\_Sale\_Budget ]

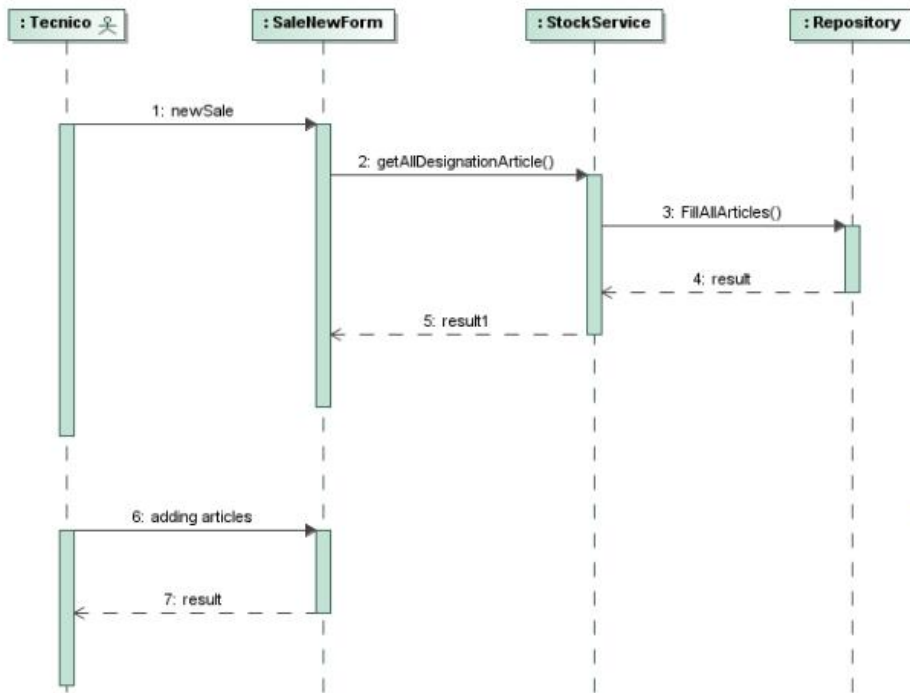



Ilustración 27 – Diagrama de secuencia – presupuesto venta

interaction Seq\_Sale\_Budget\_Print [  Seq\_Sale\_Budget\_Print ]

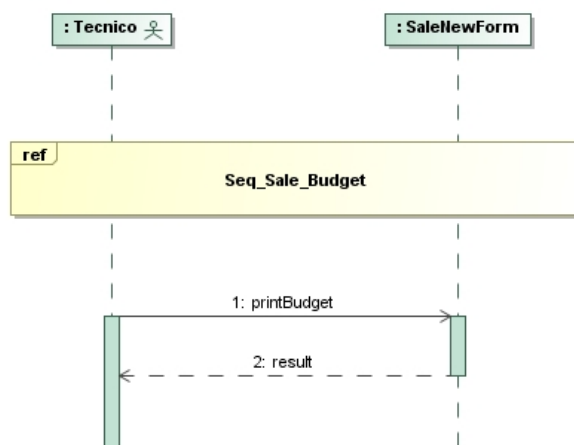


Ilustración 28 – Diagrama de secuencia – imprimir presupuesto venta

### 4.2.6. Gestión de Caja

interaction Seq\_Cash\_Deferred\_Payment [ Seq\_Cash\_Deferred\_Payment ]

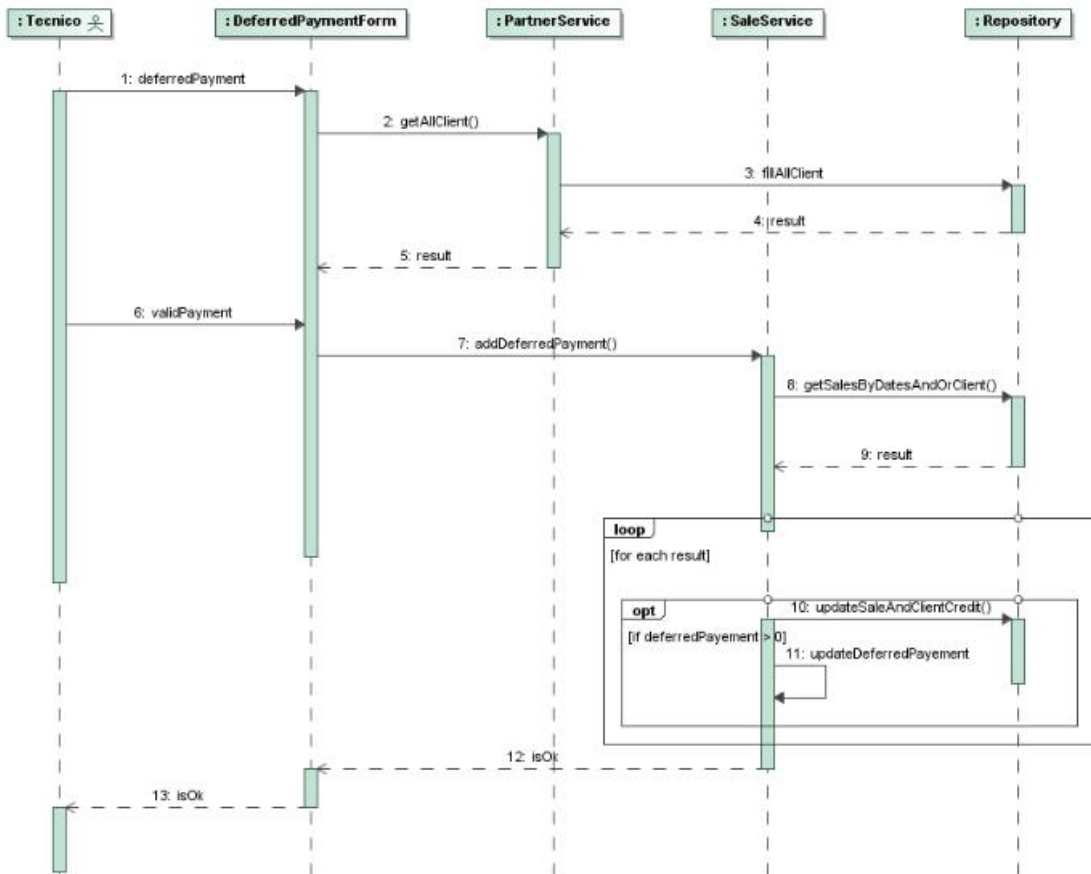


Ilustración 29 – Diagrama de secuencia – pago diferido

interaction Seq\_Cash\_Register | Seq\_Cash\_Register |

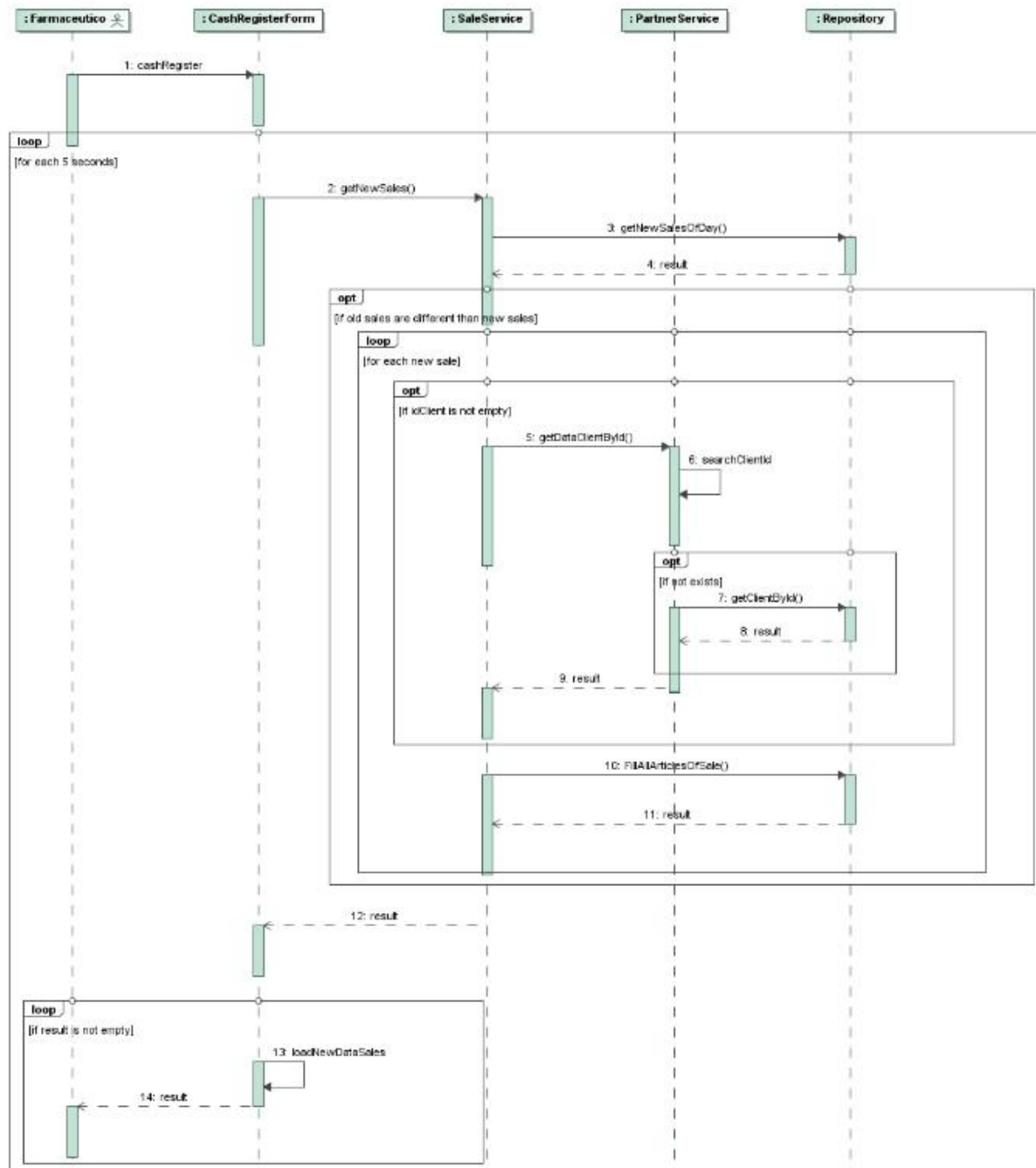


Ilustración 30 – Diagrama de secuencia – Consultar caja



interaction Seq\_Cash\_SaleEnding [ Seq\_Cash\_SaleEnding ]

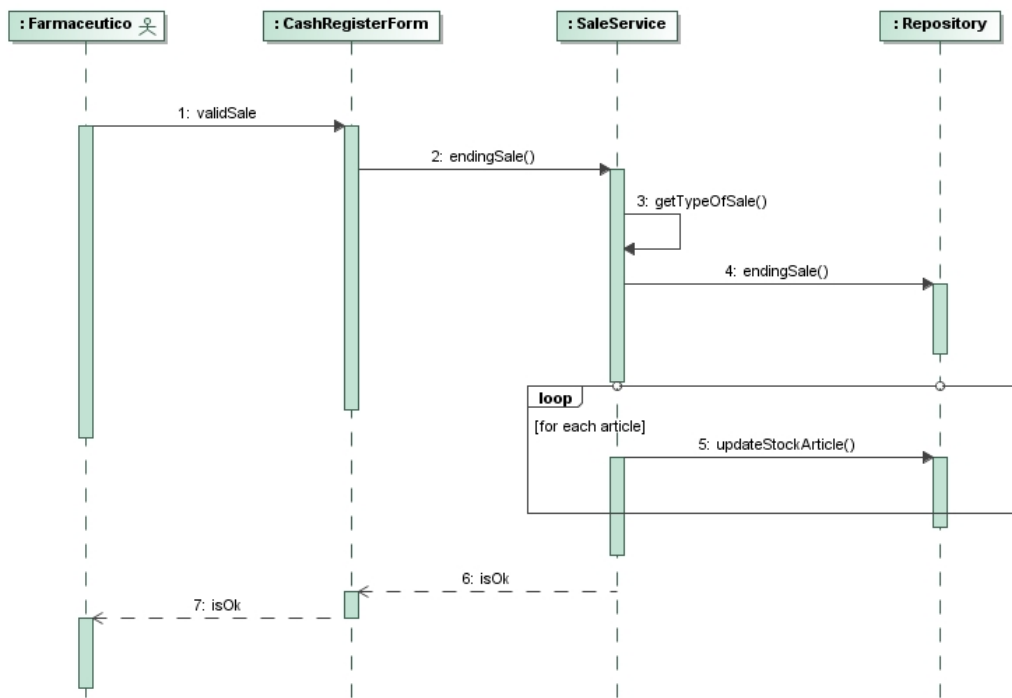


Ilustración 31 – Diagrama de secuencia – Finalizar venta

interaction Seq\_Cash\_WorkingDay [ Seq\_Cash\_WorkingDay ]

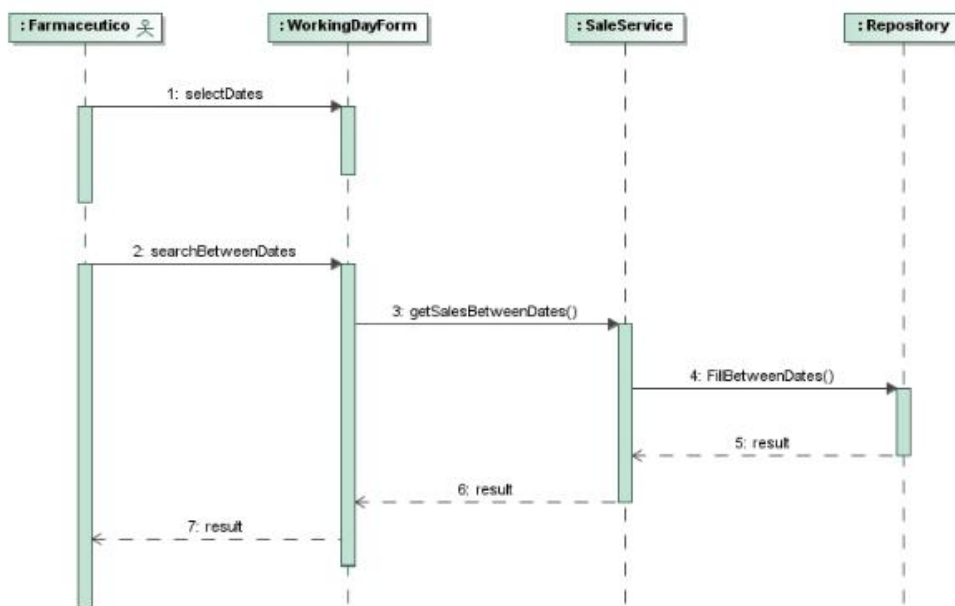


Ilustración 32 – Diagrama de secuencia – jornada

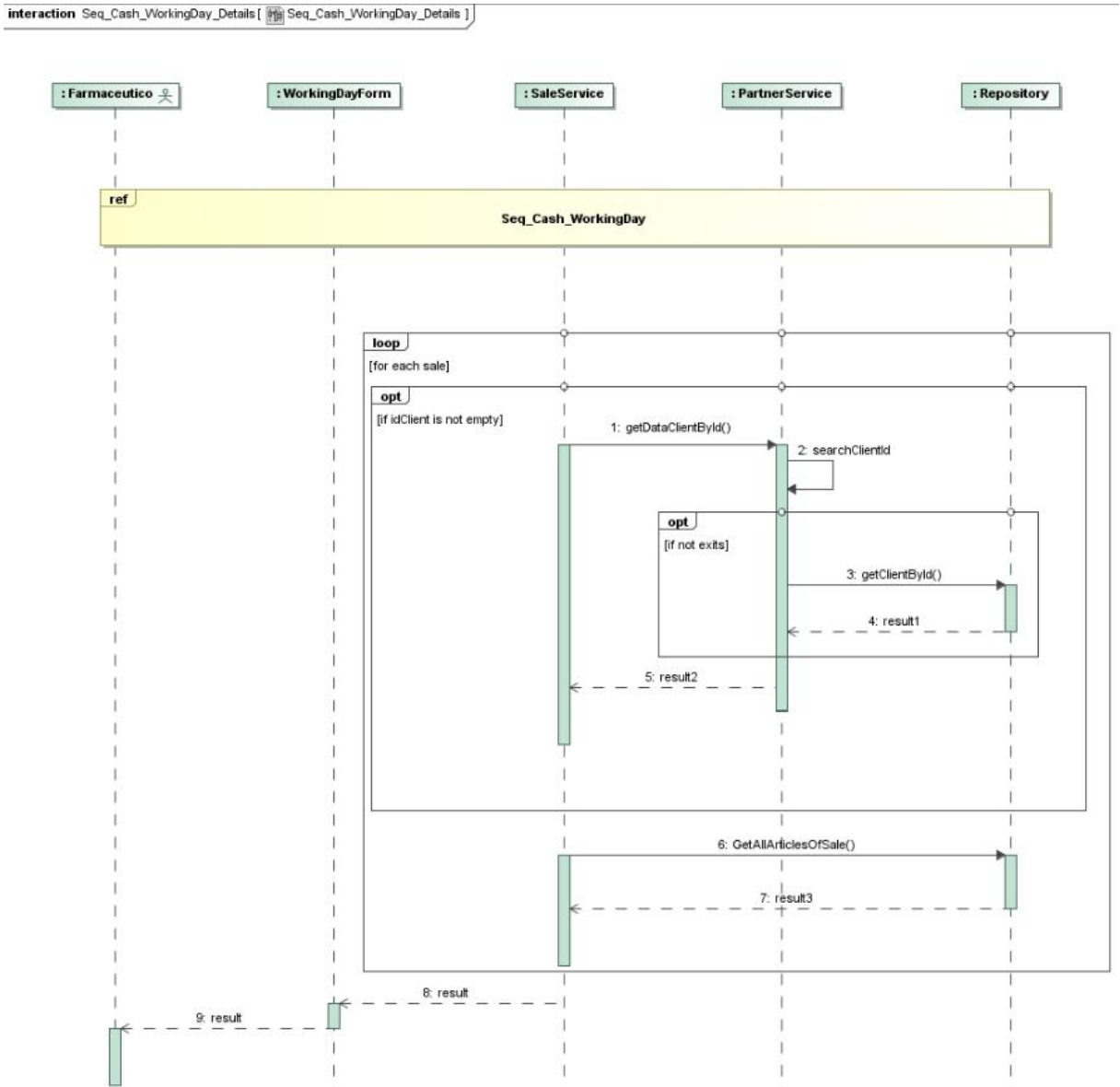


Ilustración 33 – Diagrama de secuencia – detalle de jornada

### 4.3. Diseño de base de datos

En este apartado mostraremos el diagrama de entidad relación y el diccionario de datos con la definición de las tablas.

#### 4.3.1. Diagrama de entidad relación

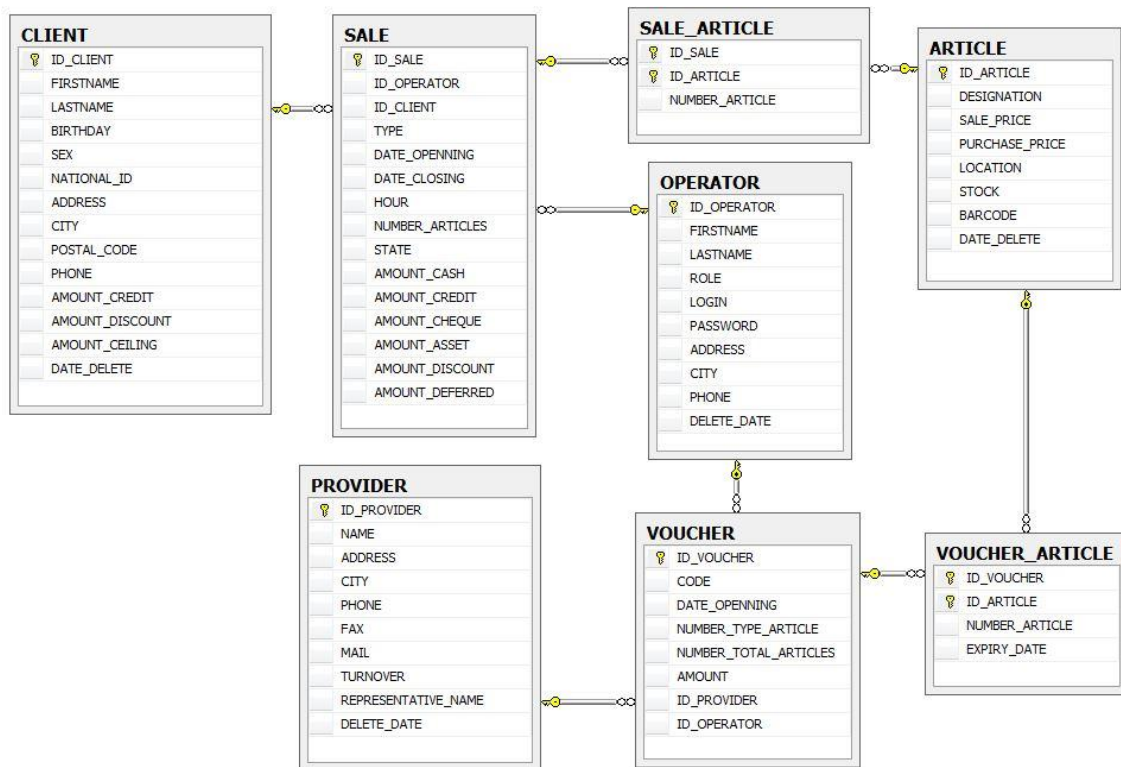


Ilustración 34 – Diagrama de entidad relación

#### 4.3.2. Definición de tablas

En este apartado se presentan las tablas de la base de datos del sistema con una breve explicación de la misma, así como de sus atributos, el tipo de dato y las restricciones de los mismos. También se definen las relaciones entre tablas con sus claves identificadoras y ajenas.

#### 4.3.2.1. Tabla Sale

Columna	Tipo	Descripción
ID_SALE	Long	Identificador auto-incremental de la venta (PK)
ID_OPERATOR	Int	Identificador del operador que realiza la venta (FK)
ID_CLIENT	Int	Identificador del cliente que realiza la venta (FK)
TYPE	Char(1)	Tipo de venta
DATE_OPENNING	Datetime	Fecha de creación de la venta
DATE_CLOSING	Datetime	Fecha de finalización de la venta
HOUR	Char(5)	Hora de creación o si procede finalización de la fecha
NUMBER_ARTICLES	Int	Número de artículos de la venta
STATE	Char(1)	Estado de la venta
AMOUNT_CASH	Float	Montante de pago en efectivo
AMOUNT_CREDIT	Float	Montante de compra por crédito
AMOUNT_CHEQUE	Float	Montante de pago con cheque
AMOUNT_ASSET	Float	Montante de pago por un haber o vale
AMOUNT_DISCOUNT	Float	Montante de descuento
AMOUNT_DEFERRED	Float	Montante de venta diferida para saldar un crédito

Tabla 11 - Definición tabla - Venta

#### 4.3.2.2. Tabla Cliente

Columna	Tipo	Descripción
ID_CLIENT	Int	Identificador del cliente (PK)
FIRSTNAME	Varchar(50)	Nombre del cliente
LASTNAME	Varchar(50)	Apellidos del cliente
BIRTHDAY	Date	Fecha de nacimiento del cliente
SEX	Char(1)	Sexo del cliente
NATIONAL_ID	Varchar(20)	Documento nacional de identidad del cliente
ADDRESSE	Varchar(50)	Dirección del cliente
CITY	Varchar(20)	Población del cliente
POSTAL_CODE	Varchar(10)	Código postal del cliente
PHONE	Varchar(20)	Número de teléfono del cliente
AMOUNT_CREDIT	Float	Montante de crédito actual del cliente
AMOUNT_DISCOUNT	Float	Montante de descuentos del cliente
AMOUNT_CEILING	Float	Límite de crédito admitido al cliente
DATE_DELETE	Date time	Fecha de supresión – borrado lógico

Tabla 12 - Definición tabla - Cliente

#### 4.3.2.3. Tabla Proveedor

Columna	Tipo	Descripción
ID_PROVEEDOR	Int	Identificador del proveedor (PK)
NAME	Varchar(50)	Nombre del proveedor
ADDRESS	Varchar(100)	Dirección del proveedor
CITY	Varchar(50)	Población del proveedor
PHONE	Varchar(50)	Número de teléfono del proveedor
FAX	Varchar(20)	Número de fax del proveedor
MAIL	Varchar(20)	Email del proveedor
TURNOVER	Float	Cifra de negocio del proveedor
REPRESENTATIVE_NAME	Varchar(50)	Nombre del representante del proveedor
DELETE_DATE	Date	Fecha de supresión – Borrado lógico

Tabla 13 - Definición tabla - Proveedor

#### 4.3.2.4. Tabla Artículo

Columna	Tipo	Descripción
ID_ARTICLE	Int	Identificador del artículo (PK)
DESIGNATION	Varchar(100)	Nombre del artículo
SALE_PRICE	Float	Precio de venta del artículo
PURCHASE_PRICE	Float	Precio de compra del artículo
LOCATION	Varchar(50)	Emplazamiento del artículo dentro de la farmacia
STOCK	Int	Número en stock del artículo
BARCODE	Varchar(50)	Código de barras del artículo
DATE_DELETE	Date	Fecha de supresión – Borrado lógico

Tabla 14 - Definición tabla - Artículo

#### 4.3.2.5. Tabla Voucher de pedidos

Columna	Tipo	Descripción
ID_VOUCHER	Int	Identificador auto-incremental (PK)
CODE	Varchar(50)	Código del pedido
DATE_OPENNING	Date	Fecha de creación
NUMBER_TYPE_ARTICLE	Int	Número de artículos distintos
NUMBER_TOTAL_ARTICLE	Int	Número total de artículos del pedido
AMOUNT	Float	Precio a pagar por el pedido
ID_PROVIDER	Int	Identificador del proveedor (FK)
ID_OPERATOR	Int	Identificador del operador que introdujo el pedido (FK)

Tabla 15 - Definición tabla – Pedido

### 4.3.2.6. Tabla Operador

Columna	Tipo	Descripción
ID_OPERADOR	Int	Identificador del operador (PK)
FIRSTNAME	Varchar(50)	Nombre del operador
LASTNAME	Varchar(50)	Apellidos del operador
ROLE	Char(1)	tipo del operador {técnico, farmacéutico, administrador}
LOGIN	Varchar(10)	Nombre de acceso del operador
PASSWORD	Varchar(20)	Contraseña de acceso del operador
ADDRESSE	Varchar(50)	Dirección del operador
CITY	Varchar(20)	Población del operador
PHONE	Varchar(20)	Número de teléfono del operador
DELETE_PHONE	Date	Fecha de supresión – Borrado lógico

Tabla 16 - Definición tabla - Operador

## 4.4. Diseño de interfaz de usuario

En este apartado, se presenta el diseño de algunas interfaces de usuario basado sobre el prototipo horizontal que ha sido validado con el usuario final en la fase de análisis.

Para la implementación de la interfaz gráfica, se va a utilizar componentes ya existentes integrados en un proyecto C#, para que sea más amigable.

El prototipo muestra una interfaz de usuario para la gestión de ventas. En la parte superior, hay tres botones de navegación: 'Nueva venta' (destacado en azul), 'Presupuesto' y 'Recordatorio de ventas'. Debajo de ellos, se encuentra un formulario de búsqueda con el título 'Buscar cliente por nombre'. Este formulario incluye tres campos de entrada etiquetados como 'Id', 'nombre' y 'apellidos', y un botón 'buscar'. A continuación, se muestra una 'Lista de artículos' con una tabla que contiene tres filas de datos. Cada fila tiene cuatro columnas: 'Designación', 'cantidad', 'precio unitario' y 'precio total'. Debajo de la tabla, hay un botón 'Precio total'. En la parte inferior del formulario, se encuentra un campo 'Añadir artículo' con una tabla de cinco columnas: 'Designación', 'Cantidad', 'Stock', 'Precio unitario' y 'Precio total'. Finalmente, en la parte inferior de la interfaz, hay dos botones: 'Cancelar' y 'Validar'.

Ilustración 35 – prototipo – nueva venta

**Buscar cliente por nombre**

Id  nombre  apellidos

De   A

Lista de artículos

NumVentas1	Operador1	precio total1	Neto a pagar1
NumVentas2	Operador2	precio total2	Neto a pagar2
NumVentas3	Operador3	precio total3	Neto a pagar3

Artículos correspondientes a la venta seleccionada

Designación1	Cantidad1	precio total1
Designación2	Cantidad2	precio total2
Designación3	Cantidad3	precio total3

**Ilustración 36 – prototipo – recordatorio de ventas**

**Buscar cliente para modificación o supresión**

nombre

**Datos del cliente**

**Ilustración 37 – prototipo – cliente**

Buscar proveedor para modificación o supresión

Datos del cliente

<input type="text" value="Nombre"/>	<input type="text" value="Dirección"/>
<input type="text" value="eMail"/>	<input type="text" value="Ciudad"/>
<input type="text" value="Teléfono"/>	<input type="text" value="Cifra de negocio"/>
<input type="text" value="Fax"/> <input type="button" value="📞"/>	<input type="text" value="Representatnte"/>

Ilustración 38 – prototipo – proveedor

Buscar artículo para modificación o supresión

Datos del artículo

<input type="text" value="Nombre"/>	<input type="text" value="Cantidad stock"/>
<input type="text" value="Precio de compra"/>	<input type="text" value="4/8/2013"/> <input type="button" value="📅"/>
<input type="text" value="Precio de venta"/>	<input type="text" value="Código de barras"/>
<input type="text" value="Emplazamiento"/>	

Ilustración 39 – prototipo – artículo



## 5. Implementación

En la implementación de este proyecto, se ha utilizado el lenguaje C# .NET que es un lenguaje de propósito general orientado a objetos creado por Microsoft para su plataforma .NET. Éste es un lenguaje de programación orientado a la rápida y sencilla creación de aplicaciones de formularios Windows, aplicaciones Web y servicios Web entre otros que funcionan sobre el Framework .NET y ofrece capacidades de diseño completamente orientado a objetos, incorporando características como herencia, sobrecarga de métodos o manejo de excepciones estructurado. Asimismo, el lenguaje de C# se ha optimizado, eliminando palabras clave obsoletas que se habían heredado y mejorando la seguridad de tipos.

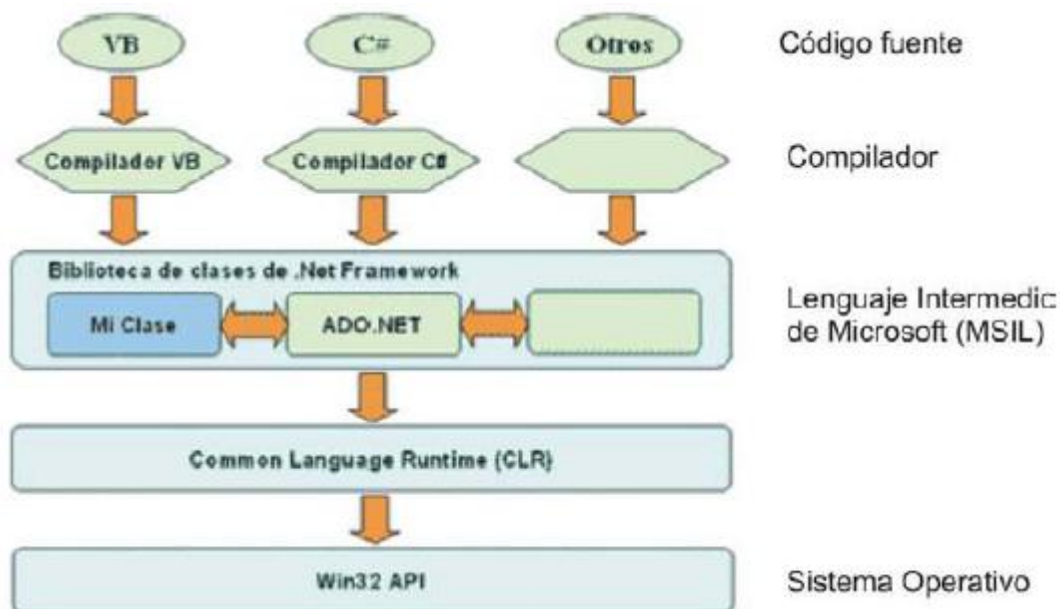


Ilustración 40 - Implementación - Esquema de compilación .NET

En cuanto al acceso de datos, la aplicación utiliza ADO .NET, una evolución de ADO (Active Data Object), con un modelo de objetos basado en XML, lo que permite una mejor interacción con la información desde y hacia las bases de datos. ADO .NET forma parte integral de .NET Framework y proporciona un extenso conjunto de clases para facilitar un acceso eficaz a los datos. Estas clases se encuentran divididas en dos ramas principales, unas están preparadas para trabajar con controladores OLE DB y

otras son específicas para trabajar con SQL Server, ésta rama es la que se ha utilizado en la elaboración de nuestro proyecto.

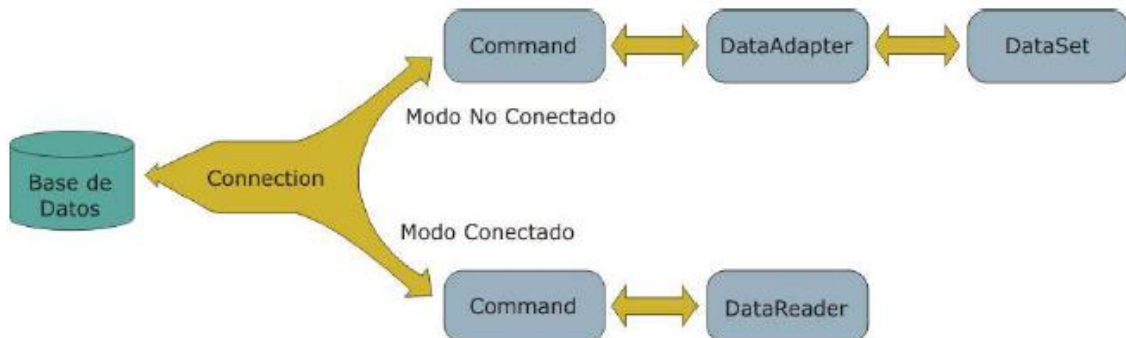


Ilustración 41 - Implementación - Esquema ADO.NET

En cuanto al entorno de desarrollo, lo detallamos como sigue.

- **RAID IDE:** Microsoft Visual Studio 2010
- **Framework:** .NET en versión 3.5
- **SGBD:** SQL Server 2008
- **Generación de informes:** Crystal Report versión 13.0.5

## 5.1. Jerarquía del proyecto

El proyecto está constituido por tres capas, de los cuales dos están integradas en el proyecto *PharmacilAB* y representan las capas de presentación y formularios y la segunda la capa del negocio, y una tercera capa que está en forma de procedimientos almacenados directamente integrada en la base de datos.

La carpeta de la capa de presentación, llamada *Forms*, está distribuida por funcionalidades como muestra el menú de la aplicación para facilitar la organización de los formularios.



Ilustración 42- Implementación - Menú

De la misma forma y siguiendo el mismo principio, la capa de negocio, llamada *Services*, respeta también la misma distribución de las principales funcionalidades del sistema.

**Nota:** Se ha hecho uso del proyecto *TabStripControlLibrary* que es en realidad una librería que no está desarrollada en este proyecto, sino que simplemente se hace referencia a ella para diseñar el menú y el submenú. Con lo cual se usa como un componente externo o una caja negra.

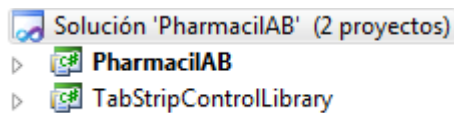


Ilustración 43 – Implementación - Proyectos C#

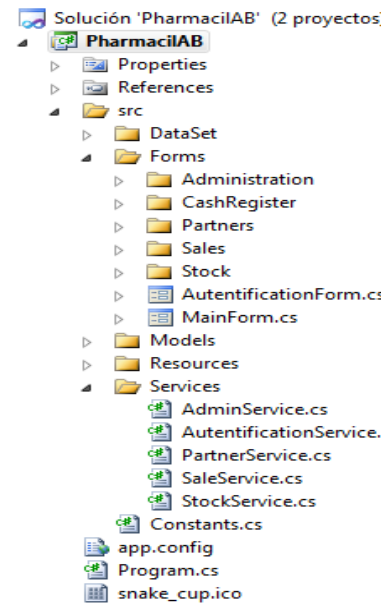


Ilustración 44 – Implementación - Capas

## 5.2. Algunos aspectos de implementación

A continuación se van a presentar algunos aspectos de implementación usados en el proyecto:

### 5.2.1. Auto completación

En todos los formularios que constituyen una búsqueda, sea de clientes, proveedores o artículos, se usa la auto completación para guiar al usuario en su búsqueda. Esto tiene un impacto ligeramente negativo en el rendimiento de la aplicación pero el beneficio es notorio, por eso se ha decidido su implantación.

```
//Getting an article list from DB
listAllArticles = Program.getInstanceStockService().getAllDesignationArticle();

//Autocomplete
AutoCompleteStringCollection acsc = new AutoCompleteStringCollection();

//build data auto complete article
if (listAllArticles != null){
    foreach (Article designation in listAllArticles){
        acsc.Add(designation.designation);
    }
}
```

```
}  
    sDesignationTB.AutoCompleteCustomSource = acsc;  
}
```

### 5.2.2. Llamadas a procedimientos almacenados

El acceso a los datos de la base de datos, se hace gracias a los procedimientos almacenados, éstos están llamados desde los servicios para consultar como para escribir en la BBDD:

```
//Adding voucher into DB  
decimal idVoucher = (decimal)Program.GetInstanceVoucherTableAdapter().  
    addOrder(voucher.code,  
    DateTime.Today, voucher.numberTypeArticle,  
    voucher.numberTotalArticle, voucher.amount,  
    voucher.idProvider, Program.theOperator.idOperator);
```

### 5.2.3. Mensajes de error y afirmación

Los mensajes que se muestran por pantalla, están consolidados en el fichero *Resources* de la aplicación:

```
<data name="es_sale_ending_credit_top" xml:space="preserve">  
    <value>Venta no permitida, umbral de crédito alcanzado</value>  
</data>  
  
sError.Text = Resources.es_sale_ending_credit_top
```

### 5.2.4. Generación de informes

La aplicación ofrece al usuario generar el informe e imprimir el presupuesto de la venta a realizar, para ello se ha utilizado *Crystal Report* que se ha integrado a nuestro IDE para este fin.

Para generar informes con *Crystal Report*, se ha creado un *DataSet* llamado *CRDataSet.xsd* que contiene los datos que se van a imprimir, luego desde el formulario

*SaleArticlePrintForm*, se rellena dicho *DataSet* y se instancia en el informe *BudgetCrystalReport* previamente creado y el que contiene el informe con el formato de los datos a presentar en el documento:

```
CRDataSet crds = new CRDataSet();

//article numbers
int numberData = saleListDGV.Rows.Count;
string[] data;
for (int i = 0; i < numberData; i++)
{
    data = new string[]{
        saleListDGV[1,i].Value.ToString(),
        saleListDGV[2,i].Value.ToString(),
        saleListDGV[3,i].Value.ToString(),
        saleListDGV[4,i].Value.ToString()
    };
    //add data to crystal report table
    crds.Tables[0].Rows.Add(data);
}

BudgetCrystalReport bcr = new BudgetCrystalReport();
//adding data source to crystal report
bcr.SetDataSource(crds);
crystalReportViewer1.ReportSource = bcr;
```

*Crystal Report* es una aplicación muy potente, aunque tiene el inconveniente de disminuir bastante el rendimiento de la aplicación que se ve alterado, sobre todo, a la hora de generar el primer informe.

## 5.1. Algunos módulos de la aplicación

En este apartado presentamos algunos módulos de la aplicación seguidos de una pequeña explicación de cómo operar, en cada módulo:

### 5.1.1. Módulo de administración - Gestión de usuarios

Esta pantalla permite dar de alta, consultar, modificar y eliminar los usuarios del sistema. Para registrar un nuevo usuario sólo hay que introducir sus datos en el área de “Datos del operador” y luego pulsar el botón “Aceptar”

Para la modificación, hay que introducir el apellido del operador previamente registrado en el área de búsqueda, esto permitirá mostrar todos sus datos mientras se vayan filtrando los usuarios y de forma sistemática en el área creación y modificación, luego sólo faltaría modificar los datos y luego validar pulsando el botón “Validar”.

Para borrar, hay que realizar la búsqueda y luego pulsar el botón “Borrar”.

Cabe destacar que la gestión de socios es prácticamente idéntica a la gestión de usuarios.

Ilustración 45 – Pantalla de gestión administrativa

### 5.1.2. Módulo de ventas – Nueva venta

Probablemente, la pantalla más utilizada por los técnicos farmacéuticos; permite añadir artículos a la venta. Esta, puede ser anónima o ligada a un cliente, ésto permite luego realizar créditos a los clientes previamente registrados en el sistema.

Para añadir un nuevo artículo, se tiene que introducir a mano en el área de introducción de artículos, luego seleccionarlo de la lista desplegable y pulsar la tecla “Intro”, finalmente introducir la cantidad deseada y pulsar de nuevo en “Intro”, esto permitirá añadir el artículo a la venta y con lo cual, se mostrará en la lista de artículos de la venta.

Una vez introducidos todos los artículos, pulsar el botón “Validar” para confirmar la venta, o sino el botón “Cancelar” para no registrar la venta en el sistema.

Para realizar un presupuesto sin compra, sólo habría que pulsar sobre el botón “Presupuesto” del submenú, introducir los artículos como se explica arriba y luego pulsar el botón “Imprimir” para generar el informe que se puede imprimir.

PharmacIAB - [Registrar venta]

VENTAS STOCK SOCIOS CAJA ADMINISTRACION SALIR ACERCA DE

Nueva Presupuesto Historico

Gestion de Ventas

**Nueva venta**

Introduzca el nombre del cliente

Código Nombre Apellidos

3 Ivonne LOPEZ

Operador PEREZ

Buscar clientes por apellidos

Lista de artículos de la venta

Nombre	Cantidad	Precio unitario	Precio total
PANADOL 125 MG X 10 SUP	2	0,69	1,38
IMOVANE 7,5 MG X 20 COMP	1	3	3
CALCIBRONAT 20 COMP EFF	3	2,71	8,13
TARDYFERON 80 MG 30 CP	4	2,84	11,36
HELICOCIN AMOXI METRONIDAZOL 2X21	7	8,57	59,99

Lista de artículos seleccionados

Añadir nuevo artículo a la venta

Nombre del artículo Cantidad Stock Precio unitario Precio total

BACICOLINE COLLYRE 3 11 2,29 6,87

Precio total de la venta Total: 83,86

Cancelar Validar

Ilustración 46 - Pantalla de nueva venta

### 5.1.3. Módulo de ventas – Histórico de ventas

Para visualizar el histórico de ventas, se puede seleccionar una búsqueda por cliente, fechas o ambos y luego pulsar el botón “Buscar”.

Se muestra la lista de ventas y artículos asociados correspondientes a la búsqueda, se pueden seleccionar las ventas de la lista lo que permitirá actualizar la lista de artículos asociados a la venta seleccionada. De la misma forma, actualizará el montante total de la venta seleccionada en el área de “Rescapitulativo”.

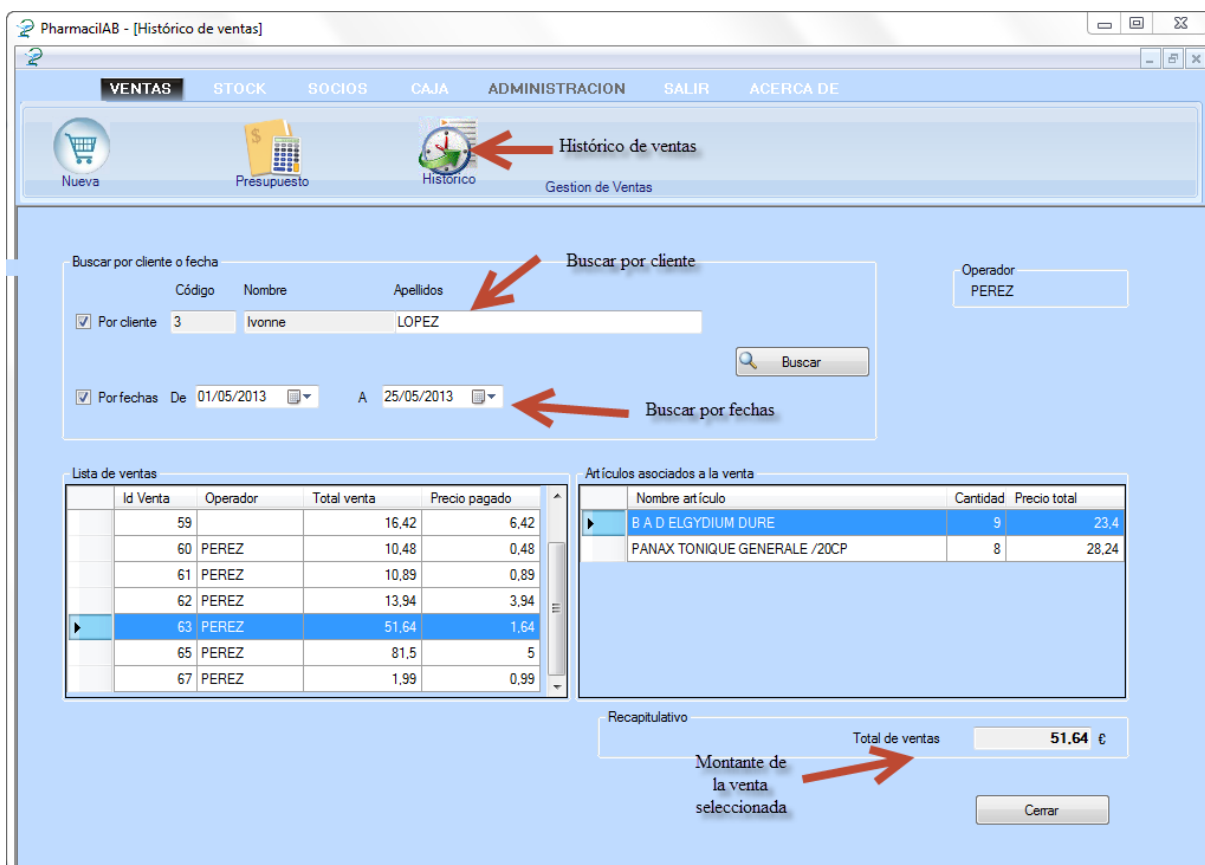


Ilustración 47 – Pantalla de histórico de ventas



### 5.1.4. Módulo de stock – Modificar pedido

Esta pantalla permite modificar y borrar pedidos previamente registrados en el sistema. Para ello, se tiene que hacer una búsqueda por nombre de proveedor, fechas o ambos y luego pulsar el botón de “Buscar”.

Obtenida la lista de albaranes con sus artículos correspondientes, se ofrecen varias posibilidades:

- Añadir un artículo al pedido introduciéndolo en el área de modificación y luego pulsar el botón de “Añadir artículo”
- Modificar un artículo del pedido haciendo clic sobre el botón de editar artículo, luego en el área de modificación cambiar la fecha de caducidad y/o cantidad, y finalmente pulsando el botón “Modificar”.
- Borrar un artículo del pedido haciendo clic sobre el botón de borrar artículo.
- Borrar el pedido seleccionado

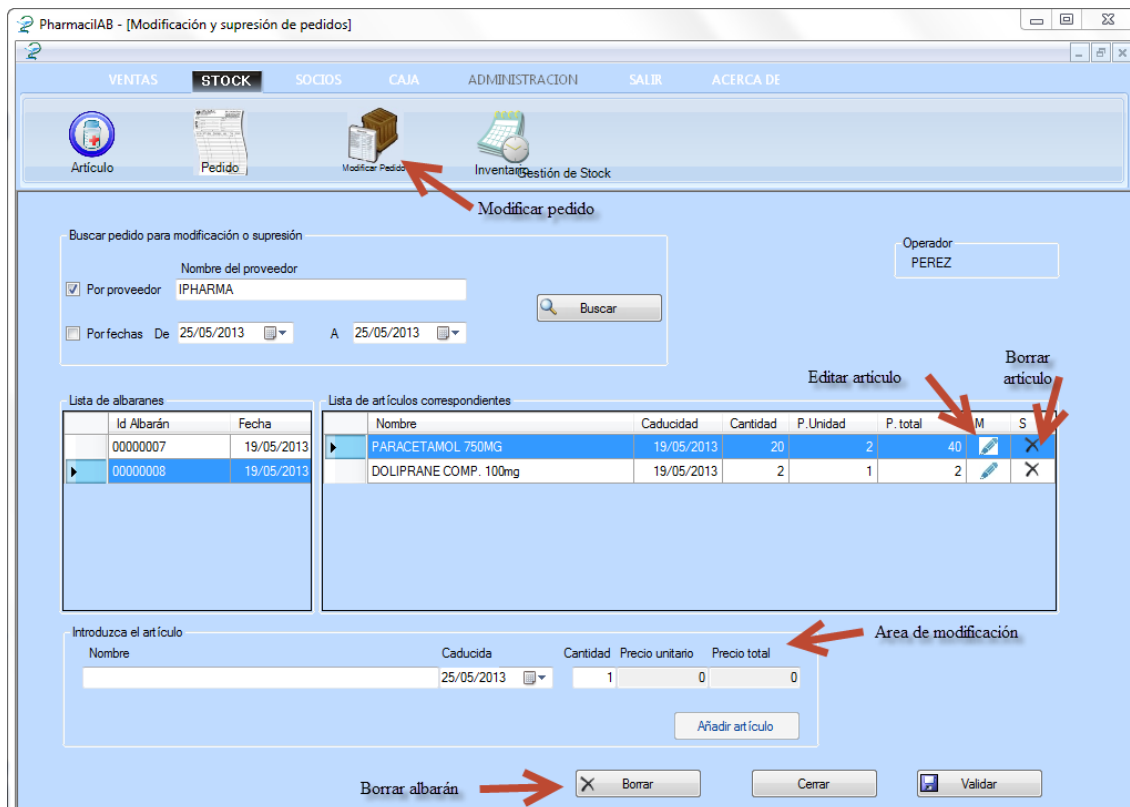


Ilustración 48 – Pantalla de modificar pedido

### 5.1.5. Módulo de caja – Caja registradora

Permite finalizar las ventas registradas previamente en las últimas 48 horas, las combinaciones de pago posibles son las siguientes:

Efectivo íntegro / Efectivo íntegro con descuento (%) / Efectivo y/o crédito para clientes / Efectivo y/o haber / Efectivo y/o cheque

Cabe destacar que no se permite devolución cuando se paga con un haber ya que éste no representa un bien monetario, y que el crédito sólo es posible para clientes previamente registrados en el sistema y asociados a la venta corriente.

Todos los clientes tienen asignado un umbral de crédito que cuando se alcanza, no se permite validar la venta a crédito.

Con lo cual para registrar una venta, hay que introducir en los campos “descuento aplicado”, “efectivo” o “crédito, haber, cheque” los montantes deseados y luego pulsar el botón “Validar venta”. O sino “Cancelar” para no registrar la venta.

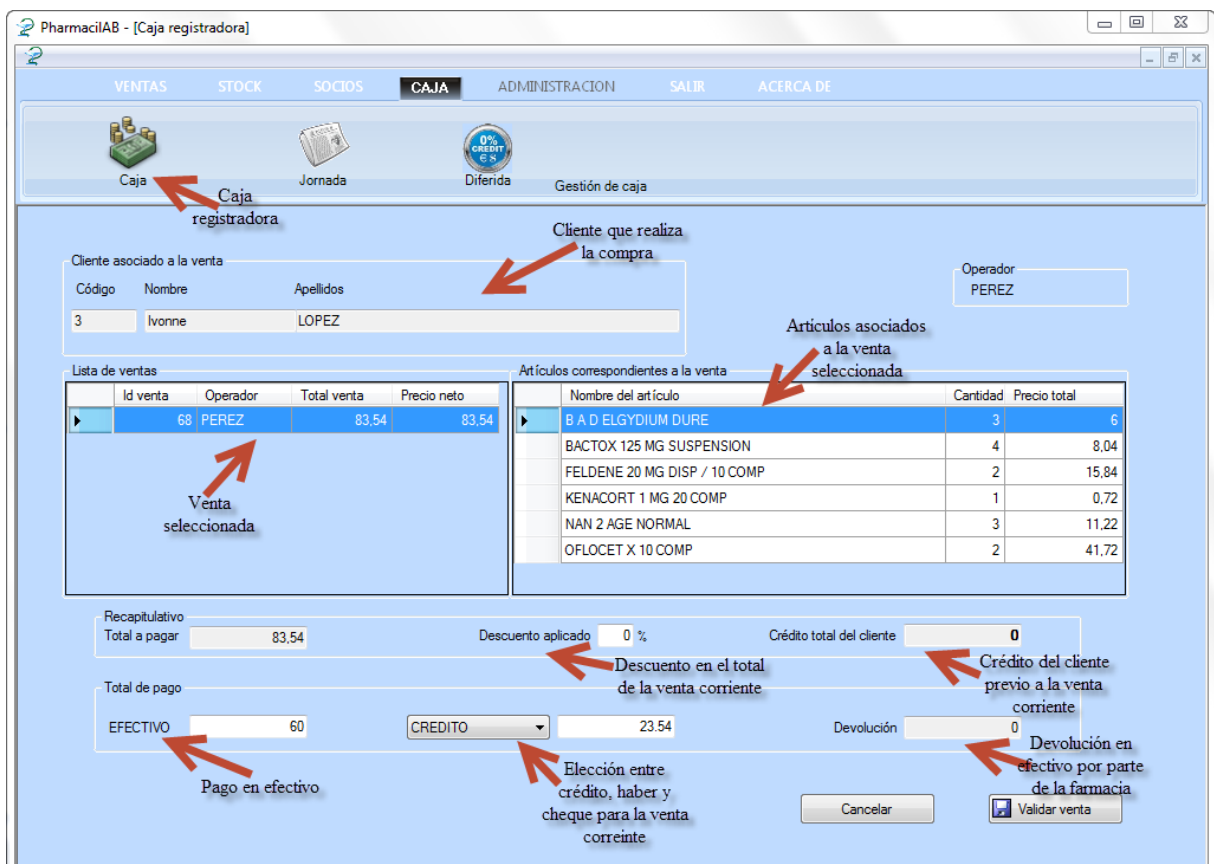


Ilustración 49 – Página caja registradora

## 6. Conclusiones

El objetivo de este proyecto técnico es poner en práctica algunos aspectos teóricos que se han adquirido a lo largo de la carrera de ingeniería informática, sea en el aspecto de gestión de proyectos en cuanto a gestión y análisis de riesgos, seguimiento y planificación de tareas como en métodos, procesos y metodologías de la ingeniería del software, así como el aspecto técnico de las fases de análisis y diseño siguiendo patrones de diseño y arquitectura de estándares tales como de IEEE.

Pues bien, este proyecto final de carrera es buen ejemplo de ello, tratándose sobre todo de un estudio de un caso real, que abarca muchos conceptos adquiridos a lo largo de la carrera, prueba de ello esta memoria que resume todo el trabajo realizado para construir un sistema completo respetando tanto los objetivos iniciales definidos en los requisitos funcionales y no funcionales como los hitos principales y en los plazos previstos llegando a entregar un producto de calidad que satisface las necesidades del cliente.

Desde el punto de vista personal, el proyecto en sí ha sido un gran reto para mí, porque habría que presentar, en un plazo muy corto de tiempo, un proyecto plagado de funcionalidades y con requisitos cambiantes y poco claros al principio del proyecto. Añadiendo que partía de unos humildes conocimientos sobre la tecnología que se va a utilizar. Gracias al gran esfuerzo personal y sobre todo a la organización del trabajo que ha sido la gran protagonista a lo largo de proyecto, se consiguió el fruto que sembrábamos al principio convirtiendo este gran esfuerzo en una enriquecedora experiencia.

## 7. Líneas de futuro

Las líneas de futuro de desarrollo con las siguientes:

- Internalización de la aplicación adoptando el multi-idioma.
- Externalizar la configuración de la aplicación como el acceso a la base de datos.
- Posibilidad de parametrizar algunos datos de la aplicación por parte de los usuarios como emplazamiento de los artículos.
- Mejorar la accesibilidad y navegación dentro de los formularios para reducir el uso del ratón.
- Añadir alertas como por ejemplo: para los artículos caducados o a caducar
- Estadísticas de ganancias netas

## 8. Bibliografía

*Moving to Microsoft Visual Studio 2010*  
por Patrice Pelland, Pascal Paré, y Ken Haine

*Una sinfonía en C#*  
<http://leomicheloni.blogspot.com>

*C# Corner*  
<http://www.c-sharpcorner.com/>

*The Code Project*  
<http://www.codeproject.com/>

*stackoverflow*  
<http://stackoverflow.com/>

*Crystal Reports Para Visual Studio Net*  
Por Blanco Luis Miguel

*Microsoft SQL Server 2008 Bible*  
Por Paul Nielsen, Mike White y Uttam Parui