



PROYECTO FONTANERIA

Jorge Victoria Andreu

Ciclo Desarrollo Aplicaciones Multiplataforma

Memoria del proyecto de DAM

IES SERRA PERENXISA. Curso 2021-22

Tutor: José Manuel Gras

Fecha: 09-Junio2022

Agradecimientos.

A Loli, mi esposa, por la paciencia que ha tenido estos 3 años y haberme permitido realizar esta formación.

A los profesores, con los que he compartido mi vida estos 3 años y que han hecho de esta “aventura” la más útil, mejor y más provechosa experiencia educativa que jamás he tenido

Contenido

1. Introducción.....	4
2. Objetivos.....	4
3. Estado del arte.....	7
4. Estudio de viabilidad.....	8
4.1. Estudio de mercado.....	8
4.1.1. Viabilidad técnica del proyecto.....	11
4.1.2. Viabilidad económica del proyecto.....	11
4.1.3. Recursos HW.....	13
4.1.4. Recursos SW.....	14
4.1.5. Recursos humanos.....	14
4.1.6. Viabilidad temporal.....	15
4.2. Planificación temporal.....	15
5. Análisis de requisitos.....	16
5.1. Descripción de requisitos.....	16
5.2. Diagrama de casos de uso.....	19
6. Diseño.....	20
6.1. Diseño conceptual E/R.....	20
6.2. Diseño Lógico Relacional.....	21
6.3. Diseño físico.....	23
6.4. Descripción de tablas y campos.....	26
6.5. Orientación a objetos.....	33
6.5.1. Diagramas de clases.....	33
6.5.2. Diagramas de secuencias.....	33
6.5.3. Diagrama de actividad.....	42
6.6. Diseño UX.....	50
6.7. Mockups.....	52
7. Codificación.....	54
7.1. Tecnologías elegidas.....	54
7.2. Documentación interna.....	55
7.2.1. Descripción de ficheros.....	55
7.2.2. Descripción de funciones.....	56
7.3. Documentación externa.....	58
7.3.1 Manual de usuario.....	58

8. Despliegue.	58
8.1. Diagramas de despliegue.....	58
8.2. Descripción de la instalación o despliegue.....	58
9. Herramientas de apoyo.....	59
10. Control de versiones.	59
11. Gestión de pruebas.	59
12. Conclusiones.	59
12.1. Conclusiones sobre el trabajo realizado.....	59
12.2. Conclusiones personales.	59
12.3. Posibles ampliaciones y mejoras.....	60
14. Bibliografía.	60
14.1 Libros, artículos y apuntes.	60
14.2 Direcciones web.....	60

1. Introducción.

Fontanería Victoria es una marca dedicada a la fontanería con sede en la localidad Torrent. Actualmente la plantilla está formada por un solo trabajador, Francisco Javier Victoria Andreu, trabajador autónomo y que es dueño de la marca.

La marca se encarga de realizar las siguientes actividades relacionadas con la fontanería: instalaciones para nuevas construcciones, reformas y reparaciones domésticas, así como mantenimiento de comunidades, residencias y naves industriales.

Además, también se encarga de la instalación y mantenimiento de otros servicios como calefacción, gas, aire acondicionado, energía solar, ACS, depuradoras de agua de piscina, descalcificadores y artículos para el tratamiento de osmosis.

En estos momentos todas las labores, ya sean referentes a la fontanería como administrativas, son llevadas a cabo por la misma persona, que en este caso es el dueño de la marca. Para otras gestiones relacionadas con contabilidad, contrataciones y demás se cuenta con los servicios de una gestoría.

El origen de este proyecto, aunque en mi caso ya lo tenía en mente desde que inicié mis estudios de FP, se acuerda en abril de 2021, tras una conversación que tuvimos ambos. En dicha conversación mostré mi interés por crear una aplicación para gestionar una pequeña empresa, dada mi necesidad de realizar un proyecto para finalizar los estudios, a la vez que él también estaba interesado en tener un programa donde centralizar la gestión básica de la empresa y además le permitiese agilizar ciertas acciones.

2. Objetivos.

El objetivo básico de este proyecto es crear una aplicación de escritorio para agilizar la gestión administrativa de la empresa, dado que en este momento todos los datos referentes a clientes, materiales y documentos de facturación son gestionados haciendo uso de diferentes documentos de Microsoft Office.

Por un lado, se almacenará toda la información en una BBDD y por otro lado se presentará una interfaz gráfica para facilitar visualizar dicha información e implementar la lógica de funcionamiento

Por falta de experiencia y debido a que el tiempo estipulado para finalizar el proyecto no se puede estimar, el objetivo inicial es presentar una primera versión básica de la aplicación ,

con una estructura de datos implementada, una interfaz gráfica sencilla y una serie de funcionalidades básicas para el correcto funcionamiento.

Posteriormente, una vez presentado el proyecto al jurado y finalizada la fase académica, se pondrá en marcha una segunda fase para estudiar junto con el cliente la inclusión de mejoras tanto en la interfaz como en la funcionalidad para mejorar la experiencia de usuario.

Sobre los datos a utilizar, obviamente debido a la ley de protección el día de la presentación del proyecto al Jurado no se presentarán datos reales. La carga de datos en el sistema se realizará posteriormente, una vez asegurado que la aplicación funciona correctamente y se puede poner en producción.

Una vez definidos los objetivos generales, podemos desgranar los mismos en dos grupos: de la aplicación y personales.

Objetivos de la aplicación por parte de la empresa:

- Centralizar la información de los clientes en una BBDD.
- Poder dar de alta, baja, consultar o modificar los datos de los clientes.
- Facilitar la generación de presupuestos, facturas y albaranes a partir de una serie de datos almacenados en la BBDD. La idea es facilitar su creación haciendo uso de los diferentes listados que tendremos almacenados: clientes, tareas y materiales, estos últimos con sus respectivos precios.
- Posibilitar el almacenamiento e impresión de dichos documentos.
- Calcular el precio final de los trabajos a realizar y reflejarlo en los documentos generados, teniendo en cuenta algunos detalles como la aplicación del IVA según el tipo de cliente.
- Almacenar un listado con las tareas y trabajos que suele realizar la empresa, acompañadas de su precio. Dicha información podrá ser editada, ya sea por cambiar el nombre de la tarea o modificar el precio en función de la fluctuación de los precios de mercado.
- Almacenar un listado con el material que se realizan los trabajos (mamparas, grifería, muebles de baño...) y toda la información referente a cada elemento del listado como puede ser su nombre, precio o alguna descripción. Dicha información podrá ser modificada.
- Implementar algún sistema de búsqueda que permita acceder a los diferentes documentos según el tipo de filtrado que deseemos. Por ejemplo, búsqueda de

facturas por clientes, o búsqueda de facturas por fechas para presentarlas posteriormente a la gestoría.

Objetivos personales:

- Dada mi inexperiencia en el sector del desarrollo y la programación, aprender a crear una aplicación desde cero y todo el trabajo que conlleva.
- Aprender a trabajar y comprender las diferentes fases que conlleva la generación de un proyecto: análisis, diseño, codificación, testeo y despliegue.
- Aprender el manejo de diferentes herramientas para la generación de documentación, diagramas, manuales...
- Implementar todos los conocimientos adquiridos durante el periodo de aprendizaje, así como recurrir, en caso necesario, a la búsqueda de información en diferentes fuentes ya sea basados en apuntes o libros o la cada vez más recurrente fuente que es Internet.
- Aprender a organizarme ante un proyecto de mucha mayor envergadura en comparación a las actividades y proyectos que se vienen trabajando durante la época escolar.
- Aprovechar la experiencia de las prácticas en empresa para intentar organizar el desarrollo del proyecto. Uso del método de framework SCRUM.
- Aprender de los errores y obstáculos que con total seguridad irán surgiendo en el camino, con el objetivo de coger experiencia y destreza para posibles proyectos futuros.

3. Estado del arte.

El mundo de la tecnología crece a un ritmo frenético. Cada vez son más los aspectos de nuestras vidas que están influenciados por la tecnología. Desde nuestro trabajo, con la implementación de nuevas tecnologías y metodologías de producción, a nuestras relaciones personales con el uso cada vez mayor de las redes sociales.

Están nuevas tecnologías ofrecen muchas oportunidades para mejorar nuestro día a día, y por eso es importante aprovecharlas para nuestro propio beneficio.

Obviamente las pequeñas empresas y Pymes también deben saber aprovecharse de estos avances tecnológicos con el fin de aumentar su producción y optimizar su rendimiento y economía. Para ello, además de los avances tecnológicos propios del sector para el que trabajen, es importante contar con un software de gestión empresarial que ayude en la administración de las empresas, por pequeña o grande que sea.

La tendencia en el mercado es el uso de software que incluya un sistema ERP, o lo que es lo mismo un sistema de planificación de recursos empresariales. Este tipo de software cubre todas las necesidades internas de cualquier empresa, desde la producción y distribución a la gestión de recursos humanos.

Las ventajas que ofrece este sistema son:

- Automatización de los procesos de la empresa.
- Almacenamiento de toda la información en una sola plataforma.
- Ahorro en tiempo y costes. Esta todo centralizado.
- En algunos casos, soluciones BI (Business Intelligence) que permite ver el estado de la empresa prácticamente en tiempo real.
- También hay tendencia al uso de CRM, o Customer Relationship Manager. Básicamente se trata de un análisis de las interacciones con los clientes con el fin de anticipar necesidades, ver tendencias, optimizar rentabilidad y ofrecer al cliente una oferta personalizada según sus necesidades y preferencias.

La principal desventaja de este tipo de software es el coste, ya que este tipo de software suele estar modularizado, con el fin de que las empresas lo personalicen según sus necesidades. A mayor nivel de personalización, mayor coste. Además, hoy la tendencia de las compañías de software es ofrecer su producto bajo un modelo de suscripción, eliminando la opción de un pago único y definitivo.

Otra desventaja, aunque cada vez menor, es la implementación del servicio y disponer de los recursos HW y de almacenamiento necesarios para poder realizar el despliegue, aunque esto

cada vez nos supone un menor quebradero de cabeza para los administradores de sistemas, primero con la aparición de la virtualización que facilita el poder separar la producción de las labores de administración y testeo, y posteriormente con la tendencia al uso de sistemas en la nube que permiten un “alquiler” externo de recursos tanto de hardware como de almacenamiento.

Como ejemplo de software que implementa ERP tenemos **HOLDED**, que ofrece un modelo de suscripción en base al volumen de clientes e información generada, y **ODOO**, que es un Open Source ERP y cuyo modelo de suscripción está basado en la cantidad de módulos que necesitemos. Existen ERP's gratuitas como Flowlu o Bitrix. Y como version más minimalista y que puede interesar a cualquier pequeña Pyme o autónomo es hacer uso de **MS-ACCESS**.

4. Estudio de viabilidad.

4.1. Estudio de mercado.

El anterior apartado creo que sirve como análisis la industria en lo que respecta a software de gestión empresarial, viendo el estado del mercado en el presente y hacia donde se dirige en el futuro, con lo cual estamos en disposición de crear un análisis DAFO para evaluar donde nos encontramos respecto al mercado y un análisis CAME para ver posibles correcciones. Obviamente al tratarse de un primer proyecto real, estamos en clara desventaja. Dichos análisis se muestran en las tablas que aparecen en las siguientes páginas.

Para terminar este apartado, una vez marcados los objetivos, haber realizado un pequeño estudio y análisis del mercado en cuanto a software de gestión empresarial y haber analizado en que posición nos encontramos dentro del mercado, el siguiente paso es realizar un análisis más exhaustivo sobre cuanto podría costar un proyecto real, aproximadamente y que recursos vamos a utilizar.

ANALISIS DAFO	
Debilidades	<ul style="list-style-type: none"> - Falta de experiencia en sector. Es mi primer proyecto. - Falta de personal para abordar el proyecto. Surgen dudas con el objetivo y el tiempo estimado para la entrega - Rentabilidad nula. Es un proyecto de fin de curso y su desarrollo va a ser gratuito. - Falta de conocimiento o soltura en el manejo de ciertas tecnologías.
Amenazas	<ul style="list-style-type: none"> - Hay una gran cantidad de software especializado en el mercado. - Los cambios continuos en la tecnología. - El uso de productos sustitutivos para gestiones en pequeñas empresas o a nivel autónomo, como hojas de cálculos sobre software de libre uso (Open Office, ...)
Fortalezas	<ul style="list-style-type: none"> - Es mi primer proyecto, no tengo presión por parte del cliente. - Experiencia en el sector de la administración de sistemas, útil para trazar un plan y tener cierta capacidad de análisis. - Mi producto será personalizado. Adaptado a la necesidad del cliente. - Los costes tanto a nivel de personal como de uso de software son nulos. No hay pérdidas.
Oportunidades	<ul style="list-style-type: none"> - La posibilidad de realizar las FCT'S en una empresa abre la oportunidad de aprender metodologías y demás. - Al cliente le interesa el producto, pues tanto el desarrollo como posterior mantenimiento serán gratuitos. - Posibilidad de sacar un buen producto y ofertarlo a otras empresas o autónomos del sector de la fontanería, construcción. Promoción.

ANÁLISIS CAME	
Corregir Debilidades	<ul style="list-style-type: none"> - Aprovechar para coger experiencia, aprender de los posibles errores. - Aprender de la experiencia para realizar una estimación mejor de los plazos de entrega, para posibles futuros proyectos. - Intentar sacar un buen producto, para sacarle rentabilidad en posteriores distribuciones. - Aprovechar para aprender el manejo de tecnologías que permitan aumentar la producción y la vez ofrecer un producto más robusto y con mejor presencia.
Afrontar Amenazas	<ul style="list-style-type: none"> - Estudiar el software que hay en el mercado, que ofrece y como lo hace. Ofrecer al cliente una experiencia similar. - Estar atentos a los cambios tecnológicos, para ver futuras tendencias y hacia donde tenemos que orientar nuestro producto. - Intentar ofrecer al cliente un producto profesional.
Mantener Fortalezas	<ul style="list-style-type: none"> - No meterme presión. Es mi primer proyecto y su objetivo es aprender. - Aprovechar mi experiencia laboral y mi metodología de trabajo usada durante años. - Intentar no salir de los requisitos establecidos por el cliente e intentar que el software sea de su agrado. - No gano dinero, pero a cambio obtengo experiencia y posibilidad de mercado.
Explotar Oportunidades	<ul style="list-style-type: none"> - Aprovechar la oportunidad de las FCT's para coger experiencia laboral y obtener conocimientos de nuevas tecnologías. - Mantener el interés del cliente. Debe de estar informado de los avances y en cierta parte hay que hacerle participe en él. - Ofrecer un buen producto, que satisfaga y aprovechar su posición el mercado como posible promoción del producto.

4.1.1. Viabilidad técnica del proyecto.

En cuanto a la viabilidad técnica del proyecto, es asequible, ya que el software va a ser desplegado en un solo puesto.

Se trata de una aplicación sencilla en cuanto al volumen de datos que se van a manejar, por lo que en principio no se requiere de un dispositivo con altas prestaciones tanto para su desarrollo como para su posterior despliegue y uso, lo que significa que con los recursos HW con los que contamos son correctos para realizar el proyecto.

No se descarta en el futuro la posibilidad de poder acceder de forma remota a la aplicación, pero habría que estudiar por un lado la seguridad de los datos, por la ley de protección de datos y por otro lado el coste económico de tener un equipo 24 horas encendido. Además, el hospedaje de la BBDD se tendría que llevar a cabo en un equipo preparado para soportar esas condiciones de trabajo, lo que aumentaría el coste del proyecto.

Se propuso al cliente la opción de uso de la nube para el hospedaje de los datos, pero esta opción fue descartada por el tema de la ley de protección de datos.

4.1.2. Viabilidad económica del proyecto

En primer lugar, es obvio que para la empresa la viabilidad económica del proyecto es factible, dado que no va a tener ningún coste, de momento. Con lo cual podríamos poner punto final a este apartado. Pero vamos a analizar cuanto podría costar el proyecto

¿Cuánto costaría hacer uso del software del mercado? Sin tener en cuenta las labores que conlleva el posterior despliegue del software o la carga de datos, vamos a analizar el coste de 2 productos que cualquier empresa puede obtener en el mercado: Holded y Odoo.

Holded ofrece un servicio de suscripción basado principalmente en la cantidad de información a manejar en base al número de facturas anuales, contactos y usuarios que van a usar el software. El software incluye módulos como facturación, CRM y ventas o gestión de proyectos. Para el tamaño de la empresa para la que voy a hacer el proyecto, el coste inicial sería el siguiente:

- Plan plus (500 facturas al año, 250 contactos, 2 usuarios) ... 9 euros/mes

Por su parte Odoo también ofrece un servicio de suscripción, pero en este caso está basado por un lado en la cantidad de usuarios que vayan a utilizar el software y por otro lado en la

selección de los módulos, dentro de una basta colección de ellos, que serán necesarios. Para contratar algo similar a Holded, el precio aproximado sería el siguiente :

- 2 usuarios..... 20 euros/mes
- Modulo CRM 12 euros/mes
- Facturación 6 euros/mes
- Ventas 6 euros/mes
- Total 44 euros/mes

En los anteriores costes no se han tenido en cuenta los costes de personal necesario para realizar las labores necesarias para la puesta en marcha de las aplicaciones, desde un análisis o diseño de cómo se van a integrar los datos hasta la carga de ellos.

Hay que tener en cuenta que por ejemplo Odoo soporta la generación de nuevos módulos, así como la personalización de los que ya hay, mediante el uso de lenguaje de programación Python, etiquetas XML...

Para realizar un cálculo estimado de cuánto puede costar el desarrollo de la aplicación en cuanto al personal necesario, tenemos que basarnos en el último convenio colectivo para la industria, la tecnología y los servicios del metal de la provincia de Valencia, publicado en el Boletín Oficial de la Provincia de Valencia el pasado 26 de enero de 2022. Según la última tabla salarial, el salario de un analista y un programador es el siguiente:

	Analista	Programador
Salario Anual	31523 € anuales	24073 € anuales
Salario por Horas	21.89 €/hora	16.71 €/hora

Una vez calculados los precios, haremos uso del diagrama de Gantt para calcular el coste del desarrollo de la aplicación.

4.1.3. Recursos HW.

Como he comentado anteriormente la aplicación se va a desplegar inicialmente en un solo equipo, cuyas características son las siguientes:

Ordenador Empresa	
Marca	HP
Modelo	All-in-One 22-df0xxx
Procesador	Intel Celeron J4025 CPU 2.00GHz
Memoria RAM	8.00 Gb
Sistema Operativo	Windows 11 Home Edition

Para recrear el mismo entorno de trabajo, dadas las posibilidades reales de poder hacerlo por los medios propios que dispongo, el equipo para el desarrollo de la aplicación es el siguiente:

Ordenador Desarrollo	
Marca	HP
Modelo	Laptop 15-da1xxx
Procesador	Intel Core i5-8265U 1.60GHz (8ª gen)
Memoria RAM	8.00 Gb
Sistema Operativo	Windows 11 Home Edition

4.1.4. Recursos SW.

El software para el desarrollo de la aplicación es gratuito y no requiere de licencia de pago para su uso. El software para a utilizar es el siguiente:

- **MariaDB** v10.7.3 : SGBD donde se hospedarán los datos que se utilizarán en la aplicación. Desarrollado a partir de un fork de MySQL y dado el cariz comercial que adquirió este último, es un software indicado para el desarrollo gratuito de aplicaciones basadas en proyectos personales o para clientes. Corre bajo licencia GPL v2. Dispone de los conectores necesarios para poder enlazar los datos con aplicaciones desarrolladas en Java.
- **Eclipse id for Java Developers** v2021-12 (4.22.0) : Id para el desarrollo la aplicación. Permite tanto la creación de la interfaz gráfica como la implementación de la lógica de programación para poder interactuar con los datos del SGBD. Es un software de código abierto que utiliza la licencia EPL, Eclipse Public License, licencia creada por la Eclipse Foundation y que viene a sustituir la licencia CPL. Permite el uso de la version OpenJDK para el desarrollo de aplicaciones basadas en Java, bajo licencia GNU y sin tener que pagar licencia para el desarrollo de software comercial.

El despliegue del software en el equipo del cliente va a ser bastante similar y no va a requerir de grandes recursos hardware ya que por un lado se le instalará la plataforma XAMPP que alberga el SGBD donde se van a almacenar los datos y por otro lado se le va a proporcionar la aplicación de escritorio con la que se va a poder trabajar.

4.1.5. Recursos humanos.

Para realizare este proyecto en el tiempo determinado para su entrega, según el calendario escolar, se necesitarían básicamente 2-3 personas: un analista y 1-2 desarrolladores.

El analista llevaría a cabo las siguientes funciones:

- Llevaría a cabo las fases de análisis y diseño
- Entrevista con el cliente, ver sus necesidades.
- Estudio y análisis del problema planteado.
- Establecer los requisitos adecuados para implementar la solución.

- Diseño tanto del sistema a implementar como de los diferentes diagramas.
- Orientar al programador en sus cometidos.

El programador llevaría a cabo las siguientes operaciones:

- Llevaría a cabo las labores de codificación, testeo y mantenimiento.
- Básicamente traducir a código todo el diseño creado por el analista.
- Implementar en la BBDD la estructura de los datos.
- Realizar la carga inicial de datos.
- Desarrollar la interfaz gráfica.
- Implementar la lógica de programación para interactuar con los datos.
- Realizar las correspondientes pruebas para comprobar el correcto funcionamiento.
- Desplegar la aplicación.
- Realizar labores de mantenimiento.

Obviamente en este proyecto voy a ser el único recurso humano, por lo que me convierto en la figura de analista-programador haciendo cargo de todas las fases por la que transcurre la creación de un proyecto de software.

4.1.6. Viabilidad temporal.

- pendiente

4.2. Planificación temporal.

- adjuntar anexo con diagrama de Gantt (pendiente)

5. Análisis de requisitos.

5.1. Descripción de requisitos.

Identificación de Actores	
Actor	Descripcion Básica de Funciones
Administrador	Sera el encargado del desarrollo y mantenimiento del software, así como de la carga y actualización en el SGBD de los diferentes los diferentes datos para su uso, como clientes iniciales, material, listado de tareas, precios...
Usuario	Será la persona que va a realizar el uso cotidiano del software. Se encargará de la gestión de usuarios, generación de documentos(albaranes, presupuestos y facturas), consultas y copias de seguridad.
Cliente	No hará ningún uso del software, pero es la persona que va a encargar la realización de servicios a la empresa.

Requisitos No Funcionales	
ID	Descripcion Básica
RNF1	El administrador será el único que puede acceder directamente al SGBD, con todos los permisos de administración.
RNF2	Se le darán los permisos necesarios al usuario para realizar las operaciones de consulta inserción, modificación o borrado de clientes dentro del SGBD.
RNF3	Se le darán los permisos necesarios al usuario para realizar las operaciones de consulta Y modificación de piezas, materiales y tareas dentro del SGBD.
RNF4	Se le darán los permisos necesarios al usuario para realizar las operaciones de generación y consulta de albaranes, presupuestos y facturas dentro del SGBD.
RNF5	Se controlará que todas las transacciones, operaciones, cambios en los datos y demás se hagan correctamente y sin ningún error.
RNF6	Se creará un procedimiento para la copia de seguridad de los datos almacenados en la BBDD, a ser posible por duplicado y en distintas localizaciones.
RNF7	Los datos serán almacenados de manera local, en el mismo dispositivo de usuario y se buscará una alternativa para una segunda copia de seguridad a través de un dispositivo externo o de un equipo remoto.
RNF8	No se hará en ningún caso uso de servicios basados en la nube para el hospedaje y almacenamiento de copias de seguridad.
RNF9	La frecuencia de la copia de seguridad de los datos se hará diaria, semanal y mensualmente.
RNF10	Se implementará una interfaz gráfica que facilite el trabajo y la interacción del usuario con los datos almacenados en el SGBD.
RNF11	La aplicación no será desplegada o entregada hasta que se compruebe su correcto funcionamiento. Se harán las pruebas correspondientes para comprobar posibles fallos y errores.

Requisitos Funcionales		
ID	Nombre	Descripcion Básica
RF1	Login	El usuario debe introducir las credenciales para poder acceder al sistema.
RF2	Login Correcto	El sistema validará las credenciales y permitirá el acceso del usuario.
RF3	Login Incorrecto	El sistema mostrará un mensaje de error si las credenciales del usuario son incorrectas. Por lo tanto, el usuario no podrá acceder al sistema.
RF4	Gestión Clientes	El usuario podrá gestionar el listado de clientes del sistema. De los clientes necesitamos un id que será generado por el Sistema, DNI, nombre, apellidos, dirección, código postal, teléfono de contacto y email.
RF5	Alta Cliente	El usuario podrá dar de alta clientes en el Sistema.
RF6	Baja Cliente	El usuario podrá eliminar clientes del Sistema.
RF7	Modificar Cliente	El usuario podrá modificar los datos de los clientes del Sistema.
RF8	Búsqueda Cliente	El usuario podrá buscar clientes en el Sistema.
RF9	Gestión Material/Trabajos	El usuario podrá gestionar el listado de material/trabajos que utiliza o realiza la empresa. De cada material/trabajo debemos conocer un id que será generado por el sistema , la categoría principal, el nombre del material/trabajo, su descripción, precio original, precio de incremento y precio final.
RF10	Alta Material/Trabajo	El usuario podrá dar de alta material o trabajos, en el Sistema.
RF11	Baja Material/Trabajo	El usuario podrá dar de baja material o trabajos, en el Sistema.
RF12	Modificar Material/Trabajo	El usuario podrá modificar los datos correspondientes de cada material o trabajo, en el Sistema
RF13	Búsqueda Material/Trabajo	El usuario podrá buscar materiales o trabajos, en el Sistema.

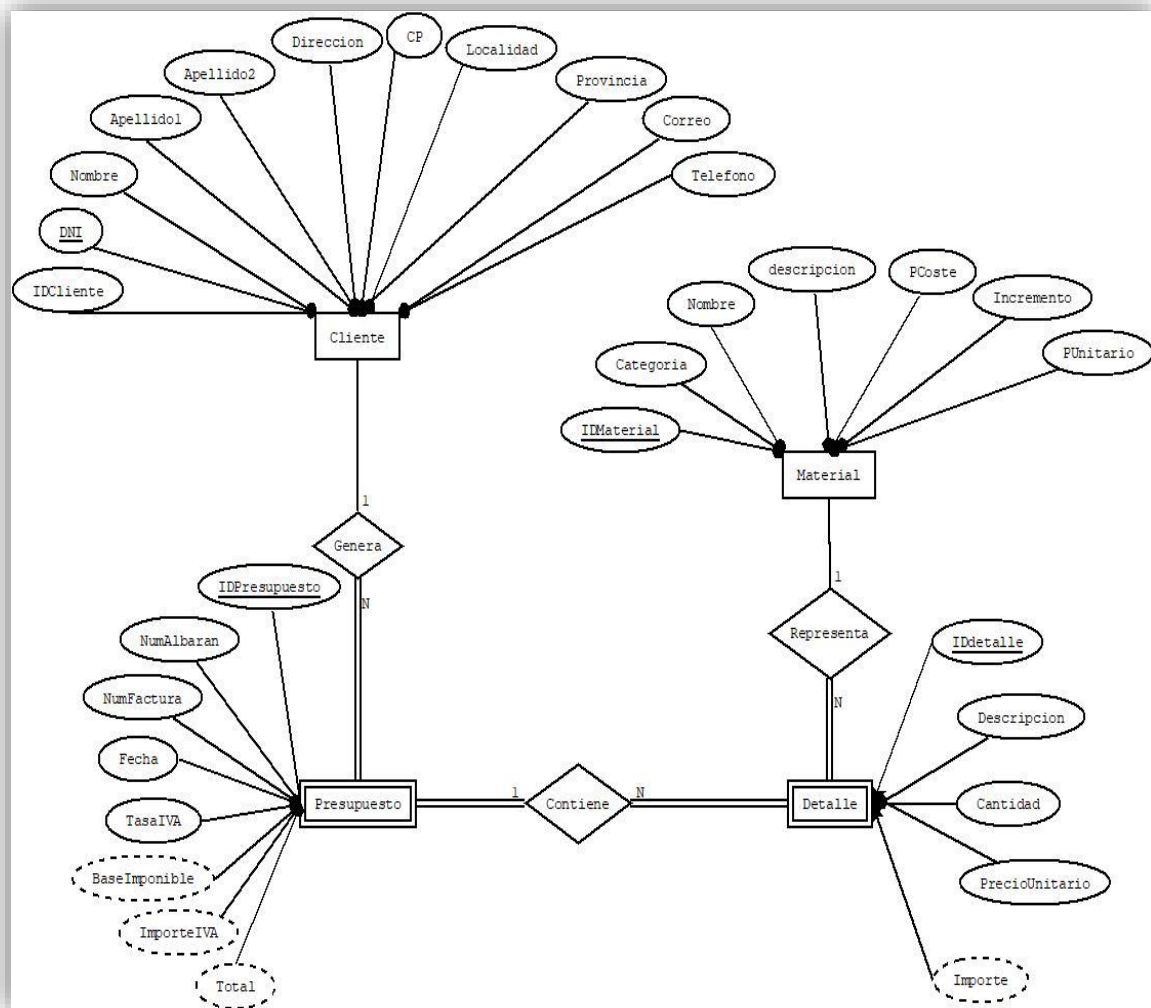
Requisitos Funcionales (continuación)		
ID	Nombre	Descripción Básica
RF14	Gestión Documentación	El usuario podrá gestionar la documentación que necesite la empresa para realizar sus servicios. La documentación está formada por presupuestos, albaranes y facturas. La cabecera consta del logo de la empresa, datos de la empresa, datos del cliente y la fecha de generación del documento. El cuerpo de la cabecera estará formado por un listado de materiales o tareas, con el nombre de la tarea, las unidades utilizadas, precio de la Unidad y precio completo. Finalmente, el pie del documento estará formado por el importe inicial, el IVA que se aplica, el importe del IVA y el importe final.
RF15	Generar Documentación	El usuario podrá generar la documentación que necesite, en el Sistema.
RF16	Modificar IVA	El usuario podrá modificar el IVA que se debe aplicar según el tipo de cliente.
RF17	Calculo precios	El sistema se encargará de calcular el importe inicial, el importe final y el importe final en cada uno de los documentos. Serán campos calculados.
RF18	Consultar Documentación	El usuario podrá consultar los documentos.
RF19	Guardar Documentación	El usuario podrá guardar la información en formato PDF para su almacenamiento y en caso necesario su impresión.

5.2. Diagrama de casos de uso.



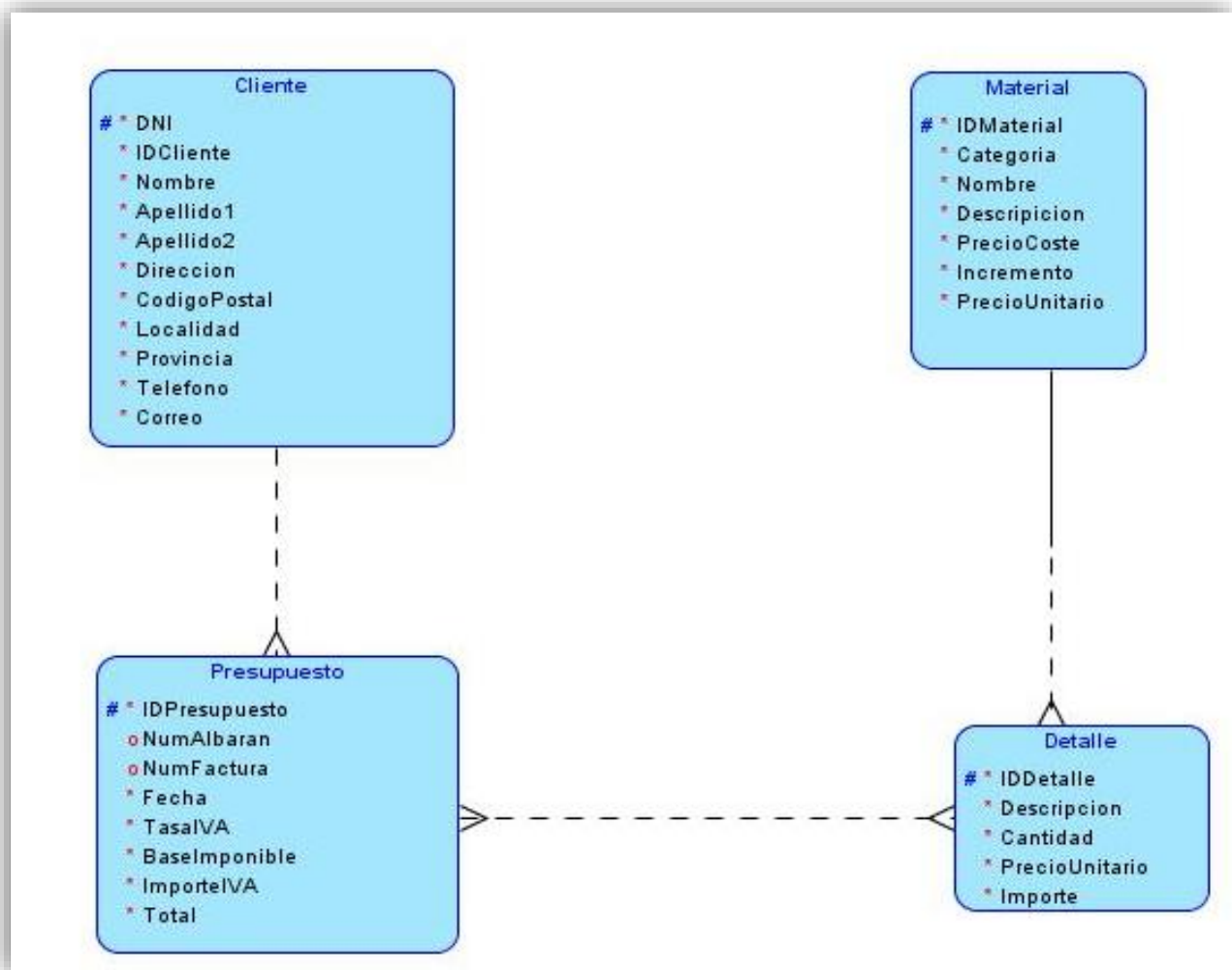
6. Diseño.

6.1. Diseño conceptual E/R.



En el diagrama podemos observar que cuatro entidades. Por un lado, tenemos cliente y material, que son entidades fuertes debido a que no necesitan o no dependen de ninguna otra entidad para su construcción. Por otro lado, tenemos presupuesto y detalle que, aunque gozan de su propia clave primaria, considero que son entidades débiles debido a que necesitan una clave ajena para su construcción. En el caso de la entidad presupuesto, no tiene sentido realizar uno sino existe un cliente y en el caso detalle, que especifica el material que se usa para realizar un trabajo y que debe ser incluido en el presupuesto, no tiene sentido sino hace referencia a un material.

6.2.Diseño Lógico Relacional.

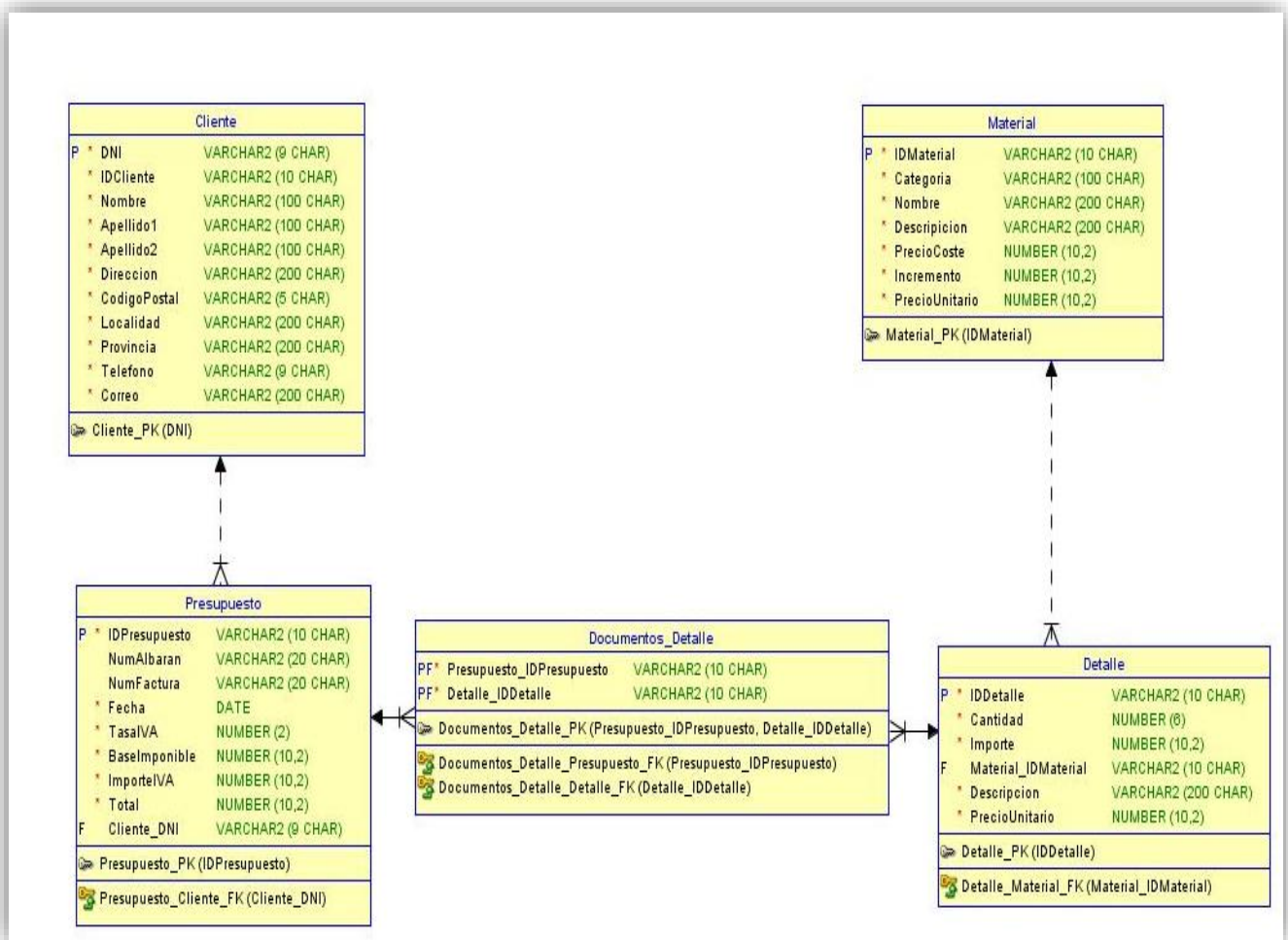


Este es el esquema lógico de la BBDD, donde se puede ver la estructura de los datos. Vemos las diferentes tablas que componen la BBDD, así como las relaciones de cardinalidad entre ellas.

Aunque en este diagrama no se muestra, para su elaboración he tenido que empezar a declarar el tipo de dato para cada uno de los campos así como las relaciones entre tablas. Esta información se muestra en el diagrama relacional de la siguiente sección.

Como podemos ver la estructura de la BBDD se divide en 4 tablas. Las tablas cliente, presupuesto y material representan los datos básicos con los que mi cliente trabaja. La tabla detalle nos va a permitir almacenar el listado de material o trabajos que se reflejan en los documentos.

En el apartado descripción de tablas se verá una descripción más detallada de cada una de las tablas y campos.



Este es el esquema relacional de la BBDD. Además de la estructura de los datos y las relaciones de cardinalidad entre las diferentes tablas, podemos ver :

- El tipo de dato de cada uno de los campos, junto con su longitud
- Las relaciones entre las diferentes mediante la definición de los campos que se van a utilizar como clave foránea.

Como novedad aparece una nueva tabla, “Documento_detalle”, que sirve para enlazar los documentos que el cliente debe generar con su composición. Intentando dar una definición más concreta, cada presupuesto, factura o albarán está compuesto por un listado de materiales o trabajos que están representado en la tabla detalle, por lo que la tabla “documento_detalle” sirve como tabla para relacionar cada documento con sus detalles correspondientes.

En el apartado descripción de tablas se verá una descripción mas detallada de cada una de las tablas y campos.

6.3. Diseño físico.

En este apartado aprovecho el código que construye la BBDD para mostrar de un modo detallado los campos que componen cada una de las tablas junto con su tamaño. También se ve la declaración de las claves primarias , y también de las secundarias para poder establecer conexión entre tablas.

En el apartado descripción de tablas se verá una descripción mas detallada de cada una de las tablas y campos.

#borramos la BBDD si existe

```
DROP DATABASE IF EXISTS fontaneria;
```

#creamos la BBDD

```
CREATE DATABASE fontaneria;
```

```
USE fontaneria;
```

#creamos la tabla cliente

```
CREATE TABLE cliente (  
    DNI VARCHAR(9),  
    IDCliente VARCHAR(10) NOT NULL,  
    Nombre VARCHAR(100) NOT NULL,  
    Apellido1 VARCHAR(100) NOT NULL,  
    Apellido2 VARCHAR(100) NOT NULL,  
    Direccion VARCHAR(200) NOT NULL,  
   CodigoPostal VARCHAR(5) NOT NULL,  
    Localidad VARCHAR(200) NOT NULL,  
    Provincia VARCHAR(100) NOT NULL,  
    Telefono VARCHAR(13) NOT NULL,  
    Correo VARCHAR(200) NOT NULL,
```



```
CONSTRAINT cliente_pk PRIMARY KEY (DNI),  
CONSTRAINT cliente_doc UNIQUE (IDCliente)  
);
```

#creamos la tabla material

```
CREATE TABLE material (  
    IDMaterial VARCHAR(14),  
    Categoria VARCHAR(100) NOT NULL,  
    Nombre VARCHAR(200) NOT NULL,  
    Descripcion VARCHAR(200) NOT NULL,  
    PrecioCoste DECIMAL (10,2) NOT NULL,  
    Incremento DECIMAL (10,2) NOT NULL,  
    PrecioUnitario DECIMAL(10,2) NOT NULL,  
    CONSTRAINT material_pk PRIMARY KEY (IDMaterial)  
);
```

#creamos la tabla detalle

```
CREATE TABLE detalle (  
    IDDetalle VARCHAR(10),  
    Material_IDMaterial VARCHAR(14) NOT NULL,  
    Descripcion VARCHAR(200),  
    Cantidad INTEGER(6) NOT NULL,  
    PrecioUnitario DECIMAL(10,2) NOT NULL,  
    Importe DECIMAL(10,2) NOT NULL,  
    CONSTRAINT detalle_pk PRIMARY KEY (IDDetalle),  
    CONSTRAINT detalle_fk FOREIGN KEY (Material_IDMaterial) REFERENCES material  
    (IDMaterial)  
);
```

#creamos la tabla presupuesto

CREATE TABLE presupuesto (

IDPresupuesto VARCHAR(10),

Cliente_DNI VARCHAR(10) NOT NULL,

NumAlbaran VARCHAR(10),

NumFactura VARCHAR(10),

Fecha DATE NOT NULL,

TasaIVA INTEGER(2) NOT NULL,

BaseImponible DECIMAL(10,2) NOT NULL,

ImporteIVA DECIMAL(10,2) NOT NULL,

Total DECIMAL(10,2) NOT NULL,

CONSTRAINT documento_pk PRIMARY KEY (IDPresupuesto),

CONSTRAINT documento_fk FOREIGN KEY (Cliente_DNI) REFERENCES cliente (DNI)

);

#creamos la tabla documento_detalle

CREATE TABLE documento_detalle (

Presupuesto_IDPresupuesto VARCHAR(10),

Detalle_IDDetalle VARCHAR(10),

CONSTRAINT documento_detalle_pk PRIMARY KEY (Presupuesto_IDPresupuesto,
Detalle_IDDetalle),

CONSTRAINT documento_detalle_fk1 FOREIGN KEY (Presupuesto_IDPresupuesto)
REFERENCES presupuesto (IDPresupuesto),

CONSTRAINT documento_detalle_fk2 FOREIGN KEY (Detalle_IDDetalle) REFERENCES
detalle (IDDetalle)

);

#Todas las restricciones se aplicarán en código Java.

#problemas con los constraint check con MySQL

6.4. Descripción de tablas y campos.

CLIENTE	
DESCRIPCION DE LA TABLA	
En esta tabla se van a almacenar los datos correspondientes a los clientes. Se compone de una serie de campos donde se registran los datos personales que permitirán identificar a los clientes y contactar con ellos en caso necesario. Adicionalmente se crea un campo con un identificador de cliente que se utilizará en los albaranes.	
CAMPOS	
NOMBRE DEL CAMPO	DNI
TIPO	VARCHAR
TAMAÑO	9 caracteres
DETALLES	Clave primaria
DESCRIPCION	Almacena el DNI del cliente.
RESTRICCIONES	Formato: 8 numeros + 1 letra de control
NOMBRE DEL CAMPO	IDCliente
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	UNIQUE, NOT NULL
DESCRIPCION	Identificador alternativo del cliente. Este código se usará exclusivamente como parte del número de albarán.
RESTRICCIONES	Formato: "C." + numero. El número será autogenerado, teniendo en cuenta el orden de los clientes en la tabla.
NOMBRE DEL CAMPO	Nombre
TIPO	VARCHAR
TAMAÑO	100 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Nombre del cliente.
RESTRICCIONES	Solo puede estar compuesto por letras.
NOMBRE DEL CAMPO	Apellido1
TIPO	VARCHAR
TAMAÑO	100 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Primer apellido del cliente.
RESTRICCIONES	Solo puede estar compuesto por letras.
NOMBRE DEL CAMPO	Apellido2
TIPO	VARCHAR(100)
TAMAÑO	100 caracteres
DETALLES	NOT NULL
DESCRIPCION	Segundo apellido del cliente.
RESTRICCIONES	Solo puede estar compuesto por letras.

CLIENTE (continuación)	
CAMPOS	
NOMBRE DEL CAMPO	Direccion
TIPO	VARCHAR
TAMAÑO	200 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Direccion del cliente.
RESTRICCIONES	Ninguna.
NOMBRE DEL CAMPO	CodigoPostal
TIPO	VARCHAR(5)
TAMAÑO	5 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Almacena el código postal de la localidad del clientes
RESTRICCIONES	Formato: 5 numeros.
NOMBRE DEL CAMPO	Localidad
TIPO	VARCHAR(200)
TAMAÑO	200 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Almacena la localidad donde reside el cliente.
RESTRICCIONES	Solo puede estar compuesto por letras
NOMBRE DEL CAMPO	Provincia
TIPO	VARCHAR(200)
TAMAÑO	200 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Almacena la provincia a la que pertenece la localidad
RESTRICCIONES	Solo puede estar compuesto por letras.
NOMBRE DEL CAMPO	Telefono
TIPO	VARCHAR(13)
TAMAÑO	13 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Telefono de contacto del cliente
RESTRICCIONES	9 dígitos obligatorios, opcional formato internacional
NOMBRE DEL CAMPO	Correo
TIPO	VARCHAR(200)
TAMAÑO	200 caracteres.
DETALLES	NOT NULL
DESCRIPCION	Direccion de correo electrónico del cliente.
RESTRICCIONES	Formato: nombre + @ + dominio

MATERIAL	
DESCRIPCION DE LA TABLA	
<p>En esta tabla se almacenan los distintos materiales y productos que ofrece o utiliza la empresa en sus tareas, las cuales también serán incluidas en dicha tabla. Además de una serie de campos que permiten identificar cada uno de los materiales, productos y tareas, también hay una serie de campos que almacenan los distintos precios de cada uno de ellos. Estos precios no son definitivos, ya que cuando la empresa deba realizar algún presupuesto, tendrá libertad para modificar estos precios dependiendo de algunos factores como por ejemplo la fluctuación de precios en el mercado.</p>	
CAMPOS	
NOMBRE DEL CAMPO	IDMaterial
TIPO	VARCHAR
TAMAÑO	14 caracteres
DETALLES	Clave primaria
DESCRIPCION	Identificador único de cada material, producto o tarea.
RESTRICCIONES	Formato creado a partir de la fecha/hora del registro.
NOMBRE DEL CAMPO	Categoria
TIPO	VARCHAR
TAMAÑO	100 caracteres
DETALLES	NOT NULL
DESCRIPCION	Representa el tipo de material (Grifo, Mueble Baño...)
RESTRICCIONES	Ninguno. Definido en un comboBox.
NOMBRE DEL CAMPO	Nombre
TIPO	VARCHAR
TAMAÑO	200 caracteres
DETALLES	NOT NULL
DESCRIPCION	Nombre específico del material, producto o tarea.
RESTRICCIONES	Ninguno
NOMBRE DEL CAMPO	Descripcion
TIPO	VARCHAR
TAMAÑO	200 caracteres
DETALLES	NOT NULL
DESCRIPCION	Descripcion del material
RESTRICCIONES	Ninguno
NOMBRE DEL CAMPO	PrecioCoste
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Precio inicial o de fabrica del material.
RESTRICCIONES	Formato numérico con decimales

MATERIAL (continuación)	
CAMPOS	
NOMBRE DEL CAMPO	Incremento
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Cantidad que se incrementa al precio coste.
RESTRICCIONES	Formato numérico.
NOMBRE DEL CAMPO	PrecioUnitario
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Precio final del material.
RESTRICCIONES	PrecioCoste + Incremento.

PRESUPUESTO	
DESCRIPCION DE LA TABLA	
<p>En esta tabla se van a almacenar los presupuestos que se van a ir generando y que permitirán posteriormente, una vez aprobados y cerrados, crear sus correspondientes albaranes y facturas. No siempre un presupuesto es cerrado, de ahí la libertad para que los campos pertenecientes al número de albarán y factura sean nulos.</p>	
CAMPOS	
NOMBRE DEL CAMPO	IDPresupuesto
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	Clave primaria
DESCRIPCION	Numero para identificar el presupuesto. Los presupuestos no son documentos obligatorios, por lo que este valor no tiene mayor importancia que identificar el documento para el funcionamiento interno del programa.
RESTRICCIONES	Valor autogenerado a partir del nº de presupuestos existentes.
NOMBRE DEL CAMPO	Cliente_DNI
TIPO	VARCHAR
TAMAÑO	9 caracteres
DETALLES	Clave ajena que referencia a la tabla cliente.
DESCRIPCION	Almacena el DNI del cliente.
RESTRICCIONES	Formato: 8 numeros + 1 letra de control

PRESUPUESTO (continuación)	
CAMPOS	
NOMBRE DEL CAMPO	NumAlbaran
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	
DESCRIPCION	Numero para identificar el albarán, que se podrá generar una vez aprobado el presupuesto.
RESTRICCIONES	Formato: IDCliente + "-" + numDoc + "/" + los dos últimos dígitos del año curso. El numDoc será el mismo para el albarán y la factura. Un ejemplo sería "C.11-001/22"
NOMBRE DEL CAMPO	NumFactura
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	
DESCRIPCION	Numero para identificar la factura, que se podrá generar una vez aprobado el presupuesto.
RESTRICCIONES	Formato: numDoc + "/" + los 4 dígitos del año curso. El numDoc será el mismo para el albarán y la factura. Un ejemplo sería "001/2022"
NOMBRE DEL CAMPO	Fecha
TIPO	DATE
DETALLES	NOT NULL
TAMAÑO	Determinado por el SGBD
DESCRIPCION	Fecha que se crea el presupuesto.
RESTRICCIONES	Ninguna.
NOMBRE DEL CAMPO	TasaIVA
TIPO	NUMBER
TAMAÑO	Precisión 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Tasa del IVA a implementar al cliente.
RESTRICCIONES	Campo numérico
NOMBRE DEL CAMPO	BaselImponible
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Importe total sin aplicar IVA
RESTRICCIONES	Campo numérico
NOMBRE DEL CAMPO	ImporteIVA
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Importe del IVA a aplicar.
RESTRICCIONES	BaselImponible * TasaIVA / 100

PRESUPUESTO (continuación)	
CAMPOS	
NOMBRE DEL CAMPO	Total
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Precio final de la factura
RESTRICCIONES	BaseImponible + ImporteIVA

DETALLE	
DESCRIPCION DE LA TABLA	
En esta tabla se va a almacenar el listado de los materiales, productos y tareas que se van utilizando o realizando y que quedarán reflejados en los distintos documentos.	
CAMPOS	
NOMBRE DEL CAMPO	IDDetalle
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	Clave primaria
DESCRIPCION	Numero para identificar el número del detalle.
RESTRICCIONES	Código autogenerado.
NOMBRE DEL CAMPO	Material_IDMaterial
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	Clave ajena que referencia a la tabla material.
DESCRIPCION	Identificador único de cada material, producto o tarea.
RESTRICCIONES	Formato creado a partir de la fecha/hora del registro.
NOMBRE DEL CAMPO	Cantidad
TIPO	NUMBER
TAMAÑO	Precisión 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Cantidad de cada material, producto o tarea.
RESTRICCIONES	Campo numérico
NOMBRE DEL CAMPO	Descripcion
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	NOT NULL
DESCRIPCION	Campo que describe el material, producto o tarea.
RESTRICCIONES	Ninguno

DETALLE (continuación)	
CAMPOS	
NOMBRE DEL CAMPO	PrecioUnitario
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Precio de cada unidad de material, producto o tarea.
RESTRICCIONES	Campo Numérico
NOMBRE DEL CAMPO	Importe
TIPO	DECIMAL
TAMAÑO	Precisión 10 numeros, escala 2 numeros
DETALLES	NOT NULL
DESCRIPCION	Precio final de cada material, producto o tarea.
RESTRICCIONES	Precio Unitario + Cantidad

DOCUMENTOS_DETALLE	
DESCRIPCION DE LA TABLA	
Tabla de apoyo para relacionar los diferentes documentos con el listado de detalles(materiales, productos o tareas) que forman parte de dichos documentos y que permitirán la generación de presupuestos, facturas y albaranes.	
CAMPOS	
NOMBRE DEL CAMPO	Detalle_IDDetalle
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	Clave primaria y clave ajena que referencia a la tabla detalle.
DESCRIPCION	Numero para identificar el número del detalle.
RESTRICCIONES	Código autogenerado.
NOMBRE DEL CAMPO	Presupuesto_IDPresupuesto
TIPO	VARCHAR
TAMAÑO	10 caracteres
DETALLES	Clave primaria y clave ajena que referencia a la tabla presupuesto.
DESCRIPCION	Numero para identificar el presupuesto. Los presupuestos no son documentos obligatorios, por lo que este valor no tiene mayor importancia que identificar el documento para el funcionamiento interno del programa.
RESTRICCIONES	Pendiente el formato

6.5. Orientación a objetos.

6.5.1. Diagramas de clases.

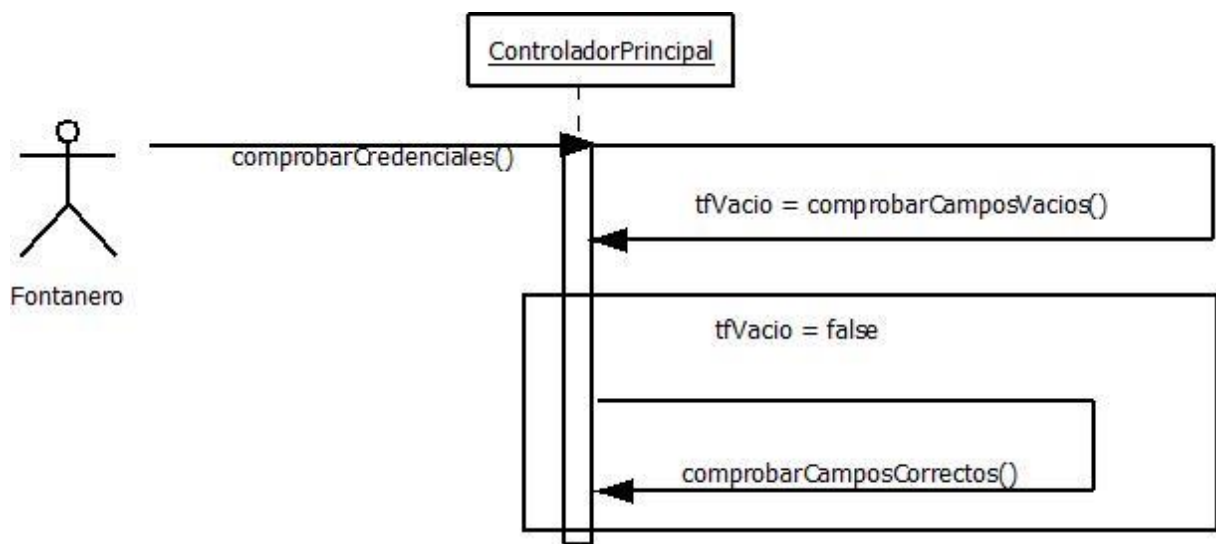
En la parte que respecta a Java, no se puede establecer ningún diagrama UML de clases para ver las relaciones entre ellas, ya que al tratarse de una aplicación desarrollada sobre JavaFX, casi todas las clases son de tipo controlador para dar funcionalidad a los distintos elementos de cada escena, así como realizar operaciones internas de cálculos y demás.

También he desarrollado una serie de interfaces donde se almacenan una serie de métodos que son comunes al resto de clases, como pueden ser operaciones contra la BBDD o métodos para controlar el formato de algunos campos.

Los elementos principales de la aplicación, como los clientes o los materiales, y sus relaciones quedan establecidas en la BBDD y se puede seguir en los anteriores diagramas.

6.5.2. Diagramas de secuencias.

- **Secuencia de login** (Requisitos RF1, RF2, RF3). El usuario llama al método `comprobarCredenciales()` de la clase `ControladorPrincipal` para comprobar usuario y password para acceder al sistema. Dentro del método se realiza una llamada al método `comprobarCamposVacios()` para comprobar que los campos usuario y password no estén vacíos. Si los campos no están vacíos, se llama al método `comprobarCamposCorrectos()` para comprobar que las credenciales de acceso sean las correctas.



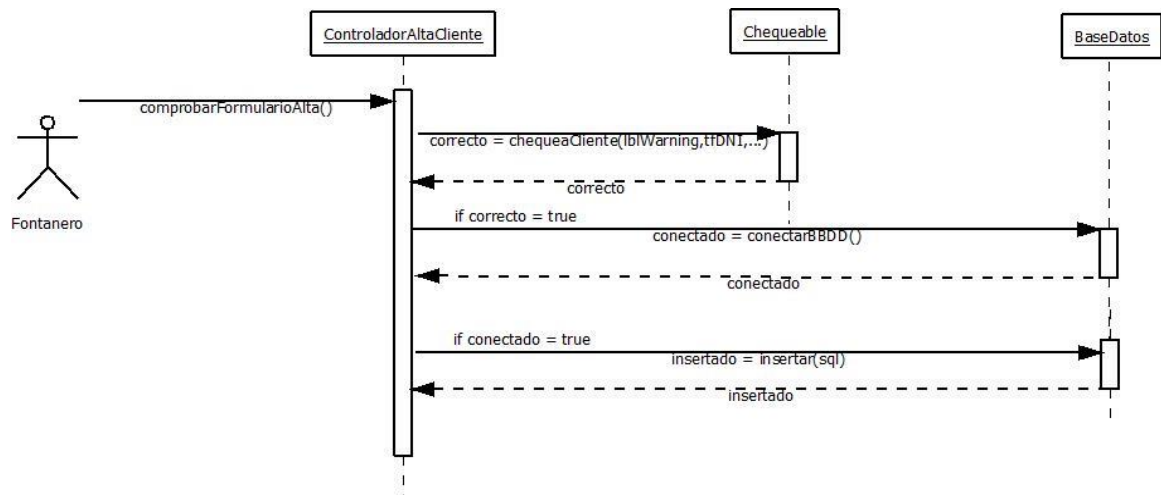
- Secuencia alta cliente (Requisitos RF5)

1º Llamada al método comprobarFormularioAlta de la clase ControladorAltaCliente para iniciar la comprobación de los datos.

2º Llamada al método chequeaCliente de la interfaz chequeable para comprobar que los datos siguen unas reglas y patrones. Se devuelve un booleano

3º Llamada al método conectar BBDD de la interfaz BaseDatos para intentar la conexión con nuestro SGBD. Devuelve un booleano

4º Llamada al método insertar de la interfaz BaseDatos para registrar los datos en la BBDD.



-Secuencia alta material (Requisitos RF5)

1º Llamada al método comprobarCampos de la clase ControladorAltaMaterial para iniciar la comprobación de los datos.

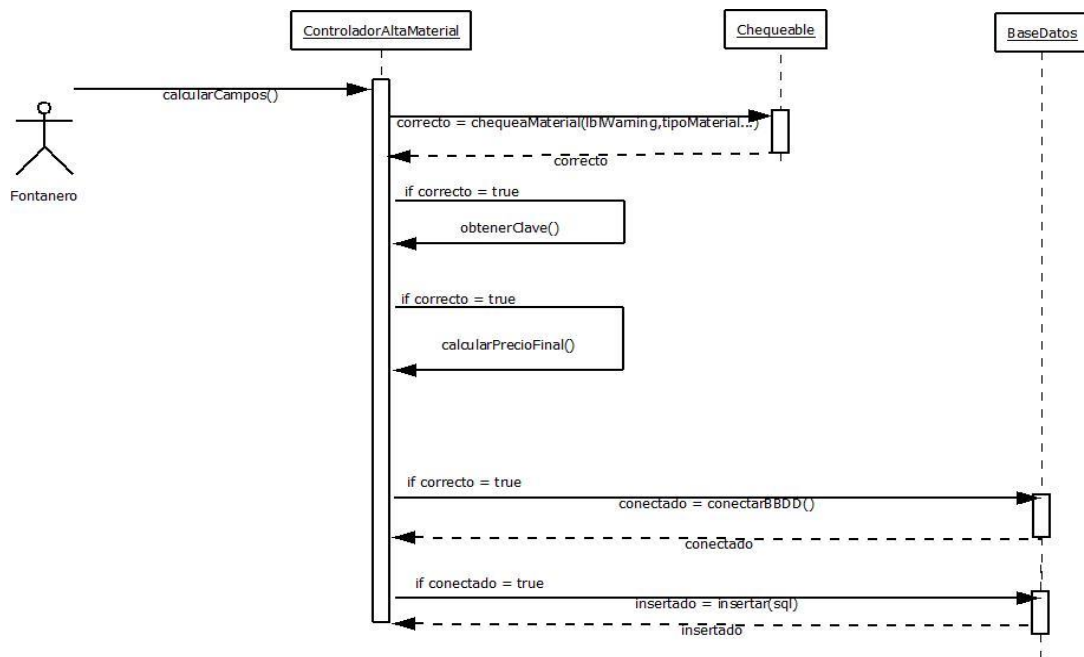
2º Llamada al método chequeaCliente de la interfaz chequeable para comprobar que los datos siguen unas reglas y patrones. Se devuelve un booleano

3º Llamada al método obtenerClave de la clase ControladorAltaMaterial para obtener la clave primaria del material.

4º Llamada al método calcularPrecioFinal de la clase ControladorAltaMaterial para calcular el precio final del material

5º Llamada al método conectar BBDD de la interfaz BaseDatos para intentar la conexión con nuestro SGBD. Devuelve un booleano

6º Llamada al método insertar de la interfaz BaseDatos para registrar los datos en la BBDD.



- Secuencia búsqueda cliente (Requisitos RF8)

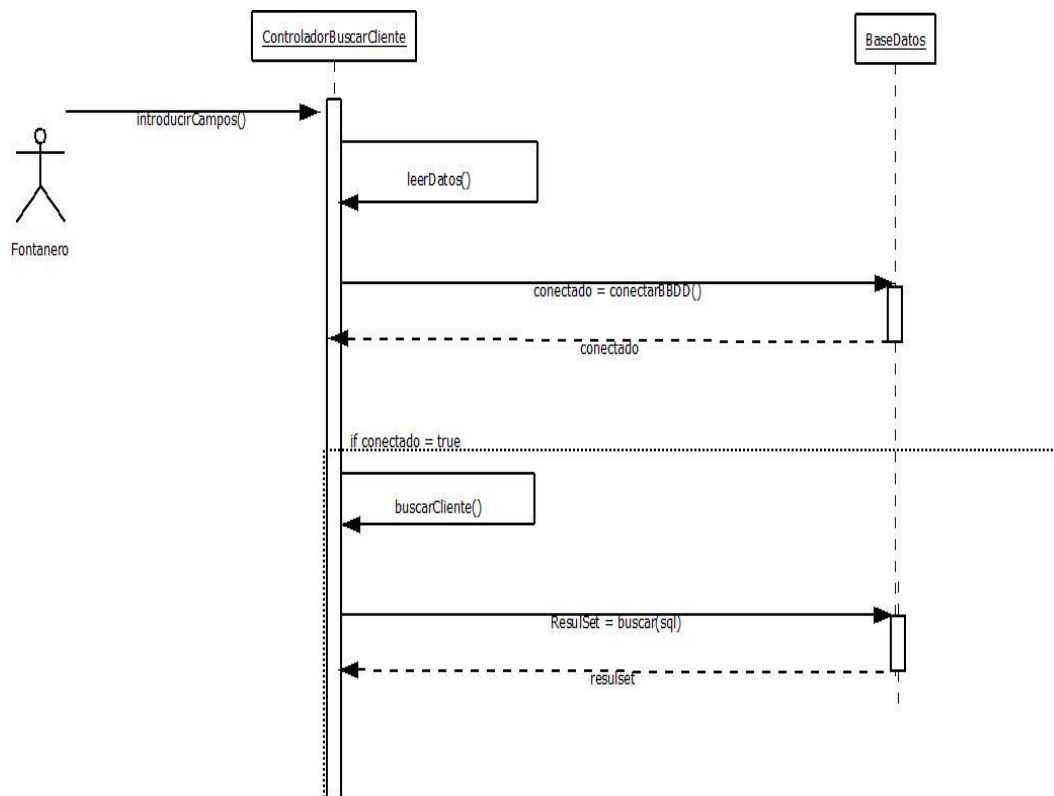
1º Llamada al método introducirCampos de la clase ControladorBsucarCliente para iniciar la búsqueda de un cliente bien por su DNI o bien por su nombre y apellidos.

2º Llamada al método leerDatos de la clase ControladorBuscarMaterial para iniciar la conexión con el SGBD.

3º Llamada al método conectar BBDD de la interfaz BaseDatos para intentar la conexión con nuestro SGBD. Devuelve un booleano.

4º Llamada al método buscarCliente de la clase ControladorBsucarCliente para construir el select.

5º Llamada al método buscar de la interfaz BaseDatos para buscar los datos en la BBDD. Devuelve un resultSet con el resultado obtenido en la búsqueda.



- Secuencia búsqueda Material (Requisitos RF13)

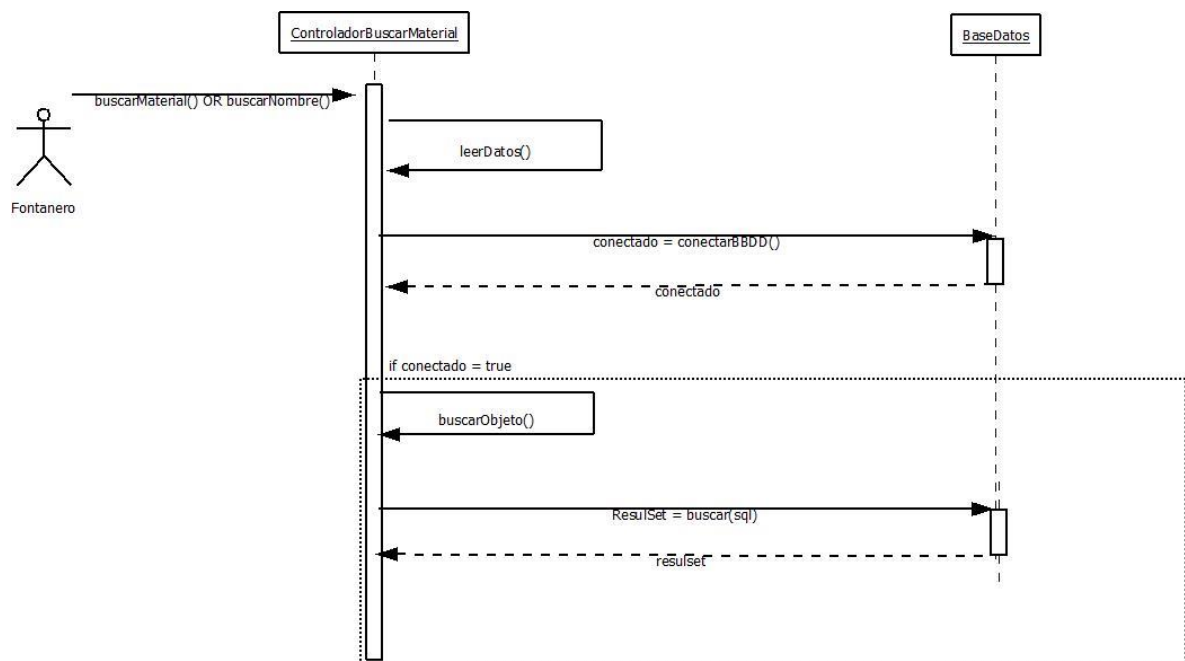
1º Llamada a uno de los métodos, `bsucarMaterial` o `buscarNombre`, de la clase `ControladorBuscarMaterial` para iniciar la búsqueda de un material bien por su tipo o bien por su nombre y apellidos.

2º Llamada al método `leerDatos` de la clase `ControladorBuscarMaterial` para iniciar la conexión con el SGBD.

3º Llamada al método `conectar BBDD` de la interfaz `BaseDatos` para intentar la conexión con nuestro SGBD. Devuelve un booleano.

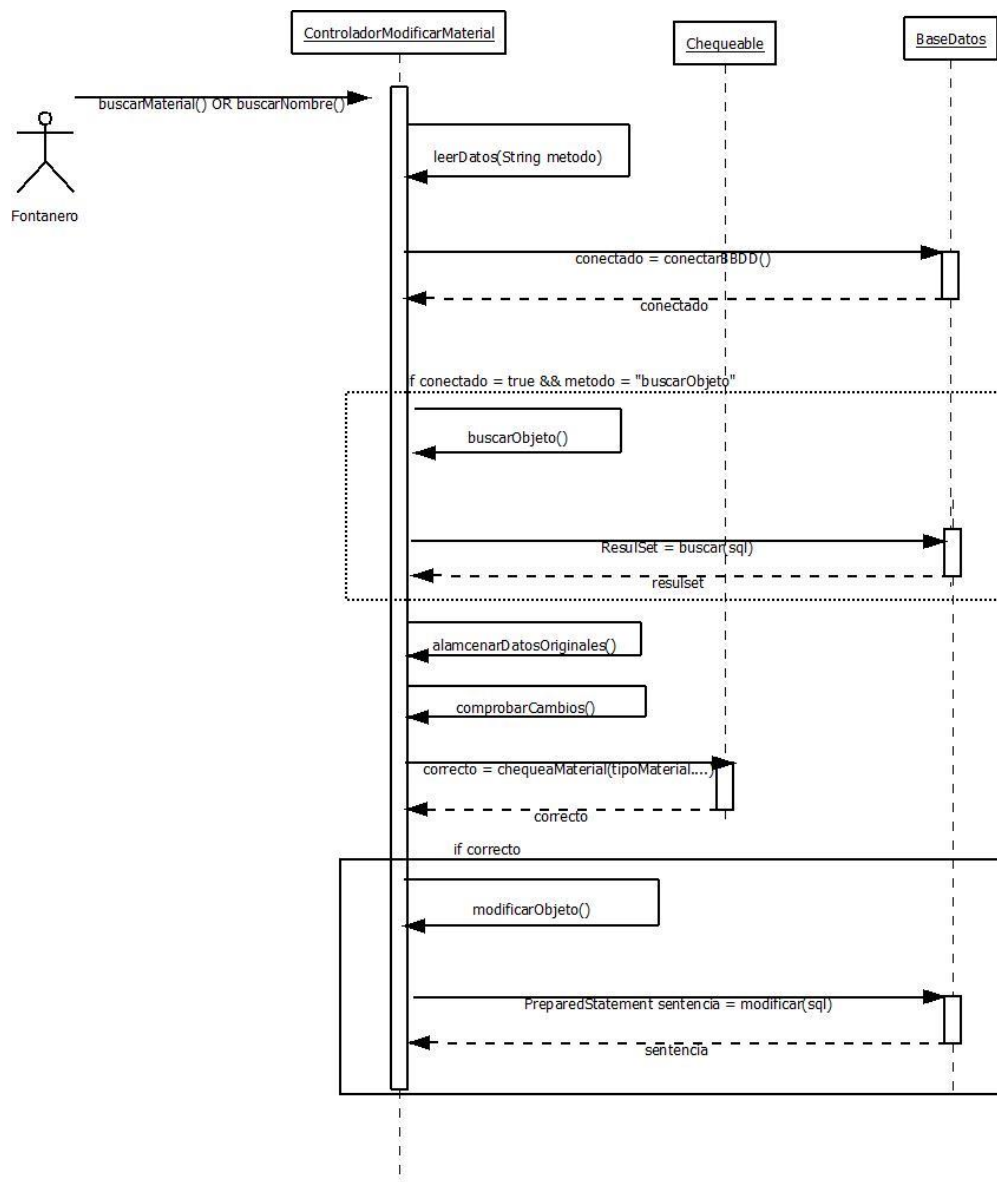
3º Llamada al método `buscarObjeto` de la clase `ControladorBsucarMaterial` para construir el select.

4º Llamada al método `buscar` de la interfaz `BaseDatos` para buscar los datos en la BBDD. Devuelve un `resultSet` con el resultado obtenido en la búsqueda.



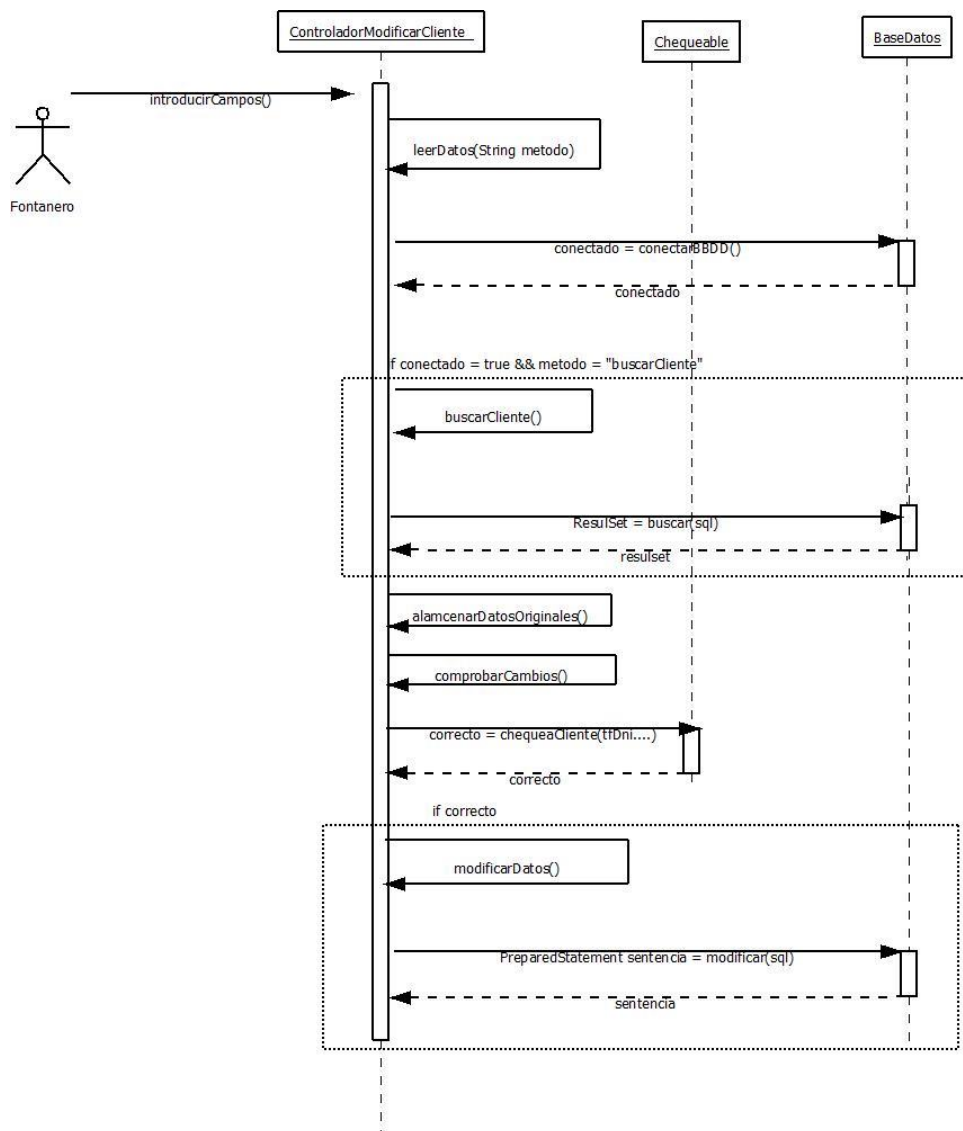
- Secuencia modificar Material (Requisitos RF12)

Básicamente es una mezcla de las secuencias de alta y búsqueda de material. En primer lugar, tenemos que buscar un material, ya sea mediante la selección de un tipo o un nombre. Tras conectarnos a la BBDD debemos realizar una consulta con los datos seleccionados. La búsqueda nos llevará a un registro de la tabla y se mostrarán los campos de dicho registro en pantalla, pudiendo ser modificados. Cada vez que encontremos un registro, almacenamos los valores en un objeto temporal y así podremos comprobar si hay cambios en alguno de los campos al pulsar el botón correspondiente. Si hay cambios, se llama a la BBDD para realizar el Update correspondiente.



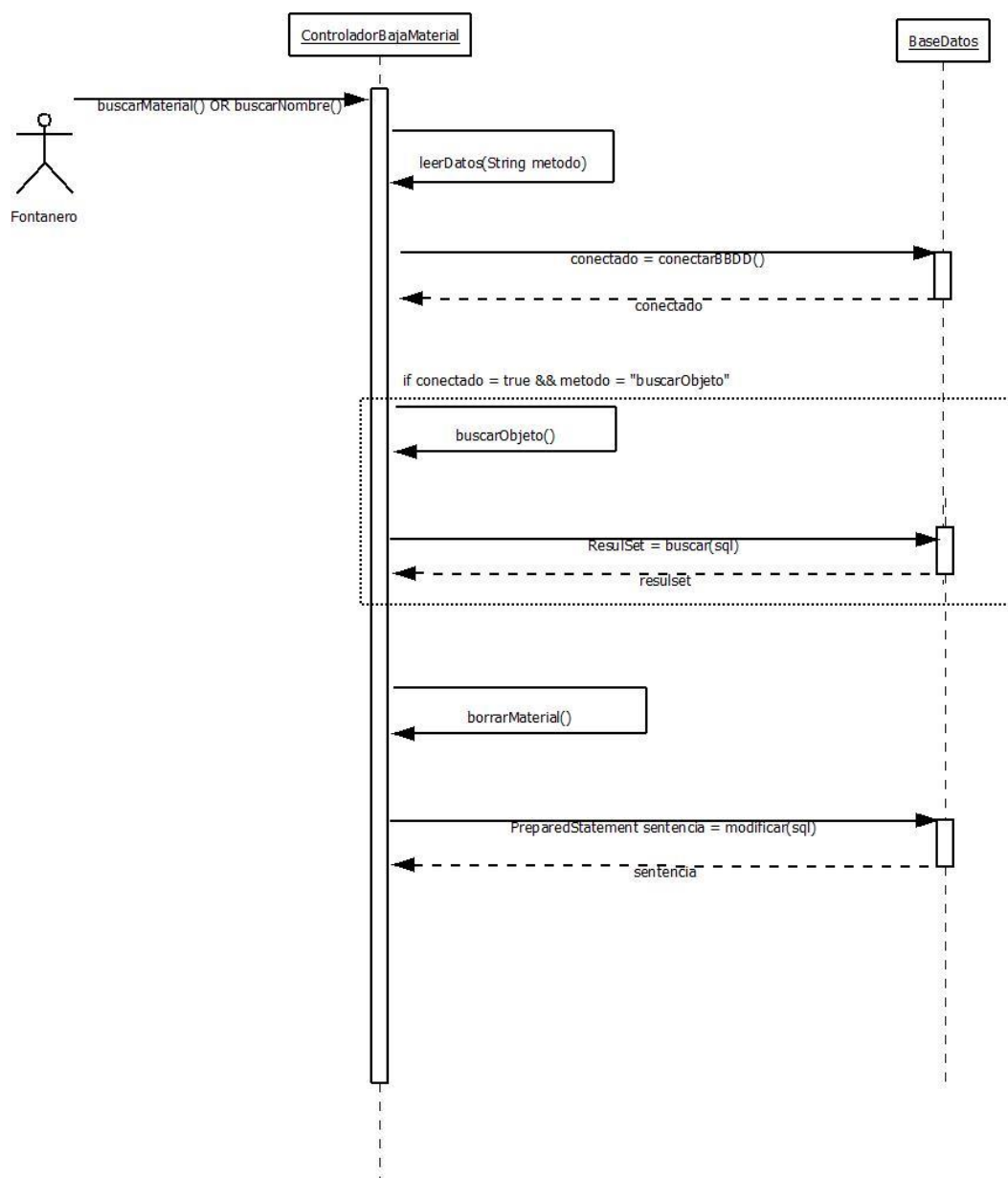
- Secuencia modificar Cliente (Requisitos RF7)

Básicamente es una mezcla de las secuencias de alta y búsqueda de cliente. En primer lugar, tenemos que buscar un cliente, ya sea mediante su DNI o su nombre y apellidos. Tras conectarnos a la BBDD debemos realizar una consulta con los datos seleccionados. La búsqueda nos llevará a un registro de la tabla y se mostrarán los campos de dicho registro en pantalla, pudiendo ser modificados. Cada vez que encontremos un registro, almacenamos los valores en un objeto temporal y así podremos comprobar si hay cambios en alguno de los campos al pulsar el botón correspondiente. Si hay cambios, se llama a la BBDD para realizar el Update correspondiente.



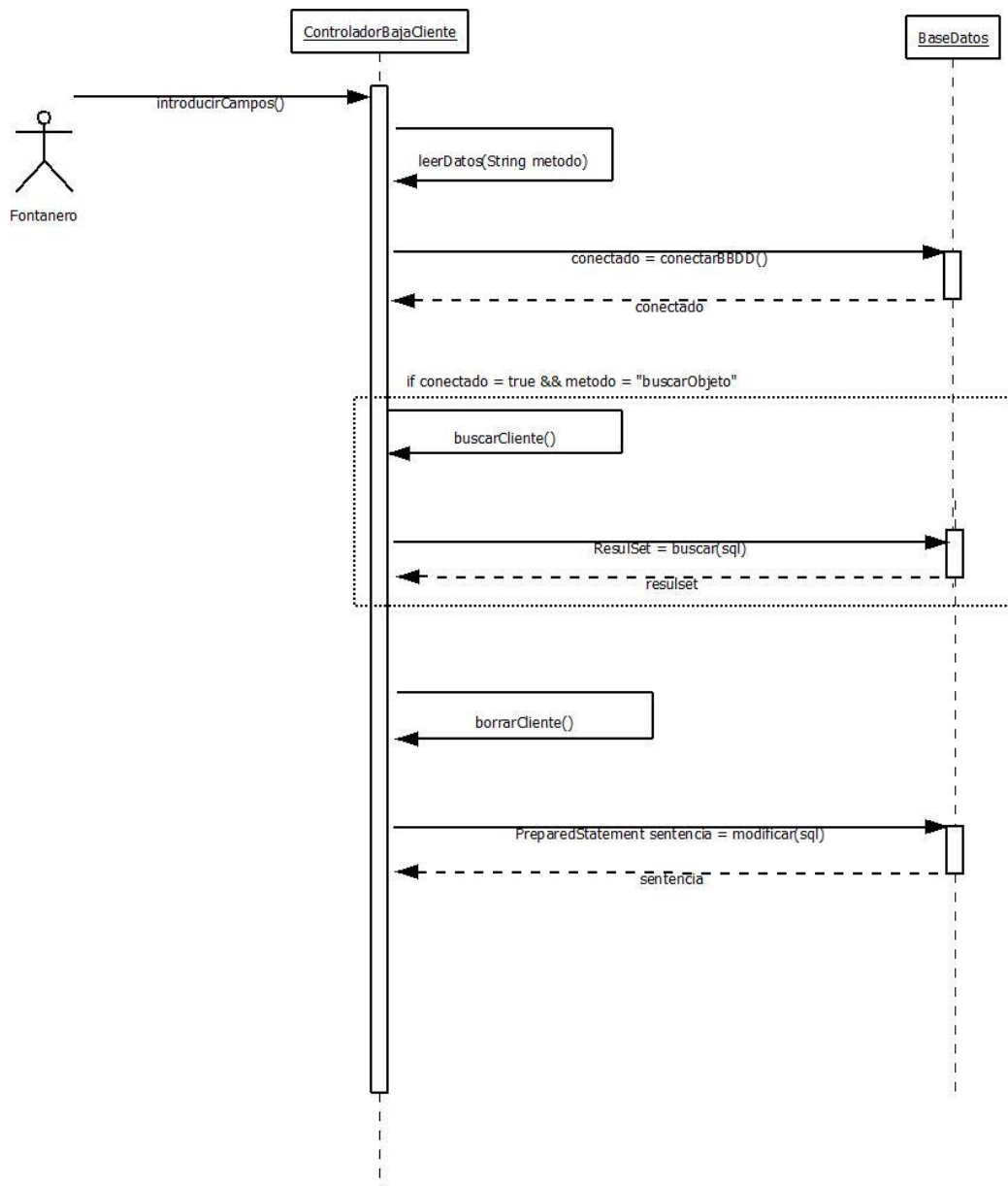
Secuencia Baja Material (Requisitos RF11)

Este apartado es similar al de búsqueda de material con el añadido que permite el borrado de un registro. En primer lugar, tenemos que buscar un material, ya sea mediante la selección de un tipo o un nombre. Tras conectarnos a la BBDD debemos realizar una consulta con los datos seleccionados. La búsqueda nos llevará a un registro de la tabla y se mostraran los campos de dicho registro en pantalla, pudiendo eliminar dicho registro. Cada vez que decidamos borrar algún elemento, se hará una llamada al SGBD para realizar la operación DELETE correspondiente.



Secuencia Baja Cliente (Requisitos RF11)

Este apartado es similar al de búsqueda de cliente con el añadido que permite el borrado de un registro. En primer lugar, tenemos que buscar un cliente, ya sea mediante su DNI o su nombre y apellidos. Tras conectarnos a la BBDD debemos realizar una consulta con los datos seleccionados. La búsqueda nos llevará a un registro de la tabla y se mostraran los campos de dicho registro en pantalla, pudiendo eliminar dicho registro. Cada vez que decidamos borrar algún elemento, se hará una llamada al SGBD para realizar la operación DELETE correspondiente.



6.5.3. Diagrama de actividad.

Secuencia de login (Requisitos RF1, RF2, RF3).

Diagrama de actividad para la secuencia de login.

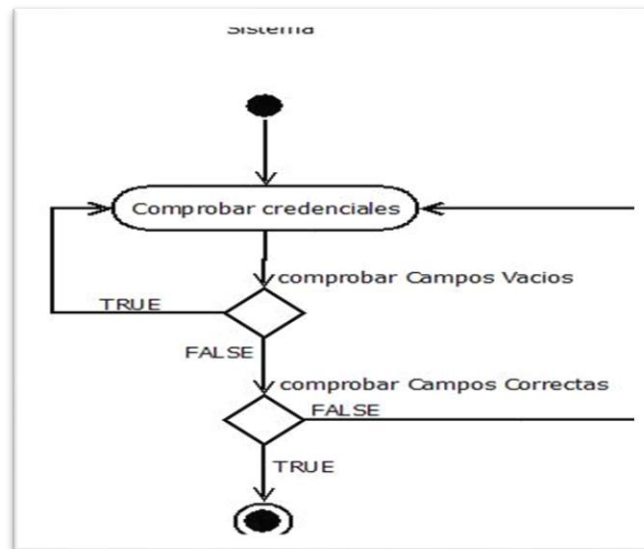
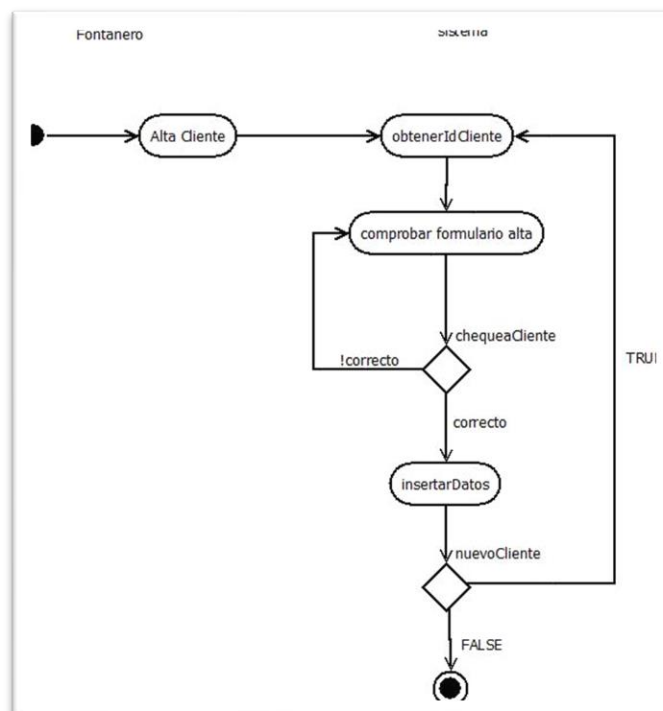
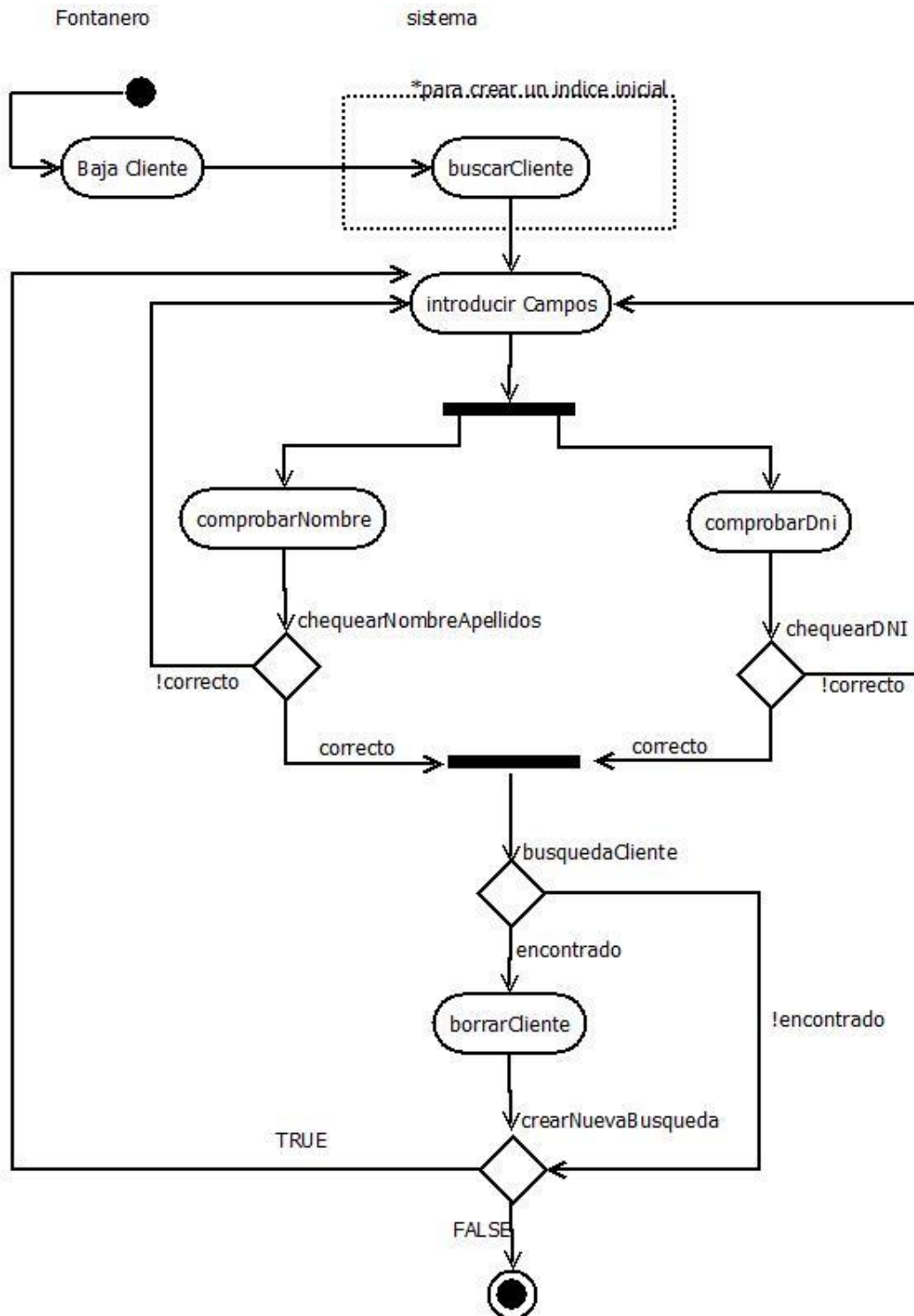
**Secuencia alta cliente** (Requisitos RF5)

Diagrama de actividad para la secuencia de alta de un cliente en el SGBD.



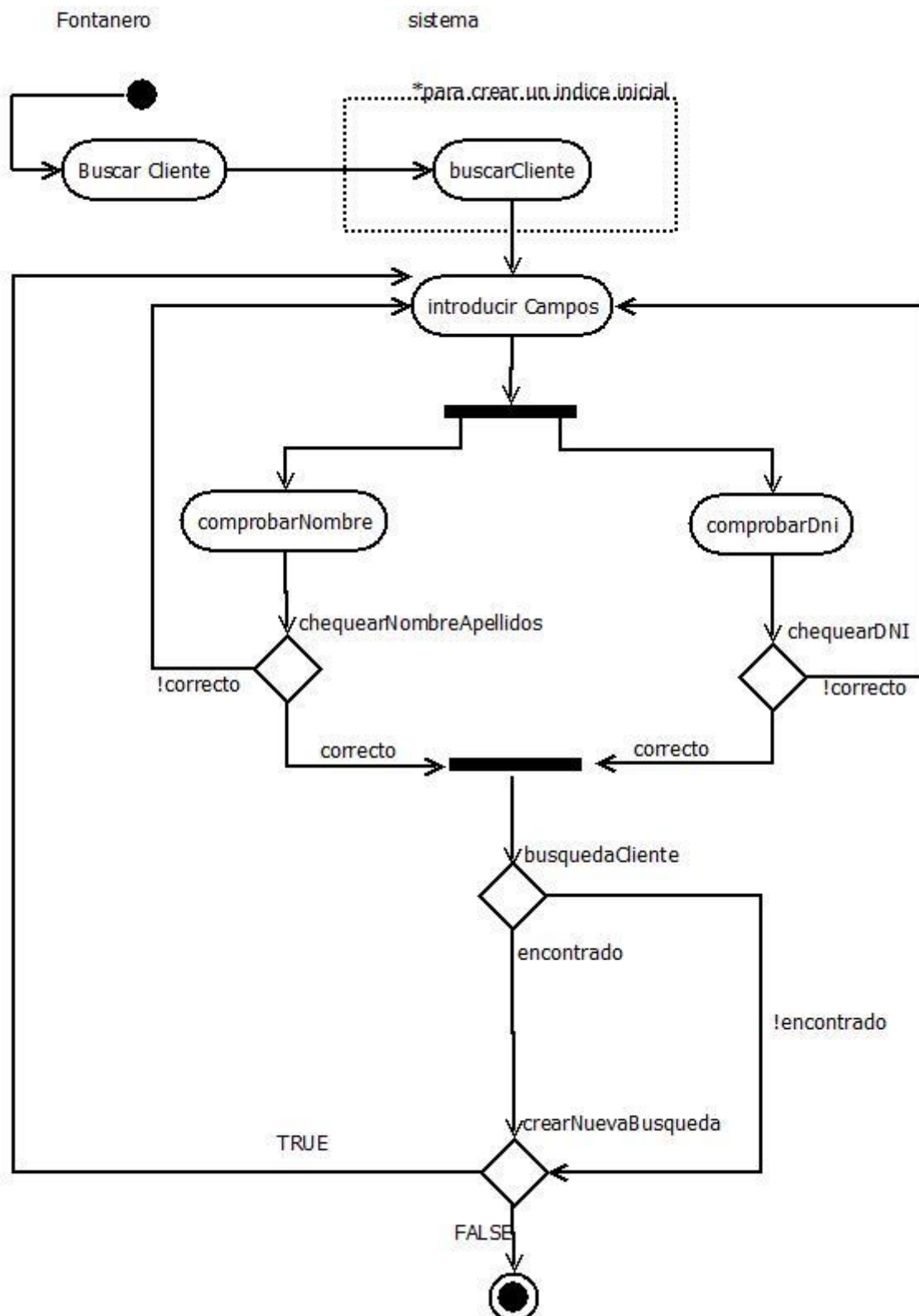
Secuencia baja cliente (Requisitos RF6)

Diagrama de actividad para la secuencia de baja de un cliente en el SGBD.



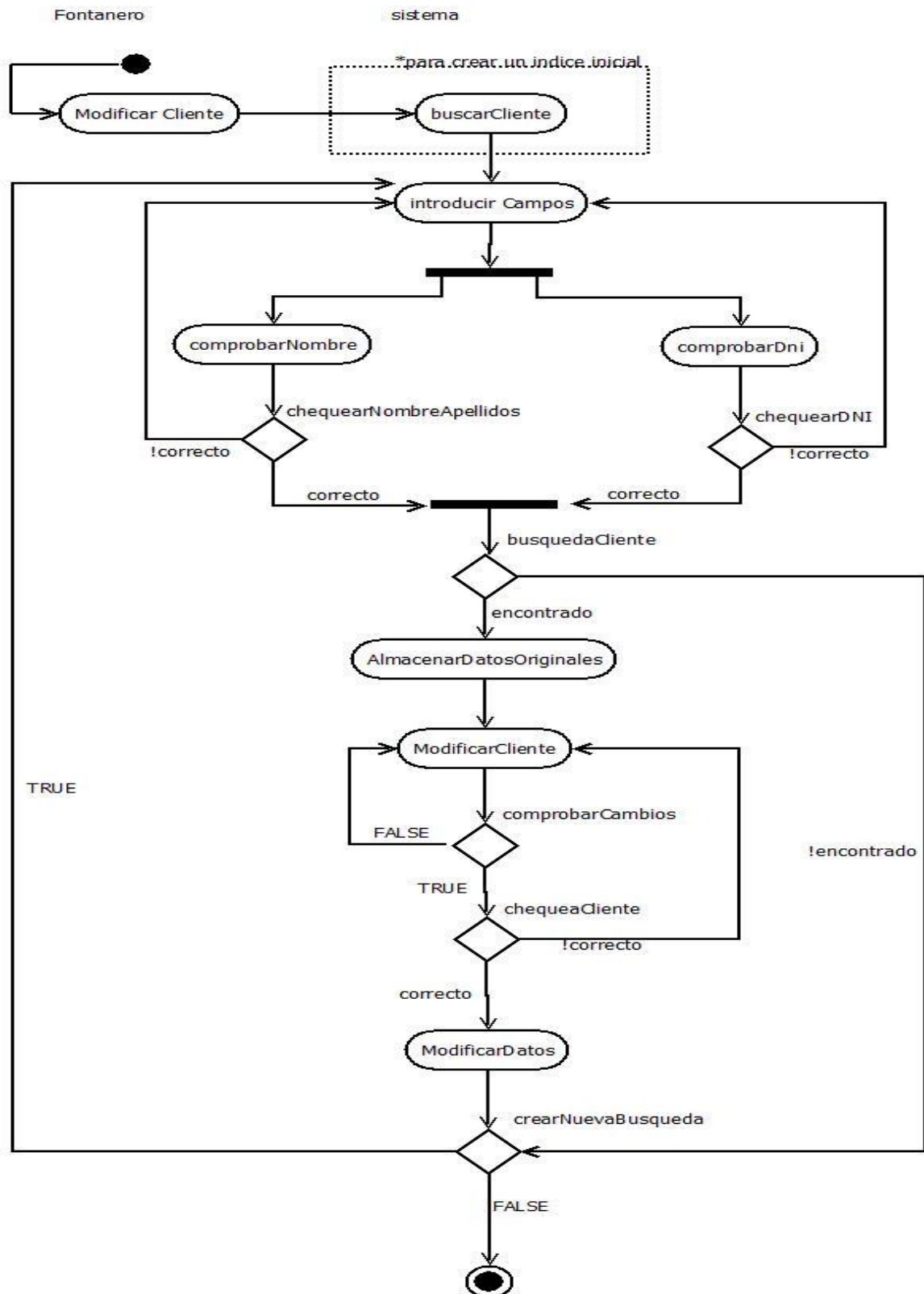
Secuencia buscar cliente (Requisitos RF8)

Diagrama de actividad para la secuencia de búsqueda de un cliente en el SGBD.



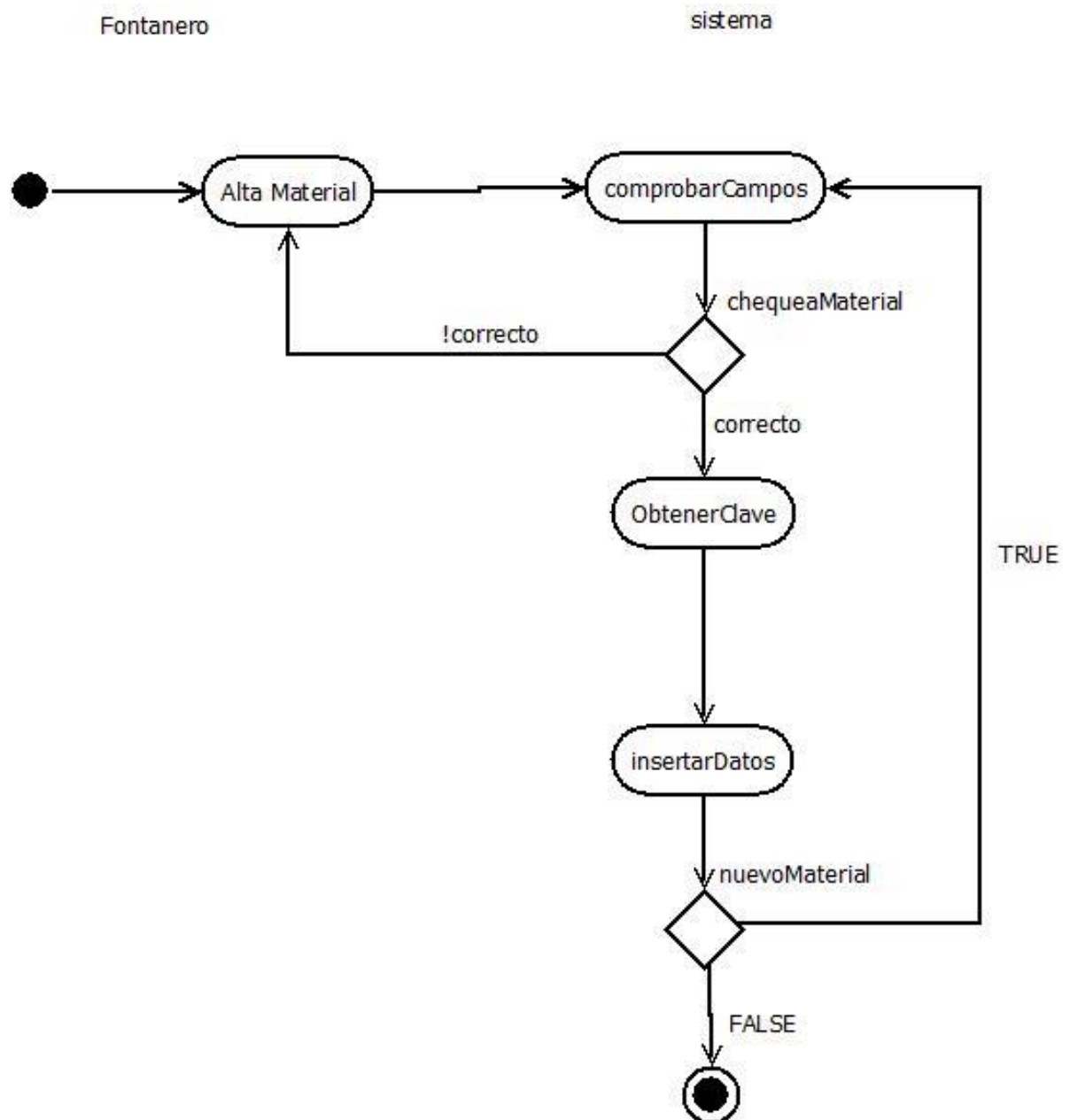
Secuencia modificar cliente (Requisitos RF7)

Diagrama de actividad para la secuencia de modificación de un cliente en el SGBD.



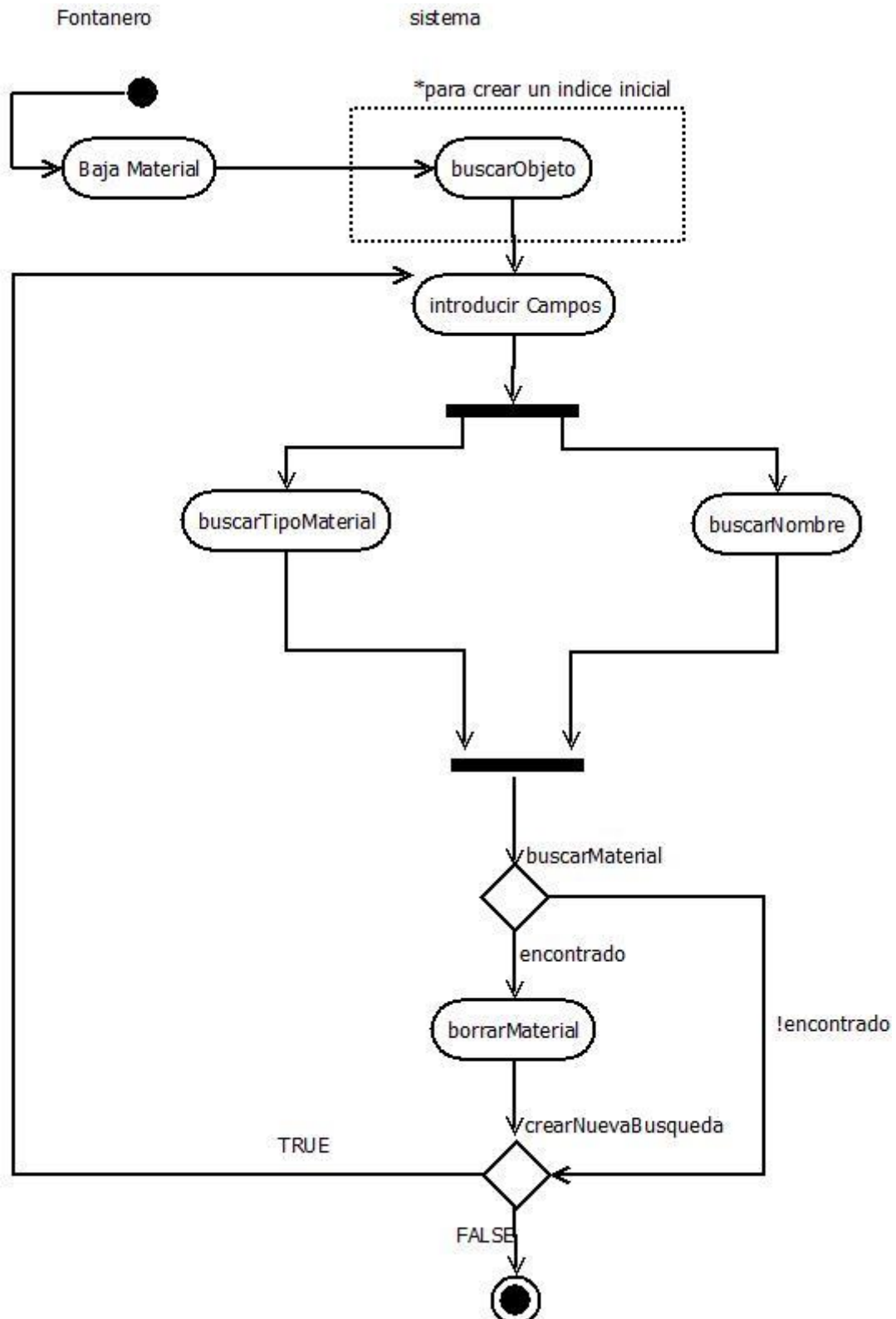
Secuencia alta material (Requisito RF10)

Diagrama de actividad para la secuencia de alta de material en el SGBD.



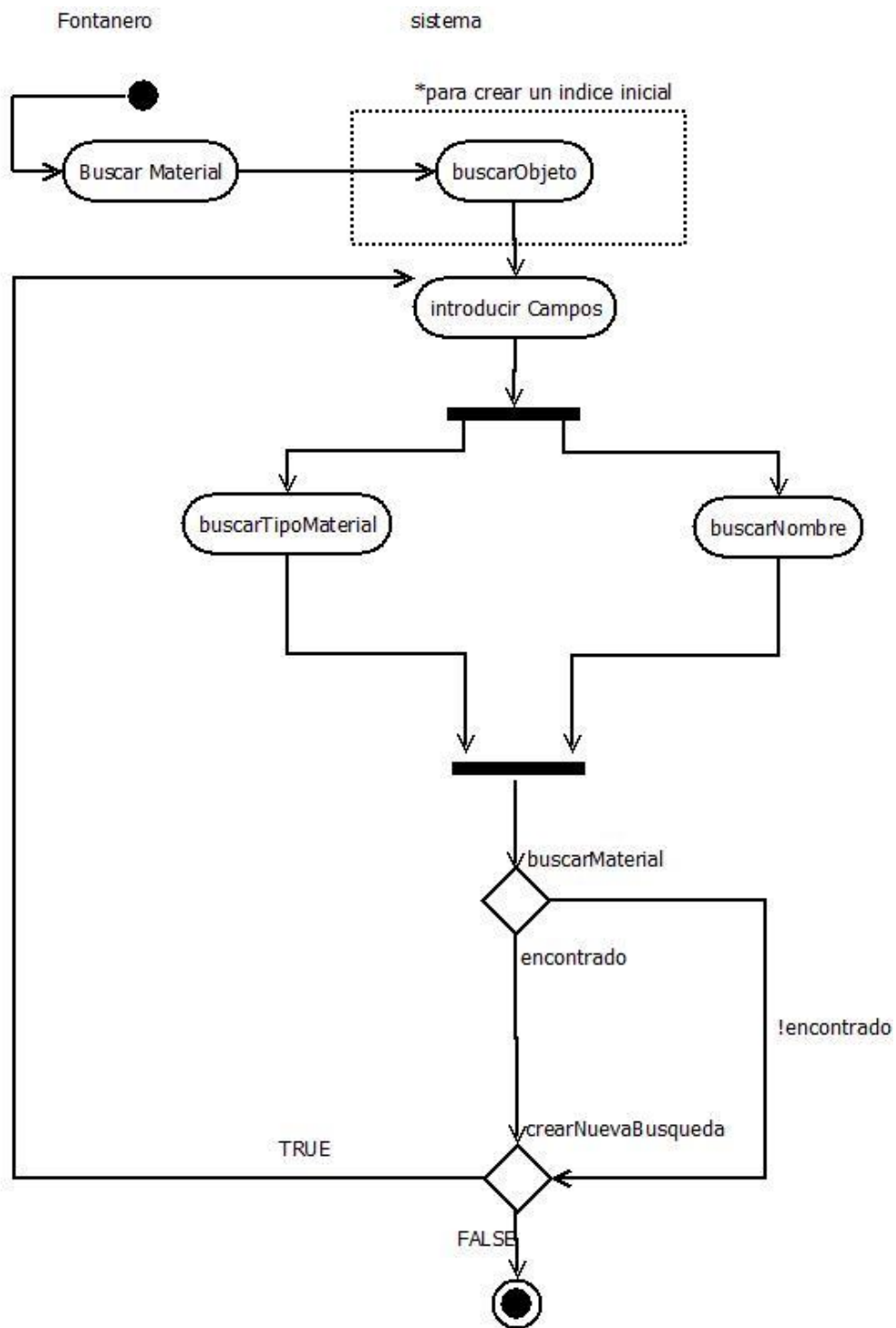
Secuencia baja material (Requisito RF11)

Diagrama de actividad para la secuencia de baja de material en el SGBD.



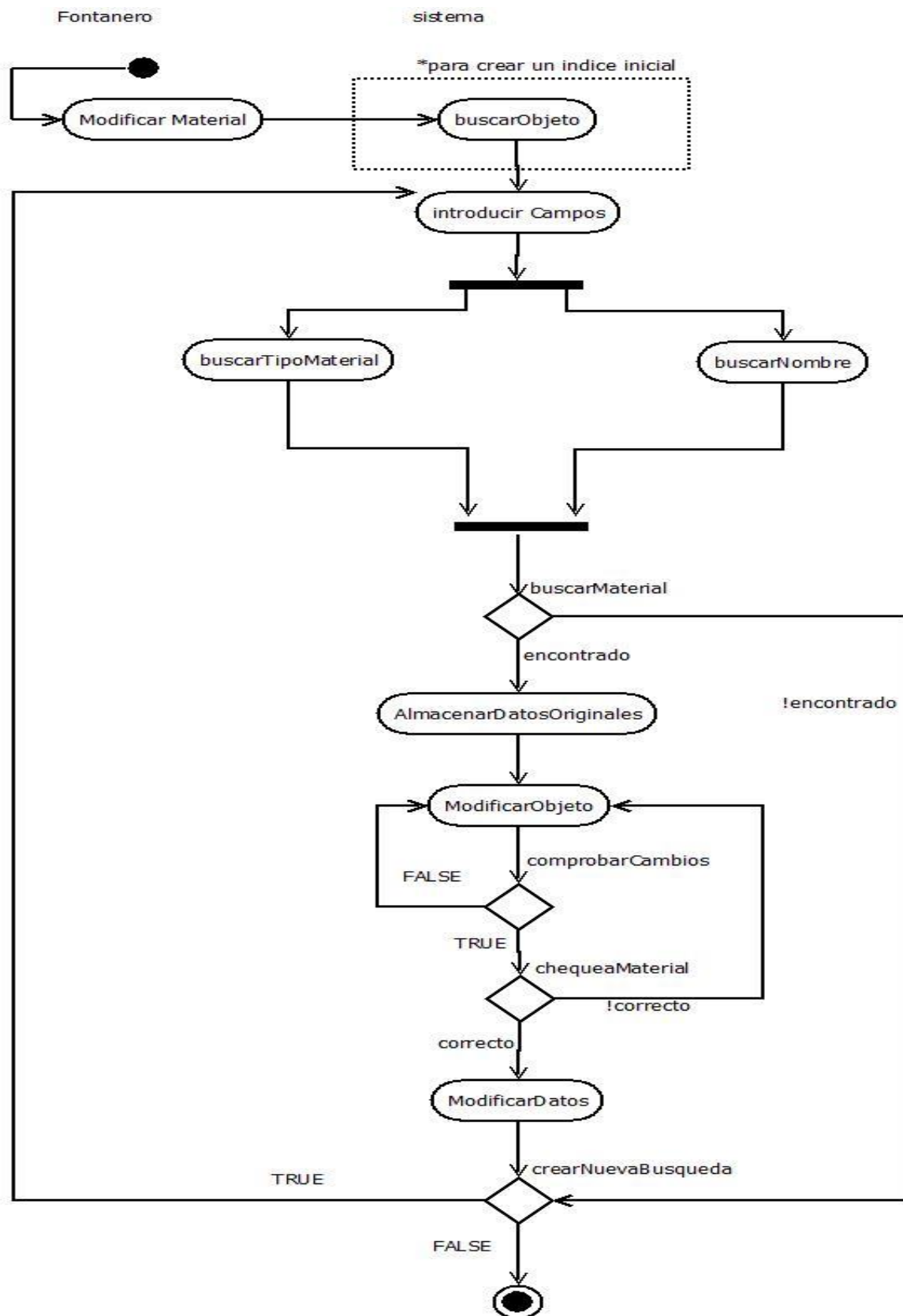
Secuencia búsqueda material (Requisito RF13)

Diagrama de actividad para la secuencia de búsqueda de material en el SGBD.



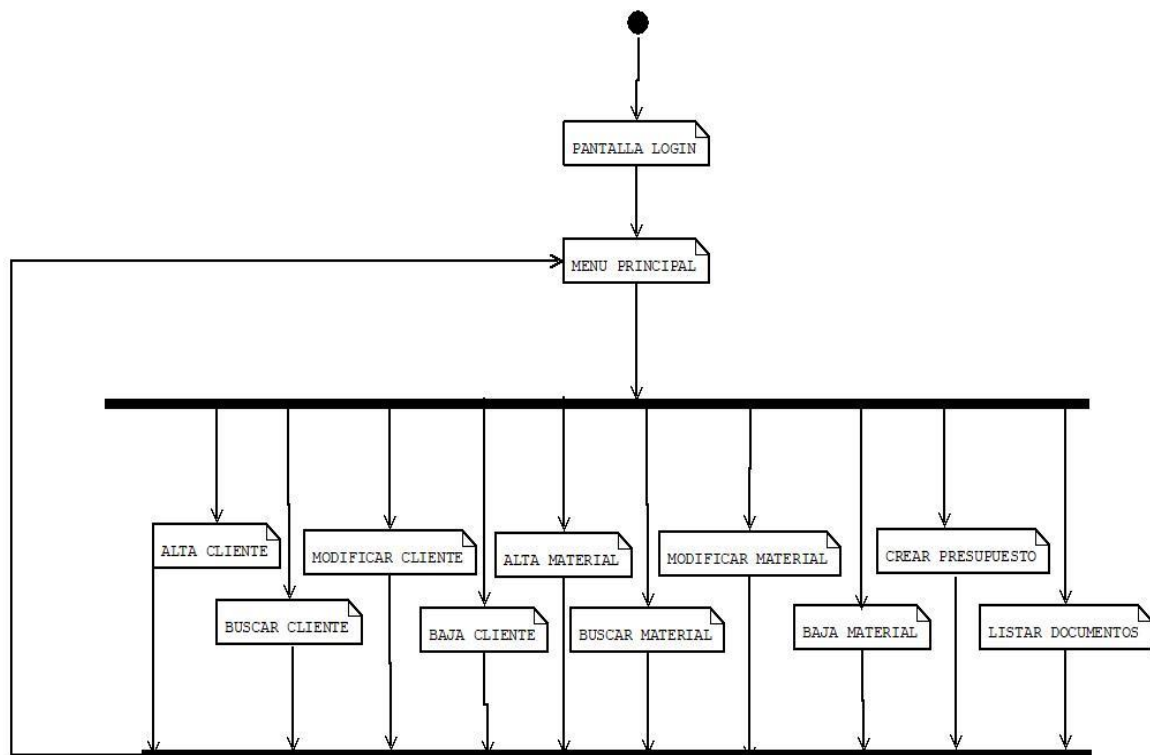
Secuencia Modificar material (Requisito RF12)

Diagrama de actividad para la secuencia de búsqueda de material en el SGBD.



6.6. Diseño UX.

A continuación, se muestra un diagrama que representa aproximadamente el flujo que puede seguir el usuario al manejar la aplicación.



Lo primero que se va a encontrar el usuario va a ser una pantalla para introducir las credenciales que le permitan acceder al sistema.

Una vez el usuario ha logrado acceder, visualizará el menú principal, desde el cual podrá seleccionar la acción a realizar.

Una vez seleccionada una opción, el sistema mostrará la pantalla o escena correspondiente a la selección y en la cual el usuario podrá realizar las operaciones correspondientes.

El bloque de opciones se puede dividir en 3 grupos, según el usuario quiera gestionar clientes, material o documentos.

Para clientes y material las opciones son iguales: alta, baja, búsqueda y modificación.

Para documentos, en principio el usuario solo podrá generarlos o acceder a un listado de ellos.

Todo este grupo de ventanas están abiertas de forma modal (WINDOW_MODAL) desde el menú principal, por lo que al cerrar cada una de ellas se vuelve a dicho menú.

The screenshot shows a web application for 'VICTORIA FONTANERÍA' with the title 'PRESUPUESTOS'. The interface includes a header with the logo and title, a top bar with 'Presupuesto N°:', 'Fecha:', and buttons 'Crear Presupuesto' and 'Guardar Presupuesto'. The main area is divided into sections: 'Datos Cliente' with 'Buscar DNI' and 'Buscar Cliente' buttons; a table with columns 'Cantidad', 'Precio Unitario', and 'Precio Total' containing a row with '1' and placeholder text, with an 'Añadir' button; a summary section with fields for 'Base Imponible', 'Tasa IVA', 'Importe Iva', 'TOTAL', 'Forma de Pago', and 'Nº Cuenta'; and a large empty table area with the text 'Tabla sin columnas'. At the bottom right are 'Borrar Producto' and 'Borrar Tabla' buttons.

Esta imagen sirve para mostrar los distintos elementos de cada escena con los que el cliente va a interactuar para poder trabajar de manera correcta.

Todos los formularios van a disponer de campos de texto (Text Fields) en los que el usuario va a poder introducir la información necesaria para realizar las distintas operaciones que necesite como dar de alta algún cliente o material, realizar búsquedas o introducir una serie de precios o valores para realizar los cálculos necesarios.

Por otro lado, la aplicación usará etiquetas (labels) para indicar el nombre de cada campo o mostrar otra información como campos calculados, warnings...

En algunos casos como por ejemplo la selección del material o su tipo, o el valor del IVA, el usuario deberá hacer uso de una serie de listados (comboBox) donde se muestra dicha información.

Para realizar las operaciones, cálculos, insertar datos en el SGBD y demás, el usuario dispone de una serie de botones con una indicación clara de la utilidad de cada uno. En aquellas escenas en las que haya sido necesaria alguna búsqueda se dispondrá de botones para el avance y retroceso dentro del índice.

Por último, y como elemento exclusivo del apartado de documentación, se hace uso de una tabla (table view) donde se irán mostrando los materiales incluidos en cada uno de los documentos y que podremos usar para seleccionar algún registro que necesitemos borrar.

6.7. Mockups.

A continuación, vamos a mostrar unos ejemplos de las distintas escenas en las que el cliente va a poder interactuar. Como color para las fuentes, bordes y otros elementos se ha elegido el color azul que tiene el logotipo de la marca. Para diferenciar las algunas zonas de cada uno de los formularios se han utilizado colores de tipo crema.

Pantalla para alta de clientes



VICTORIA FONTANERÍA

ALTA CLIENTES

IDCliente DNI

Nombre Apellido 1 Apellido 2

Direccion

C. Postal Localidad Provincia

Email Telefono

Pantalla Búsqueda Cliente



VICTORIA FONTANERÍA

BUSCAR CLIENTES

IDCliente DNI



Nombre Apellido 1 Apellido 2

Direccion

C. Postal Localidad Provincia

Email Telefono

Pantalla Baja Material

**BAJA MATERIAL**

IDMaterial

Tipo Material

Seleccione una tipo

Nombre

Descripcion

Precio Coste

Incremento

PrecioUnitario




Buscar por Material

Nueva Busqueda

Buscar por Nombre

Borrar Material



Pantalla Búsqueda Material

**BUSCAR MATERIAL**

IDMaterial

Tipo Material

Seleccione una tipo

Nombre

Descripcion

Precio Coste

% Incremento

Incremento

PrecioUnitario



Buscar por Material

Nueva Busqueda

Buscar por Nombre

Modificar Material

Recalcular Campos



7. Codificación.

7.1. Tecnologías elegidas.

Básicamente son 2 los pilares en los que me he basado para desarrollar este proyecto. Por un lado, el uso de un SGBD para el almacenamiento de los datos, y por otro Java como lenguaje de desarrollo para crear la interfaz de usuario. Además, en ambas cosas se trata de tecnologías con las que tengo mas experiencia como desarrollador y me permiten tener una base con la que poder trasladar la idea original del proyecto a algo codificable.

Como SGBD estoy usando la plataforma XAMPP, que entre otras herramientas tiene a MariaDB, que es bastante similar a MySQL. Son varias las razones que me han llevado al uso de esta tecnología, pero principalmente me interesaba un software que no fuese muy exigente con el rendimiento del hardware, teniendo en cuenta que el equipo del cliente tiene unas características muy básicas.

El lenguaje SQL me permite trabajar con las herramientas necesarias para realizar las operaciones de consulta, insertado de datos y demás contra el SGBD. Además, permite adaptación y escalabilidad para ajustar la estructura de las tablas a las necesidades de trabajo.

El lenguaje Java me aporta el conector para poder integrar el SGBD con la aplicación, además de una serie de clases e interfaces que me permiten hacer uso del lenguaje DML contra la BBDD. También me permite, aunque en esta ocasión no ha sido necesario, la creación de clases que sean idénticas a las tablas, lo cual facilita el trabajo. Además, me permite aplicar y controlar las restricciones necesarias para almacenar los datos en el Sistema como puede ser el formato de una cadena o un valor numérico.

También he usado la tecnología JavaFX, que permite implementar el patrón o modelo vista-controlador. Por un lado, el uso de la herramienta SceneBuilder permite tanto el diseño de la interfaz de usuario como la colocación y disposición de los distintos elementos que vamos a necesitar. Por otro lado, cada escena lleva asociada una clase que permite controlar las acciones de cada uno de los elementos de dicha escena y permite aplicar las operaciones o acciones necesarias.

Todo lo relacionado con Java está implementado bajo la tecnología Apache Maven, la cual inicialmente ofrece una estructura de directorios que facilita la distribución de las distintas clases o ficheros para su trabajo. Además, mediante el uso de arquetipos, como por ejemplo los que tiene para JavaFX, permite tener una configuración inicial correcta y lista para empezar a trabajar con el proyecto.

Por último, para el control de versiones he hecho uso de GitHub y más concretamente de su versión Desktop. Al ser yo el único desarrollador y dado que el proceso de desarrollo ha sido bastante lineal en todo momento he hecho uso de la rama máster, pero donde he descubierto todo su potencial ha sido en la empresa donde estoy realizando las FCT's. Además, el uso de GitHub como repositorio del proyecto permite que otras personas, puedan consultar el estado del proyecto de forma remota.

7.2. Documentación interna.

Se adjunta documento con la estructura de las clases y sus métodos.

7.2.1. Descripción de ficheros.

CLASES Y CONTROLADORES

Fichero .JAVA	Descripción	Fichero XML asociado
App	clase para iniciar la aplicación de escritorio a partir de una scene inicial	
ControladorAltaCliente	clase para controlar la escena de alta de clientes	PantallaAltaCliente
ControladorAltaMaterial	clase para controlar la escena de alta de clientes	PantallaAltaMaterial
ControladorBajaCliente	clase para controlar la escena de consulta de la tabla clientes	PantallaBajaCliente
ControladorBajaMaterial	clase para controlar la escena de consulta de la tabla clientes	PantallaBajaMaterial
ControladorBuscarCliente	clase para controlar la escena de consulta de la tabla clientes	PantallaBuscarCliente
ControladorBuscarDocumento	clase para controlar la escena de consulta de documentos	PantallaBuscarDocumento
ControladorBuscarMaterial	clase para controlar la escena de consulta de la tabla clientes	PantallaBuscarMaterial
ControladorCrearDocumento	clase para controlar la creación de documentos de tipo presupuesto o facturas	PantallaCrearDocumento
ControladorMenu	Clase para controlar la apertura de escenas en función del botón pulsado	PantallaMenu
ControladorModificarCliente	clase para controlar la escena de consulta de la tabla clientes	PantallaModificarCliente
ControladorModificarMaterial	clase para controlar la escena de modificación de material	PantallaModificarMaterial
ControladorPrincipal	clase controlador del menú login	PantallaLogin
FacturaBase	clase para construir los elementos que componen la factura	

INTERFACES

Fichero .JAVA	Descripcion
BaseDatos	clase de tipo interfaz para las operaciones DML contra la BBDD
Chequeable	clase de tipo interfaz para comprobar el formato de los datos introducidos en los campos de los formularios
Comprobable	interfaz para implementar diferentes métodos que nos permitirán comprobar patrones en el formato de los diferentes campos de los formularios
Fichero	Interfaz que implementa un método para obtener el identificador del cliente

7.2.2. Descripción de funciones.

En el documento JAVADOC adjunto vienen documentados todas las clases que se utilizan en el proyecto, junto con sus métodos y funciones y una breve descripción de cada uno de ellos.

Por destacar y comentar alguna función, me quedo con una que permite controlar en tiempo real el número de caracteres que se pueden escribir en un text field.

En primer lugar, dentro del método initialize de la clase ControladorAltaCliente, hacemos uso del método setTextFormatter para indicar el formato que queremos que tengan los campos DNI, teléfono y código postal. En concreto, además de que estos campos tengan un patrón determinado, también nos interesa restringir su número de caracteres. Para configurar este formato hacemos una llamada a la interfaz Comprobable, y mas en concreto a su método getFormatter al cual se le pasa el rango de caracteres de la cadena.

```
//esta parte del codigo controla que el campo DNI no tenga mas de 9 caracteres
tfDNI.setTextFormatter(Comprobable.getFormatter("{0,9}"));

//esta parte del codigo controla que el campo Telefono no tenga mas de 13 caracteres
tfTelefono.setTextFormatter(Comprobable.getFormatter("{0,13}"));

//esta parte del codigo controla que el campoCodigo Postal no tenga mas de 5 caracteres
tfCPostal.setTextFormatter(Comprobable.getFormatter("{0,5}"));
```

Una vez dentro de la interfaz y más concretamente en el método que hemos llamado, podemos hacer uso de la clase TextFormatter para limitar en este caso el tamaño del texto que debemos introducir en cada uno de los campos.

```
/**
 * metodo para comprobar que en algunos campos no pueda escribir mas caracteres de los permitidos
 * @param cantidadCaracteres el rango de caracteres permitidos
 * @return el texto con el numero de caracteres correspondiente
 */
@SuppressWarnings({ "unchecked", "rawtypes" })
public static TextFormatter getFormatter(String cantidadCaracteres) {

    Pattern pattern = Pattern.compile(cantidadCaracteres);
    TextFormatter formatter = new TextFormatter((UnaryOperator<TextFormatter.Change>) change -> {
        return pattern.matcher(change.getControlNewText()).matches() ? change : null;
    });

    return formatter;
}
```

También me gustaría destacar el trabajo de algunas funciones de la interfaz comprobable para comprobar el patrón que deben seguir algunos campos como el DNI del usuario, el teléfono, el código postal, etc... A continuación, muestro alguno de ellos.

```
/**
 * metodo para comprobar que el DNI cumple con el formato correcto
 * @param tfDNI, dni del cliente
 * @return true or false, en funcion de si cumple o no el formato
 */
public static boolean comprobarFormatoDNI(String tfDNI){

    //creamos el formato que deseamos tener para el DNI
    Pattern pat = Pattern.compile("[0-9]{8}[A-Z a-z]");

    //creamos el comparador para ver que coincide con la cadena que le pasamos
    Matcher mat = pat.matcher(tfDNI);

    //devolvemos el resultado de la comparacion
    return mat.matches();

}

/**
 * metodo para ver si el nombre o los apellidos cumplen con el patrón
 * @param nombre, nombre o apellidos
 * @return true or false, en funcion de si se cumple el patrón
 */
public static boolean comprobarNombres(String nombre) {

    //creamos un pattern con los caracteres que queremos validar, incluyendo acentos
    Pattern patron = Pattern.compile("[ A-Za-zñÑáéíóúÁÉÍÓÚ]{3,50}");

    //comparamos el patron con la cadena recibida como parametro
    Matcher comprobar = patron.matcher(nombre);

    //en funcion de la comparación se devolvera true or false
    if (comprobar.matches()) return true;
    else return false;

}

/**
 * metodo para comprobar que el campo Codigo Postal cumple un patrón correcto
 * @param cPostal dirección del cliente
 * @return true or false, en funcion de si se cumple el patrón
 */
public static boolean comprobarCodigoPostal(String cPostal) {

    //creamos un pattern con los caracteres que queremos validar
    Pattern patron = Pattern.compile("[0-5][0-9]{4}");

    //comparamos el patron con la cadena recibida como parametro
    Matcher comprobar = patron.matcher(cPostal);

    //en funcion de la comparación se devolvera true or false
    if (comprobar.matches()) return true;
    else return false;

}
```

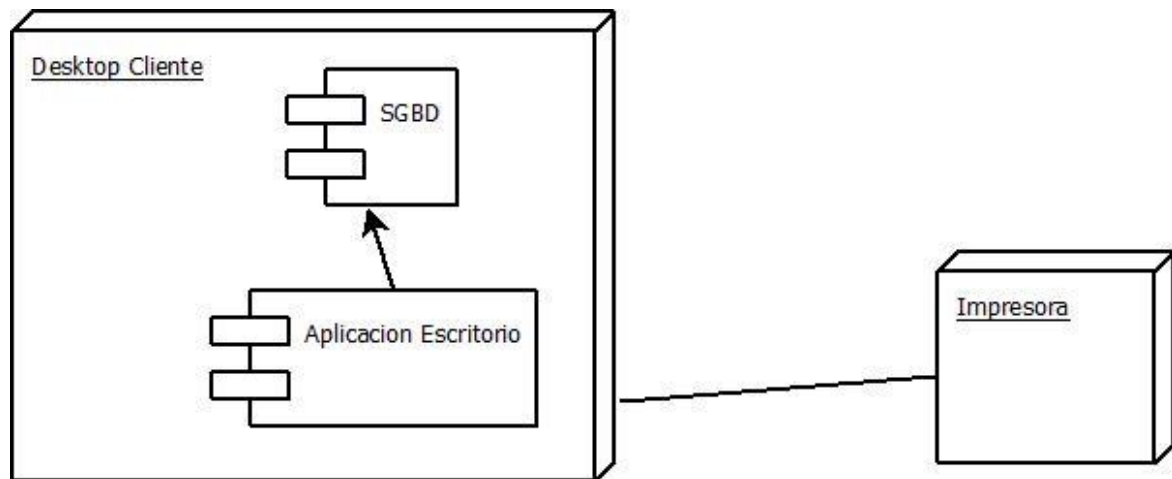
7.3. Documentación externa.

7.3.1 Manual de usuario.

Ver documento anexo.

8. Despliegue.

8.1. Diagramas de despliegue.



8.2. Descripción de la instalación o despliegue.

El despliegue de la aplicación en teoría es bastante sencillo ya que tanto el SGBD como la aplicación de escritorio van a ser instalados en el ordenador del cliente. Por un lado, se instalará la plataforma XAMPP con el SGBD el cual almacenará los datos de la empresa, mientras que por otro lado se instalará en el mismo ordenador la aplicación de escritorio que permitirá la conexión con la BBDD para poder realizar las operaciones que se necesiten.

Como dispositivo externo se hará uso de una impresora para que el usuario pueda imprimir todos los documentos que necesite, ya sean presupuestos, facturas o albaranes.

En principio la opción de usar algún servicio remote para el almacenamiento de datos o la opción de poder acceder de manera remota a la aplicación han quedado descartadas por parte del usuario.

9. Herramientas de apoyo.

- **Microsoft Word:** Para la creación de la documentación.
- **Microsoft Excel:** Para la creación del diagrama de Gantt.
- **Adobe Reader:** Para la lectura de los documentos PDF presentados.
- **DIA:** Software para la elaboración de diagramas UML, E/R...
- **SceneBuilder:** Software para el diseño JavaFX de las escenas.
- **GitHub Desktop:** Control de versiones y acceso al repositorio GitHub

10. Control de versiones.

Como he comentado anteriormente he usado GitHub como sistema de control de versiones, usando su versión Desktop que facilita la interacción con el repositorio mediante interfaz gráfica. En principio solo he hecho uso de la rama master como rama única del proyecto. Para la siguiente fase del proyecto sí que será necesario el uso de otras ramas para mejorar la aplicación.

11. Gestión de pruebas.

Pendiente incluir pruebas Junit en el proyecto.

12. Conclusiones.

12.1. Conclusiones sobre el trabajo realizado.

De momento no he podido completar todos los requisitos funcionales que había establecido al inicio de este proyecto, quedando pendiente el poder imprimir los documentos y mostrar los listados. También es necesario dar un repaso a la gestión de la documentación para orquestar mejor la generación de facturas, presupuestos y albaranes. Esta parte ha sido la mas complicada de afrontar. Por lo demás, hasta que se produzca la review con el cliente, todo lo relacionado a la gestión de clientes y materiales está completo y funcional.

12.2. Conclusiones personales.

Personalmente el desarrollo del proyecto me ha resultado bastante útil para mejorar mi experiencia como desarrollador. El proyecto es en parte ambicioso, pero me ha ayudado a aprender todas las fases necesarias para su desarrollo, así como la necesidad previa de tener todo bien diseñado con anterioridad. Esta experiencia y junto con todo lo aprendido en la empresa en cuanto a planificación, metodología y creación de metas me han servido para, ahora sí, verme capacitado para afrontar un proyecto de esta envergadura.

12.3. Posibles ampliaciones y mejoras.

Como he comentado anteriormente el apartado que en principio necesita una revisión es el de la gestión de documentos para crear facturas, presupuestos y albaranes, ya que es un poco caótico entender el funcionamiento de su generación y orquestar todo es algo complicado.

Por lo demás, estoy a la espera de que el cliente pruebe el producto para que me indique mejoras funcionales, de diseño, etc...

14. Bibliografía.

14.1 Libros, artículos y apuntes.

- **Apuntes de clase** para consulta concretas sobre la elaboración de diagramas.
- **Empresa e iniciativa emprendedora (Editorial Aula Digital)**

Libro de la asignatura Empresa e Iniciativa Emprendedora y que me ha servido para realizar tareas como el estudio de mercado o los análisis DAFO Y CAME.

14.2 Direcciones web.

- [Lucidchart Español - YouTube](#)
- [nicosiored - YouTube](#)

Ambos son canales de YouTube que he seguido como tutorial para la elaboración de algunos diagramas, como el de secuencias o el de actividades.

- [JavaFX Tutorial \(jenkov.com\)](#)

Página dedicada al desarrollo de Java, contiene un apartado bastante completo sobre las diferentes herramientas y utilidades que componen JavaFx, con útiles ejemplos. Me ha servido para trabajo con tableViews.

- [Using JavaFX UI Controls: About This Tutorial | JavaFX 2 Tutorials and Documentation \(oracle.com\)](#)

Página oficial de Oracle con documentación oficial sobre JavaFx. La he usado para consultar como configurar algunos elementos JavaFx o como interactuar con los table View para borrar datos de la tabla o realizar cálculos con ellos.

- [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)

Página para realizar consultas específicas sobre código, funciones...