

Trabajo Fin de Grado

Aplicación de Trazabilidad para la Gestión de Biomasa: Entrada de Materias Primas y Procesos

Autor

Eduardo Fonte Polo

Directores

Dr. Luis Carlos Aparicio Cardiel

Dr. Guillermo Azuara Guillén

Escuela Universitaria Politécnica de Teruel

Teruel

Septiembre de 2017

APLICACIÓN DE TRAZABILIDAD PARA LA GESTIÓN DE BIOMASA: ENTRADA DE MATERIAS PRIMAS Y PROCESOS

RESUMEN

La trazabilidad es el conjunto de procedimientos que permite seguir el proceso de evolución de un producto en cada una de sus etapas. Por lo tanto, es una herramienta muy útil para la gestión de cualquier empresa u organismo.

El punto de partida de este proyecto surge de la necesidad de automatizar la obtención de la trazabilidad en una supuesta empresa que trabaja con productos madereros. Para lo cual se ha desarrollado la aplicación “BiomasaEUP”, un programa genérico aplicable no solo al sector maderero.

Dado que es un proyecto complejo, el desarrollo y trabajo se ha dividido en dos partes. En este Trabajo Fin de Grado se aborda la primera parte del proyecto, centrada en la construcción de un prototipo funcional para obtener la trazabilidad en el proceso de entrada de materias primas y transformación, es decir, la gestión de permisos, usuarios, clientes, proveedores y recepciones de materias primas, así como la obtención de la trazabilidad de las recepciones y de las materias primas. Todo ello, a través de una aplicación de escritorio conectada a una base de datos SQL Server, programada en C# usando la tecnología Entity Framework para el manejo de los datos.

En la segunda parte del proyecto se implementarán los procesos de fabricación de productos terminados y ventas de productos envasados, así como la obtención de la trazabilidad completa del sistema.

Palabras clave: trazabilidad, Entity Framework, identificación, etiquetas.

TRACEABILITY APPLICATION FOR BIOMASS MANAGEMENT: INPUT OF RAW MATERIALS AND PROCESSES

ABSTRACT

Traceability is the set of procedures that allow to follow the evolution process of a product in each of its stages. Therefore, it is a very useful tool for the management of any company or organization.

The starting point of this project comes from the need to automate the obtaining of traceability in a supposed timber company. For this purpose, the application “BiomasaEUP” has been developed, a generic program applicable not only to the timber sector.

As it is a complex project, development and work has been divided into two parts. In this Final Degree Project, the first part of the project is addressed, focused on the construction of a functional prototype to obtain traceability in the process of raw material entry and transformation, that is to say, the management of permissions, users, customers, suppliers and raw material receptions, as well as obtaining traceability of receptions and raw materials. All this, through a desktop application connected to a SQL Server database, programmed in C# using Entity Framework technology for data management.

In the second part of the project, the manufacturing processes of finished products and sales of packaged products will be implemented, as well as obtaining complete traceability of the system.

Key words: traceability, Entity Framework, identification, tags.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer la labor de mis directores, Luis Carlos Aparicio Cardiel y Guillermo Azuara Guillén, así como todo su esfuerzo, dedicación y ayuda prestada, cada uno en su especialidad, que me han proporcionado para que este Trabajo Fin de Grado se haya podido llevar a cabo. Con el desarrollo del mismo, he podido mejorar en mi formación como ingeniero informático, adquiriendo ciertas competencias que me resultarán muy útiles para el mundo laboral.

En segundo lugar, me gustaría agradecerle a mi amigo y compañero de grado Javier Bielsa Hernández, la implicación en este proyecto. Él ha colaborado en el diseño de la base de datos, así como en la implementación del prototipo desarrollado.

Y, por último, no quisiera olvidarme de mi familia, en especial de mi hermano Alberto, por sus útiles consejos a la hora de la realización de este trabajo.

ÍNDICE

1. Introducción	1
1.1. Contexto	2
1.2. Trazabilidad	3
1.2.1. Códigos de identificación únicos de captura automática	4
1.3. Motivación, problemas y objetivos	6
2. Estado del Arte	8
2.1. Introducción	8
2.2. Aplicaciones de trazabilidad en el mercado.....	8
2.3. Conclusiones.....	9
3. Análisis.....	10
3.1. Casos de Uso	10
3.1. Especificación de Requisitos de Software	13
3.1.1. Requisitos comunes de los interfaces	13
3.1.2. Requisitos funcionales.....	15
3.1.3. Requisitos no funcionales	19
4. Diseño del sistema	21
4.1. Perspectiva global	21
4.2. Arquitectura Software.....	21
4.2.1. Microsoft .NET.....	22
4.2.2. C#.....	22
4.2.3. XAML	23
4.2.4. MVVM	23
4.3. Estructura del modelo de datos	24
4.3.1. Elección de SQL Server como SGBD	24
4.3.2. Configuración de SQL Server en MV remota	25
4.4. Entity Framework	27
4.4.1. Creación de la Base de Datos	28
4.4.2. Migraciones	35
4.4.3. Generación de la trazabilidad usando PDF	39
4.5. Prototipos de la Interfaz de Usuario	40
5. Mejoras Futuras	41
6. Conclusiones.....	42
7. Bibliografía	43
ANEXOS	47

ÍNDICE DE FIGURAS

Figura 1. Logotipo de BiomasaEUPTE.....	1
Figura 2. Estructura del código GS1-128.	5
Figura 3. Diagrama de Casos de Uso del usuario Super Administrador.....	11
Figura 4. Diagrama de Casos de Uso del usuario Administrativo.	11
Figura 5. Diagrama de Casos de Uso del usuario Administrador.	12
Figura 6. Diagrama de Casos de Uso del usuario Técnico.	12
Figura 7. Ejemplo de consulta en LINQ.	23
Figura 8. Relación entre los componentes del patrón MVVM.	24
Figura 9. Habilitación del protocolo TCP/IP para conexión remota con SQL Server.....	26
Figura 10. Reinicio del proceso de SQL Server.....	26
Figura 11. Habilitación del puerto TCP/IP para la conexión remota.....	26
Figura 12. Diagrama de la arquitectura de Entity Framework.....	28
Figura 13. Creación del contexto de la BD con Entity Framework.....	29
Figura 14. Contexto de Entity Framework.	29
Figura 15. Cadena de conexión a la BD ubicada en el fichero "App.config".....	29
Figura 16. Clase POCO de la tabla "Usuarios".....	31
Figura 17. Clase POCO de la tabla "TiposUsuarios".	31
Figura 18. Relación entre las tablas "Recepcion" y "MateriaPrima".	32
Figura 19. Diagrama de la BD. 1ª parte.....	33
Figura 20. Diagrama de la BD. 2ª parte.....	34
Figura 21. Árbol de directorios generado por el comando "Enable-Migrations".	35
Figura 22. Tablas Usuarios y TiposUsuarios generadas en la BD con Entity Framework.....	36
Figura 23. Inserción de datos en la BD usando migraciones.	37
Figura 24. Parte de los ficheros CSV usados en las migraciones.....	38
Figura 25. Contenido del fichero CSV para la tabla "Usuarios".	38
Figura 26. Parte de código ubicado en el fichero BiomasaEUPTE.csproj.....	38
Figura 27. Creación de un código de barras con iText7.	39
Figura 28. Informe PDF de una materia prima.....	39
Figura 29. Creación de un formulario usando Balsamiq Mockups.	40

ANEXO C

Figura C 1. Parte del contenido de la clase BiomasaEUPTEContextInitializer.	1
Figura C 2. Inicializador de la BD desde el BiomasaEUPTEContext.....	2

ANEXO D

Figura D 1. Auditoría usando EntityFramework-Plus.	2
---	---

ANEXO E

Figura E 1. Representación gráfica de la tabla de la BD "TipoUsuario".....	1
Figura E 2. Representación gráfica de la tabla de la BD "Usuario".....	1
Figura E 3. Representación gráfica de la tabla de la BD "Permiso".	1
Figura E 4. Representación gráfica de la tabla de la BD "TipoCliente".	2
Figura E 5. Representación gráfica de la tabla de la BD "GrupoCliente".....	2

Figura E 6. Representación gráfica de la tabla de la BD “Cliente” .	2
Figura E 7. Representación gráfica de la tabla de la BD “TipoProveedor” .	2
Figura E 8. Representación gráfica de la tabla de la BD “Proveedor” .	3
Figura E 9. Representación gráfica de la tabla de la BD “Pais” .	3
Figura E 10. Representación gráfica de la tabla de la BD “Comunidad” .	3
Figura E 11. Representación gráfica de la tabla de la BD “GrupoMateriaPrima” .	3
Figura E 12. Representación gráfica de la tabla de la BD “Provincia” .	4
Figura E 13. Representación gráfica de la tabla de la BD “Municipio” .	4
Figura E 14. Representación gráfica de la tabla de la BD “Recepcion” .	4
Figura E 15. Representación gráfica de la tabla de la BD “EstadoRecepcion” .	4
Figura E 16. Representación gráfica de la tabla de la BD “MateriaPrima” .	5
Figura E 17. Representación gráfica de la tabla de la BD “TipoMateriaPrima” .	5
Figura E 18. Representación gráfica de la tabla de la BD “SitioRecepcion” .	5
Figura E 19. Representación gráfica de la tabla de la BD “HuecoRecepcion” .	6
Figura E 20. Representación gráfica de la tabla de la BD “HistorialHuecoRecepcion” .	6

ANEXO F

Figura F 1. Prototipo de la ventana inicio de sesión.	1
Figura F 2. Prototipo de la ventana gestión de usuarios.	2
Figura F 3. Prototipo del formulario usuario.	3
Figura F 4. Prototipo de la ventana de gestión de clientes.....	4
Figura F 5. Prototipo del formulario cliente.....	5
Figura F 6. Prototipo del formulario tipo de cliente.	5
Figura F 7. Prototipo de la ventana gestión de proveedores.....	6
Figura F 8. Prototipo de la ventana gestión de recepciones.....	7
Figura F 9. Prototipo del formulario recepción.....	8
Figura F 10. Prototipo del formulario materia prima.	8
Figura F 11. Prototipo de la ventana obtención de trazabilidad.	9
Figura F 12. Prototipo de la ventana de gestión de elaboraciones.	10
Figura F 13. Prototipo del formulario orden de elaboración.....	10
Figura F 14. Prototipo del formulario producto terminado.....	10
Figura F 15. Prototipo de la ventana de gestión de pedidos.	11
Figura F 16. Prototipo del formulario pedido.	11
Figura F 17. Prototipo del formulario producto envasado.	11

ÍNDICE DE TABLAS

ANEXO A

Tabla A 1. Caso de Uso - Iniciar sesión.	1
Tabla A 2. Caso de Uso - Cerrar sesión.	1
Tabla A 3. Caso de Uso - Gestión usuarios.	2
Tabla A 4. Caso de Uso - Registrar usuario.	2
Tabla A 5. Caso de Uso - Modificar usuario.	2
Tabla A 6. Caso de Uso - Borrar usuario.	2
Tabla A 7. Caso de Uso - Gestión clientes.	3
Tabla A 8. Caso de Uso - Añadir cliente.	3
Tabla A 9. Caso de Uso - Modificar cliente.	3
Tabla A 10. Caso de Uso - Borrar cliente.	3
Tabla A 11. Caso de Uso - Gestión proveedores.	4
Tabla A 12. Caso de Uso - Gestión recepciones.	4
Tabla A 13. Caso de Uso - Añadir recepción.	4
Tabla A 14. Caso de Uso - Modificar recepción.	4
Tabla A 15. Caso de Uso - Borrar recepción.	5
Tabla A 16. Caso de Uso - Añadir materia prima.	5
Tabla A 17. Caso de Uso - Modificar materia prima.	5
Tabla A 18. Caso de Uso - Borrar materia prima.	6
Tabla A 19. Caso de Uso - Obtener trazabilidad.	6
Tabla A 20. Caso de Uso - Gestión permisos.	6
Tabla A 21. Caso de Uso - Cambiar ajustes	6
Tabla A 22. Caso de Uso - Cambiar contraseña	7
Tabla A 23. Caso de Uso - Cambiar color del tema	7
Tabla A 24. Caso de Uso - Cambiar ajustes ventana	7
Tabla A 25. Caso de Uso - Gestión elaboraciones.	7
Tabla A 26. Caso de Uso - Añadir elaboración.	8
Tabla A 27. Caso de Uso - Modificar elaboración.	8
Tabla A 28. Caso de Uso - Borrar elaboración.	8
Tabla A 29. Caso de Uso - Añadir producto terminado.	9
Tabla A 30. Caso de Uso - Modificar producto terminado.	9
Tabla A 31. Caso de Uso - Borrar producto terminado.	9
Tabla A 32. Caso de Uso - Gestión ventas.	10
Tabla A 33. Caso de Uso - Añadir pedido.	10
Tabla A 34. Caso de Uso - Modificar pedido.	10
Tabla A 35. Caso de Uso - Borrar pedido.	10
Tabla A 36. Caso de Uso - Añadir producto envasado.	11
Tabla A 37. Caso de Uso - Modificar producto envasado.	11
Tabla A 38. Caso de Uso - Borrar producto envasado.	11

ANEXO B

Tabla B 1. Comparativa de las distintas versiones de Visual Studio.	2
--	---

GLOSARIO

API	<i>Application Programming Interface</i> , Interfaz de Programación de Aplicaciones
BAML	<i>Binary Application Markup Language</i> , Lenguaje de Marcado para Aplicaciones Binarias
BD	Base de Datos
CU	Caso de Uso
DTO	<i>Data Transfer Objects</i> , Objeto de Transferencia de Datos
EF	Entity Framework
GNU	<i>GNU's Not Unix</i> , GNU no es Unix
GTIN	<i>Global Trade Item Number</i> , Número Mundial de un Artículo Comercial
IDE	<i>Integrated Development Environment</i> , Entorno de Desarrollo Integrado
LINQ	<i>Language Integrated Query</i> , Lenguaje de Consultas Integrado
MDX	<i>MultiDimensional eXpressions</i> , Expresiones Multidimensionales
MV	Máquina Virtual
MVC	<i>Model View Controller</i> , Modelo Vista Controlador
MVVM	<i>Model View ViewModel</i> , Modelo Vista VistaModelo
OLAP	<i>OnLine Analytical Processing</i> , Procesamiento Analítico en Línea
OLTP	<i>OnLine Transactions Processing</i> , Procesamiento de Transacciones en Línea
ORM	<i>Object-Relational Mapping</i> , Mapeo Objeto-Relacional
POCO	<i>Plain Old CLR Object</i>
RFID	<i>Radio Frequency Identification</i> , Identificación por Radio Frecuencia
RQF	Requisito Funcional
RQI	Requisito de interfaz
RQNF	Requisito no Funcional
SGBD	Sistema Gestor de Bases de Datos
SQL	<i>Structured Query Language</i> , Lenguaje de consulta estructurada
SQL-DMO	<i>Structured Query Language-Distributed Management Objects</i> Objetos de Administración de base de datos de SQL
SSMS	SQL Server Management Studio
TCP	<i>Transmission Control Protocol</i> , Protocolo de Control de Transmisión
TFG	Trabajo Fin de Grado
T-SQL	<i>Transact Structured Language</i> , Lenguaje de Búsquedas Estructurado
UNE	Una Norma Española
WPF	<i>Windows Presentation Foundation</i>
XAML	<i>eXtensible Application Markup Language</i> , Lenguaje Extensible de Formato para Aplicaciones
XML	<i>eXtensible Markup Language</i> , Lenguaje de Marcado Extensible

1. INTRODUCCIÓN

“BiomasaEUPB” (Figura 1) es un prototipo de aplicación de escritorio para Sistemas Operativos Windows 7 o superior, que hace uso del estilo visual *Material Design* (Google), que permite obtener la trazabilidad del proceso de elaboración de distintos tipos de productos derivados de la industria maderera.



FIGURA 1. Logotipo de BiomasaEUPB.

Antes de continuar con la introducción, es necesario aclarar el concepto de trazabilidad para poder entender los fundamentos de este proyecto. La trazabilidad sirve para conocer todos los elementos que intervienen en la elaboración de un producto (materias primas, procesos intermedios, etc.) y todas las fases por las que pasa dicho producto (adquisición, almacenamiento, producción, elaboración, almacenaje, distribución, etc.). Por lo tanto, es una herramienta muy útil en cualquier empresa productiva o proveedora de servicios para conseguir el logro de sus objetivos de gestión.

Así pues, la trazabilidad es el conjunto de medidas, acciones y procedimientos que permiten registrar e identificar un determinado producto desde su nacimiento hasta su destino final. Y consiste en la capacidad para reconstruir la historia, recorrido o aplicación de un determinado producto, identificando:

- El origen de sus componentes.
- La historia de los procesos aplicados al producto.
- Y la distribución y localización después de su entrega.

Al contar con esta información es posible entregar productos definidos a mercados específicos, con la garantía de conocer con certeza el origen y la historia del mismo.

A lo largo de esta memoria, se exponen distintos capítulos que describen el funcionamiento de la aplicación, así como otras cuestiones de interés.

En una primera fase se realiza un análisis de la situación, haciendo un estudio del arte, redactando el documento de especificación de requisitos y haciendo una explicación de las distintas tecnologías utilizadas en este proyecto. En el estado del arte, lo que se pretende es indagar sobre las distintas aplicaciones que puedan existir en el mercado, con la intención de ver, en la medida de lo posible, cómo realizan la trazabilidad de los procesos.

El siguiente paso es la redacción del documento de especificación de requisitos, donde se ha intentado detallar todas las restricciones y características que este prototipo tiene. Para

que resulte más explicativo, se acompañan con diagramas de casos de uso donde se analizan las acciones de los distintos tipos de usuarios.

En cuanto al diseño de la aplicación, en el capítulo relativo a la arquitectura y su funcionamiento, se explican con detalle los distintos aspectos técnicos de los procesos con mayor importancia en la aplicación, tales como el filtrado de datos, la inserción de los distintos tipos de datos o cómo se ha realizado la trazabilidad de todo el proceso. Asimismo, se explican las distintas tecnologías que han sido utilizadas, así como el papel que aportan al proyecto.

Dado que es un prototipo y no una aplicación final, la última parte de este Trabajo Fin de Grado (en adelante TFG) se centra en las mejoras futuras a realizar.

Por último, se incluye una serie de anexos que aportan información complementaria sobre algunos apartados tratados en el TFG. Cabe destacar el Anexo G, donde se presenta el manual de usuario de la aplicación BiomasaEUP.T.

1.1. Contexto

La idea de desarrollar este proyecto surge de la necesidad de una empresa del sector maderero. Esta empresa contactó con la Escuela Universitaria Politécnica de Teruel (EUP.T) para buscar personal que desarrollara una aplicación real que obtuviese trazabilidad en la industria maderera y que pudiese ser instalada en sus equipos informáticos. Pues querían automatizar la labor que sus empleados realizan a mano.

La empresa explicó a groso modo a los tutores cómo estaba organizada su actividad y el modo de trabajo dentro de la misma. Sin embargo, la empresa decidió cortar la comunicación con la universidad cuando el proyecto ya estaba iniciado, por lo que se decidió continuar el proyecto suponiendo que se desarrollaba para una empresa del sector maderero. Pero realizándolo de la manera más genérica posible debido a que no se tenían conocimientos de dicho sector y, además, de este modo, podría ser fácilmente adaptable a cualquier otro sector empresarial en un futuro.

Explicación del funcionamiento de la empresa

La empresa (sobre la que se basa la idea del proyecto) se dedica a la elaboración de productos cuya materia prima es la madera. Para ello, ésta cuenta con unos **proveedores** que le suministran las **materias primas** que serán usadas posteriormente para fabricar dichos productos.

Los proveedores entregan las materias primas a través de camiones (**recepciones**). Dichas materias primas entregadas por el camión serán de un mismo proveedor.

Al descargar las materias primas de cada recepción, éstas deberán almacenarse en almacenes o explanadas (**sitios de recepción**). Cada una de estas explanadas estará subdividida en partes (**huecos de recepción**) para que así no se mezclen las materias primas de las diferentes recepciones.

Cada hueco de recepción constará de una capacidad total que puede medirse en volumen (m^3) o en unidades (dependiendo de la materia prima que haya almacenada). Si la cantidad de materias primas que vinieron en una recepción es muy grande y no cabe en un solo hueco, deberá ser almacenada en varios huecos.

Posteriormente, se crearán **órdenes de elaboración** mediante las cuales, usando las materias primas, se fabricarán los **productos terminados**. Una orden podrá tener varios productos terminados para poder empezar con el proceso de fabricación.

Las órdenes de elaboración serán encargadas cuando no se disponga de stock de los productos terminados o cuando un **cliente** haya realizado un **pedido**.

Por último, los clientes harán una serie de encargos o pedidos a la empresa. Si se tiene stock de dichos productos terminados, éstos se envasarán y serán vendidos. Si no, se deberán crear las órdenes de elaboración oportunas para fabricar los productos reclamados por el cliente, de forma que cuando sean fabricados le sean entregados al cliente.

En cuanto a la trazabilidad, puede realizarse en cada momento del proceso de fabricación con el fin de obtener la información que corresponda en cada etapa (recepción, elaboración y venta).

1.2. Trazabilidad

Definición

Según el diccionario de la RAE (Real Academia Española) se define la trazabilidad como *“Posibilidad de identificar el origen y las diferentes etapas de un proceso de producción y distribución de bienes de consumo”* [1].

La norma UNE-EN ISO 9000 de la Asociación Española de Normalización y Certificación (AENOR) define trazabilidad como la *“Capacidad para seguir la historia, la aplicación o la localización de todo aquello que está bajo consideración”* [2].

Según la AECOC (Asociación Española de Codificación Comercial) se define la trazabilidad como el *“Conjunto de procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministro, en un momento dado y a través de herramientas determinadas”* [3].

Se podría llegar a la conclusión, a partir de las definiciones anteriores, que la trazabilidad está compuesta por procesos prefijados que se llevan a cabo para determinar los diversos pasos que recorre un producto, desde su nacimiento hasta su ubicación actual en la cadena de abastecimiento.

El sistema de trazabilidad que se va a desarrollar en este Trabajo de Fin de Grado se soporta en tres pilares:

1. Identificación: Un sistema que permita identificar los productos de manera única.
2. Sistemas para la captura y comunicación de datos: Un sistema que permita tanto adquirir información acerca del producto, como comunicárselo a los demás agentes de la cadena.
3. Gestión de datos: Consiste en un sistema informático donde se pueda realizar la gestión de la información acerca de la trazabilidad del producto en cada momento. Por ejemplo, que pueda proporcionar información o imprimir las etiquetas con los códigos de barras.

Legislación

El concepto de trazabilidad que debe aplicarse en el ámbito que abarca el proyecto se encuentra reflejado en la norma UNE-EN ISO 9001 [4], que dice:

“Cuando sea apropiado, la organización debe identificar el producto por medios adecuados, a través de toda la realización del producto.

La organización debe identificar el estado del producto con respecto a los requisitos de seguimiento y medición a través de toda la realización del producto.

Cuando la trazabilidad sea un requisito, la organización debe controlar la identificación única del producto y mantener registros”.

Según la Ley 17/2011, de 5 de julio, del BOE (Boletín Oficial del Estado) de seguridad alimentaria y nutrición [5], en algunos sectores, como es el caso de la alimentación, es obligatorio, en todas las etapas de producción, transformación y distribución, garantizar la trazabilidad de los alimentos, piensos, animales destinados a la producción de alimentos y de cualquier sustancia o producto que se incorpore o pueda incorporarse a los alimentos o los piensos.

Importancia

En una empresa, el hecho de tener un proceso de trazabilidad completo y fiable, a lo largo de la cadena de suministro de un producto, es una de las herramientas imprescindibles a la hora de prevenir y detectar un problema y lugar donde se produce, suponiendo un ahorro de costes.

1.2.1. Códigos de identificación únicos de captura automática

Para llevar a cabo el proceso de obtención de la trazabilidad es necesario hacer uso de códigos de identificación único de captura automática. Estos códigos ayudan en la identificación inequívoca de cada unidad o producto elaborado, donde, en una aplicación real, se utilizarían claves de identificación del estándar Global GS1, especialmente el GTIN (Número Mundial del Artículo Comercial) y un número de serie único por producto.

En una aplicación real también se requiere que los datos sean capturados mediante la inclusión de alguna de las distintas tecnologías de identificación automática (códigos de barras, códigos QR, etiquetas RFID –*Radio Frequency Identification*, Identificación por Radio Frecuencia–). De esta manera se automatiza la lectura de éstos mediante dispositivos electrónicos (ej.: lector de códigos de barras, de códigos QR, etc.).

A continuación, se explican dos tipos de códigos que son usados habitualmente.

Código de barras GS1-128 (código lineal)

Código de barras lineal, al igual que *Data Matrix*, pudiendo ser impreso sobre un estuche o en una etiqueta para su posterior aplicación [6].

Pese a la clara desventaja que tiene en la relación tamaño/densidad de información, cuenta con una difundida y extensa utilización, debido principalmente a que puede emplearse cualquier lector de código de barras para la captura del dato.

Código Data Matrix ECC200 2D

Tecnología óptica que surge como la evolución natural del código de barras convencional. A diferencia de su antecesor, permite la codificación de un gran volumen de información en un espacio reducido [6].

Este código bidimensional debe ser colocado en el exterior del estuche del producto a identificar, pudiendo ser impreso directamente en el producto, el estuche o añadido mediante la colocación de una etiqueta autoadhesiva.

La captura de datos se realiza con un escáner óptico de imagen, requiriendo el procesamiento de las unidades en forma individual.

Para este proyecto se ha utilizado el código de barras GS1-128, ya que, a diferencia del código de barras Data Matrix ECC200, el primero se genera de manera muy fácil con respecto al segundo.

Una de las desventajas que tiene utilizar estos códigos es que es necesario pedir autorización para la utilización del código GS1, donde son asignados los rangos que la empresa, aplicación u organización puedan utilizar.

En una aplicación real se seguiría, por ejemplo, el código GS1-128 cuyo formato [7] sería el especificado en la Figura 2.

Este código contiene dos dígitos para el identificador de la aplicación, un dígito de extensión, siete dígitos para el prefijo GS1 que asigna a la compañía, tres dígitos que pueden corresponder al prefijo GS1 o al número de serie basado en la longitud del prefijo anterior, seis dígitos para el número de serie y un dígito de control.



FIGURA 2. Estructura del código GS1-128.

Por el momento, se utilizará un código de barras cuyo número corresponda íntegramente al código de identificación de 10 dígitos que le corresponderá según la fase del proyecto en el que se introduzca, incrementándose de forma secuencial, como, por ejemplo:

- 1000000001 → Identificación de las materias primas, a través de un código que comienza en 1.
- 2000000001 → Identificación de los productos terminados, a través de un código que comienza en 2.
- 3000000001 → Identificación de los productos de ventas, a través de un código que comienza en 3.

Se ha optado por usar este formato porque es un prototipo sin fines comerciales y porque no valdría la numeración, ya que en el formato GS1-128 el número de serie tiene una longitud de 6 dígitos y en el proyecto son necesarios 10 dígitos, lo que hace imposible su utilización puesto que no cabría.

1.3. Motivación, problemas y objetivos

Una de las principales razones por las que se ha escogido este TFG es porque debía realizarse una aplicación de escritorio con la condición de que no fuera en Java, por lo tanto, se debían usar nuevas herramientas y lenguajes que no se habían estudiado en el Grado, esto es el lenguaje C#, SQL Server, Visual Studio, ...

Asimismo, una de las motivaciones ha sido la realización de dicho trabajo entre dos personas, dividiendo claramente cada una de las tareas a realizar por cada uno de los estudiantes en fases, así como todo lo aprendido durante el desarrollo del mismo.

Pese a la motivación de aprender a usar nuevas herramientas y tecnologías este proyecto se ha alargado más de lo necesario porque la tecnología XAML usada en la vista, además de ser muy potente es muy compleja y requiere mucho más tiempo para dominarla. Es por ello que el prototipo desarrollado podría contar con alguna funcionalidad más si se hubiese realizado con lenguajes más sencillos como es el caso de PHP y HTML.

Respecto a los objetivos principales del proyecto son:

- Realizar un prototipo de aplicación de trazabilidad que dé solución a las necesidades de una industria de la biomasa (reutilizable en otro tipo de sectores).
- Diseñar e implementar un prototipo que permita realizar la trazabilidad de un producto desde la entrada de las materias primas hasta su venta.
- Aprender a desarrollar en el lenguaje de programación C# y SQL Server.
- Identificar el origen y las diferentes etapas del proceso de producción y distribución de los bienes producidos por la empresa.
- Realizar la trazabilidad del proceso final, es decir, desde el almacenamiento del proceso hasta las ventas.

Además de los objetivos principales existen otros objetivos secundarios que se especifican a continuación:

- Diseñar una interfaz de usuario lo más usable, sencilla e intuitiva posible, para que los trabajadores puedan efectuar las tareas de manera cómoda y fácil.
- Proporcionar un entorno de implantación del sistema y funcionamiento de fácil mantenimiento y bajo coste económico.
- Desarrollar una aplicación eficaz en la realización de las distintas consultas a la Base de Datos y con control de errores.

2. ESTADO DEL ARTE

2.1. Introducción

El estado del arte implica el desarrollo de una metodología resumida en tres grandes pasos: contextualización, clasificación y categorización [8]. En este caso se trata de una recopilación de diferentes aplicaciones existentes en el mercado que pueden ser de utilidad para conocer lo que ofrece el mercado.

Es importante mencionar que, tanto el software *Open Source* desarrollado para este proyecto como el software comercial que se expone en el siguiente subapartado, necesita del manejo de grandes volúmenes de datos para que este control se realice de manera eficaz.

2.2. Aplicaciones de trazabilidad en el mercado

Sage Murano

Conjunto de módulos ERP (*Enterprise Resource Planning*) que conforman Sage Murano. El módulo Sage Logic Class es el que genera la trazabilidad. Permite asegurar el óptimo control de productos y la trazabilidad de dicho control cuando se utiliza en cualquier proceso industrial [9].

También permite obtener trazabilidad en procesos laborales, tales como en la elaboración de la hoja de salario, así como el cálculo del % de IRPF (Impuesto sobre la Renta de las Personas Físicas).

Ilean Tracer

Ofrece soluciones para la obtención de la trazabilidad en cualquier proceso industrial mediante el uso de pantallas táctiles y con dispositivos móviles. También se apoya en que puede ser usado por personal sin formación informática, mediante la impresión por parte del usuario de las distintas etiquetas a utilizar [10].

Gestión5 SQL Trazabilidad y Lotes

Conjunto de módulos ERP para la gestión completa de cualquier empresa del sector alimentario, gestionando el proceso de compra, stock y almacén, fabricación y composiciones, ventas y distribuciones e informes sobre calidad. Otro punto fuerte de esta aplicación es que integra el etiquetado con EAN128 y que posibilita la asignación de lotes automáticamente y su posterior captura con lectores móviles [11].

Trazatec

Software de trazabilidad de la producción y gestión de la cadena de suministro para la industria en general y los sectores de la alimentación y distribución en particular.

Es totalmente flexible e integrable con las bases de datos estándar, permitiendo complementar a cualquier ERP, puesto que éstos no cubren particularidades específicas de entornos productivos complejos [12].

SolinCloud

Facilita la gestión de la trazabilidad en la nube tanto de empresas que se dedican exclusivamente a la comercialización y distribución de productos como a las que también se encargan del proceso de fabricación, elaboración o transformación de sus propios productos a partir de las materias primas, permitiendo el seguimiento de cualquier lote o partida (tanto comprado como vendido), su origen, proceso y destino [13].

IOAlimentación

Sistema ERP para el sector cárnico que incorpora el software para la gestión de la trazabilidad alimentaria. Integrado completamente en el sistema ERP Microsoft Dynamics NAV y especialmente diseñado para la trazabilidad del sector alimentario [14].

Alba IBS

Software ERP que contiene distintos módulos entre ellos uno para la gestión de la trazabilidad para distintos sectores (alimentación, hostelería, droguerías, ...). Controla lotes, fechas, históricos por lote, asignación e impresión de códigos de barra a lotes, control de trazabilidad en las ventas, lectura de etiquetas y otras funcionalidades [15].

2.3. Conclusiones

Del análisis de los diferentes productos software estudiados en este apartado se deduce que la gran mayoría del software desarrollado está especializado en la industria alimentaria. La razón es muy simple, la normativa española e internacional así lo exige para toda empresa que se dedique a la producción, distribución o elaboración de comida.

También se puede ver que la trazabilidad es muy importante en todo proceso industrial que se desarrolle, de ahí que se integren en módulos ERP. Esto tiene todo el sentido, puesto que, con este tipo de sistema de información, se integran y manejan muchos de los negocios asociados con las operaciones de producción y los aspectos de distribución de una compañía en la producción de bienes o servicios.

El prototipo desarrollado en este TFG se centra en la trazabilidad de un proceso industrial concreto. En este caso, relacionado con la industria maderera, pero modulable, de manera que se pueda utilizar en cualquier proceso industrial como los analizados anteriormente.

3. ANÁLISIS

Una vez tratado el estado del arte, el siguiente apartado del proyecto es el análisis. En este punto se especificarán los requisitos funcionales y no funcionales que deberá cumplir el prototipo desarrollado.

En este Trabajo Fin de Grado, el rol de cliente lo han adoptado tanto los dos tutores como los dos estudiantes que lo hemos desarrollado. Con los que se han mantenido una serie de reuniones a lo largo del proceso de desarrollo del mismo.

3.1. Casos de Uso

Teniendo claro los requisitos y sus restricciones, se pasará a tratar los distintos Casos de Uso (CU) que representan las interacciones de los distintos usuarios, así como las funcionalidades que tendrá el prototipo.

En primer lugar, se representarán gráficamente usando diagramas UML los casos de uso, posteriormente se hará una descripción más detallada de los mismos.

Diagramas

Como se puede observar en las cuatro ilustraciones siguientes existen cuatro actores que corresponden con los cuatro tipos de usuarios predefinidos en el sistema.

Para diferenciar los CU correspondientes a cada uno de los dos participantes en el desarrollo del prototipo, se muestran de color blanco los que corresponden a las funcionalidades implementadas por Javier Bielsa Hernández, mientras que los realizados en este TFG se representan de color azul. Además, el CU de obtención de la trazabilidad está coloreado de color más claro debido a que es común.

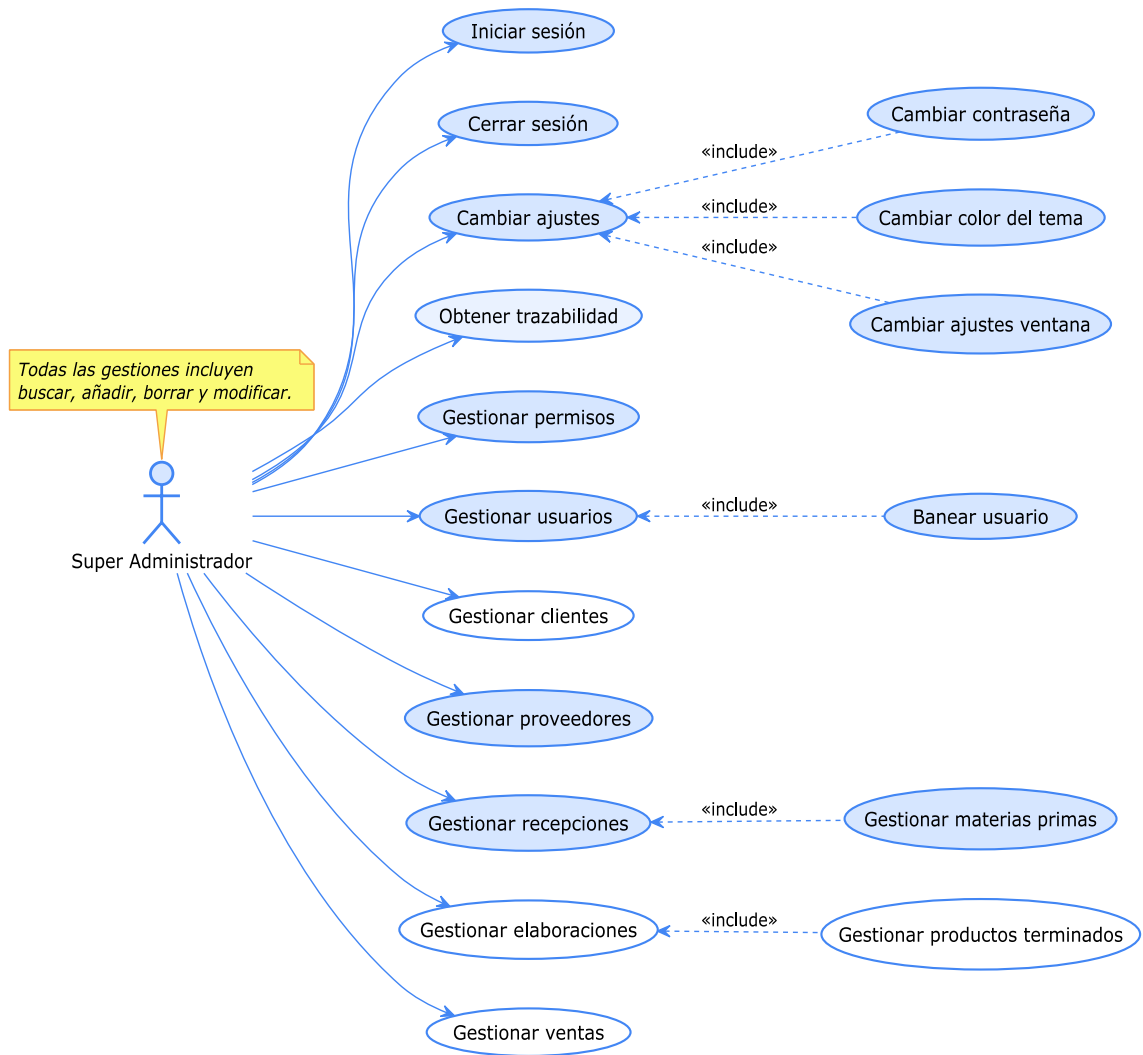


FIGURA 3. Diagrama de Casos de Uso del usuario Super Administrador.

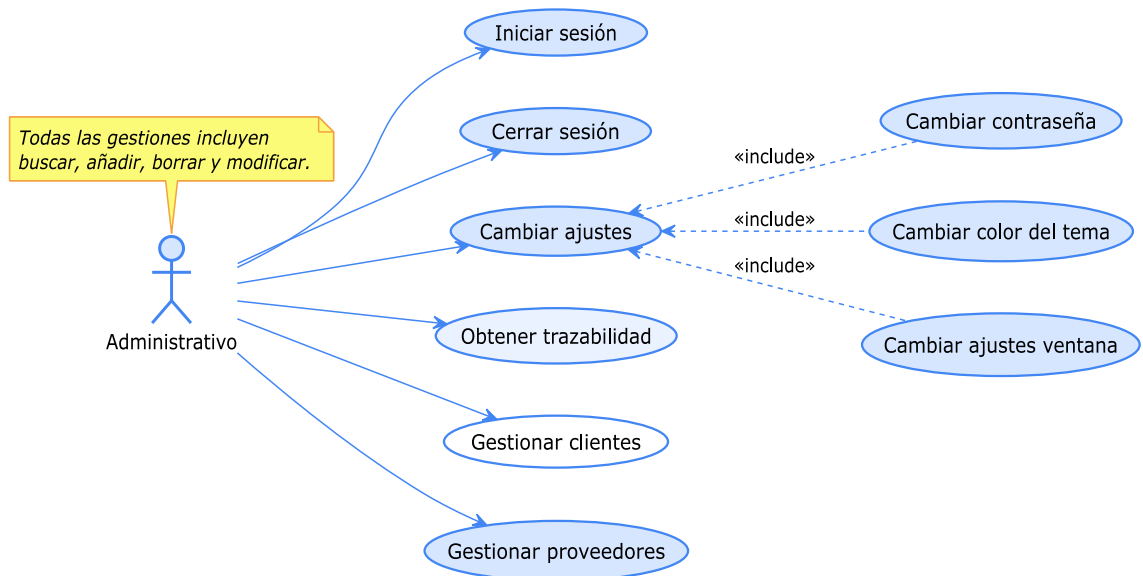


FIGURA 4. Diagrama de Casos de Uso del usuario Administrativo.

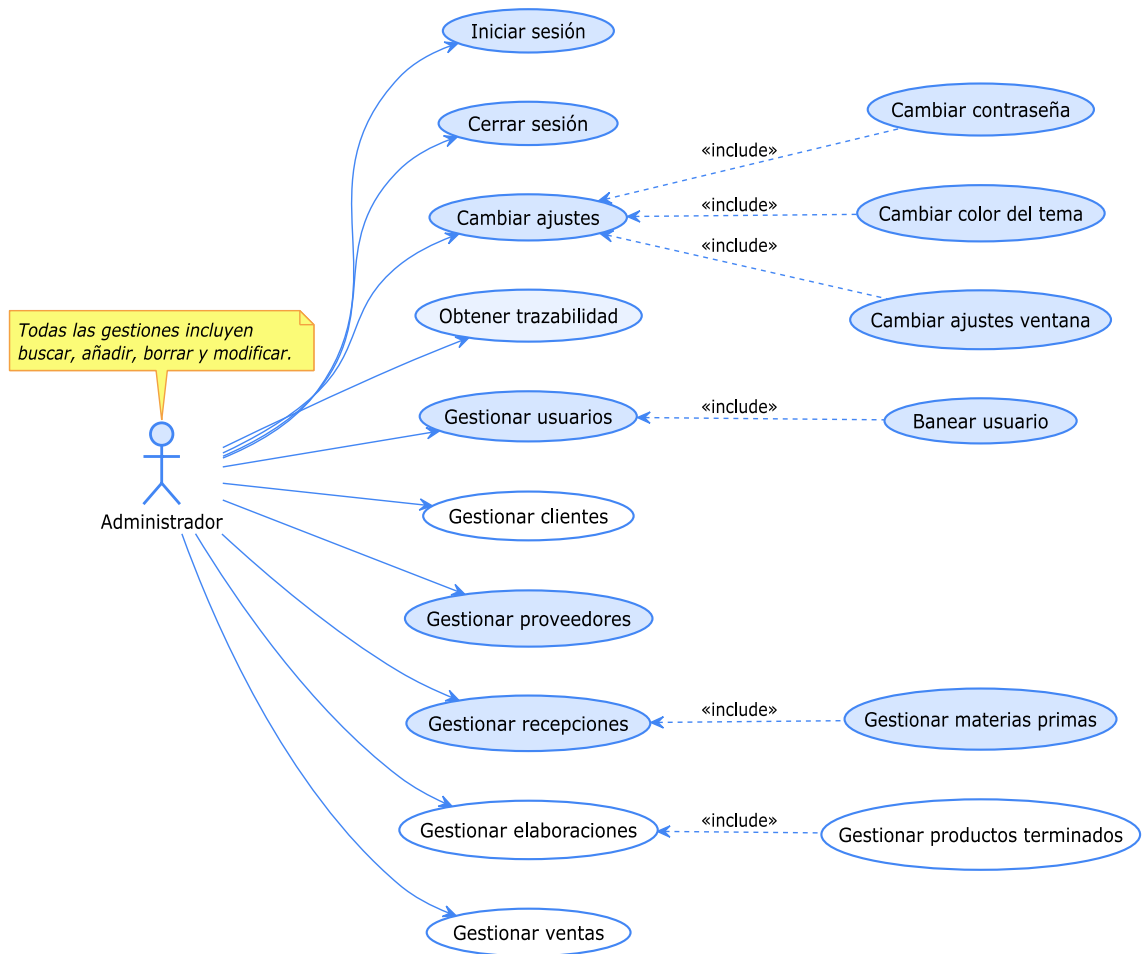


FIGURA 5. Diagrama de Casos de Uso del usuario Administrador.

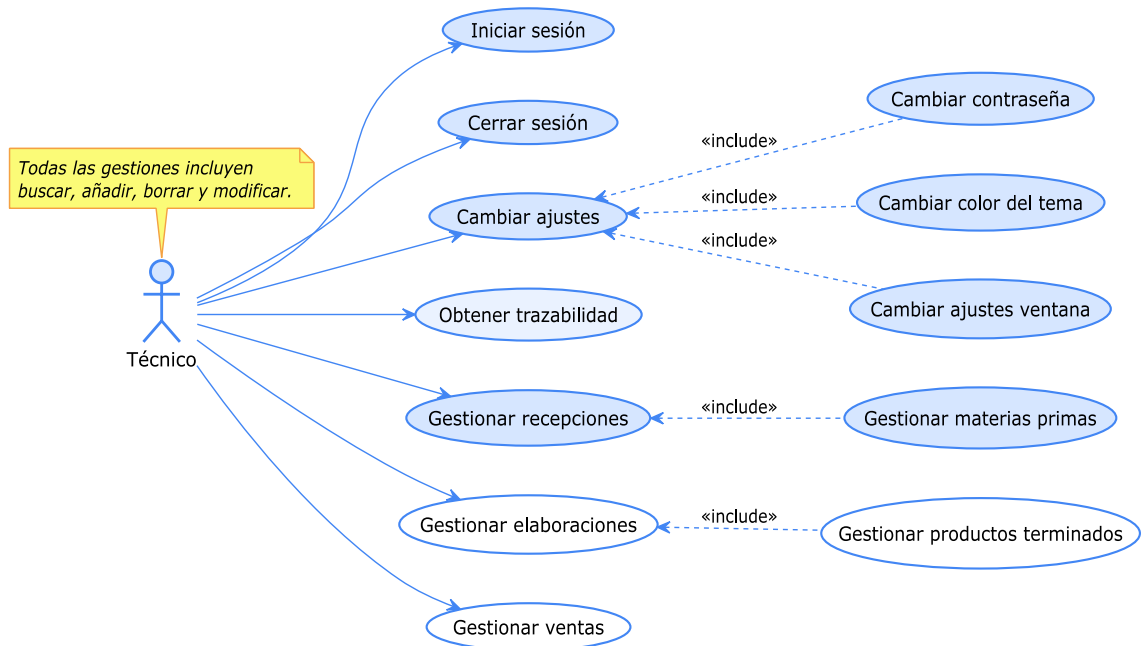


FIGURA 6. Diagrama de Casos de Uso del usuario Técnico.

Descripción

A continuación, se describe textualmente cada uno de los casos de uso que han sido implementado en este TFG.

- Iniciar Sesión: Antes de poder hacer uso de la aplicación, el usuario deberá iniciar sesión.
- Cerrar Sesión: En todo momento el usuario podrá cerrar su sesión regresando a la ventana de inicio de sesión.
- Gestionar usuarios: Si se posee el permiso para gestionar los usuarios, se podrán registrar, modificar y borrar usuarios en el sistema, así como buscarlos (filtrar).
- Gestionar clientes: Si se posee el permiso para gestionar los clientes, el usuario podrá añadir, modificar o borrar clientes en el sistema. Además de poder buscar clientes.
- Gestionar proveedores: Al igual que con la gestión de clientes, se podrán añadir, modificar o borrar proveedores en el sistema.
- Gestionar recepciones: Si se posee el permiso para gestionar recepciones, el usuario podrá añadir, modificar o borrar recepciones de materias primas en el sistema.
- Añadir materia prima: Para cada recepción, el usuario podrá añadir, modificar o borrar sus materias primas.
- Obtener trazabilidad: El usuario podrá obtener la trazabilidad completa del sistema, ya sea a partir del número de albarán de una recepción, a partir del código de una materia prima, a partir del código de un producto terminado o a partir del código de un producto envasado. En este TFG sólo se obtendrá la trazabilidad de las recepciones y de las materias primas.
- Gestionar permisos: El usuario que cuente con la capacidad de gestionar los permisos podrá añadir, modificar o borrar los distintos tipos de usuarios que haya en el sistema.
- Cambiar ajustes: Cualquier usuario podrá cambiar los ajustes de la aplicación, es decir, podrá cambiar su contraseña, el color del tema (tanto el color primario y secundario como activar el modo nocturno) o cambiar los ajustes de la ventana (iniciar siempre maximizado, recordar última pestaña activa, ...).

Si se desea un mayor detalle de cada uno de los casos de uso, ver el Anexo A “Casos de Uso”.

3.1. Especificación de Requisitos de Software

En este apartado se especificarán los distintos requisitos de software es los que se detalla la funcionalidad de la aplicación y sus restricciones con un nivel de abstracción más bajo.

3.1.1. Requisitos comunes de los interfaces

Interfaces de usuario

RQI01. La aplicación deberá presentar una interfaz intuitiva, siguiendo criterios de usabilidad y accesibilidad para que sea intuitiva y de fácil acceso.

- RQI02.** La interfaz de usuario debe ser orientada a ventanas, manejándose con teclado y ratón.
- RQI03.** Todas las ventanas de la aplicación se diseñarán siguiendo el mismo estilo. Concretamente usando *Material Design* (normativa de diseño, desarrollada por Google, inicialmente para el sistema operativo Android).
- RQI04.** En todo momento existirá la posibilidad de que el usuario pueda salir de la aplicación.
- RQI05.** Los mensajes de errores serán claros y concisos para que cualquier usuario pueda entenderlos.
- RQI06.** La aplicación tendrá una resolución por defecto de 900x650 píxeles. Pudiéndose agrandar y adaptar el tamaño de las ventanas a la resolución de la pantalla donde se ejecute.
- RQI07.** Los iconos usados deberán ser simples e intuitivos.
- RQI08.** El idioma de la aplicación deberá ser en castellano, ya que está pensado para el mercado español.
- RQI09.** Se podrán filtrar datos en las tablas.

Interfaces de hardware

- RQI10.** El software deberá mostrar información al usuario a través de la pantalla del monitor.
- RQI11.** El software deberá interactuar con el movimiento del ratón y los botones de éste.
- RQI12.** El software deberá interactuar con las pulsaciones del teclado.
- RQI13.** El software actuará de forma diferente al pulsar los atajos de teclado Ctrl+Shift+A (para abrir el formulario de añadir un registro), Ctrl+Shift+B (para borrar uno o más registros de la tabla), Ctrl+Shift+M (para modificar un registro) y Ctrl+Shift+R (para refrescar los datos de la pantalla) en función de la pestaña en la que esté situado el usuario.
- RQI14.** El software actuará al pulsar el atajo de teclado Alt+Letra (siendo la letra la que aparece subrayada) en el menú situado en la parte superior de la aplicación, de manera que se puedan controlar las funcionalidades de esas opciones mediante el teclado.

Interfaces de software

- RQI15.** La aplicación se programará en C# y siguiendo las directrices del Paradigma de Orientación a Objetos.
- RQI16.** Se usará SQL Server para las cuestiones relacionadas con la base de datos.

Interfaces de comunicación

- RQI17.** La comunicación entre la aplicación y la base de datos se realizará mediante el *framework* para C# Entity Framework, usando los métodos que tiene definidos además de los propios del lenguaje de programación (consultas, modificaciones, inserciones y borrados) así como la notación LINQ.
- RQI18.** La base de datos permanecerá activa constantemente, atendiendo las peticiones de los clientes que deseen realizar alguna acción.

3.1.2. Requisitos funcionales

Se exponen a continuación todos los requisitos encargados de especificar todas las funcionalidades que el sistema debe satisfacer.

Inicio de sesión

- RQF01.** Para poder usar la aplicación el usuario deberá iniciar sesión.
- RQF02.** El prototipo permitirá recordar el nombre de usuario y contraseña para poder iniciar sesión de forma automática al arrancar la aplicación.
- RQF03.** Dependiendo del usuario que inicie sesión, la aplicación cargará las pestañas correspondientes a los permisos que tenga dicho tipo de usuario.
- RQF04.** Si un usuario está suspendido (baneado) no podrá iniciar sesión.
- RQF05.** Se permitirá cerrar la sesión en cualquier momento desde el menú superior.

Gestión de permisos

- RQF06.** El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) los tipos de usuarios.
- RQF07.** Para poder gestionar los tipos de usuarios, el usuario deberá tener el permiso de gestión de permisos.
- RQF08.** Cada tipo de usuario contará con un nombre, una descripción y una serie de permisos.
- RQF09.** Los permisos disponibles para los tipos de usuarios son: gestión de permisos, gestión de usuarios, gestión de clientes, gestión de proveedores, gestión de recepciones, gestión de elaboraciones, gestión de ventas y obtención de trazabilidad.
- RQF10.** La aplicación deberá contar con un tipo de usuario, con al menos un usuario asignado, el cual tenga el permiso para poder gestionar los permisos.
- RQF11.** Por defecto en el prototipo existirán los siguientes tipos: Super Administrador (con todos los permisos), Administrador (con todos los permisos menos el de poder gestionar los propios permisos), Administrativo (con los permisos para gestionar los clientes y los proveedores, así como obtener la trazabilidad) y Técnico (con los permisos para gestionar las recepciones, las elaboraciones y las ventas, así como obtener la trazabilidad).
- RQF12.** Un usuario que no cuente con la capacidad de otorgar permisos, pero sí con la de gestionar usuarios, no podrá seleccionar de los tipos de usuarios disponibles los tipos que cuenten con la gestión de permisos. Es decir, un usuario no puede asignar más privilegios que los que él tenga.

Gestión de usuarios

- RQF13.** El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) los usuarios del sistema.
- RQF14.** Para poder gestionar los usuarios, el usuario deberá tener el permiso de gestión de usuarios.

RQF15. No se permitirá a los propios usuarios registrarse. Para ello deberán registrarse a través de la ventana de gestión de usuarios.

RQF16. Cada usuario contará con un nombre, una contraseña, una dirección de correo electrónico, un tipo de usuario, una fecha de alta en el sistema, una fecha de baja del sistema, una fecha con el último cambio de su contraseña.

RQF17. A los usuarios se les podrá prohibir el acceso al sistema.

RQF18. Existirá un contador de los tipos de usuarios en la pestaña de gestión de usuarios, con el fin de controlar la cantidad de usuarios que hay de cada tipo en el sistema.

Gestión de clientes y proveedores

RQF19. El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) los cliente y proveedores del sistema.

RQF20. Para poder gestionar los clientes o proveedores, el usuario deberá tener el permiso de gestión de clientes o proveedores respectivamente.

RQF21. Cada cliente y proveedor contará con una razón social, un NIF (podrá ser un NIF de persona o de empresa), una dirección de correo electrónico, un municipio con la correspondiente calle y un tipo de cliente/proveedor.

RQF22. Cada cliente y proveedor podrá tener una serie de observaciones.

RQF23. Los clientes estarán asociados a un grupo de clientes.

RQF24. Los tipos de clientes y proveedores tendrán un nombre y una descripción, al igual que los grupos de clientes.

Gestión de recepciones de materias primas

RQF25. El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) las recepciones de materias primas del sistema.

RQF26. Para poder gestionar las recepciones, el usuario deberá tener el permiso de gestión de recepciones.

RQF27. Cada recepción contará con su fecha de entrega, un número de albarán, un estado y estará asociada a un proveedor.

RQF28. Los estados de una recepción serán: disponible (se tienen que añadir las materias primas) o aceptada (todas las materias primas fueron registradas en el sistema).

RQF29. Cada recepción podrá contener una o varias materias primas.

RQF30. Cuando se añada una recepción al sistema, no se podrá seleccionar su estado ya que se seleccionará por defecto el estado disponible.

RQF31. Cada materia prima contará con una cantidad (volumen o unidades), una fecha de baja (de caducidad o cuando la cantidad sea cero), estará asociada a un tipo de materia prima, una procedencia y un código de identificación único.

RQF32. Las materias primas podrán contener observaciones.

- RQF33.** Al añadir una materia prima, deberá especificarse una cantidad mayor de cero y se deberán añadir suficientes huecos de recepción para poder descargar la cantidad de materia especificada.
- RQF34.** Si el estado de la recepción es “aceptada” entonces no se le podrán añadir nuevas materias primas.
- RQF35.** Los sitios de recepción deberán tener un nombre y una descripción.
- RQF36.** Los huecos de un sitio de recepción deberán tener un nombre y una descripción.
- RQF37.** Las procedencias de las materias primas deberán tener un nombre y una descripción.
- RQF38.** Los grupos de materias primas deberán tener un nombre y una descripción.
- RQF39.** Cada tipo de materia prima estará asociado a un grupo de materias primas. Contará con un nombre y una descripción.

Gestión de elaboraciones

Estos requisitos corresponden al TFG de Javier Bielsa Hernández.

- RQF40.** El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) las elaboraciones con el fin de poder generar productos terminados.
- RQF41.** Para poder gestionar las elaboraciones, el usuario deberá tener el permiso de gestión de elaboraciones.
- RQF42.** Cada orden de elaboración contará con una fecha de elaboración de dicha orden de elaboración, un estado y una descripción.
- RQF43.** Los estados de una orden de elaboración serán: nueva (la orden de elaboración acaba de ser creada), procesando (se añaden materias primas para obtener un producto terminado y/o se pueden seguir añadiendo) o finalizada (no se pueden añadir más materias primas porque ese producto ya ha sido fabricado).
- RQF44.** Cada producto terminado podrá contener una o varias materias primas de distintos huecos de recepción.
- RQF45.** Cuando se añada una orden de elaboración al sistema, no se podrá seleccionar su estado ya que se seleccionará por defecto el estado nuevo mientras no se añadan materias primas.
- RQF46.** Cada producto terminado contará con una cantidad (volumen o unidades), una fecha de baja (de caducidad o cuando la cantidad en cero), estará asociada a un tipo y grupo de producto terminado y un código de identificación único.
- RQF47.** Los productos terminados podrán contener observaciones.
- RQF48.** Al añadir un producto terminado, deberá especificarse una cantidad mayor de cero y se deberán añadir suficientes huecos de almacenaje para poder depositar la cantidad de producto terminado especificado.
- RQF49.** Si el estado de la orden de elaboración es “procesando”, entonces no se le podrán añadir nuevos productos terminados.

RQF50. Los sitios de almacenaje deberán tener un nombre y una descripción.

RQF51. Los huecos de un sitio de almacenaje deberán tener un nombre y una descripción.

RQF52. Los grupos de productos terminados deberán tener un nombre y una descripción.

RQF53. Cada tipo de producto terminado estará asociado a un grupo de productos terminados, contando con un nombre y una descripción.

Gestión de ventas

Estos requisitos corresponden al TFG de Javier Bielsa Hernández.

RQF54. El prototipo permitirá gestionar (añadir, modificar, borrar y refrescar) las ventas con el fin de poder generar productos envasados mediante pedidos.

RQF55. Para poder gestionar las elaboraciones, el usuario deberá tener el permiso de gestión de ventas.

RQF56. Cada pedido contará con una fecha de pedido, un estado, su fecha de finalización y un cliente asociado al mismo.

RQF57. Los estados de un pedido serán: pendiente (el pedido acaba de ser creado), procesando (se añaden productos terminados del hueco que corresponda para crear un producto envasado) o completo (no se pueden añadir más productos terminados porque ese producto ya ha sido envasado).

RQF58. Cada producto envasado podrá contener uno o varios productos terminados de distintos huecos de almacenaje.

RQF59. Cuando se añada un pedido al sistema, no se podrá seleccionar su estado ya que se seleccionará por defecto el estado “pendiente” mientras no se añadan productos terminados.

RQF60. Cada producto envasado contará con una cantidad (volumen), estará asociada a un tipo y grupo de producto terminado, un código de identificación único y el *picking* donde se preparará dicho producto para que le sea entregado al cliente.

RQF61. Los productos envasados podrán contener observaciones.

RQF62. Al añadir un producto envasado, deberá especificarse una cantidad mayor de cero y se deberán añadir suficientes huecos de almacenaje para poder depositar la cantidad de producto envasado especificado.

RQF63. Si el estado del pedido es “procesando”, entonces no se le podrán añadir nuevos productos terminados.

RQF64. El *picking* contará con un nombre, un volumen total y un volumen restante

RQF65. Los grupos de productos terminados deberán tener un nombre y una descripción.

RQF66. Cada tipo de producto terminado estará asociado a un grupo de productos terminados, contando con un nombre y una descripción.

Obtención de la trazabilidad

RQF67. Se podrá obtener la trazabilidad de una recepción usando su número de albarán.

RQF68. Se podrá obtener la trazabilidad de una materia prima usando su código de identificación único.

RQF69. El prototipo tendrá la posibilidad de exportar cada trazabilidad a un fichero PDF.

RQF70. En la trazabilidad deberá representarse de forma gráfica el código de identificación único (código de barras) de las materias primas.

Cambio de los ajustes

RQF71. El prototipo contará con un menú de ajustes.

RQF72. Se accederá al menú de ajustes a través de la opción correspondiente ubicada en el menú superior.

RQF73. En el menú de ajustes se permitirá cambiar la contraseña del usuario.

RQF74. En el menú de ajustes se permitirá cambiar la apariencia de la aplicación. Deberá contar con un modo nocturno además de permitir cambiar los colores.

RQF75. En el menú de ajustes se permitirá iniciar la aplicación con la ventana maximizada, recordar el tamaño de la ventana y posición de esta.

RQF76. En el menú de ajustes se permitirá seleccionar el directorio donde se guardarán los ficheros de los informes en PDF.

RQF77. En el menú de ajustes se permitirá activar o desactivar la búsqueda de actualizaciones al arrancar la aplicación.

3.1.3. Requisitos no funcionales

Se exponen a continuación los requisitos que representan restricciones, necesidades o condiciones de las funcionalidades del sistema, o algunos aspectos de seguridad del mismo.

Rendimiento

RQNF01. El tiempo de respuesta de la aplicación será el aceptable para una aplicación de escritorio, es decir, cuando se haga clic en algún componente de la interfaz se deberá ver el resultado al instante (ej.: si se hace clic en el botón “añadir usuario” se deberá abrir el formulario para añadir un nuevo usuario).

RQNF02. La aplicación debe ser compatible y totalmente funcional con los sistemas operativos de Microsoft (Windows 7, Windows 8, Windows 8.1 y Windows 10).

RQNF03. La aplicación exigirá unos requisitos de memoria de 250 Mbytes.

RQNF04. La aplicación requerirá la versión 4.5.2 o superior de .NET Framework.

Desarrollo

RQNF05. Se requiere el uso de la notación UML en los diagramas de diseño.

RQNF06. Deberá proporcionarse junto a la aplicación un manual de usuario.

RQNF07. El ciclo de vida elegido para desarrollar la aplicación será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funcionalidades.

Seguridad

RQNF08. Sólo se podrá acceder a la aplicación si el usuario se encuentra dado de alta en la base de datos.

RQNF09. Las contraseñas de los usuarios deberán almacenarse cifradas usando el algoritmo de cifrado SHA-256.

RQNF10. Se deben comprobar en los formularios que estén rellenos todos los campos con carácter obligatorio.

RQNF11. Se deben comprobar la validez de los distintos campos de los formularios.

Mantenimiento

RQNF12. La aplicación se programará usando las guías de estilo de los diferentes lenguajes usados.

RQNF13. Las variables, funciones y clases tendrán nombres descriptivos.

RQNF14. El código fuente tendrá la indentación dictada por *C# Coding Conventions* o por *C# Programming Guide* que ofrece Microsoft.

RQNF15. La aplicación será actualizable, es decir, se modificará cada cierto tiempo, según cambien las necesidades del proceso de fabricación y/o de la trazabilidad del producto.

4. DISEÑO DEL SISTEMA

Una vez presentado el análisis del sistema en el apartado anterior se expondrá el diseño de la aplicación.

4.1. Perspectiva global

El prototipo software desarrollado se puede dividir en 2 partes que se describen a continuación:

- BiomasaEUP: Aplicación WPF (*Windows Presentation Foundation*) de código libre para escritorio con la que el usuario gestionará la recepción de las materias primas, el proceso, la elaboración de un producto terminado y su venta, y con la que obtendrá la trazabilidad de un producto según un código asignado.
- Base de datos: Almacenará toda la información que el prototipo necesita para poder tratar los datos.

Dado que es un prototipo para ser usado a menudo, el usuario no necesita tener conocimientos sobre informática para poder utilizar la aplicación, pero sí es necesario que sepa cómo funciona el proceso de fabricación del producto maderero, ya que, de no ser así, no podrá entender la trazabilidad (ver Anexo G “Manual de Usuario”).

De esta manera, la funcionalidad de los botones y demás elementos de la interfaz es bastante simple, pudiendo verse el efecto de la pulsación al momento en la mayoría de los casos. A pesar de que la aplicación tiene una gran cantidad de opciones, es sencilla de utilizar, además de cumplir con los patrones de usabilidad (toda la aplicación tiene el mismo estilo, las pestañas son similares, etc.)

4.2. Arquitectura Software

Este punto se centrará en la definición de componentes y en la interacción de los mismos. La aplicación cuenta con tres partes bien diferenciadas:

- La interfaz de interacción con los usuarios.
- Los datos almacenados en la base de datos.
- La lógica de unión entre las dos partes.

Con esta definición se deduce que uno de los patrones de arquitectura software más adecuados a emplear es el conocido y exitoso *Model View Controller* (MVC). Pero dada la tecnología empleada se usará un patrón más específico que se amolda aún mejor [16], el *Model View ViewModel* (también conocido como MVVM). De esta manera, cuando se amplíen los módulos y la funcionalidad, sólo se verán afectadas las partes que se quieran modificar y las demás podrán funcionar igual que antes sin hacer ningún cambio sustancial.

4.2.1. Microsoft .NET

El *framework* .NET es necesario para poder ejecutar la aplicación, así como para poder desarrollarla, ya que es indispensable cuando se programa en el lenguaje C#.

Es parte de la empresa multinacional Microsoft Corporation. Surgió principalmente como respuesta al gran auge de la plataforma Java, perteneciente a Oracle Corporation, usada en el creciente mercado de los negocios en entornos Web y también en los distintos *frameworks* basados en PHP [17].

Esta plataforma hace un gran énfasis en la transparencia de redes, así como en la independencia del hardware. Además, permite un desarrollo rápido y económico de aplicaciones (llamadas soluciones) a la vez que de forma robusta y segura.

Puede instalarse en los Sistemas Operativos de la familia Windows tanto para escritorio (Windows XP, Windows 7, Windows 8, Windows 8.1, Windows 10), como para servidores (Windows Server 2003, Windows Server 2008) así como para Windows Mobile (aunque en versión reducida).

Gracias al Proyecto Mono, que pretende implementar la plataforma .NET con código abierto, se pueden ejecutar aplicaciones .NET en Sistemas Operativos basados en GNU/Linux.

Actualmente la plataforma soporta los lenguajes C#, F# y VB.NET.

4.2.2. C#

La aplicación está desarrollada usando el lenguaje de programación C#. Es uno de los lenguajes más usados actualmente, es por ello una de las principales razones de la elección de este lenguaje para el desarrollo del prototipo. También se ha escogido este lenguaje porque en este TFG se pretendían aprender nuevas tecnologías.

C# es un lenguaje de programación, de propósito general, orientado a objetos desarrollado y estandarizado por Microsoft [18]. Es parte de la plataforma .NET.

La sintaxis deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes. La versión más reciente es C# 7, la cual ha sido lanzada en 2017 junto con Visual Studio 2017.

Hay que destacar el componente LINQ (Lenguaje de Consultas Integrado, *Language Integrated Query*) el cual agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET [19]. En la Figura 7 se muestra un ejemplo de su uso, filtrándose en un *array* de enteros, los enteros mayores de 80 e imprimiéndose por la consola [20].

```
class EjemploLINQ
{
    static void Main()
    {
        // Se especifica el origen de los datos.
        int[] puntuaciones = new int[] { 97, 92, 81, 60 };

        // Se define la expresión de la consulta.
        IEnumerable<int> puntuacionesConsulta =
            from puntuacion in puntuaciones
            where puntuacion > 80
            select puntuacion;
    }
}
```

```

        // Se ejecuta la consulta.
        foreach (int p in puntuacionesConsulta)
        {
            Console.Write(p + " ");
        }
    }
}
// Salida: 97 92 81

```

FIGURA 7. Ejemplo de consulta en LINQ.

4.2.3. XAML

Cuando se desarrollan aplicaciones en WPF, el diseño y funcionamiento de la interfaz se realiza utilizando el lenguaje de marcado XAML.

XAML (Lenguaje Extensible de Formato para Aplicaciones - *eXtensible Application Markup Language*) es un lenguaje declarativo basado en XML, desarrollado por Microsoft [21]. Es usado principalmente para desarrollar la interfaz en aplicaciones WPF, Silverlight (similar a Adobe Flash) y en UWP (Aplicaciones para la Plataforma Universal de Windows - *Universal Windows Platform apps*) [22].

Al compilarse un fichero XAML se genera un fichero BAML (Lenguaje de Marcado para Aplicaciones Binario - *Binary Application Markup Language*) el cual puede ser insertado como un recurso dentro de un ensamblado del *framework* .NET.

4.2.4. MVVM

Modelo-Vista-VistaModelo (*Model-View-ViewModel*, MVVM) es el patrón de diseño utilizado cuando se implementan aplicaciones WPF es por ello que BiomasaEUPT lo usará.

Los tres componentes principales del patrón MVVM [23] son:

- El Modelo (*Model*), es el encargado de representar la lógica de negocio, el modelo y la validación de los datos. Dentro del modelo están incluidas las clases POCO (*Plain Old CLR Objects*) [24] o los objetos de transferencias de datos entre procesos (*Data Transfer Objects*, DTO) [25].
- La Vista (*View*), es la responsable de definir la estructura, diseño y apariencia de lo que el usuario va a ver en pantalla, es decir, de la parte visual de la aplicación. No se ocupa en ningún momento del manejo de los datos. Lo ideal sería que la vista esté definida únicamente con XAML, sin código por debajo (*code-behind*).
- La VistaModelo (*ViewModel*), es la que actúa de intermediario entre la vista y el modelo. Normalmente, el *ViewModel* interactúa con el modelo invocando sus métodos. Este obtiene los datos del modelo y los hace visibles a través de la vista.

La Figura 8 muestra las relaciones entre los tres componentes. La Vista XAML asocia comandos (*Binding Commands*) que están definidos en la VistaModelo. A su vez, la VistaModelo notifica a la Vista mediante eventos o notificación de propiedades (*Property Change*

Notification). Por otra parte, el Modelo es actualizado por la VistaModelo y al actualizarse el Modelo se envían de nuevo notificaciones vía eventos.

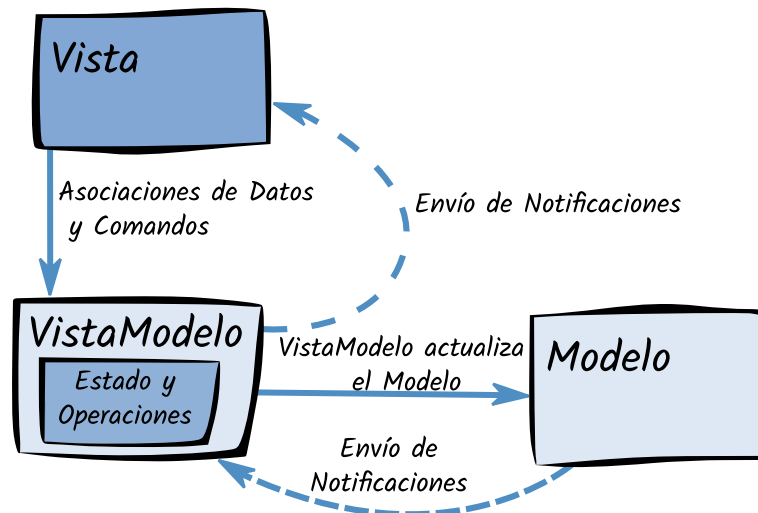


FIGURA 8. Relación entre los componentes del patrón MVVM.

4.3. Estructura del modelo de datos

Una de las partes más importantes de este Trabajo Fin de Grado, por no decir la más importante, es el adecuado almacenamiento de los datos para que su posterior recuperación sea lo más eficiente posible. Es por eso que, estas líneas tratan de abordar las razones por las que se ha elegido SQL Server como Sistema Gestor de Bases de Datos (SGBD), con sus ventajas e inconvenientes.

Además, se explica de manera detallada como se ha configurado SQL Server en una máquina virtual remota, como se ha creado la base de datos (BD) con Entity Framework, así como el posterior insertado de datos de prueba mediante migraciones (*migrations*).

4.3.1. Elección de SQL Server como SGBD

Microsoft SQL Server está basado en una arquitectura cliente/servidor, donde la información y datos se encuentran en el servidor y los clientes pueden acceder a esa información sin tener que guardarla de forma local [26].

Maneja de forma sencilla la seguridad en cuanto al acceso a la información, posibilitando contener múltiples bases de datos para diferentes usuarios. Esto permite el acceso a dichas bases de datos sólo a aquellos usuarios que tengan los correspondientes permisos.

Entre otras, éstas son sus mejores características:

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite administrar información de otros servidores de datos.

- Administra los datos para ambientes OLTP (*OnLine Transactions Processing*) y OLAP (*OnLine Analytical Processing*).
- Permite utilizar T-SQL (*Transact Structured Query Language*), XML (*eXtensible Markup Language*), MDX (*MultiDimensional Expressions*) o SQL-DMO (*Structured Query Language-Distributed Management Objects*) para solicitudes de procesamientos.

4.3.2. Configuración de SQL Server en MV remota

La base de datos del TFG se encuentra alojada en una Máquina Virtual (MV) de la Universidad de Zaragoza. De esta manera se puede acceder a la BD desde cualquier lugar, lo que hace que la aplicación se pueda utilizar en cualquier sitio.

Debido a que el servidor que aloja la MV cuenta con pocos recursos hardware el retardo entre la aplicación y la BD es lo suficientemente grande como para apreciarse (un par de segundos para consultas pequeñas). En una puesta en marcha real de la aplicación, habría que usar un servidor dedicado, mejorar el hardware del servidor o los recursos de la MV.

El primer paso a realizar es habilitar la conexión TCP en SQL Server para que permita operaciones desde conexiones remotas. Posteriormente, habrá que configurar el Firewall de Windows para que se permita el tráfico por dicha conexión TCP.

A continuación, se muestran los pasos que son necesarios realizar.

Habilitar la conexión TCP en SQL Server

1. En el menú de **Inicio**, se escribe **SQL Server Configuración Manager** y, a continuación, se pulsa la tecla **Intro**.
2. Se hace clic en el árbol de la parte izquierda de la pantalla, denominado **Administrador de configuración de SQL Server (Local)** y, se despliega la rama de **Configuración de red de SQL Server**.
3. Una vez efectuado este paso, es necesario hacer clic sobre la rama **Protocolos de SQLEXPRESS**, viéndose el contenido de la Figura 9.
4. A continuación, se hace clic con el botón derecho del ratón sobre la opción **TCP/IP** y se hace clic en **Habilitar**.
5. Para continuar, es necesario reiniciar el proceso en ejecución de SQL Server. Para ello se hace clic en la rama **Servicios de SQL Server** y en el proceso **SQL Server(SQLEXPRESS)**, que está en **Ejecución**, se hace clic con el botón derecho, pulsando **Reiniciar** (ver Figura 10).
6. A continuación, hay que añadir un puerto específico para la conexión. Esto es necesario para el siguiente paso, puesto que hay que añadir una regla en el Firewall que permita la conexión, tal y como se verá en el siguiente apartado. Para ello, se despliega la rama de **Configuración de red de SQL Server**, al igual que en los pasos 2 y 3.
7. Se hace clic en el botón derecho sobre la hoja de **TCP/IP**, pero ahora se hace clic en **Propiedades**.

8. Seleccionando la pestaña **Direcciones IP**, se elimina el puerto del campo **TCP Dynamic Ports** y se añade en el campo **TCP Port** el puerto **49170** (ver Figura 11).
9. Se hace clic en **Aceptar** y se repite el paso 5.

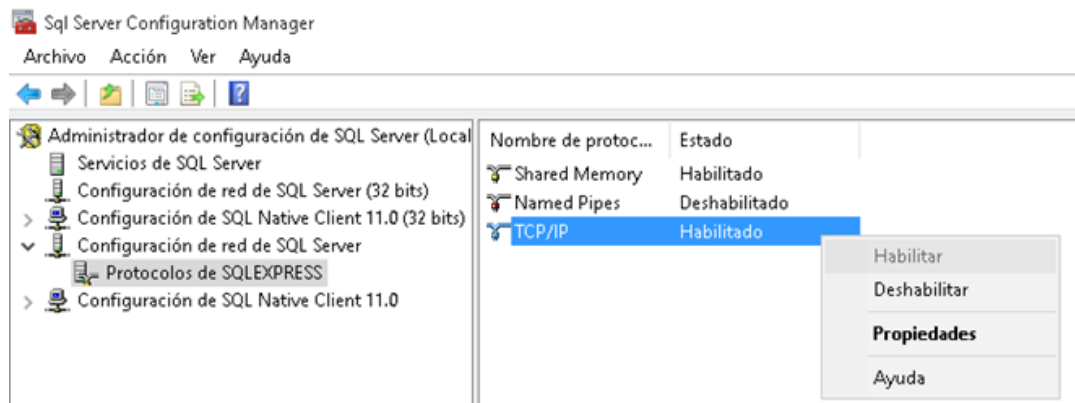


FIGURA 9. *Habilitación del protocolo TCP/IP para conexión remota con SQL Server.*

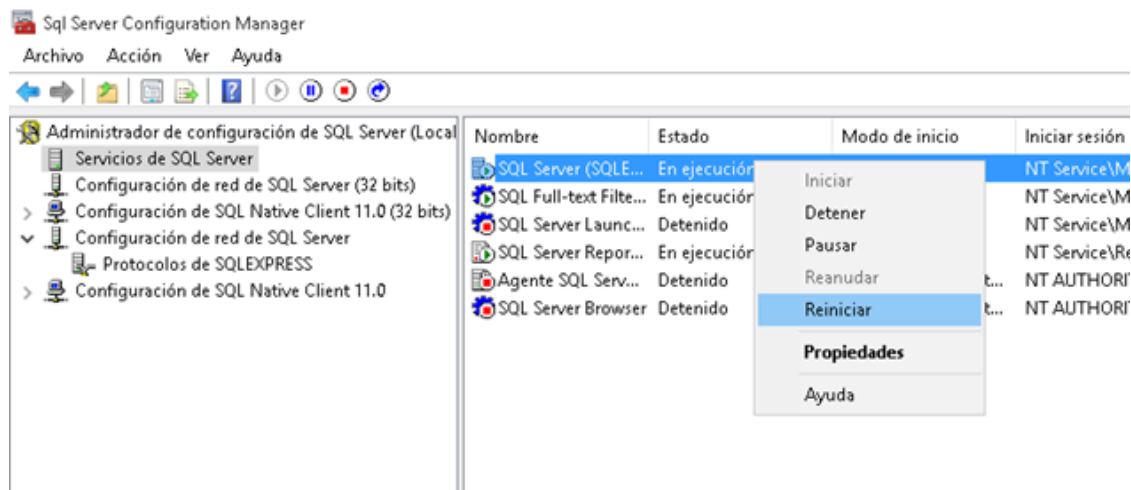


FIGURA 10. *Reinicio del proceso de SQL Server.*

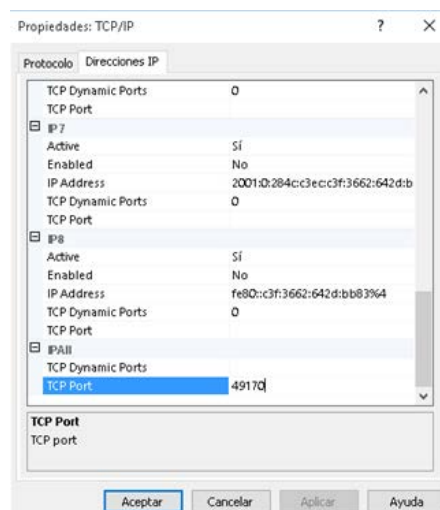


FIGURA 11. *Habilitación del puerto TCP/IP para la conexión remota.*

Abrir un puerto en el Firewall de Windows para el acceso TCP.

Los siguientes pasos son para Windows 10 de tal forma que puede variar ligeramente la explicación en otros Sistemas Operativos.

1. En el menú **Inicio**, se hace clic en **Ejecutar**, escribiendo **WF.msc** y, a continuación, se hace clic en **Aceptar**.
2. En la opción **Firewall de Windows con seguridad avanzada** del panel izquierdo, se hace clic con el botón derecho en **Reglas de entrada** y, luego, clic en **Nueva regla** en el panel de acciones.
3. En el cuadro de diálogo **Tipo de regla**, se selecciona **Puerto** y, a continuación, se hace clic en **Siguiente**.
4. En el cuadro de diálogo **Protocolo y puertos**, se selecciona **TCP**. A continuación, se selecciona **Puertos locales específicos** y se escribe el número de puerto de la instancia del Motor de base de datos, por ejemplo, **1433** para la instancia predeterminada. En este caso, se añade el puerto **49170**. A continuación, clic en **Siguiente**.
5. En el cuadro de diálogo **Acción**, se selecciona **Permitir la conexión** y, a continuación, se hace clic en **Siguiente**.
6. En el cuadro de diálogo **Perfil**, se seleccionan los perfiles que describan el entorno de conexión del equipo cuando se desee conectarse al Motor de base de datos y, a continuación, se hace clic en **Siguiente**.
7. En el cuadro de diálogo **Nombre**, se escribe un nombre y una descripción para esta regla. Por último, clic en **Finalizar**.

4.4. Entity Framework

Para la creación de la base de datos y para la comunicación entre la aplicación y la BD se ha usado Entity Framework (EF).

Es un *framework* de código abierto ORM (Mapeo Objeto Relacional, *Object-Relational Mapping*) que inicialmente era parte de la plataforma .NET concretamente de ADO.NET [27].

ADO.NET es un conjunto de componentes software para poder acceder a datos y a servidores de datos. Es usado comúnmente para acceder a datos almacenados en Sistemas Gestores de Bases de Datos Relacionales, aunque puede usarse para acceder a datos en fuentes no relacionales [28].

Provee un conjunto de API (Interfaz de Programación de Aplicaciones, *Application Programming Interface*) que permiten el desarrollo de aplicaciones orientadas a datos, en las cuales los desarrolladores pueden trabajar con un nivel alto de abstracción. Además, EF permite crear y mantener aplicaciones orientadas a datos con menos código que las aplicaciones tradicionales. A partir de la versión 6, EF está separado del *framework* .NET.

El diagrama de la Figura 12 muestra la arquitectura de EF para el acceso a datos.

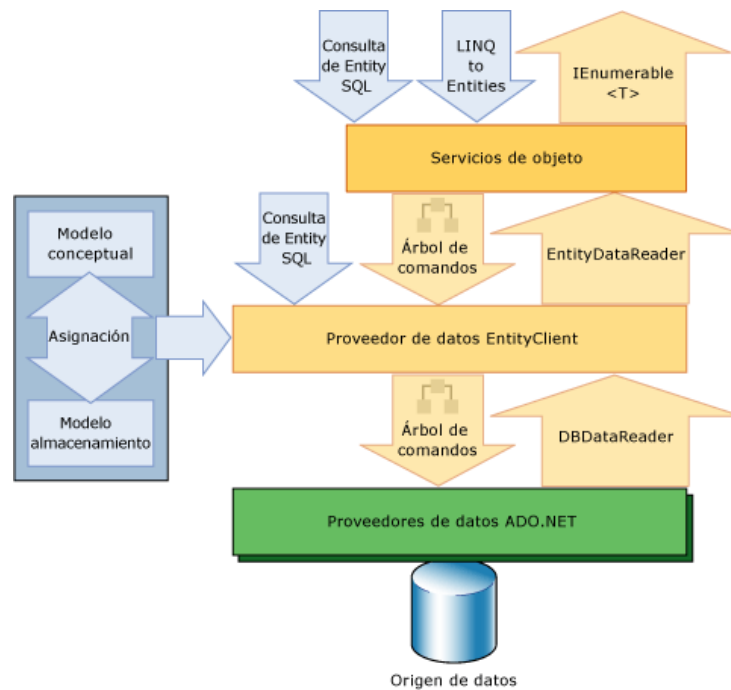


FIGURA 12. Diagrama de la arquitectura de Entity Framework.

Hay tres enfoques diferentes para poder hacer uso de EF [29]:

- Database First: Usado cuando la base de datos y las tablas están ya creadas/diseñadas o cuando el equipo de la BD es diferente al del proyecto. EF generará las distintas entidades POCO (*Plain Old CLR Object*) [30].
- Model First: Usado cuando ya se tiene el modelo y se desea construir la BD. No permite realizar cambios directamente en la BD. Si se desean modificar tablas, relaciones, etc. se deberán realizar los correspondientes cambios en el modelo y después Entity Framework los añadirá en la BD.
- Code First: Permite construir la BD desde el código. Usando este enfoque no haría falta (si no se quisiera) modificar la BD manualmente ya que se podrían realizar dichos cambios desde el código. Es necesario crear las distintas entidades POCO como modelo de datos. Es muy popular entre los programadores MVC.

Como en este proyecto había que crear tanto la aplicación como la base de datos desde cero se ha optado por utilizar el enfoque *Code First* permitiendo, de esta manera, construir conjuntamente la BD e ir realizando los cambios oportunos durante todo el proceso de desarrollo.

4.4.1. Creación de la Base de Datos

El primer paso para poder crear la BD es instalar Entity Framework usando el administrador de paquetes NuGet de Visual Studio. Para ello se hace clic derecho en uno de los proyectos que se tenga abierto en la solución y se selecciona la opción "Administrar paquetes NuGet...". En la nueva pestaña que se ha abierto se busca Entity Framework y se procede a su

instalación haciendo clic en el botón “Instalar”. Otra opción es instalarlo usando la “Consola de Administración de paquetes” insertando el siguiente comando:

```
PM> Install-Package EntityFramework
```

Una vez instalado EF, hay que crear el contexto de la base de datos para así poder interactuar con ella. Para ello hay que añadir al proyecto un nuevo elemento de tipo “ADO.NET Entity Data Model”, y escoger la opción “Modelo vacío de Code First” tal como se indica en la Figura 13.

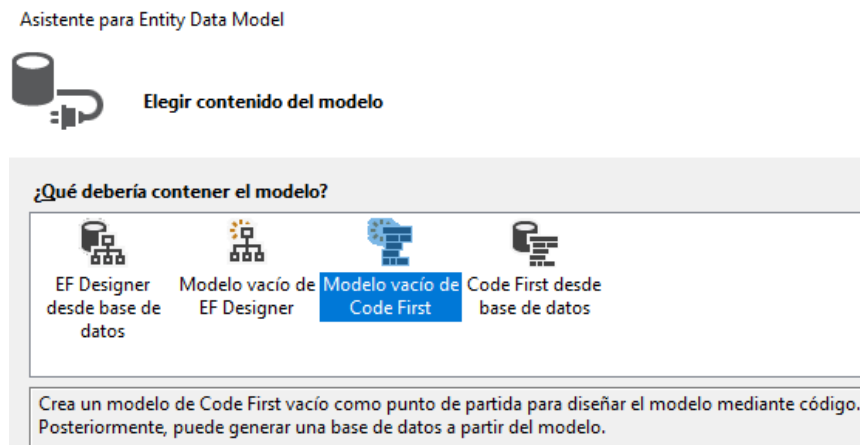


FIGURA 13. Creación del contexto de la BD con Entity Framework.

En el proyecto se habrá creado la clase `BiomasaEUPTContext` en la que hay que ir añadiendo cada una de las clases POCO correspondientes a cada una de a tablas de la BD (ver Figura 14). También, en el fichero de configuración `App.config` se habrá añadido la cadena de conexión (Figura 15) [31].

```
public class BiomasaEUPTContext : DbContext
{
    public BiomasaEUPTContext()
        : base("name=BiomasaEUPTContext")
    {
    }

    public DbSet<Usuario> Usuarios { get; set; }
    public DbSet<TipoUsuario> TiposUsuarios { get; set; }

    // ...
}
```

FIGURA 14. Contexto de Entity Framework.

```
<connectionStrings>
  <add name="BiomasaEUPTContext"
        connectionString="data source=(LocalDb)\MSSQLLocalDB;
                           initial catalog=BiomasaEUPT;
                           integrated security=True;Connect Timeout=10;
                           MultipleActiveResultSets=True;App=EntityFramework"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

FIGURA 15. Cadena de conexión a la BD ubicada en el fichero “App.config”.

Después, se procede a la creación de las distintas clases POCO que conformarán las distintas tablas de la base de datos. Hay que usar una serie de anotaciones para poder indicar, por ejemplo, cómo será el nombre de la tabla, el tipo de columna o las relaciones entre diferentes tablas.

A continuación, se describen brevemente las distintas anotaciones que se han usado en el proyecto:

- **Table:** Indica que la clase es una tabla, se le puede especificar el nombre que tendrá la tabla en la BD.
- **Key:** La propiedad es una clave primaria.
- **Required:** La propiedad es un campo obligatorio.
- **StringLength:** Tamaño máximo de caracteres que tendrá la propiedad. Se le puede indicar con el argumento **MinimumLength** el tamaño mínimo.
- **Index:** Crea un índice en la propiedad, útil si se desea, por ejemplo, restringir por valor único usando el atributo **IsUnique**.
- **DisplayName:** Nombre que será mostrado cuando se llame a la propiedad en las vistas, por ejemplo, en un formulario creado en XAML.
- **EmailAddress:** La propiedad será de tipo *email*. Entity Framework lo validará usando una expresión regular.
- **DatabaseGenerated:** Indica que el valor de la propiedad será generado por la Base de Datos. Esta propiedad es muy útil para ser usado junto a los *triggers*.
- **ForeignKey:** Usado para especificar que la propiedad es una clave ajena.
- **RegularExpression:** La propiedad debe cumplir la expresión regular que se haya especificado.

En la Figura 16 y la Figura 17 se muestran dos ejemplos de la estructura de las clases POCO correspondientes a las tablas “Usuarios” y “TiposUsuarios”.

```
[Table("Usuarios")]
public class Usuario
{
    public Usuario()
    {
        Baneado = false;
    }

    [Key]
    public int UsuarioId { get; set; }

    [Required]
    [StringLength(Constants.LONG_MAX_NOMBRE_USUARIO,
        MinimumLength = Constants.LONG_MIN_NOMBRE_USUARIO)]
    [Index(IsUnique = true)]
    [DisplayName("Nombre"), Display(Name = "Nombre")]
    public string Nombre { get; set; }
```

```

[Required]
[StringLength(Constants.LONG_MAX_HASH_CONTRASENA,
               MinimumLength = Constants.LONG_MIN_HASH_CONTRASENA)]
[DisplayName("Contraseña"), Display(Name = "Contraseña")]
public string Contraseña { get; set; }

[Required]
[StringLength(Constants.LONG_MAX_EMAIL,
               MinimumLength = Constants.LONG_MIN_EMAIL)]
[Index(IsUnique = true)]
[EmailAddress]
[DisplayName("Email"), Display(Name = "Email")]
public string Email { get; set; }

[DatabaseGenerated(DatabaseGeneratedOption.Computed)]
[DisplayName("Fecha alta"), Display(Name = "Fecha alta")]
public DateTime? FechaAlta { get; set; }

[DatabaseGenerated(DatabaseGeneratedOption.Computed)]
[DisplayName("Fecha baja"), Display(Name = "Fecha baja")]
public DateTime? FechaBaja { get; set; }

[DisplayName("Fecha contraseña"), Display(Name = "Fecha contraseña")]
public DateTime? FechaContraseña { get; set; }

[DisplayName("Baneado"), Display(Name = "Baneado")]
public bool? Baneado { get; set; }

public int TipoId { get; set; }

[ForeignKey("TipoId")]
public virtual TipoUsuario TipoUsuario { get; set; }
}

```

FIGURA 16. Clase POCO de la tabla "Usuarios".

```

[Table("TiposUsuarios")]
public class TipoUsuario
{
    [Key]
    public int TipoUsuarioId { get; set; }

    [Required]
    [StringLength(Constants.LONG_MAX_NOMBRE,
                   MinimumLength = Constants.LONG_MIN_NOMBRE)]
    [Index(IsUnique = true)]
    [DisplayName("Nombre"), Display(Name = "Nombre")]
    public string Nombre { get; set; }

    [Required]
    [StringLength(Constants.LONG_MAX_DESCRIPCION,
                   MinimumLength = Constants.LONG_MIN_DESCRIPCION)]
    [DisplayName("Descripción"), Display(Name = "Descripción")]
    public string Descripcion { get; set; }

    public virtual List<Permiso> Permisos { get; set; }

    public virtual List<Usuario> Usuarios { get; set; }
}

```

FIGURA 17. Clase POCO de la tabla "TiposUsuarios".

A continuación, se representa el diagrama completo de la base de datos. Para una mejor representación, el diagrama se ha dividido en dos partes, relacionándose mediante las tablas de recepción y materia prima (ver Figura 18). En las dos páginas siguientes se representa el diagrama (Figura 19 y Figura 20).

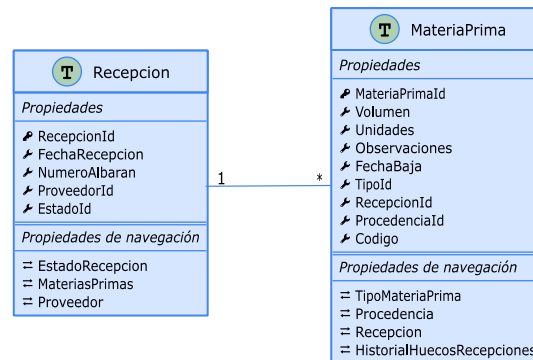


FIGURA 18. Relación entre las tablas "Recepcion" y "MateriaPrima".

Para diferenciar las tablas que corresponden a cada uno de los dos miembros que desarrollamos el proyecto, estas se representan usando dos colores distintos. Las tablas de color blanco corresponden al TFG de Javier Bielsa Hernández mientras que las tablas de color azul a este TFG (Eduardo Fonte Polo).

En el Anexo E "Tablas de la Base de Datos" están representadas con más detalle cada una de las columnas que tiene cada tabla.

Tanto los diagramas de casos de uso como los diagramas de la base de datos están realizados usando la herramienta *Open Source* PlantUML, que permite generar diagramas a partir de una descripción textual.

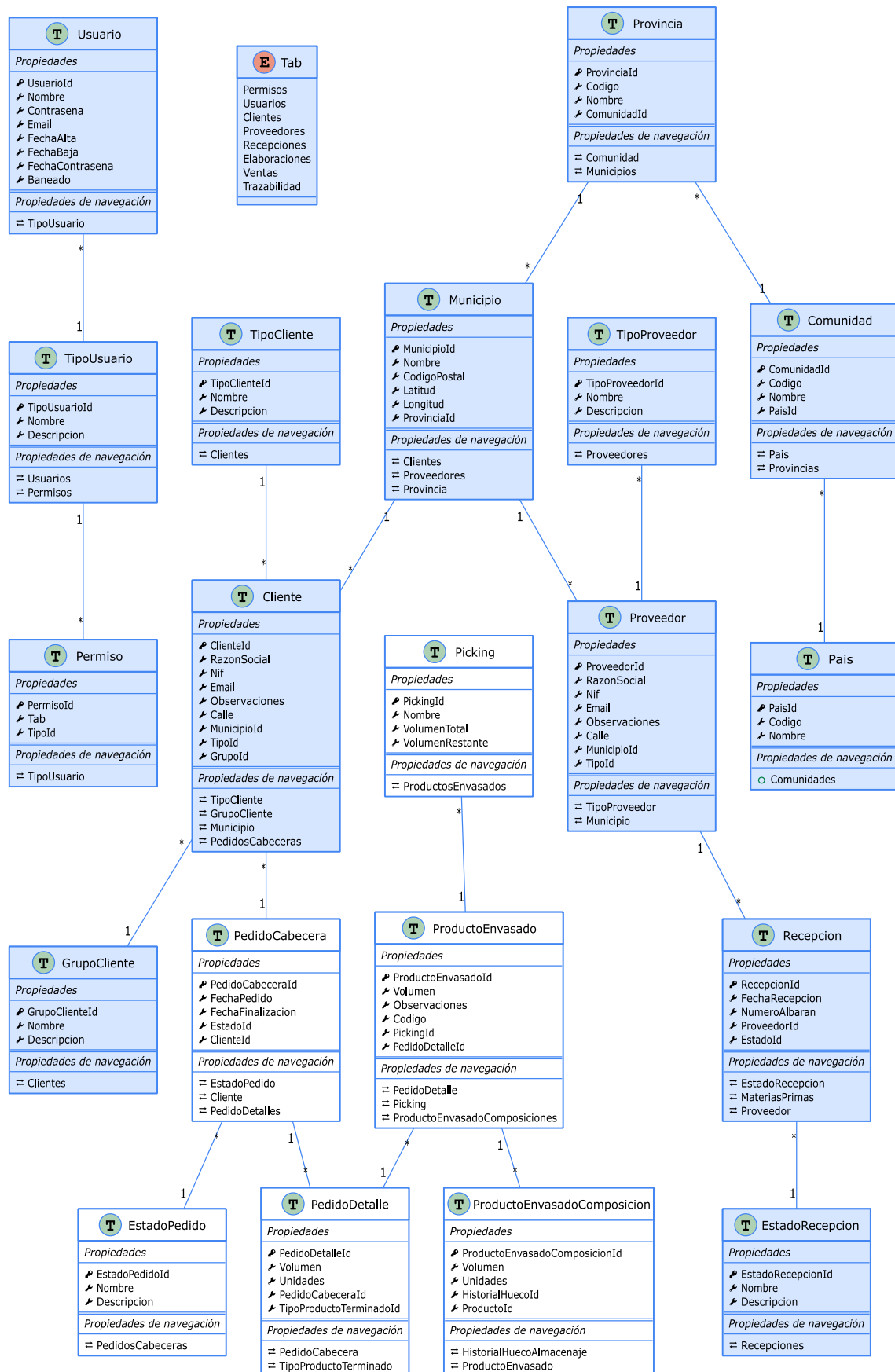


FIGURA 19. Diagrama de la BD. 1ª parte.

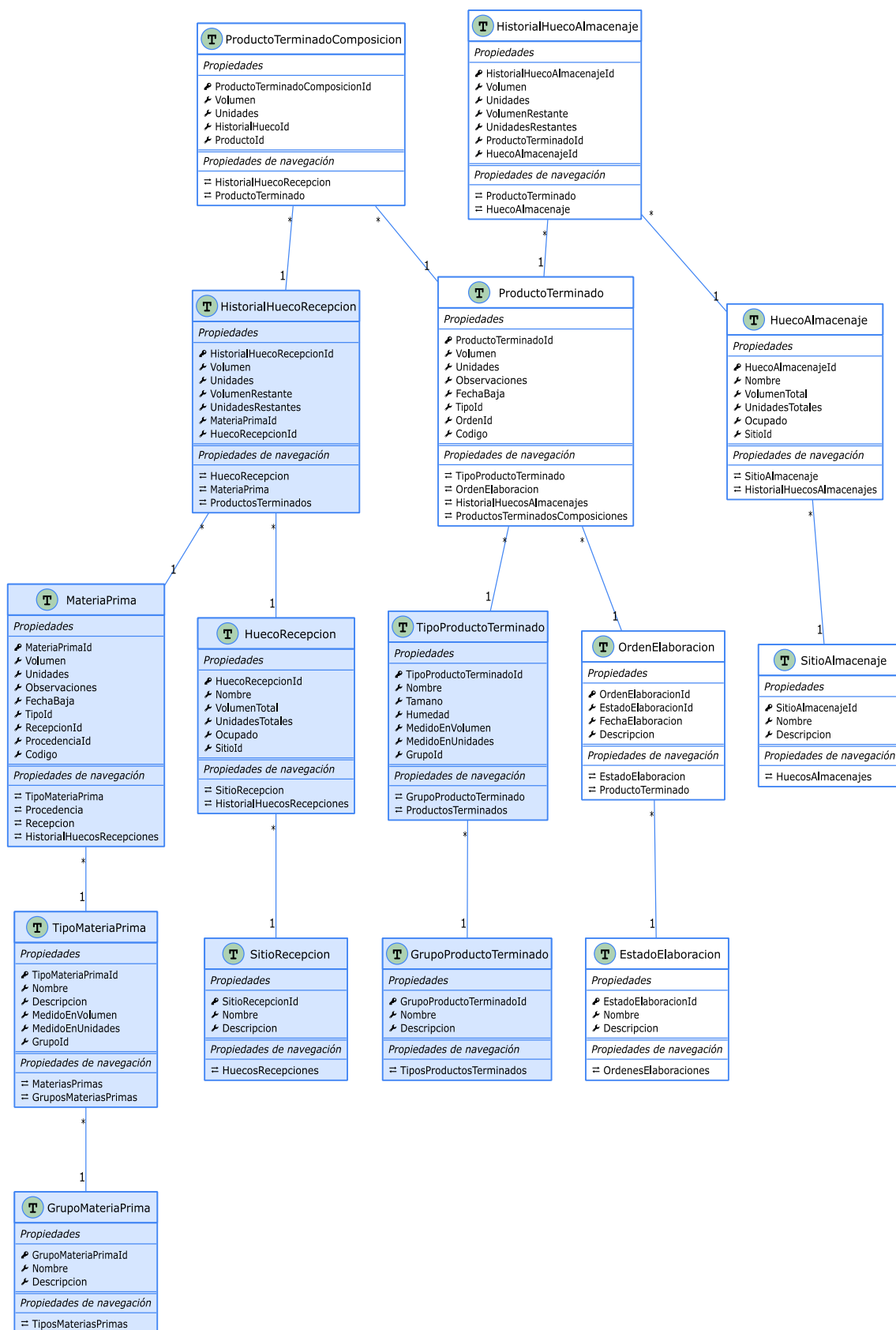


FIGURA 20. Diagrama de la BD. 2ª parte.

4.4.2. Migraciones

Al haber creado la base de datos con el enfoque *Code First* de Entity Framework se pueden hacer uso de migraciones para añadir datos en las distintas tablas de la base de datos. Además, se permite generar la BD e ir actualizando los cambios en ésta según se vayan modificando las clases POCO. Si se hubieran escogido alguno de los otros dos enfoques esto no sería posible.

Para comenzar a hacer uso de las migraciones el primer paso es generar la carpeta con la correspondiente clase “Configuration”, para ello se debe ejecutar el siguiente comando desde la “Consola del Administrador de paquetes” [32] [33] [34].

```
PM> Enable-Migrations
```

```
Checking if the context targets an existing database...
```

```
Code First Migrations enabled for project BiomasaEUP.T.
```

Tras ejecutar dicho comando, se habrá creado el árbol de directorios mostrado en la Figura 21. Ahora hay que generar la base de datos tal y como se detalla en el siguiente subapartado.

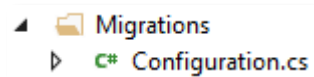


FIGURA 21. Árbol de directorios generado por el comando "Enable-Migrations".

Creación y actualización de la Base de Datos

Como la base de datos no está ni siquiera creada habría que generarla mediante una migración usando el siguiente comando:

```
PM> Add-Migration CreaciónBD
```

```
Scaffolding migration 'CreaciónBD'.
```

Se puede observar que el comando “Add-Migration” añade una nueva migración con el nombre especificado (parámetro opcional), en este caso “CreaciónBD”, el cual crea un fichero en la carpeta “Migrations” con el siguiente nombre Fecha_Nombre.cs (Ej.: 201704100816494_CreaciónBD).

Para subir los cambios a la Base de Datos hay que ejecutar la migración con el comando Update-Database.

```
PM> Update-Database
```

```
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
```

```
Applying explicit migrations: [201704100816494_CreaciónBD].
```

```
Applying explicit migration: 201704100816494_CreaciónBD.
```

```
Running Seed method.
```

En la Figura 22 puede verse como se han generado las tablas `TiposUsuarios` y `Usuarios` además de la tabla especial `__MigrationHistory`, la cual contiene la información de cada migración con el fin de poder detectar los cambios que se hicieron en dichas migraciones y así poder volver al estado de la base de datos en el momento de la migración (usando el comando `Update-Database -TargetMigration:<nombre migración>`).

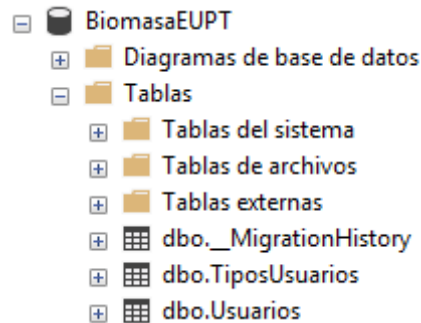


FIGURA 22. *Tablas Usuarios y TiposUsuarios generadas en la BD con Entity Framework.*

Cada vez que se realiza un cambio en las clases POCO, ya sea añadiendo una nueva clase (tabla) o modificando las ya existentes, es necesario ejecutar los comandos `Add-Migration` y posteriormente `Update-Database`. Esto puede ser poco eficiente al principio del desarrollo de la base de datos ya que se harán cambios constantemente en ella. Para simplificar este proceso se puede activar la propiedad `AutomaticMigrationsEnabled=true` en el constructor de la clase “`Configuration`” [35].

Al haber activado las migraciones automáticas con tal de ejecutar el comando `Update-Database` se creará la clase con la migración (ya no estará ubicada en la carpeta “`Migrations`”; EF la borra al final del proceso de actualización) y se actualizará la base de datos.

PM> Update-Database

Specify the '-Verbose' flag to view the SQL statements being applied to the target database.

No pending explicit migrations.

Applying automatic migration: 201704100986494_AutomaticMigration.

Running Seed method.

Entity Framework mostrará un aviso y no dejará ejecutar el comando `Update-Database` si en las tablas donde se van a modificar hay datos, por lo que para que se permita ejecutar dicho comando habrá que añadir el parámetro `-Force`.

Añadir datos a través de migraciones

La clase “`Configuration`” encargada de las migraciones cuenta con el método `Seed` el cual permite insertar datos en las tablas de la BD. En la siguiente figura se muestra un ejemplo de uso añadiendo un tipo de usuario con dos permisos.

```

internal sealed class Configuration: DbMigrationsConfiguration<BiomasaEUPContext>
{
    public Configuration()
    {
        AutomaticMigrationsEnabled = true;
    }

    protected override void Seed(BiomasaEUPContext context)
    {
        context.TiposUsuarios.AddOrUpdate(
            tu => tu.Nombre,
            new TipoUsuario()
            {
                Nombre = "Administrador",
                Descripcion = "Usuario con máximos privilegios"
            });
        context.SaveChanges();

        context.Permisos.AddOrUpdate(
            p => p.PermisoId,
            new Permiso()
            {
                PermisoId = 1,
                Tab = Tab.Usuarios,
                TipoId = context.TiposUsuarios.Local
                    .Single(u => u.Nombre == "Administrador")
                    .TipoUsuarioId
            },
            new Permiso()
            {
                PermisoId = 2,
                Tab = Tab.Clientes,
                TipoId = context.TiposUsuarios.Local
                    .Single(u => u.Nombre == "Administrador")
                    .TipoUsuarioId
            }
            // ...
        );
        context.SaveChanges();
    }
}

```

FIGURA 23. Inserción de datos en la BD usando migraciones.

Para simplificar un poco el proceso de creación de los datos que serán insertados en la BD, se ha creado dentro a la carpeta “Migrations” una serie de ficheros CSV los cuales contienen los datos de cada tabla, posteriormente se ha implementado una clase que recopila los datos de cada CSV, analizándolos (*parser*), e insertándolos en el Seed. En la Figura 24 se observan parte de los ficheros CSV y en la Figura 25 el contenido del fichero Usuarios.csv.

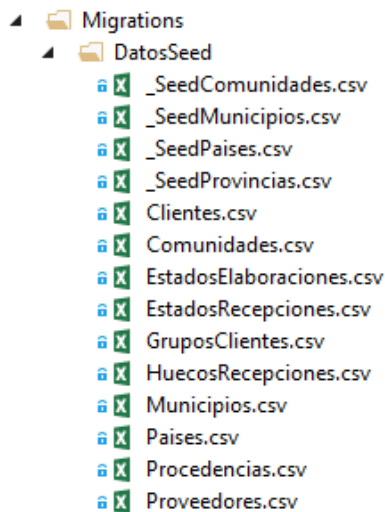


FIGURA 24. Parte de los ficheros CSV usados en las migraciones.

UsuarioId	Nombre	Email	TipoId
1	superadmin	superadmin@biomasaeupt.es	1
2	admin	admin@biomasaeupt.es	2
3	efonte	efonte@biomasaeupt.es	3
4	jbielsa	jbielsa@biomasaeupt.es	4

FIGURA 25. Contenido del fichero CSV para la tabla "Usuarios".

También hay que tener en cuenta que cuando la aplicación es compilada en modo Release (Visual Studio permite compilar en modo Debug, ofreciendo una compilación sin optimizar entre otras características; y en modo Release, cuando el programa va a ser distribuido al público) el programa no debería incluir tanto la librería dependiente del análisis de los ficheros CSV (CsvHelper.dll) como los propios ficheros CSV o las clases de la carpeta "Migrations". Al incluirse estos ficheros aumenta el tamaño del programa además de disminuir la seguridad (por incluirse las migraciones).

Para que esto no ocurra hay que establecer en el fichero del proyecto (BiomasaEUP.T.csproj) una serie de condiciones para que, dependiendo de cómo es compilado el proyecto, se incluyan o no estos ficheros CSV, clases y librerías. Una parte del código puede verse en la Figura 26.

```
<ItemGroup Condition=" '$(Configuration)' == 'Debug' ">
  <Reference Include="CsvHelper, Version=2.0.0.0, Culture=neutral,
    PublicKeyToken=8c4959082be5c823,
    processorArchitecture=MSIL">
    <HintPath>
      ..\packages\CsvHelper.2.16.3.0\lib\net45\CsvHelper.dll
    </HintPath>
  </Reference>
  <Compile Include="Migrations\Configuration.cs" />
  <Compile Include="Migrations\SeedCodigosPostales.cs" />
  <Compile Include="Migrations\SeedTablas.cs" />
  <Compile Include="Modelos\BiomasaEUP.TContextInitializer.cs" />
  <EmbeddedResource Include="Migrations\DatosSeed\Clientes.csv" />
  <EmbeddedResource Include="Migrations\DatosSeed\Usuarios.csv" />
  <EmbeddedResource Include="Migrations\DatosSeed\TiposClientes.csv" />
  <!-- ... -->
  <EmbeddedResource Include="Migrations\DatosSeed\SitiosAlmacenajes.csv" />
  <EmbeddedResource Include="Migrations\DatosSeed\Procedencias.csv" />
</ItemGroup>
```

FIGURA 26. Parte de código ubicado en el fichero BiomasaEUP.T.csproj.

En el Anexo C "Triggers en Entity Framework" así como en el Anexo D "Auditoría en Entity Framework" se detallan partes del código para crear *triggers* en la base de datos y para mantener una auditoría de la BD respectivamente.

4.4.3. Generación de la trazabilidad usando PDF

Para la realización de los códigos de barras se ha utilizado, al igual que para la generación de los PDF, la librería iText, concretamente iText7.

iText es una adaptación de la biblioteca de Java de código abierto iText para la generación de PDF escrita totalmente en C# para la plataforma .NET. Antes se llamaba iTextSharp pero desde la versión 5 paso a llamarse igual que la versión para Java [36].

Es una biblioteca *Open Source*, la cual es usada para crear y manipular archivos PDF. Además de permitir generar PDF desde ficheros HTML.

Para generar un código de barras, lo único que hay que hacer es crear una nueva instancia de la clase Barcode128 y pasarle mediante distintas funciones (SetCodeType(), SetCode(), ...) el tipo de código de barras y el texto a partir del cual lo generará [37]. En la Figura 27 puede verse un fragmento de código para poder generar el código de barras de la materia prima con código "1000000001".

```
PdfWriter writer = new PdfWriter("C:\\Informe.pdf");
PdfDocument pdfDoc = new PdfDocument(writer);
Document doc = new Document(pdfDoc, PageSize.A4.Rotate());

var codigoBarras = new Barcode128(pdfDoc);
codigoBarras.SetCodeType(Barcode128.CODE128);
codigoBarras.SetCode("1000000001");
Image imagenCodigo = new Image(codigoBarras.CreateFormXObject(pdfDoc))
    .SetWidth(100);
doc.Add(imagenCodigo);
doc.Close();
```

FIGURA 27. Creación de un código de barras con iText7.

En la siguiente figura se puede observar un ejemplo del informe PDF que la aplicación genera, correspondiente a una materia prima.

PROVEEDOR					RECEPCIÓN			
RAZON SOCIAL	Maderas Pepe, S.L.				FECHA RECEPCIÓN	20/01/2017 13:20		
NIF	B-36381563				Nº DE ALBARÁN	A-010VB		
TIPO	Ecológico							
MATERIA PRIMA								
TIPO	Tronco Pino		CANTIDAD	44 ud.		 1000000004		
Recepción				Elaboración				
Sitio	Hueco	Capacidad Total	Cantidad Almacenada	Cantidad Utilizada	Producto Terminado	Cantidad Producida	Sitio	Hueco
Sitio C	C02	35 m³ / 45 ud.	44 ud.	20 ud.	Tablón pino	80 ud.	Sitio A	A01
								A05

FIGURA 28. Informe PDF de una materia prima.

4.5. Prototipos de la Interfaz de Usuario

Los prototipos de media fidelidad, diseñados para la interfaz de usuario, se han realizado usando la herramienta Balsamiq Mockups, esta permite ir seleccionando componentes desde un menú desplegable y otorgando a cada uno de ellos una serie de propiedades (color, tipo de fuente, ...). En la Figura 29 se muestra la interfaz de esta aplicación.

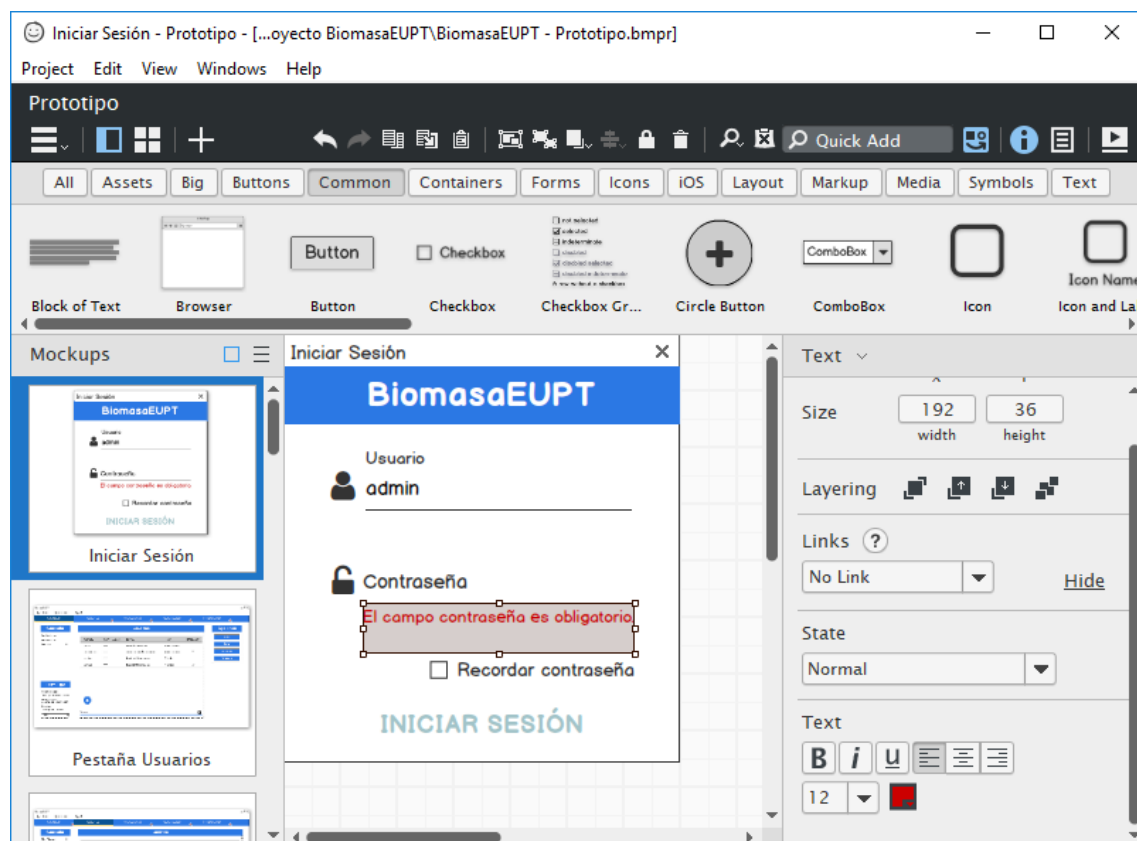


FIGURA 29. Creación de un formulario usando Balsamiq Mockups.

Balsamiq Mockups no cuenta por defecto con el tema Material Design de Google por lo que se han tenido que crear dicha apariencia cambiando la propiedad de los componentes.

En el Anexo E “Tablas de la Base de Datos” se representan cada uno de los prototipos de media fidelidad.

5. MEJORAS FUTURAS

El prototipo desarrollado cumple con todos los objetivos que se habían marcado para su realización. Aunque es posible añadir nuevas funcionalidades, así como mejoras a las ya existentes, algunas de las cuales se enumeran a continuación:

- Mejoras en la gestión de los permisos de cada tipo de usuario. Por ahora sólo se puede limitar la vista y gestión de una pestaña entera, pero sería mejor limitar partes concretas de cada una de ellas.
- Si la aplicación tiene éxito, se podría desarrollar para los sistemas operativos Android e iOS mediante la herramienta Xamarin, para desarrollar aplicaciones móviles multiplataforma, desde el código fuente C#. Además, permite utilizar Entity Framework [38].
- Sería útil añadir soporte multi-idioma con el fin de que el prototipo pueda ser usado en mercados externos al español.
- Utilización de los productos ya terminados como materias primas. Esto está pensado para reintroducir en el proceso de fabricación los sobrantes, excesos y/o productos que no hayan sido vendidos.
- Como se puede observar durante la ejecución del prototipo, este funciona muy lento debido a que se utiliza una máquina virtual en un servidor de la EUPT el cual aloja a otras varias máquinas virtuales. Sería bueno usar un servidor de pago para la base de datos con el fin de minimizar el tiempo de respuesta de cada solicitud entre la aplicación y la BD.

Dado que la segunda parte del prototipo, realizada por Javier Bielsa Hernández, aún está sin finalizar, las siguientes mejoras podrían ser implementadas por él:

- Al añadir un producto envasado, si no se disponen de suficientes productos terminados, se debería crear una nueva orden de elaboración con la que se fabriquen las cantidades necesarias de ese producto terminado.
- Detallar con precisión el proceso de fabricación en cada una de sus etapas.

6. CONCLUSIONES

El prototipo desarrollado en este Trabajo Fin de Grado, BiomasaEUP, ofrece los servicios de gestión de gran parte del proceso de fabricación en una empresa tipo de la industria maderera, y permite obtener la trazabilidad de las recepciones de materias primas.

Las funcionalidades, implementadas en este TFG, que posee la aplicación de escritorio BiomasaEUP son:

- Gestión de usuarios, con los correspondientes permisos.
- Gestión de clientes y proveedores.
- Gestión de recepciones de materias primas.
- Obtención de la trazabilidad de las recepciones mediante la generación de informes en PDF.
- Gestión de ajustes.
- Actualización automática del programa.

Las principales herramientas y lenguajes utilizados para el desarrollo del prototipo han sido:

- Entity Framework, para la comunicación entre la base de datos y la aplicación, así como para la generación de la BD.
- XAML, empleado en las interfaces de usuario.
- C#, lenguaje en el que está programado la aplicación.

Por último, cabe destacar que la aplicación final es totalmente funcional, de fácil implantación, solo se necesita un servidor con SQL Server para el almacenamiento de los datos y un Sistema Operativo Windows 7 o superior para la ejecución de la aplicación; y de fácil mantenimiento porque hace uso del patrón de diseño MVVM. También se ha intentado que sea lo más genérica posible con el fin de que pueda ser utilizada, no sólo en la industria maderera, sino en cualquier otra.

7. BIBLIOGRAFÍA

- [1] RAE, «Trazabilidad - Real Academia Española,» [En línea]. Available: <http://dle.rae.es/srv/search?m=30&w=trazabilidad>. [Último acceso: 30 Agosto 2017].
- [2] «Trazabilidad - AEC,» [En línea]. Available: <https://www.aec.es/web/guest/centro-conocimiento/trazabilidad>. [Último acceso: 30 Agosto 2017].
- [3] AEOC, «Trazabilidad - AEOC,» [En línea]. Available: <https://www.aecoc.es/servicios/implantacion/trazabilidad>. [Último acceso: 30 Agosto 2017].
- [4] J. Pereiro, «Producción y prestación del servicio en ISO 9001:2000 - Portal Calidad,» 17 Marzo 2017. [En línea]. Available: http://www.portalcalidad.com/articulos/58-produccion_y_prestacion_del_servicio_iso_9001_2000. [Último acceso: 30 Agosto 2017].
- [5] BOE, «Ley 17/2011, de 5 de julio, de seguridad alimentaria y nutrición.,» *BOE*, nº 160, pp. 71283-71319, 2011.
- [6] «Reglamentaciones, manuales y estándares de trazabilidad - TrazaMed,» [En línea]. Available: <http://www.trazamed.com/reglamentaciones-manuales-y-estandares-de-trazabilidad/>. [Último acceso: 30 Agosto 2017].
- [7] «SSCC-18 Barcodes - GS1-128 INFO,» [En línea]. Available: <http://www.gs1-128.info/sscc-18/>. [Último acceso: 30 Agosto 2017].
- [8] N. P. Molina Montoya, «¿Qué es el estado del arte?,» *Ciencia y Tecnología para la Salud Visual y Ocular*, nº 5, pp. 73-75, 2005.
- [9] «Sage Murano - BMS para pymes que mejora la gestión de recursos,» [En línea]. Available: <http://www.sage.es/software/erp/mediana-empresa/sage-murano>. [Último acceso: 30 Agosto 2017].
- [10] «iLEAN Tracer - Software de Trazabilidad,» [En línea]. Available: <http://www.ilean.net/trazabilidad-caracteristicas.html>. [Último acceso: 2017 Agosto 2017].
- [11] «Trazabilidad y Gestión de Lotes - Gestion5 SQL,» [En línea]. Available: <http://www.gestion5.com/modulos/alimentacion-y-distribucion/trazabilidad-y-gestion-de-lotes/1109755>. [Último acceso: 30 Agosto 2017].
- [12] «Trazatec - Software de Trazabilidad,» [En línea]. Available: <http://www.asintec.es/software-de-trazabilidad/article/14/17>. [Último acceso: 30 Agosto 2017].
- [13] «Características - SolinCloud,» [En línea]. Available: <https://www.solincloud.com/demo.php>. [Último acceso: 30 Agosto 2017].

- [14] «IOAlimentación - Software para el sector de la alimentación,» [En línea]. Available: <http://www.ioalimentacion.com/index.php/solucion-alimentaria.html>. [Último acceso: 30 Agosto 2017].
- [15] «Software ERP para trazabilidad y control de lotes - Alba IBS,» [En línea]. Available: <https://www.albaibs.es/programa-gestion-comercial-alba/software-gestion-de-trazabilidad/>. [Último acceso: 30 Agosto 2017].
- [16] «Diferencia entre MVC y MVVM - MSDN - Microsoft,» 30 Noviembre 2012. [En línea]. Available: <https://social.msdn.microsoft.com/Forums/es-ES/ff728fb8-177c-460d-bad5-25a9a1582afe/no-acabo-de-entender-muy-bien-la-diferencia-entre-mvc-y-mvvm?forum=windowsphone7>. [Último acceso: 30 Agosto 2017].
- [17] «.NET Framework - Wikipedia,» 9 Marzo 2017. [En línea]. Available: https://en.wikipedia.org/wiki/.NET_Framework. [Último acceso: 30 Agosto 2017].
- [18] «C Sharp (programming language) - Wikipedia,» [En línea]. Available: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)). [Último acceso: 30 Agosto 2017].
- [19] «Language-Integrated Query (LINQ) - Microsoft Docs,» 30 Noviembre 2016. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/csharp/linq/>. [Último acceso: 30 Agosto 2017].
- [20] «Language Integrated Query - Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Language_Integrated_Query. [Último acceso: 30 Agosto 2017].
- [21] «XAML - Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/XAML>. [Último acceso: 30 Agosto 2017].
- [22] «¿Qué es XAML? - MSDN - Microsoft,» [En línea]. Available: <https://msdn.microsoft.com/es-es/library/cc295302.aspx>. [Último acceso: 30 Agosto 2017].
- [23] «El Patrón Model - View - ViewModel (MVVM) - Cup Coffee,» [En línea]. Available: <http://cup-coffe.blogspot.com.es/2012/03/el-patron-model-view-viewmodel-mvvm.html>. [Último acceso: 30 Agosto 2017].
- [24] «What is POCO in Entity Framework? - Stack Overflow,» 27 Enero 2017. [En línea]. Available: <https://stackoverflow.com/questions/16075245/what-is-poco-in-entity-framework>. [Último acceso: 30 Agosto 2017].
- [25] «Data transfer object - Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Data_transfer_object. [Último acceso: 30 Agosto 2017].
- [26] «Microsoft SQL Server - Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Microsoft_SQL_Server. [Último acceso: 30 Agosto 2017].

- [27] «Entity Framework - Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Entity_Framework. [Último acceso: 30 Agosto 2017].
- [28] «ADO.NET - Wikipedia,» [En línea]. Available: <https://en.wikipedia.org/wiki/ADO.NET>. [Último acceso: 30 Agosto 2017].
- [29] «Code-first vs Model/Database-first - Stack Overflow,» 16 Noviembre 2016. [En línea]. Available: <https://stackoverflow.com/questions/5446316/code-first-vs-model-database-first>. [Último acceso: 30 Agosto 2017].
- [30] «Plain Old CLR Object - Wikipedia,» 6 Abril 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Plain_Old_CLR_Object. [Último acceso: 30 Agosto 2017].
- [31] I. Endres, «Programmatic Connection Strings in Entity Framework 6 - Cosairus,» 10 Marzo 2015. [En línea]. Available: <http://www.cosairus.com/Blog/2015/3/10/programmatic-connection-strings-in-entity-framework-6>. [Último acceso: 30 Agosto 2017].
- [32] «Entity Framework Tutorial,» [En línea]. Available: <http://www.entityframeworktutorial.net/>. [Último acceso: 30 Agosto 2017].
- [33] «Entity Framework Code First Migrations - MSDN - Microsoft,» 23 Octubre 2016. [En línea]. Available: [https://msdn.microsoft.com/en-us/library/jj591621\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/jj591621(v=vs.113).aspx). [Último acceso: 30 Agosto 2017].
- [34] D. Cañizares Corrales, «Relaciones y migraciones con el Entity Framework (ASP.NET MVC) - The Science of Code,» 29 Septiembre 2016. [En línea]. Available: <http://thescienceofcode.azurewebsites.net/Articles/Show/57ed8b7d79e39e54f0b64026>. [Último acceso: 30 Agosto 2017].
- [35] «Automated Migration - Entity Framework Tutorial,» [En línea]. Available: <http://www.entityframeworktutorial.net/code-first/automated-migration-in-code-first.aspx>. [Último acceso: 30 Agosto 2017].
- [36] «iTextSharp vs iText - Stack Overflow,» 14 Febrero 2017. [En línea]. Available: <https://stackoverflow.com/questions/42212245/itextsharp-vs-itext>. [Último acceso: 30 Agosto 2017].
- [37] «Examples hub - iText Developers,» [En línea]. Available: <http://developers.itextpdf.com/examples>. [Último acceso: 30 Agosto 2017].
- [38] J. Douglas, «Building Android Apps with Entity Framework - Xamarin Blog,» 9 Febrero 2017. [En línea]. Available: <https://blog.xamarin.com/building-android-apps-with-entity-framework/>. [Último acceso: 30 Agosto 2017].
- [39] «SQL Server Management Studio (SSMS) - Microsoft Docs,» 2 Febrero 2017. [En línea]. Available: <https://docs.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms>. [Último acceso: 30 Agosto 2017].

- [40] «Usar SQL Server Management Studio - Microsoft TechNet,» [En línea]. Available: [https://technet.microsoft.com/es-es/library/ms174173\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms174173(v=sql.105).aspx). [Último acceso: 30 Agosto 2017].
- [41] «Microsoft Visual Studio - Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio. [Último acceso: 30 Agosto 2017].
- [42] «Comparar los IDE de Visual Studio 2017 - Visual Studio,» [En línea]. Available: <https://www.visualstudio.com/es/vs/compare/>. [Último acceso: 30 Agosto 2017].
- [43] D. Snider, «SQL Server Trigger After Update for a Specific Value - SQL Server Tips,» 10 Septiembre 2015. [En línea]. Available: <https://www.mssqltips.com/sqlservertip/4024/sql-server-trigger-after-update-for-a-specific-value/>. [Último acceso: 30 Agosto 2017].
- [44] S. Weisfeld, «Entity Framework Code First: Executing SQL files on database creation,» 15 Julio 2011. [En línea]. Available: <http://www.drowningintechndealt.com/ShawnWeisfeld/archive/2011/07/15/entity-framework-code-first-executing-sql-files-on-database-creation.aspx>. [Último acceso: 30 Agosto 2017].
- [45] «EntityFramework-Plus - GitHub Wiki,» 13 Diciembre 2016. [En línea]. Available: <https://github.com/zzzprojects/EntityFramework-Plus/wiki>. [Último acceso: 30 Agosto 2017].
- [46] «EF property of type List<enum> not created in db - Stack Overflow,» 10 Febrero 2015. [En línea]. Available: <https://stackoverflow.com/questions/28429945/ef-property-of-type-listenum-not-created-in-db>. [Último acceso: 30 Agosto 2017].
- [47] «Model-view-viewmodel - Wikipedia,» [En línea]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>. [Último acceso: 30 Agosto 2017].
- [48] «What is the difference between MVC and MVVM? - Stack Overflow,» 29 Abril 2015. [En línea]. Available: <https://stackoverflow.com/questions/667781/what-is-the-difference-between-mvc-and-mvvm>. [Último acceso: 30 Agosto 2017].
- [49] «The MVVM Pattern - MSDN - Microsoft,» 10 Febrero 2012. [En línea]. Available: <https://msdn.microsoft.com/en-us/library/hh848246.aspx>. [Último acceso: 30 Agosto 2017].
- [50] «Updating one to many relationship tables using Entity Framework 6 - Yasser Shaikh,» 21 Junio 2015. [En línea]. Available: <http://yassershaikh.com/updating-one-to-many-relationship-tables-using-entity-framework-6/>. [Último acceso: 30 Agosto 2017].

ANEXOS

- Anexo A. Casos de Uso
- Anexo B. Tecnologías Utilizadas
 - B1. SQL Server Management Studio
 - B2. Microsoft Visual Studio 2017 Community
- Anexo C. Triggers en Entity Framework
- Anexo D. Auditoría en Entity Framework
- Anexo E. Tablas de la Base de Datos
- Anexo F. Prototipos de la Interfaz de Usuario
 - F1. Inicio de Sesión
 - F2. Gestión de usuarios
 - F3. Gestión de clientes
 - F4. Gestión de proveedores
 - F5. Gestión de recepciones
 - F6. Obtención de la trazabilidad
 - F7. Gestión de elaboraciones
 - F8. Gestión de ventas
- Anexo G. Manual de Usuario

Anexo A. CASOS DE USO

En este anexo, se describen textualmente cada uno de los casos de uso mediante una tabla que contiene los siguientes campos:

- Identificador y nombre: permite reconocer cada caso de uso inequívocamente. El identificador no tiene por qué guardar relación con el identificador del caso de uso anterior.
- Actores: tipo de usuario o usuarios que pueden realizar el CU.
- Descripción: detalla el objetivo del CU.
- Precondiciones: para poder realizarse el CU es necesario que se cumplan estas condiciones.
- Escenario básico: cada uno de los pasos o etapas que se deben seguir para realizarse el CU.
- Escenario alternativo: posibles pasos o etapas alternativas al escenario básico.
- Postcondiciones: tras finalizarse el CU, se realizarán estas condiciones.

TABLA A 1. Caso de Uso - Iniciar sesión.

CU01 – Iniciar sesión	
Actores	Todos
Descripción	El usuario deberá iniciar sesión para poder usar la aplicación.
Precondiciones	Ninguna
Escenario Básico	1. Se ejecuta la aplicación. 2. Se introduce el nombre de usuario y la contraseña. 3. Se pulsa el botón de iniciar sesión.
Escenario Alternativo	3. No se introduce la contraseña y se le pide que la introduzca para poder continuar.
Escenario Alternativo	3. Se marca recordar sesión. 4. Se pulsa el botón de iniciar sesión.
Postcondiciones	Se accede a la aplicación.

TABLA A 2. Caso de Uso - Cerrar sesión.

CU02 – Cerrar sesión	
Actores	Todos
Descripción	El usuario cerrará su sesión.
Precondiciones	Haber iniciado sesión previamente.
Escenario Básico	1. Se escoge en el menú superior "Archivo → Cerrar Sesión".
Escenario Alternativo	N/A
Postcondiciones	Se cierra la sesión del usuario.

TABLA A 3. *Caso de Uso - Gestión usuarios.*

CU03 – Gestionar usuarios	
Actores	Super Administrador y Administrador
Descripción	El usuario podrá gestionar los distintos usuarios que haya en el sistema, podrá modificarlos (incluido banear), borrarlos o añadir nuevos usuarios.
Precondiciones	Tener el permiso de gestión de usuarios.
Escenario Básico	1. Se accede a la pestaña de gestión de usuarios.
Escenario Alternativo	Ver Tabla A 4, Tabla A 5 y Tabla A 6.
Postcondiciones	Ninguna

TABLA A 4. *Caso de Uso - Registrar usuario.*

CU04 – Registrar usuario	
Actores	Super Administrador y Administrador
Descripción	El usuario añadirá un nuevo usuario al sistema.
Precondiciones	Tener el permiso de gestión de usuarios.
Escenario Básico	1. Se accede a la pestaña de gestión de usuarios. 2. Se accede al formulario de registro del usuario. 3. Se rellena el formulario correctamente. 4. Se envían los datos.
Escenario Alternativo	3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se crea un nuevo usuario en el sistema.

TABLA A 5. *Caso de Uso - Modificar usuario.*

CU05 – Modificar usuario	
Actores	Super Administrador y Administrador
Descripción	El usuario modificará al usuario seleccionado.
Precondiciones	Tener el permiso de gestión de usuarios.
Escenario Básico	1. Se accede a la pestaña de gestión de usuarios. 2. Se selecciona el usuario a modificar. 3. Se accede al formulario de edición del usuario. 4. Se modifican los datos del formulario correctamente (podrá banearse dicho usuario). 5. Se envían los nuevos datos.
Escenario Alternativo	4. Los nuevos datos del usuario no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos del usuario seleccionado.

TABLA A 6. *Caso de Uso - Borrar usuario.*

CU06 – Borrar usuario	
Actores	Super Administrador y Administrador
Descripción	El usuario borrará al usuario seleccionado.
Precondiciones	Tener el permiso de gestión de usuarios.
Escenario Básico	1. Se accede a la pestaña de gestión de usuarios. 2. Se selecciona el usuario a eliminar. 3. Se pulsa el botón de borrado. 4. Se le pregunta si desea borrar el usuario. 5. Se confirma el borrado del usuario.
Escenario Alternativo	3. Se pulsa la tecla Supr.
Postcondiciones	Se borra el usuario seleccionado.

TABLA A 7. Caso de Uso - Gestión clientes.

CU07 – Gestionar clientes	
Actores	Super Administrador, Administrador y Administrativo
Descripción	El usuario podrá gestionar los distintos clientes que haya en el sistema, podrá modificarlos, borrarlos o añadir nuevos clientes.
Precondiciones	Tener el permiso de gestión de clientes.
Escenario Básico	1. Se accede a la pestaña de gestión de clientes.
Escenario Alternativo	Ver Tabla A 8, Tabla A 9 y Tabla A 10.
Postcondiciones	Ninguna

TABLA A 8. Caso de Uso - Añadir cliente.

CU08 – Añadir cliente	
Actores	Super Administrador, Administrador y Administrativo
Descripción	El usuario añadirá un nuevo cliente al sistema.
Precondiciones	Tener el permiso de gestión de clientes.
Escenario Básico	1. Se accede a la pestaña de gestión de clientes. 2. Se accede al formulario para añadir un nuevo cliente. 3. Se rellena el formulario correctamente. 4. Se envían los datos.
Escenario Alternativo	3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se crea un nuevo cliente en el sistema.

TABLA A 9. Caso de Uso - Modificar cliente.

CU09 – Modificar cliente	
Actores	Super Administrador, Administrador y Administrativo
Descripción	El usuario modificará al cliente seleccionado.
Precondiciones	Tener el permiso de gestión de clientes.
Escenario Básico	1. Se accede a la pestaña de gestión de clientes. 2. Se selecciona el cliente a modificar. 3. Se accede al formulario de edición del cliente. 4. Se modifican los datos del formulario correctamente. 5. Se envían los nuevos datos.
Escenario Alternativo	4. Los nuevos datos del cliente no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos del cliente seleccionado.

TABLA A 10. Caso de Uso - Borrar cliente.

CU10 – Borrar cliente	
Actores	Super Administrador, Administrador y Administrativo
Descripción	El usuario borrará al cliente seleccionado.
Precondiciones	Tener el permiso de gestión de clientes.
Escenario Básico	1. Se accede a la pestaña de gestión de clientes. 2. Se selecciona el cliente a eliminar. 3. Se pulsa el botón de borrado. 4. Se le pregunta si desea borrar el cliente. 5. Se confirma el borrado del cliente.
Escenario Alternativo	3. Se pulsa la tecla Supr.
Postcondiciones	Se borra el cliente seleccionado si no tiene pedidos, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 11. Caso de Uso - Gestión proveedores.

CU11 – Gestionar proveedores	
Actores	Super Administrador, Administrador y Administrativo.
Descripción	El usuario podrá gestionar los distintos proveedores que haya en el sistema, podrá modificarlos, borrarlos o añadir nuevos proveedores.
Precondiciones	Tener el permiso de gestión de proveedores.
Escenario Básico	1. Se accede a la pestaña de gestión de proveedores.
Escenario Alternativo	Ídem a la Tabla A 8, a la Tabla A 9 y a la Tabla A 10. En el caso de borrado de un proveedor deberá no tener ninguna recepción.
Postcondiciones	Ninguna

TABLA A 12. Caso de Uso - Gestión recepciones.

CU12 – Gestionar recepciones	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario podrá gestionar las distintas recepciones de materias primas que haya en el sistema, podrá modificarlas, borrarlas o añadir nuevas recepciones, además de poder gestionar las materias primas que hay en cada recepción.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	1. Se accede a la pestaña de gestión de recepciones.
Escenario Alternativo	Ver Tabla A 13, Tabla A 14, Tabla A 15, Tabla A 16, Tabla A 17 y Tabla A 18.
Postcondiciones	Ninguna

TABLA A 13. Caso de Uso - Añadir recepción.

CU13 – Añadir recepción	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá una nueva recepción de materias primas al sistema.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	1. Se accede a la pestaña de gestión de recepciones. 2. Se accede al formulario de añadir una nueva recepción. 3. Se rellena el formulario correctamente. 4. Se envían los datos.
Escenario Alternativo	3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se crea una nueva recepción en el sistema.

TABLA A 14. Caso de Uso - Modificar recepción.

CU14 – Modificar recepción	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará la recepción seleccionada de materias primas.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	1. Se accede a la pestaña de gestión de recepciones. 2. Se selecciona la recepción a modificar. 3. Se accede al formulario de edición de la recepción. 4. Se modifican los datos del formulario correctamente. 5. Se envían los nuevos datos.
Escenario Alternativo	4. Los nuevos datos de la recepción no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos de la recepción seleccionada.

TABLA A 15. Caso de Uso - Borrar recepción.

CU15 – Borrar recepción	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará la recepción seleccionada de materias primas.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de recepciones. 2. Se selecciona la recepción a eliminar. 3. Se pulsa el botón de borrado. 4. Se le pregunta si desea borrar la recepción. 5. Se confirma el borrado de la recepción.
Escenario Alternativo	<ol style="list-style-type: none"> 3. Se pulsa la tecla Supr.
Postcondiciones	Se borra la recepción seleccionada si no tiene materias primas asignadas, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 16. Caso de Uso - Añadir materia prima.

CU16 – Añadir materia prima	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá una nueva materia prima al sistema.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de recepciones. 2. Se selecciona la recepción que contendrá esa materia prima. 3. Se accede al formulario de añadir una nueva materia prima. 4. Se rellena el formulario correctamente. 5. Se envían los datos.
Escenario Alternativo	<ol style="list-style-type: none"> 4. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se asigna una nueva materia prima, a la recepción seleccionada, en el sistema.

TABLA A 17. Caso de Uso - Modificar materia prima.

CU17 – Modificar materia prima	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará la materia prima seleccionada.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de recepciones. 2. Se selecciona la recepción que contiene la materia prima. 3. Se selecciona la materia prima modificar. 4. Se accede al formulario de edición de la materia prima. 5. Se modifican los datos del formulario correctamente. 6. Se envían los nuevos datos.
Escenario Alternativo	<ol style="list-style-type: none"> 5. Los nuevos datos de la materia prima no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos de la materia prima seleccionada. Si con dicha materia prima se ha elaborado algún producto terminado no podrá modificarse la cantidad (unidades o volumen) ni su ubicación (huecos de recepción).

TABLA A 18. Caso de Uso - Borrar materia prima.

CU18 – Borrar materia prima	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará la materia prima seleccionada.
Precondiciones	Tener el permiso de gestión de recepciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de recepciones. 2. Se selecciona la recepción que contiene la materia prima. 3. Se selecciona la materia prima. 4. Se pulsa el botón de borrado. 5. Se le pregunta si desea borrar la materia prima. 6. Se confirma el borrado de la recepción.
Escenario Alternativo	<ol style="list-style-type: none"> 4. Se pulsa la tecla Supr.
Postcondiciones	Se borra la materia prima seleccionada si no se han fabricado productos terminados con ella, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 19. Caso de Uso - Obtener trazabilidad.

CU19 – Obtener trazabilidad	
Actores	Super Administrador, Administrador, Administrativo y Técnico.
Descripción	El usuario obtendrá la trazabilidad de una recepción, de una materia prima, de un producto terminado o de un pedido.
Precondiciones	Tener el permiso de obtención de trazabilidad.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de obtención de trazabilidad. 2. Se escribe el número de albarán de la recepción de materias primas o el código de la materia prima. 3. Se obtiene todo el proceso llevado a cabo en dicha recepción o materia prima. 4. El usuario consulta los datos de un ítem concreto (proveedor, hueco de recepción, ...).
Escenario Alternativo	<ol style="list-style-type: none"> 4. El usuario hace clic en el botón de generar un informe en PDF. 5. El usuario obtiene el informe completo en PDF.
Postcondiciones	La aplicación obtendrá todos los datos del ítem consultado (recepción, materia prima, ...). Si se escoge el informe en PDF se generará el archivo.

TABLA A 20. Caso de Uso - Gestión permisos

CU20 – Gestionar permisos	
Actores	Super Administrador.
Descripción	El usuario podrá gestionar los permisos de cada uno de los tipos de usuarios, así como modificar, borrar o añadir nuevos tipos.
Precondiciones	Tener el permiso de gestión de permisos.
Escenario Básico	N/A
Escenario Alternativo	Ídem Gestión usuarios.
Postcondiciones	Ninguna.

TABLA A 21. Caso de Uso - Cambiar ajustes

CU21 – Cambiar ajustes	
Actores	Todos.
Descripción	El usuario podrá cambiar los distintos ajustes de la aplicación.
Precondiciones	Ninguna.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la ventana de ajustes.
Escenario Alternativo	Ver Tabla A 22, Tabla A 23 y Tabla A 24.
Postcondiciones	Ninguna.

TABLA A 22. Caso de Uso - Cambiar contraseña

CU22 – Cambiar contraseña	
Actores	Todos.
Descripción	El usuario podrá cambiar su contraseña.
Precondiciones	Ninguna.
Escenario Básico	<ol style="list-style-type: none"> 1. Dentro de la ventana de ajustes se accede al apartado de cambio de contraseña. 2. Se introduce la nueva contraseña. 3. Se confirma la nueva contraseña. 4. Se envía el hash de la nueva contraseña a la BD.
Escenario Alternativo	3. Se confirma la nueva contraseña de forma errónea y, al no ser iguales las dos contraseñas, se le vuelve a pedir que la confirme.
Postcondiciones	La contraseña del usuario será actualizada.

TABLA A 23. Caso de Uso - Cambiar color del tema

CU23 – Cambiar color del tema	
Actores	Todos.
Descripción	El usuario podrá cambiar el color de la aplicación, así como escoger un tema oscuro.
Precondiciones	Ninguna.
Escenario Básico	<ol style="list-style-type: none"> 1. Dentro de la ventana de ajustes se accede al apartado de apariencia. 2. Se escoge un tema de color azul.
Escenario Alternativo	2. Se escoge el tema oscuro.
Postcondiciones	La aplicación actualizará su color.

TABLA A 24. Caso de Uso - Cambiar ajustes ventana

CU24 – Cambiar ajustes ventana	
Actores	Todos.
Descripción	El usuario podrá cambiar distintos ajustes relativos a la ventana.
Precondiciones	Ninguna.
Escenario Básico	<ol style="list-style-type: none"> 1. Dentro de la ventana de ajustes se accede al apartado de la ventana. 2. El usuario selecciona “iniciar la aplicación maximizada”.
Escenario Alternativo	2. El usuario selecciona “recordar pestaña activa”.
Postcondiciones	Dependiendo de lo que se modificó en los ajustes, la siguiente vez que se inicie la aplicación, la ventana podrá estar maximizada, podrá tener la pestaña activa de la última vez, etc.

Las siguientes tablas corresponden al TFG de Javier Bielsa Hernández por lo que pueden variar ligeramente los escenarios de los CU en la versión final.

TABLA A 25. Caso de Uso - Gestión elaboraciones.

CU25 – Gestionar elaboraciones	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario podrá gestionar las distintas elaboraciones para generar productos terminados que haya en el sistema, podrá modificarlas, borrarlas o añadir nuevas órdenes de elaboración, además de poder gestionar los productos terminados que hay en cada orden de elaboración.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	1. Se accede a la pestaña de gestión de elaboraciones.
Escenario Alternativo	Ver Tabla A 26, Tabla A 27, Tabla A 28, Tabla A 29, Tabla A 30 y Tabla A 31.
Postcondiciones	Ninguna.

TABLA A 26. Caso de Uso - Añadir elaboración.

CU26 – Añadir elaboración	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá una nueva orden de elaboración al sistema.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se accede al formulario de añadir una nueva orden de elaboración. 3. Se rellena el formulario correctamente. 4. Se envían los datos.
Escenario Alternativo	<ol style="list-style-type: none"> 3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se crea una nueva orden de elaboración en el sistema.

TABLA A 27. Caso de Uso - Modificar elaboración.

CU27 – Modificar elaboración	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará la orden de elaboración seleccionada de productos terminados.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona la orden de elaboración a modificar. 3. Se accede al formulario de edición de la orden de elaboración. 4. Se modifican los datos del formulario correctamente. 5. Se envían los nuevos datos.
Escenario Alternativo	<ol style="list-style-type: none"> 4. Los nuevos datos de la orden de elaboración no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos de la orden de elaboración seleccionada.

TABLA A 28. Caso de Uso - Borrar elaboración.

CU28 – Borrar elaboración	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará la orden de elaboración seleccionada de productos terminados.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona la orden de elaboración a eliminar. 3. Se pulsa el botón de borrado. 4. Se le pregunta si desea borrar la orden de elaboración. 5. Se confirma el borrado de la orden de elaboración.
Escenario Alternativo	<ol style="list-style-type: none"> 3. Se pulsa la tecla Supr.
Postcondiciones	Se borra la orden de elaboración seleccionada si no tiene productos terminados asignados, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 29. Caso de Uso - Añadir producto terminado.

CU29 – Añadir producto terminado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá un nuevo producto terminado al sistema.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona la orden de elaboración que contendrá ese producto terminado. 3. Se accede al formulario de añadir un nuevo producto terminado. 4. Se rellena el formulario correctamente. 5. Se envían los datos.
Escenario Alternativo	<ol style="list-style-type: none"> 3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se asigna un nuevo producto terminado, a la orden de elaboración, en el sistema.

TABLA A 30. Caso de Uso - Modificar producto terminado.

CU30 – Modificar producto terminado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará el producto terminado seleccionado.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona la orden de elaboración que contiene el producto terminado. 3. Se selecciona el producto terminado a modificar. 4. Se accede al formulario de edición del producto terminado. 5. Se modifican los datos del formulario correctamente. 6. Se envían los nuevos datos.
Escenario Alternativo	<ol style="list-style-type: none"> 4. Los nuevos datos del producto terminado no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos del producto terminado seleccionado. Si con dicho producto terminado se ha elaborado algún producto envasado no podrá modificarse la cantidad (unidades o volumen) ni su ubicación (huecos de almacenaje).

TABLA A 31. Caso de Uso - Borrar producto terminado.

CU31 – Borrar producto terminado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará el producto terminado seleccionado.
Precondiciones	Tener el permiso de gestión de elaboraciones.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona la orden de elaboración que contiene el producto terminado. 3. Se selecciona el producto terminado. 4. Se pulsa el botón de borrado. 5. Se le pregunta si desea borrar el producto terminado. 6. Se confirma el borrado de la orden de elaboración.
Escenario Alternativo	<ol style="list-style-type: none"> 3. Se pulsa la tecla Supr.
Postcondiciones	Se borra el producto terminado seleccionado si no se han fabricado productos envasados con él, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 32. Caso de Uso - Gestión ventas.

CU32 – Gestionar ventas	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario podrá gestionar las distintas ventas para generar productos envasados que haya en el sistema, podrá modificarlas, borrarlas o añadir nuevos pedidos, además de poder gestionar los productos envasados que hay en cada pedido.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	1. Se accede a la pestaña de gestión de ventas.
Escenario Alternativo	Ver Tabla A 33, Tabla A 34, Tabla A 35, Tabla A 36, Tabla A 37 y Tabla A 38.
Postcondiciones	Ninguna.

TABLA A 33. Caso de Uso - Añadir pedido.

CU33 – Añadir pedido	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá un nuevo pedido al sistema.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	1. Se accede a la pestaña de gestión de ventas. 2. Se accede al formulario de añadir un pedido. 3. Se rellena el formulario correctamente. 4. Se envían los datos.
Escenario Alternativo	3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se crea un nuevo pedido en el sistema.

TABLA A 34. Caso de Uso - Modificar pedido.

CU34 – Modificar pedido	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará el pedido seleccionado de productos envasados.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	1. Se accede a la pestaña de gestión de ventas. 2. Se selecciona el pedido a modificar. 3. Se accede al formulario de edición de pedido. 4. Se modifican los datos del formulario correctamente. 5. Se envían los nuevos datos.
Escenario Alternativo	4. Los nuevos datos del pedido no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos del pedido seleccionado.

TABLA A 35. Caso de Uso - Borrar pedido.

CU35 – Borrar pedido	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará el pedido seleccionado de productos envasados.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	1. Se accede a la pestaña de gestión de ventas. 2. Se selecciona el pedido a eliminar. 3. Se pulsa el botón de borrado. 4. Se le pregunta si desea borrar el pedido. 5. Se confirma el borrado del pedido.
Escenario Alternativo	3. Se pulsa la tecla Supr.
Postcondiciones	Se borra el pedido seleccionado si no tiene productos envasados asignados, sino se muestra un mensaje informando que no se ha podido borrar.

TABLA A 36. Caso de Uso - Añadir producto envasado.

CU36 – Añadir producto envasado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario añadirá un nuevo producto envasado al sistema.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de ventas. 2. Se selecciona el pedido que contendrá ese producto envasado. 3. Se accede al formulario de añadir un nuevo producto envasado. 4. Se rellena el formulario correctamente. 5. Se envían los datos.
Escenario Alternativo	3. El usuario no rellena todos los campos obligatorios del formulario y se le pide que los complete para poder continuar.
Postcondiciones	Se asigna un nuevo producto envasado al pedido en el sistema.

TABLA A 37. Caso de Uso - Modificar producto envasado.

CU37 – Modificar producto envasado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario modificará el producto envasado seleccionado.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de ventas. 2. Se selecciona el pedido que contiene el producto envasado. 3. Se selecciona el producto envasado a modificar. 4. Se accede al formulario de edición del producto envasado. 5. Se modifican los datos del formulario correctamente. 6. Se envían los nuevos datos.
Escenario Alternativo	4. Los nuevos datos del producto envasado no son correctos y se le pide al usuario que los corrija.
Postcondiciones	Se actualizan los campos del producto envasado seleccionado. Si con dicho producto envasado se ha utilizado en el <i>picking</i> no podrá modificarse la cantidad (unidades o volumen).

TABLA A 38. Caso de Uso - Borrar producto envasado.

CU38 – Borrar producto envasado	
Actores	Super Administrador, Administrador y Técnico.
Descripción	El usuario borrará el producto envasado seleccionado.
Precondiciones	Tener el permiso de gestión de ventas.
Escenario Básico	<ol style="list-style-type: none"> 1. Se accede a la pestaña de gestión de elaboraciones. 2. Se selecciona el pedido que contiene el producto envasado. 3. Se selecciona el producto envasado. 4. Se pulsa el botón de borrado. 5. Se le pregunta si desea borrar el producto envasado. 6. Se confirma el borrado del pedido.
Escenario Alternativo	3. Se pulsa la tecla Supr.
Postcondiciones	Se borra el producto envasado seleccionado si no se ha utilizado en el <i>picking</i> , sino se muestra un mensaje informando que no se ha podido borrar.

Anexo B. TECNOLOGÍAS UTILIZADAS

Este anexo contiene diferentes tecnologías o herramientas utilizadas para la realización de este TFG.

B1. SQL Server Management Studio

SQL Server Management Studio (SSMS) es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se utiliza para configurar y administrar todos los componentes de Microsoft SQL Server. La herramienta incluye tanto editores de guiones como herramientas gráficas que trabajan con objetos y características del servidor [39].

SQL Server Management Studio combina las características del Administrador corporativo, el Analizador de consultas y *Analysis Manager*, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, funciona con todos los componentes de SQL Server, como *Reporting Services*, *Integration Services* y *SQL Server Compact 3.5 SP2*. Los programadores obtienen una experiencia familiar y los administradores de bases de datos una única herramienta completa que combina herramientas gráficas fáciles de usar con funcionalidad de *scripting* enriquecida [40].

Al igual que SQL Server, esta herramienta merece una mención especial, puesto que permite generar toda la infraestructura de la base de datos gráficamente o manualmente mediante sentencias SQL. De esta manera se facilita muchísimo toda la labor de gestión de la BD.

B2. Microsoft Visual Studio 2017 Community

Microsoft Visual Studio es un IDE (entorno de desarrollo integrado, *Integrated Development Environment*) para sistemas operativos Windows. Soporta múltiples lenguajes de programación (C#, C++, Visual Basic .NET, Python, ...) [41].

El IDE cuenta con tres versiones, una de ellas, la Community, al ser gratuita es la utilizada en la realización de este proyecto.

En la Tabla B 1 puede observarse la comparativa entre las tres versiones existentes [42].

TABLA B 1. Comparativa de las distintas versiones de Visual Studio.

	COMMUNITY	PROFESSIONAL	ENTERPRISE
Escenarios de uso admitidos	●●●○	●●●●	●●●●
Desarrolladores individuales	●	●	●
Aprendizaje en clase	●	●	●
Investigación académica	●	●	●
Contribución a proyectos de código abierto	●	●	●
Organizaciones no empresariales, máximo 5 usuarios	●	●	●
Enterprise		●	●
Compatibilidad de la plataforma de desarrollo	●●●○	●●●○	●●●●
Entorno de desarrollo integrado	●●●○	●●●○	●●●●
Validación de dependencias en vivo			●
Diagramas de capas arquitectónicas			●
Validación de arquitectura			●
Clon de código			●
CodeLens		●	●
Definición de Peek	●	●	●
Refactorización	●	●	●
Implementación web con un solo clic	●	●	●
Visor de recursos de modelo	●	●	●
Sol. con gráficos de dependencias y mapas de código	●	●	●
Compatibilidad con múltiples versiones	●	●	●
Depuración y diagnóstico avanzados	●●●○	●●●○	●●●●
IntelliTrace			●
Integración del depurador del mapa de código			●
Análisis de volcado de memoria de .NET			●
Métrica del código	●	●	●
Depuración de gráficos	●	●	●
Análisis de código estático	●	●	●
Concentrador de rendimiento y diagnósticos	●	●	●
Herramientas de pruebas	●○○○	●○○○	●●●●
Twenty-five percent			●
Live Unit Testing			●
Administración de casos de pruebas			●
Pruebas de rendimiento y carga web			●
IntelliTest			●
Microsoft Fakes (aislamiento de prueba unitaria)			●
Cobertura de código			●
Lab Management			●
Pruebas de UI codificada			●
Pruebas manuales con Microsoft Test Manager			●
Pruebas exploratorias con Microsoft Test Manager			●
Avance rápido en Microsoft Test Manager			●
Pruebas unitarias	●	●	●
Desarrollo multiplataforma	●●●○	●●●○	●●●●
Ensamblados insertados			●
Inspección de Xamarin			●
Xamarin Profiler			●
Simulador de iOS remoto para Windows			●
Compartir código entre Android e iOS con Xamarin	●	●	●
Diseñadores de UI nativos de Android e iOS	●	●	●
Xamarin.Forms	●	●	●
Xamarin Instant Player	●	●	●
Herramientas y características de colaboración	●●●●	●●●●	●●●●
Guiones gráficos para PowerPoint	●	●	●
Revisión del código	●	●	●
Suspensión/reanudación de tareas	●	●	●
Team Explorer	●	●	●

Anexo C. TRIGGERS EN ENTITY FRAMEWORK

Este anexo contiene una breve explicación de cómo crear *triggers* o disparadores en la base de datos usando el enfoque Code First de Entity Framework.

Entity Framework no permite añadir *triggers* usando anotaciones (como las usadas para la creación de las tablas) por lo que hay que crearlas usando sentencias SQL [43] [44].

Lo primero que hay que hacer es crear una clase que herede de `DropCreateDatabaseAlways<BiomasaEUPContext>` la cual al instanciarse borrará la base de datos y la creará de nuevo, si se desea también se puede heredar de la clase `CreateDatabaseIfNotExists<BiomasaEUPContext>` la cual sólo borrará la BD si se han producido cambios en las clases POCO.

```
public class BiomasaEUPContextInitializer : DropCreateDatabaseAlways<BiomasaEUPContext>
{
    protected override void Seed(BiomasaEUPContext context)
    {
        context.Database.ExecuteSqlCommand(
@"ALTER TABLE Usuarios ADD CONSTRAINT DF_UsuariosFechaAlta DEFAULT GETDATE() FOR FechaAlta;
ALTER TABLE Usuarios ADD CONSTRAINT DF_UsuariosBaneado DEFAULT 0 FOR Baneado;

SET ANSI_NULLS ON;
SET QUOTED_IDENTIFIER ON;

EXEC dbo.sp_executesql @statement = N'
CREATE TRIGGER [dbo].[TR_Usuarios_U]
ON [dbo].[Usuarios]
AFTER UPDATE
AS
BEGIN
SET NOCOUNT ON;
IF UPDATE(Baneado)
BEGIN
    UPDATE u
    SET FechaBaja = CASE WHEN i.Baneado = 1 THEN GETDATE() ELSE NULL END
    FROM Usuarios u
    JOIN inserted i ON u.UsuarioId = i.UsuarioId;
END
IF UPDATE(Contrasena)
BEGIN
    UPDATE u
    SET FechaContrasena = CASE WHEN i.Contrasena IS NOT NULL THEN GETDATE() ELSE NULL END
    FROM Usuarios u
    JOIN inserted i ON u.UsuarioId = i.UsuarioId;
END
END
'" );
        base.Seed(context);
    }
}
```

FIGURA C 1. Parte del contenido de la clase *BiomasaEUPContextInitializer*.

Para hacer uso de esta clase hay que instanciarla desde el constructor del contexto de la base de datos BiomasaEUPTContext.

```
public class BiomasaEUPTContext : DbContext
{
    public BiomasaEUPTContext()
    : base("name=BiomasaEUPTContext")
    {
        // CUIDADO -> Descomentar para borrar la tabla y la volverla a crear
        Database.SetInitializer(new BiomasaEUPTContextInitializer());

        // ...
    }

    // ...
}
```

FIGURA C 2. Inicializador de la BD desde el BiomasaEUPTContext.

Anexo D. AUDITORÍA EN ENTITY FRAMEWORK

El siguiente apartado explica cómo realizar una auditoría de la base de datos a través de Entity Framework.

EF no permite realizar una auditoría de la BD pero existe el paquete EntityFramework-Plus el cual además de ofrecer mejoras como son funciones para el insertado de datos masivos o mejoras en las consultas LINQ, ofrece métodos para realizar la auditoría [45].

Para usar EntityFramework-Plus hay que añadirlo al proyecto desde el administrador de paquetes NuGet o desde la “Consola de Administrador de Paquetes” usando el siguiente comando.

```
PM> Install-Package Z.EntityFramework.Plus.EF6
```

Posteriormente hay que sobrescribir la función SaveChanges() para hacer uso de la clase Audit y así conseguir la auditoría de la BD. Si se desea puede cambiarse el nombre de las tablas que crea por defecto (AuditEntry y AuditEntryProperty) desde el método OnModelCreating().

```
public class BiomasaEUPContext : DbContext
{
    public BiomasaEUPContext()
    : base("name=BiomasaEUPContext")
    {
        // ....

        AuditManager.DefaultConfiguration.AutoSavePreAction = (ctx, audit) =>
        // añadir "Where(x => x.AuditEntryID == 0)" para permitir
        // múltiples SaveChanges con el mismo Audit()
        (ctx as BiomasaEUPContext).AuditoriaTablas.AddRange(audit.Entries);
    }

    public override int SaveChanges()
    {
        var auditoria = new Audit()
        {
            CreatedBy = Properties.Settings.Default.usuario
        };

        auditoria.PreSaveChanges(this);
        var filasAfectadas = base.SaveChanges();
        auditoria.PostSaveChanges();

        if (auditoria.Configuration.AutoSavePreAction != null)
        {
            auditoria.Configuration.AutoSavePreAction(this, auditoria);
            base.SaveChanges();
        }

        return filasAfectadas;
    }
}
```

```

protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    // Cambio del nombre por defecto de las tablas de auditoría
    modelBuilder.Entity<AuditEntry>().ToTable("AuditoriaTablas");
    modelBuilder.Entity<AuditEntryProperty>()
        .ToTable("AuditoriaDatosTablas");
}
// ...

// Auditoría
public DbSet<AuditEntry> AuditoriaTablas { get; set; }
public DbSet<AuditEntryProperty> AuditoriaDatosTablas { get; set; }
}







```

FIGURA D 1. Auditoría usando EntityFramework-Plus.

Anexo E. TABLAS DE LA BASE DE DATOS

En este anexo, se describen individualmente cada una de las tablas que hay en la base de datos, exceptuando las tablas pertenecientes al TFG de Javier Bielsa Hernández debido a que estas pueden variar respecto a la versión final.











FIGURA E 1. Representación gráfica de la tabla de la BD “TipoUsuario”.

 TipoUsuario
Propiedades
 TipoUsuarioId
 Nombre
 Descripcion
Propiedades de navegación
 Usuarios
 Permisos

Tipo de usuario según el rol que desempeña. Cada tipo (super administrador, administrador, administrativo o técnico) tiene unas funciones específicas en el sistema.

- Nombre: Nombre del tipo de usuario. Debe ser único.
- Descripción: Breve descripción del rol que desempeña.






FIGURA E 2. Representación gráfica de la tabla de la BD “Usuario”.

 Usuario
Propiedades
 UsuarioId
 Nombre
 Contraseña
 Email
 FechaAlta
 FechaBaja
 FechaContraseña
 Baneado
Propiedades de navegación
 TipoUsuario

Cada uno de los usuarios que hay en el sistema. Si no se es un usuario registrado y activo no se podrá usar la aplicación.

- Nombre: Nombre de usuario necesario para iniciar sesión junto con la contraseña. Debe ser único.
- Contraseña: Hash SHA-256 de la contraseña.
- Email: Correo electrónico del usuario. Debe ser único.
- FechaAlta: Fecha de registro. El valor es rellenado por un *trigger* al añadir el usuario (INSERT).
- FechaBaja: Fecha de cuando un usuario fue baneado. El valor es rellenado por un *trigger* al poner el campo “Baneado” a “true”.
- FechaContraseña: Fecha del último cambio de la contraseña. El valor es rellenado por un *trigger* al actualizar el campo “Contraseña” (UPDATE).
- Baneado: Campo booleano que informa si un usuario está dado de baja en el sistema. Si es “true” no podrá iniciar sesión.

FIGURA E 3. Representación gráfica de la tabla de la BD “Permiso”.

 Permiso
Propiedades
 PermisoId
 Tab
 TipoId
Propiedades de navegación
 TipoUsuario

Indica los permisos que tienen cada uno de los tipos de usuarios.

- Tab: Enum que contiene las diferentes pestañas de la aplicación (usuarios, clientes, proveedores, recepciones, permisos, ...). Es necesario crear esta tabla porque EF no permite tener una lista de *enum* directamente [46].

FIGURA E 4. Representación gráfica de la tabla de la BD “TipoCliente”.

T TipoCliente
<i>Propiedades</i>
<ul style="list-style-type: none"> TipoClienteId Nombre Descripcion
<i>Propiedades de navegación</i>
⇒ Clientes

Distintos tipos de clientes que puede haber.

- Nombre: Nombre del tipo de cliente. Debe ser único.
- Descripcion: Breve descripción del tipo.

FIGURA E 5. Representación gráfica de la tabla de la BD “GrupoCliente”.

T GrupoCliente
<i>Propiedades</i>
<ul style="list-style-type: none"> GrupoClienteId Nombre Descripcion
<i>Propiedades de navegación</i>
⇒ Clientes

Distintos grupos de clientes que puede haber. Un tipo de cliente no tiene por qué estar asociado a un grupo.

- Nombre: Nombre del grupo de cliente. Debe ser único.
- Descripcion: Breve descripción del grupo.

FIGURA E 6. Representación gráfica de la tabla de la BD “Cliente”.

T Cliente
<i>Propiedades</i>
<ul style="list-style-type: none"> ClienteId RazonSocial Nif Email Observaciones Calle MunicipioId TipoId GrupoId
<i>Propiedades de navegación</i>
⇒ TipoCliente ⇒ GrupoCliente ⇒ Municipio

Clientes que realizan pedidos de productos envasados. Estos pueden ser empresas o particulares.

- RazonSocial: Si el cliente es una empresa será la razón social de la empresa, si es un particular será el nombre de la persona. Debe ser único.
- Nif: NIF de la empresa o del particular (DNI). Debe ser único.
- Email: Correo electrónico del cliente. Debe ser único.
- Observaciones: Breve observación del cliente. Es un campo opcional.
- Calle: Calle del municipio donde reside la empresa o el particular.












FIGURA E 7. Representación gráfica de la tabla de la BD “TipoProveedor”.

T TipoProveedor
<i>Propiedades</i>
<ul style="list-style-type: none"> TipoProveedorId Nombre Descripcion
<i>Propiedades de navegación</i>
⇒ Proveedores

Distintos tipos de proveedores que puede haber.

- Nombre: Nombre del tipo de proveedor. Debe ser único.
- Descripcion: Breve descripción del tipo.






FIGURA E 8. Representación gráfica de la tabla de la BD “Proveedor”.

 Proveedor
<i>Propiedades</i>
 ProveedorId  RazonSocial  Nif  Email  Observaciones  Calle  MunicipioId  TipoId
<i>Propiedades de navegación</i>
 TipoProveedor  Municipio

Proveedores que suministran las materias primas.

- RazonSocial: Ídem razón social del cliente.
- Nif: NIF de la empresa o del particular (DNI). Debe ser único.
- Email: Correo electrónico del proveedor. Debe ser único.
- Observaciones: Breve observación del proveedor. Es un campo opcional.
- Calle: Ídem calle del cliente.








FIGURA E 9. Representación gráfica de la tabla de la BD “Pais”.

 Pais
<i>Propiedades</i>
 PaisId  Codigo  Nombre
<i>Propiedades de navegación</i>
 Comunidades

Distintos países necesarios para los clientes y proveedores.

- Codigo: Código ISO 3166-1 del país (ej.: ES para España). Debe ser único.
- Nombre: Nombre del país.






FIGURA E 10. Representación gráfica de la tabla de la BD “Comunidad”.

 Comunidad
<i>Propiedades</i>
 ComunidadId  Codigo  Nombre  PaisId
<i>Propiedades de navegación</i>
 Pais  Provincias

Comunidades de los países almacenados en la tabla “Pais”.

- Codigo: Código de la comunidad. Debe ser único.
- Nombre: Nombre de la comunidad.


FIGURA E 11. Representación gráfica de la tabla de la BD “GrupoMateriaPrima”.

 GrupoMateriaPrima
<i>Propiedades</i>
 GrupoMateriaPrimaId  Nombre  Descripcion
<i>Propiedades de navegación</i>
 TiposMateriasPrimas

Grupos de materias primas.

- Nombre: Nombre del grupo.
- Descripcion: Breve descripción del grupo.


FIGURA E 12. Representación gráfica de la tabla de la BD “Provincia”.

 Provincia
<i>Propiedades</i>
<ul style="list-style-type: none"> ProvinciaId Codigo Nombre ComunidadId
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> Comunidad Municipios

Provincias de las comunidades almacenadas en la tabla “Comunidad”.

- Codigo: Código de la provincia. Debe ser único.
- Nombre: Nombre de la provincia.


FIGURA E 13. Representación gráfica de la tabla de la BD “Municipio”.

 Municipio
<i>Propiedades</i>
<ul style="list-style-type: none"> MunicipioId Nombre CodigoPostal Latitud Longitud ProvinciaId
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> Cientes Proveedores Provincia

Municipios de las provincias almacenadas en la tabla “Provincia”.

- Nombre: Nombre del municipio. Puede haber varios municipios con el mismo nombre.
- CodigoPostal: Código postal del municipio. Puede haber varios municipios con el mismo código postal.
- Latitud: Latitud del municipio.
- Longitud: Longitud del municipio.


FIGURA E 14. Representación gráfica de la tabla de la BD “Recepcion”.

 Recepcion
<i>Propiedades</i>
<ul style="list-style-type: none"> RecepcionId FechaRecepcion NumeroAlbaran ProveedorId EstadoId
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> EstadoRecepcion MateriasPrimas Proveedor

Recepción de materias primas suministrada por un proveedor.

- FechaRecepcion: Fecha en la que la recepción llegó.
- NumeroAlbaran: Número de albarán que tiene la recepción. Debe ser único.

FIGURA E 15. Representación gráfica de la tabla de la BD “EstadoRecepcion”.

 EstadoRecepcion
<i>Propiedades</i>
<ul style="list-style-type: none"> EstadoRecepcionId Nombre Descripcion
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> Recepciones

Posibles estados en los que se puede encontrar una recepción.

- Nombre: Nombre del estado.
- Descripcion: Breve descripción del estado.

FIGURA E 16. Representación gráfica de la tabla de la BD “MateriaPrima”.

T MateriaPrima	Son las distintas materias primas que llegaron en una recepción.
<i>Propiedades</i>	
<ul style="list-style-type: none"> MaterialPrimaId Volumen Unidades Observaciones FechaBaja TipoId RecepcionId ProcedenciaId Codigo 	<ul style="list-style-type: none"> <u>Volumen</u>: Volumen en metros cúbicos (m³) de la materia prima. <u>Unidades</u>: Unidades de materia prima. <u>Observaciones</u>: Observaciones que puede tener la materia prima. Es un campo opcional. <u>FechaBaja</u>: Fecha de caducidad de la materia prima. También puede ser cuando la cantidad de la materia prima se queda en cero (unidades o m³). <u>Codigo</u>: Código de identificación único de dicha materia prima. Es rellenado por un <i>trigger</i> al hacer una inserción (INSERT).
<i>Propiedades de navegación</i>	
<ul style="list-style-type: none"> TipoMateriaPrima Procedencia Recepcion HistorialHuecosRecepciones 	


FIGURA E 17. Representación gráfica de la tabla de la BD “TipoMateriaPrima”.

T TipoMateriaPrima	Tipos posibles de materias primas. Los tipos están agrupados en grupos.
<i>Propiedades</i>	
<ul style="list-style-type: none"> TipoMateriaPrimaId Nombre Descripcion MedidoEnVolumen MedidoEnUnidades GrupoId 	<ul style="list-style-type: none"> <u>Nombre</u>: Nombre del tipo. <u>Descripcion</u>: Breve descripción del tipo. <u>MedidoEnVolumen</u>: Boleando que indica si el tipo es medido en volumen (m³). <u>MedidoEnUnidades</u>: Boleando que indica si el tipo es medido en unidades.
<i>Propiedades de navegación</i>	
<ul style="list-style-type: none"> MateriasPrimas GruposMateriasPrimas 	

FIGURA E 18. Representación gráfica de la tabla de la BD “SitioRecepcion”.

T SitioRecepcion	Sitios o explanadas donde se almacenarán las materias primas.
<i>Propiedades</i>	
<ul style="list-style-type: none"> SitioRecepcionId Nombre Descripcion 	<ul style="list-style-type: none"> <u>Nombre</u>: Nombre del sitio. <u>Descripcion</u>: Breve descripción del tipo.
<i>Propiedades de navegación</i>	
<ul style="list-style-type: none"> HuecosRecepciones 	

FIGURA E 19. Representación gráfica de la tabla de la BD “HuecoRecepcion”.

 HuecoRecepcion
<i>Propiedades</i>
<ul style="list-style-type: none"> 🔑 HuecoRecepcionId ✍ Nombre ✍ VolumenTotal ✍ UnidadesTotales ✍ Ocupado ✍ SitioId
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> ⇒ SitioRecepcion ⇒ HistorialHuecosRecepciones

Huecos en los que está dividido cada sitio de recepción. En cada hueco sólo habrá una materia prima (de distinto tipo y distinta recepción).

- Nombre: Nombre del hueco.
- VolumenTotal: Volumen de materia prima (en m³) que puede caber en dicho hueco.
- UnidadesTotales: Unidades de materia prima que puede caber en dicho hueco.
- Ocupado: Booleano que indica si el hueco está ocupado por materias primas. El valor es añadido por un *trigger* al añadir el volumen o las unidades en la tabla HistorialHuecoRecepcion. Si la cantidad de unidades o volumen (dependiendo en que se mida) es mayor de cero entonces el campo valdrá “true” sino “false”.

FIGURA E 20. Representación gráfica de la tabla de la BD “HistorialHuecoRecepcion”.


 HistorialHuecoRecepcion
<i>Propiedades</i>
<ul style="list-style-type: none"> 🔑 HistorialHuecoRecepcionId ✍ Volumen ✍ Unidades ✍ VolumenRestante ✍ UnidadesRestantes ✍ MateriaPrimaId ✍ HuecoRecepcionId
<i>Propiedades de navegación</i>
<ul style="list-style-type: none"> ⇒ HuecoRecepcion ⇒ MateriaPrima ⇒ ProductosTerminados

Tabla necesaria para mantener la trazabilidad. Las materias primas que hay en cada hueco no pueden almacenarse en la tabla HuecoRecepcion porque dicha tabla indica el hueco físico.

En cambio, esta tabla indica en todo momento la cantidad de materia prima que hay en cada hueco o que hubo en dicho hueco en el pasado. Además de indicar la cantidad de materia prima que ya ha sido usada y la que hay disponible.

- Volumen: Volumen de materia prima (en m³) que hay almacenados en el hueco.
- Unidades: Unidades de materia prima que hay en el hueco.
- VolumenRestante: Volumen de materia prima que sigue quedando en el hueco. Este campo es necesario ya que al fabricarse productos terminados se irán cogiendo materias primas de los huecos.
- UnidadesRestantes: Unidades de materia prima que sigue quedando en el hueco. Este campo es necesario ya que al fabricarse productos terminados se irán cogiendo materias primas de los huecos.

Anexo F. PROTOTIPOS DE LA INTERFAZ DE USUARIO

Este anexo contiene los distintos prototipos de media fidelidad que han sido diseñados para la aplicación BiomasaEUP. El aspecto de los prototipos puede variar ligeramente del de la aplicación final debido a que fueron diseñados cuando aún no se tenían en cuenta todas las funcionalidades que se iban a implementar; por ejemplo, no están los prototipos correspondientes a la gestión de permisos o a la gestión de los huecos de recepción.

F1. Inicio de Sesión

En la Figura F 1 se muestra la ventana de inicio de sesión. Si los datos que se introducen son incorrectos, se informa con un mensaje de color rojo el error producido y hasta que no se solucione el botón “Iniciar sesión” estará deshabilitado.



El prototipo de la ventana de inicio de sesión de BiomasaEUP presenta un diseño limpio y funcional. La ventana tiene un título "Iniciar Sesión" en la parte superior izquierda y un botón de cierre "X" en la esquina superior derecha. Debajo del título, hay una barra azul con el texto "BiomasaEUP" en blanco. El formulario principal contiene dos campos de entrada: "Usuario" con un ícono de persona y el texto "admin" ingresado, y "Contraseña" con un ícono de candado. Debajo del campo de contraseña, hay un mensaje de error en rojo que dice "El campo contraseña es obligatorio.". Debajo de los campos, hay un checkbox con el texto "Recordar contraseña". En la parte inferior del formulario, hay un botón grande y azul con el texto "INICIAR SESIÓN".

FIGURA F 1. Prototipo de la ventana inicio de sesión.

F2. Gestión de usuarios

En la Figura F 2 se observan los distintos usuarios que hay en el sistema, los cuales se pueden editar haciendo doble clic en cada una de las celdas de la tabla. A la derecha hay un panel con distintas opciones para poder añadir, borrar, modificar o refrescar los distintos usuarios. Además de mostrarse a la izquierda un contador y un filtro. Situando el cursor en el botón circular (\oplus), ubicado debajo de la tabla, aparecerán las mismas opciones que en el menú de la derecha.

BiomasaEUPTE

Archivo

Opciones

Ayuda

USUARIOS

CLIENTES

PROVEEDORES

RECEPCIONES

ELABORACIONES

VENTAS

TRAZABILIDAD

Contador

Administrador

1

Administrativo

1

Técnico

2

Filtro Tipo

Administrador

Descripción Administrado

Administrativo

Descripción Administrativo

Técnico

Descripción Técnico

Usuarios

NOMBRE	CONTRASEÑA	EMAIL	TIPO	BANEADO
admin	****	admin@biomasa.es	Administrador	<input type="checkbox"/>
administrativo	****	administrativo@biomasa.es	Administrativo	<input type="checkbox"/>
tecnicoa	****	tecnicoa@biomasa.es	Técnico	<input type="checkbox"/>
tecnicob	****	tecnicob@biomasa.es	Técnico	<input checked="" type="checkbox"/>

+

Buscar...

Opciones

Añadir

Borrar

Modificar

Refrescar

FIGURA F 2. Prototipo de la ventana gestión de usuarios.

Formulario del usuario

En la Figura F 3 se representa el formulario para poder añadir o modificar a los usuarios. Si no se completan los campos obligatorios el botón “Aceptar” estará deshabilitado. También, si los datos no son los correctos se mostrará un mensaje de error en cada uno de ellos.

Cuando se modifique un usuario los distintos campos estarán rellenos con los valores que tenía dicho cliente y el título tendrá el texto “Modificar usuario”.

El formulario 'NUEVO USUARIO' está dividido en dos columnas. La columna izquierda contiene los campos 'Nombre' (con el mensaje de error 'El campo nombre es obligatorio.'), 'Tipo' (un menú desplegable con 'Administrador' seleccionado) y 'Contraseña' (con asteriscos). La columna derecha contiene 'Email' (con el mensaje 'El campo email no tiene formato válido.'), un checkbox 'Baneado' (seleccionado) y 'Confirmar Contraseña' (con el mensaje 'El campo confirmar contraseña es obligatorio.'). En la parte inferior hay dos botones: 'ACEPTAR' y 'CANCELAR'.

FIGURA F 3. Prototipo del formulario usuario.

F3. Gestión de clientes

La ventana correspondiente a la gestión de clientes (ver Figura F 4), al igual que la de los usuarios, cuenta con una tabla con todos los clientes que hay en el sistema, un botón circular (⊕) ubicado debajo de la tabla para poder añadir, modificar, borrar o refrescar los clientes. Respecto a la gestión de usuarios el filtro de tipo, cuenta con un menú que se hará visible al hacer clic en el botón de los tres puntos (:). Este menú permitirá gestionar los tipos de clientes.

BiomasaEUP

Archivo

Opciones

Ayuda

USUARIOS

CLIENTES

PROVEEDORES

RECEPCIONES

ELABORACIONES

VENTAS

TRAZABILIDAD

Contador

Tipo Cliente 1

2

Tipo Cliente 2

0

Tipo Cliente 3

0

Filtro Tipo :

Tipo Cliente 1

Descripción Tipo Cliente

Tipo Cliente 2

Descripción Tipo Cliente

Tipo Cliente 3

Descripción Tipo Cliente

GRUPOS

Clientes

RAZÓN SOCIAL	NIF	EMAIL	TIPO	GRUPO	CALLE	DIRECCIÓN
Cliente 1	A-11111111	cliente1@biomasaesuptes	Tipo Cliente 1	Grupo Cliente 1	Calle 1	44003 Teruel (Teruel), Aragón - España
Cliente 2	11111111-B	cliente2@biomasaesuptes	Tipo Cliente 1	Grupo Cliente 2	Calle 2	44003 Teruel (Teruel), Aragón - España

+

Buscar...

FIGURA F 4. Prototipo de la ventana de gestión de clientes.

Formulario del cliente

En la Figura F 5 se representa el formulario para poder añadir a los nuevos clientes. Todos los campos del formulario son obligatorios excepto el de las observaciones.

Cuando se selecciona un país, el contenido del *ComboBox* de las comunidades cambia a las que haya en el país seleccionado. De igual forma con los municipios y provincias.

En el formulario de edición de un cliente los campos saldrán rellenos con los datos que tenga dicho cliente y el título tendrá el texto “Modificar cliente”.

Formulario Tipo

La Figura F 6 muestra el formulario para poder añadir o modificar los tipos de clientes. Los dos campos son obligatorios mostrándose un mensaje de error, de color rojo, si el contenido es incorrecto. El formulario para los grupos de clientes es similar.

NUEVO CLIENTE

Razón Social

El campo razón social es obligatorio.

Email

El campo email es obligatorio.

Grupo

Grupo Cliente 1

País

España

Municipio

4003 (Teruel)

Observaciones

NIF

aaaaaaaaaa

El campo NIF no tiene formato válido (L-NNNNNNNN o NNNNNNNN-L).

Calle

Calle 1

Tipo

Tipo Cliente 1

Comunidad

Aragón

Provincia

Teruel

ACEPTAR

CANCELAR

FIGURA F 5. Prototipo del formulario cliente.

NUEVO TIPO

Nombre

El campo nombre es obligatorio.

Descripción

aaa

El campo descripción es menor de 5 caracteres.

ACEPTAR

CANCELAR

FIGURA F 6. Prototipo del formulario tipo de cliente.

F4. Gestión de proveedores

La Figura F 7 muestra la pestaña para poder gestionar los proveedores. Es similar a la gestión de los clientes exceptuando que los proveedores no tienen grupos. Se han omitidos los prototipos que representan los formularios del proveedor y de los tipos de proveedores ya que son iguales los de los clientes.

BiomasaEUP.T

Archivo

Opciones

Ayuda

USUARIOS

CLIENTES

PROVEEDORES

RECEPCIONES

ELABORACIONES

VENTAS

TRAZABILIDAD

Contador

Tipo Proveedor 1

2

Tipo Proveedor 2

0

Tipo Proveedor 3

0

Filtro Tipo :

Tipo Proveedor 1

Descripción Tipo Provee

Tipo Proveedor 2

Descripción Tipo Provee

Tipo Proveedor 3

Descripción Tipo Provee

GRUPOS

Proveedores

RAZÓN SOCIAL	NIF	EMAIL	TIPO	CALLE	DIRECCIÓN
Proveedor 1	A-11111111	proveedor1@biomasaeupt.es	Tipo Proveedor 1	Calle 1	44003 Teruel (Teruel), Aragón - España
Proveedor 2	1111111-B	proveedor2@biomasaeupt.es	Tipo Proveedor 1	Calle 2	44003 Teruel (Teruel), Aragón - España

+

Buscar...

FIGURA F 7. Prototipo de la ventana gestión de proveedores.

F5. Gestión de recepciones

La gestión de recepciones es un poco diferente a las pestañas mostradas anteriormente. Como se puede observar en la Figura F 8 hay dos tablas. La que tiene la cabecera “Recepciones” contiene las recepciones que hay en el sistema, una vez se haga clic en una fila (una recepción), se mostrarán en la tabla de la derecha las materias primas que hay en esa recepción. Al igual que las pestañas anteriores se podrán añadir, modificar o borrar los datos.

ALBARÁN	FECHA	PROVEEDOR	ESTADO
A-00001	10/01/2017 08:15	Proveedor 1	Disponible
A-00002	15/01/2017 09:37	Proveedor 1	Aceptada

TIPO	GRUPO	VOLUMEN /UNIDADES	PROCEDENCIA	F. BAJA CADUCIDAD
Tipo MP 1	Grupo MP 1	40 m³	Procedencia 1	S/D
Tipo MP 2	Grupo MP 1	40 ud.	Procedencia 2	15/10/2017 10:15

FIGURA F 8. Prototipo de la ventana gestión de recepciones

Formulario de recepción

La Figura F 9 representa el formulario para poder añadir una nueva recepción, así como para poder añadir las ya existentes. En el momento de añadir una nueva habrá tres campos el número de albarán, la fecha de recepción y el proveedor. Cuando se modifique alguna ya existente aparecerá el *ComboBox* estado.



Formulario de recepción con el título "NUEVA RECEPCIÓN".

Nº Albarán	Fecha
<input type="text" value="El campo nº albarán es obligatorio."/>	<input type="text" value="17/02/2017 16:05"/>
Proveedor	
<input type="text" value="Proveedor 1"/>	

Botones: ACEPTAR, CANCELAR

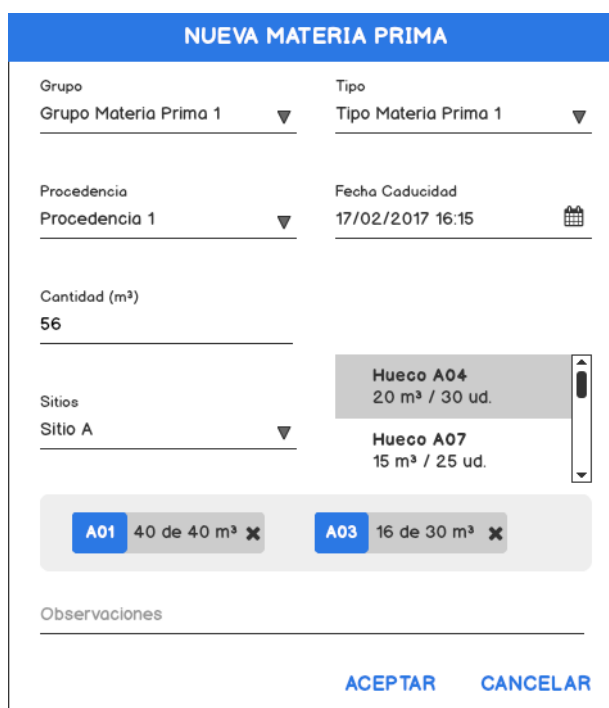
FIGURA F 9. Prototipo del formulario recepción.

Formulario de materia prima

La Figura F 10 muestra el formulario para poder añadir una materia prima a una recepción. El usuario deberá escoger el tipo de materia prima que desea añadir, su procedencia, la fecha de caducidad (campo opcional), y la cantidad (m^3 o en unidades).

Se deberán arrastrar los distintos huecos a la superficie rectangular justo encima de las observaciones de tal forma que haya suficientes huecos como para poder almacenar la cantidad indicada más arriba.

En el formulario de edición de una materia prima los huecos estarán deshabilitados si se ha utilizado esa materia prima para elaborar un producto terminado.



Formulario de materia prima con el título "NUEVA MATERIA PRIMA".

Grupo	Tipo
<input type="text" value="Grupo Materia Prima 1"/>	<input type="text" value="Tipo Materia Prima 1"/>
Procedencia	Fecha Caducidad
<input type="text" value="Procedencia 1"/>	<input type="text" value="17/02/2017 16:15"/>
Cantidad (m^3)	
<input type="text" value="56"/>	
Sitios	
<input type="text" value="Sitio A"/>	

Hueco A04
20 m^3 / 30 ud.

Hueco A07
15 m^3 / 25 ud.

Botones: A01 40 de 40 m^3 ✕, A03 16 de 30 m^3 ✕

Observaciones

Botones: ACEPTAR, CANCELAR

FIGURA F 10. Prototipo del formulario materia prima.

F6. Obtención de la trazabilidad

La pestaña para obtener la trazabilidad (ver Figura F 11) permite obtener la trazabilidad de una recepción dado su número de albarán o de una materia prima dado su código. En la izquierda se representa en forma de árbol todo el proceso que tuvo la recepción o materia prima que se buscó (en la esquina superior derecha). Si se hace clic en un ítem del árbol se muestra información más detallada de dicho ítem. Si se hace clic en el icono del PDF se puede descargar un PDF con la trazabilidad.

Prototipo de la ventana de obtención de trazabilidad. La interfaz muestra una barra de navegación superior con pestañas: Archivo, Opciones, Ayuda, USUARIOS, CLIENTES, PROVEEDORES, RECEPCIONES, ELABORACIONES, VENTAS, y TRAZABILIDAD (seleccionada). Debajo de la barra, a la izquierda, hay un árbol de navegación con los siguientes ítems: Proveedor - Nombre, Número Albarán Recepción, Nombre Materia Prima, Nombre Hueco Recepción 1, Nombre Producto Terminado 1, Nombre Hueco Almacenaje 1, Nombre Producto Terminado 2, y Nombre Hueco Recepción 2. A la derecha del árbol, hay un panel de información con el título 'Información' y el sub-título 'PROVEEDOR'. El panel contiene los siguientes datos: Razón Social: Proveedor 1, NIF: 11111111-A, Email: proveedor1@biomasaeuptes, Tipo: Tipo 1, Dirección: 44003 Teruel (Teruel), Aragón - España, Calle: Calle 1, y Observaciones: S/D. En la esquina superior derecha del panel de información, hay un icono de PDF y el número 1000000001.

FIGURA F 11. Prototipo de la ventana obtención de trazabilidad.

F7. Gestión de elaboraciones

Este prototipo pertenece al TFG de Javier Bielsa Hernández por lo que la apariencia de dicha ventana puede variar en la versión final.

FIGURA F 12. Prototipo de la ventana de gestión de elaboraciones.

FIGURA F 13. Prototipo del formulario orden de elaboración.

FIGURA F 14. Prototipo del formulario producto terminado.

F8. Gestión de ventas

Este prototipo pertenece al TFG de Javier Bielsa Hernández por la apariencia de dicha ventana puede variar en la versión final.

FECHA PEDID	FECHA FINALIZACI	ESTADO PEDID	CLIENT
10/01/2017 08	10/01/2017 08:15	Disponible	Cliente 1
15/01/2017 09	15/01/2017 09:37	Aceptada	Cliente 2

PICKING	VOLUMEN	CÓDIGO	INFO PEDIDO
Picking 1	20 m³	300000001	Info 1
Picking 2	15 m³	300000002	Info 2

FIGURA F 15. Prototipo de la ventana de gestión de pedidos.

NUEVO PEDIDO

Fecha pedido: 17/02/2017 16:05

Fecha: 17/02/2017 16:05

Cliente: Cliente 1

Tipo PT 1: Unidades

Tipo PT 2: Unidades

ACEPTAR CANCELAR

FIGURA F 16. Prototipo del formulario pedido.

NUEVO PRODUCTO ENVASADO

Picking: Picking 1

Volumen (m³):

El campo volumen es obligatorio.

Observaciones:

ACEPTAR CANCELAR

FIGURA F 17. Prototipo del formulario producto envasado.

Anexo G. MANUAL DE USUARIO

Este anexo contiene la parte del manual de usuario de la aplicación BiomasaEUPPT que concierne al presente Trabajo Fin de Grado. Debido a que la aplicación aún no está terminada, porque falta implementar la parte del TFG de Javier Bielsa Hernández, este manual es por lo tanto un borrador, y es susceptible de sufrir modificaciones.

Requisitos mínimos

Para el correcto funcionamiento del programa es necesario tener cubiertos una serie de requisitos, tanto hardware como software:

- Sistema Operativo Windows (a partir de Windows 7).
- Versión de .NET Framework 4.5.2 o superior.
- Procesador, al menos, Intel Celeron de 2,13 GHz o AMD, ...
- Al menos 250 MBytes de memoria RAM.
- Conexión a Internet: Tarjeta Ethernet o WIFI.
- *Software* que permita visualizar documentos PDF.

Instalador

El primer paso es descargar el instalador con el nombre `setup_BiomasaEUPPT_VERSION.exe` desde la dirección github.com/FONT3/BiomasaEUPPT/releases.

Si se desea puede descargarse el archivo ZIP para tener el programa portable, en ese caso deberá omitirse este apartado.

Debido a que el programa no tiene editor y no está firmado puede salir un mensaje de advertencia en Windows al tener activado *SmartScreen* en Windows Defender (Figura G 1). Lo único que hay que hacer es pulsar en el botón “Ejecutar de todas formas” para iniciar el instalador.

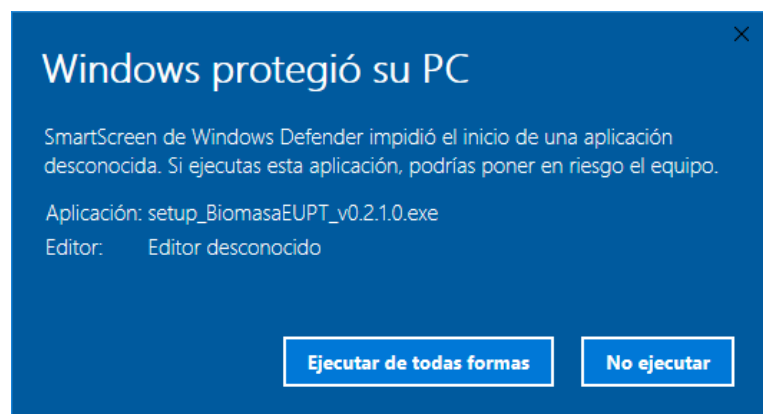


FIGURA G 1. Advertencia de SmartScreen.

Cuando comience el instalar se podrá escoger la ruta donde será instalado el programa, así como crear un acceso directo en el escritorio (Figura G 2).

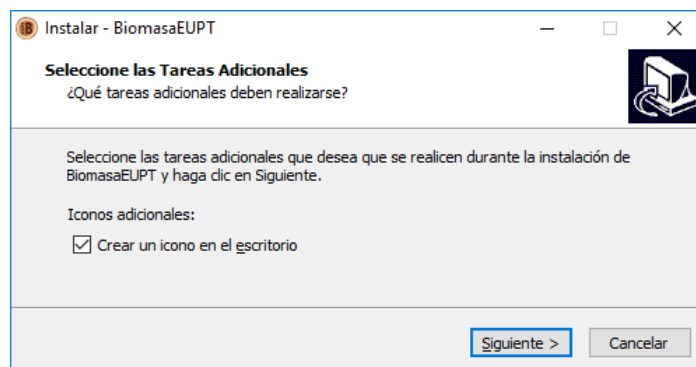


FIGURA G 2. *Instalador de BiomasaEUPt.*

Splash de inicio

Nada más ejecutar el programa se mostrará una pantalla de carga (*Splash*). En esta pantalla se informará si se han encontrado actualizaciones, si se ha podido conectar a la base de datos y cuando el programa ya está listo para iniciarse (Figura G 3).



FIGURA G 3. *Pantalla de carga.*

Si se encuentra una actualización el programa descargará dicha actualización, actualizará el programa y se reiniciará con la nueva versión. Todo esto de manera automática e informándose en todo momento en la barra de progreso.

Ventana de inicio de sesión

La primera pantalla que aparece después del *Splash* de inicio de la aplicación es la ventana de inicio de sesión, donde el usuario deberá introducir su usuario y contraseña. Si desea que la aplicación recuerde su contraseña deberá seleccionar el *checkbox* “Recordar contraseña”.

Cuando se marque el *checkbox* la próxima vez que se arranque la aplicación, se iniciará sesión automáticamente.

Durante toda la aplicación, si algún dato de cualquier formulario está incompleto o es incorrecto se mostrará un mensaje de error. Además, estará deshabilitado el botón para poder enviar los datos de dicho formulario (Figura G 4).

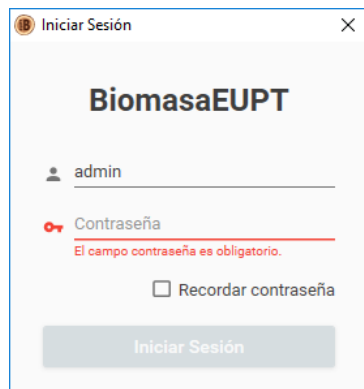


FIGURA G 4. Botón de inicio de sesión deshabilitado.

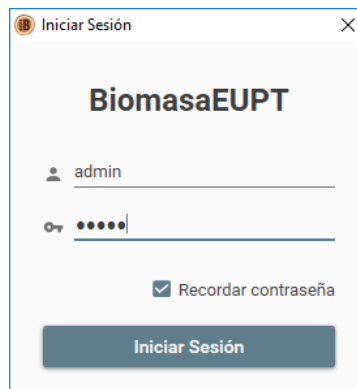


FIGURA G 5. Botón de inicio de sesión habilitado.

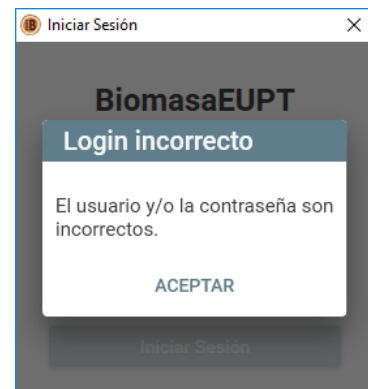


FIGURA G 6. Mensaje de inicio de sesión incorrecto.

Tras iniciar sesión correctamente, dependiendo del tipo de usuario que haya iniciado sesión, se mostrarán distintas pestañas (funcionalidades) en la ventana principal de la aplicación.

Para el usuario Super Administrador:



FIGURA G 7. Pestañas del usuario Super Administrador

Para el usuario Administrador:



FIGURA G 8. Pestañas del usuario Administrador.

Para el usuario Administrativo:

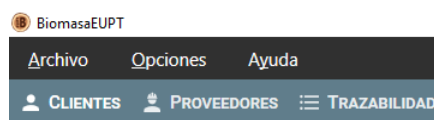


FIGURA G 9. Pestañas del usuario Administrativo.

Para el usuario Técnico:



FIGURA G 10. Pestañas del usuario Técnico.

En este manual, se va a detallar el funcionamiento de la aplicación usando el tipo de usuario “Super Administrador” porque es el que cuenta con toda la funcionalidad del sistema.

Elementos comunes en las pestañas

Con el fin de mantener la estética común en la aplicación, todas las pestañas que cuentan con una tabla tienen una serie de elementos comunes.

- **Botón de opciones circular:** Al situar el cursor del ratón sobre el botón, se mostrarán a la derecha un conjunto de opciones (Figura G 12). Al situar el botón sobre cada botón se mostrará un *tooltip* descriptivo. Algunos botones estarán deshabilitados como son los de borrar o modificar, hasta que no se seleccionen una o más filas de la tabla (Figura G 13).



FIGURA G 11. Botón de opciones colapsado.



FIGURA G 12. Botón de opciones expandido.



FIGURA G 13. Botón de borrado deshabilitado.

- **Filtro de búsqueda:** En el inferior de cada tabla hay un filtro que permite filtrar los elementos de la tabla (Figura G 14). A la derecha del filtro hay un icono de ajustes que permite modificar por qué campos se desea buscar (Figura G 15).



FIGURA G 14. Filtro de las tablas.

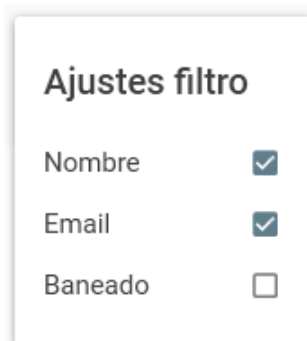


FIGURA G 15. Ventana de configuración del filtro.

- **Descripción de los elementos (*tooltip*):** Algunos elementos (textos, botones, ...) cuentan con una breve descripción al pasar el ratón por encima suyo (Figura G 16).



FIGURA G 16. Tooltip dentro de la sección del contador

- **Atajos de teclado:** Para abrir los formularios usando el teclado basta con pulsar en cualquier pestaña los siguientes atajos de teclado:
 - Ctrl+Shift+A: Formulario para añadir un nuevo registro en la tabla.
 - Ctrl+Shift+M: Formulario para modificar la fila seleccionada de la tabla.
 - Ctrl+Shift+B: Borra la fila o filas seleccionadas de la tabla, al igual que la tecla Supr.
 - Ctrl+Shift+R: Refresca los datos de la tabla.
- **Menú lateral izquierdo en las tablas:** Si las tablas tienen algún campo que es demasiado extenso como para ser mostrado usando una fila, estas tendrán un menú lateral (Figura G 17). Al hacer clic en él, este se desplegará expandiéndose hacia la izquierda y mostrándose su contenido (Figura G 18).



FIGURA G 17. Menú lateral izquierdo de las tablas colapsado.

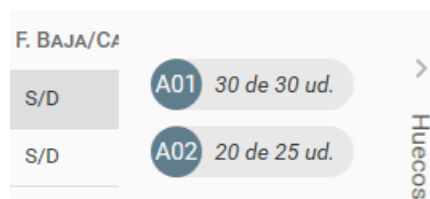


FIGURA G 18. Menú lateral izquierdo de las tablas expandido.

- **Confirmación de borrado:** La aplicación pedirá en todo momento una confirmación cuando se desee borrar algún elemento (Figura G 19).



FIGURA G 19. Mensaje de confirmación de borrado.

- **Filtro lateral:** En algunas ocasiones a la izquierda de la ventana habrá un filtro para poder filtrar los componentes de las tablas (Figura G 20). Se pueden seleccionar varios elementos a la vez (Figura G 21).

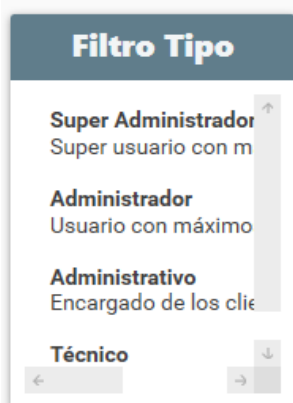


FIGURA G 20. Filtro lateral.

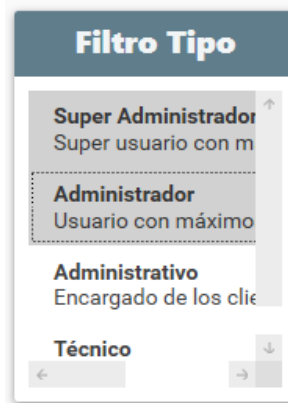


FIGURA G 21. Filtro lateral con varios elementos seleccionados.

- **Mensajes validación formularios:** Cuando un campo en algún formulario es incorrecto se mostrará un mensaje informativo indicando el error (Figura G 22). Hasta que no se solucionen los campos no se podrán enviar los datos del formulario.

Nombre df	Email asasa.com	Contraseña ●●●●
La longitud del campo nombre es menor de 3 caracteres.	El campo email no tiene formato válido.	El campo contraseña y contraseña confirmación no son iguales.

FIGURA G 22. Diferentes mensajes de error de los formularios.

- **Edición de los elementos desde las filas:** En ciertas pestañas (gestión usuarios, clientes y proveedores) se pueden editar las celdas haciendo doble clic en ellas. En las demás pestañas al hacer doble clic se abrirá el formulario de edición (ver Figura G 23 y Figura G 24).

TIPO	GRUPO	CALLE	DIRECCIÓN
Empresa	Pago tarjeta	Ctra. Alcañiz, 10	44003 Teruel (Teruel), Aragón - España
Empresa	Pago tarjeta	Paseo de la independencia, 12	<div> País 🇪🇸 España </div> <div> Comunidad Aragón </div> <div> Provincia Teruel </div> <div> Código Postal 44003 (Teruel) </div>
Particular	Pago efectivo	Plaza Mansuetos, 1	
Empresa	Pago tarjeta	Ctra. Alcañiz, 7	

FIGURA G 23. Edición de la dirección de un cliente desde la tabla.

NOMBRE	CONTRASEÑA	EMAIL	TIPO	BANEADO
superadmin	●●●●●●	superadmin@biomasaeupt.es	Super Administrador	<input type="checkbox"/>
admin	●●●●●●	admin@biomasaeupt.es	Administrador	<input type="checkbox"/>
efonte	●●●●●●	efonte@biomasaeupt.es	Administrativo	<input type="checkbox"/>
jbielsa	●●●●●●	jbielsa@biomasaeupt.es	<div> Técnico </div>	<input type="checkbox"/>

FIGURA G 24. Edición del tipo de usuario desde la tabla.

- **Paginación en las tablas:** en la parte inferior de ciertas tablas habrá un menú para permitir paginar los datos (Figura G 25). Esto es útil cuando se van a tener muchos registros en ellas.

El icono << permite ir a la página principal, el icono < permite volver a la página anterior, el icono > sirve para ir a la página siguiente y el icono >> salta a la última página.

Los iconos permanecerán deshabilitados si ya no hay más páginas a las que ir, además, a la derecha, hay un *combobox* que permite escoger el número de registros que se quieren mostrar por página.



FIGURA G 25. *Menú de paginación de las tablas.*

Gestión de los permisos

En la pestaña de permisos (Figura G 26), se podrá gestionar los tipos de usuarios que hay en sistema. A cada tipo de usuario se le asignarán uno o más permisos. A la izquierda hay un contador que indica el número de usuarios que hay de cada tipo.

Es aconsejable que no se asigne el permiso para poder gestionar los demás permisos a otros tipos de usuarios ya que, si esto se hiciese, el nuevo tipo de usuario tendría máximos privilegios (él mismo se podría modificar sus privilegios).

BiomasaEUPPT

Archivo

Opciones

Ayuda

PERMISOS

USUARIOS

CLIENTES

PROVEEDORES

RECEPCIONES

ELABORACIONES

VENTAS

TRAZABILIDAD

Contador

Super Administr...

1

Administrador

1

Administrativo

1

Técnico

1

Tipos de Usuarios

Descripción

Super Administrador

Permisos

Usuarios

Clientes

Proveedores

Recepciones

Elaboraciones

Ventas

Trazabilidad

Administrador

Usuarios

Clientes

Proveedores

Recepciones

Elaboraciones

Ventas

Trazabilidad

Administrativo

Clientes

Proveedores

Trazabilidad

Técnico

Recepciones

Elaboraciones

Ventas

Trazabilidad

+

Buscar...

Opciones

Añadir

Borrar

Modificar

Refrescar

FIGURA G 26. BiomasaEUPPT - Pestaña permisos.

G8

El formulario para añadir un nuevo tipo de usuario tendrá el campo nombre de usuario y observaciones (Figura G 27). Para añadir los permisos se deberán arrastrar tantos permisos como se requieran desde la lista superior derecha hasta el rectángulo central que está sombreado (Figura G 28).

El formulario 'Nuevo Tipo de Usuario' tiene un encabezado azul con el título 'Nuevo Tipo de Usuario'. A la izquierda, hay dos campos de texto: 'Nombre' con un mensaje de error 'El campo nombre es obligatorio.' y 'Descripción' con un mensaje de error 'La longitud del campo descripción es menor de 5 caracteres.'. A la derecha, hay una lista de permisos: 'Gestión de Permisos', 'Gestión de Usuarios' y 'Gestión de Clientes'. Debajo de la lista, hay un mensaje: 'Por favor, arrastre los permisos que desee asignar al tipo de usuario.'. En la parte inferior, hay dos botones: 'ACEPTAR' y 'CANCELAR'.

FIGURA G 27. Formulario del tipo de usuario.

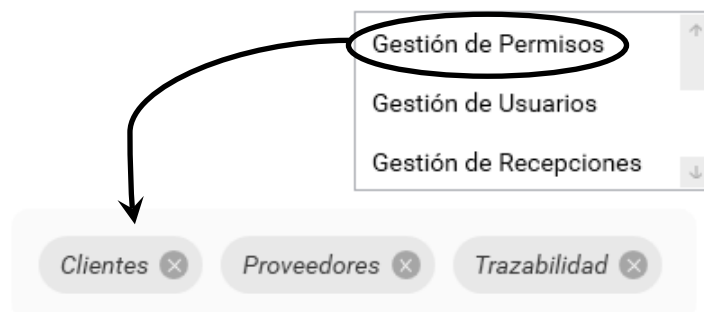


FIGURA G 28. Arrastrado de distintos permisos.

Gestión de los usuarios

En la pestaña de usuarios (Figura G 29), se pueden gestionar los distintos usuarios que hay en el sistema. Además de poder borrar un usuario, este se puede bloquear (banear) con el fin de que ya no puede entrar en el sistema al iniciar sesión. Esto es muy útil si se desea restringir temporalmente el acceso de ciertos usuarios.

Esta pestaña permite modificar los distintos usuarios desde la propia tabla para ello, hay que hacer doble clic en la celda que se quiera modificar y la celda cambiará a modo editable. Si se desean cancelar los cambios realizados basta con pulsar la tecla Esc.

Al hacer clic en el botón “Añadir usuario” se abrirá el formulario de inserción de un nuevo usuario (Figura G 30). Este requerirá un nombre de usuario (en minúsculas), una dirección de correo electrónico (se recomienda usar biomasauept.es), una contraseña (asegúrese de que las contraseñas son iguales) y un tipo de usuario. También es posible crear un usuario no activo marcando el *checkbox* baneado.

Al modificar un usuario, los dos campos relativos a la contraseña estarán si rellenar, debido a que el sistema no tiene almacenadas las contraseñas de los usuarios (están cifradas), por lo que será necesario especificar una nueva contraseña o introducir la anterior. Si no se quiere cambiar la contraseña al modificar un usuario será necesario modificarlo usando la tabla.

El formulario "Nuevo Usuario" contiene los siguientes campos:

- Nombre:** Campo de texto con el mensaje de error "El campo nombre es obligatorio."
- Email:** Campo de texto con el mensaje de error "El campo email es obligatorio."
- Tipo:** Selector de lista desplegable con la opción "Super Administrador" seleccionada.
- Baneado:** Casilla de verificación desactivada.
- Contraseña:** Campo de texto con el mensaje de error "El campo contraseña es obligatorio."
- Confirmar Contraseña:** Campo de texto con el mensaje de error "El campo contraseña confirmación es obligatorio."

En la parte inferior del formulario hay dos botones: "ACEPTAR" (desactivado) y "CANCELAR" (activo).

FIGURA G 30. Formulario del usuario.

Gestión de los clientes

En la pestaña de clientes, se podrán gestionar los clientes que hay en el sistema. Esta pestaña es similar a la de los usuarios por lo que se permiten editar los clientes desde un formulario o desde la propia tabla.

Respecto a la pestaña anterior, en esta se pueden gestionar los tipos y grupos de clientes. En el filtro de la izquierda hay un menú superior (ver icono **:** de la Figura G 31) el cual permite añadir, modificar o borrar los tipos. Para acceder al grupo de clientes (Figura G 32) hay que hacer clic en la flecha inferior (**GRUPOS →**).

El "Filtro Tipo" muestra una lista de tipos de clientes:

- Empresa:** El cliente es...
- Particular:** El cliente es...

Un menú desplegado está abierto sobre la lista, mostrando las opciones: "Añadir", "Editar" y "Borrar".

En la parte inferior del filtro hay un botón con la etiqueta "GRUPOS" y una flecha hacia la derecha.

FIGURA G 31. Filtro de Tipos con el menú desplegado.

El "Filtro Grupo" muestra una lista de grupos de clientes:

- Pago efectivo:** Clientes que pagan en €
- Pago tarjeta:** Clientes que pagan con

En la parte inferior del filtro hay un botón con la etiqueta "TIPOS" y una flecha hacia la izquierda.

FIGURA G 32. Filtro de Grupos con el menú oculto.

FIGURA G 33. BiomasaEUP^T - pestaña clientes.

El formulario para añadir nuevos clientes requiere introducir la razón social del cliente, el NIF (ej.: A-11111111 para una empresa o 11111111-A para un particular), una dirección de correo electrónico, el tipo de cliente, el grupo de cliente y el código postal del municipio donde reside con su calle (Figura G 34). El campo observaciones es opcional.

Cuando se cambia el país, el *combobox* de las comunidades cambiará respecto al país seleccionado, así mismo con las provincias y los códigos postales.

El formulario 'Nuevo Cliente' está dividido en dos columnas. La columna izquierda contiene los campos: 'Razón social' (obligatorio), 'Email' (obligatorio), 'Tipo' (dropdown con 'Empresa' seleccionado), 'País' (dropdown con 'España' y bandera de España), 'Provincia' (dropdown con 'Teruel') y 'Observaciones' (campo de texto). La columna derecha contiene: 'NIF' (obligatorio), 'Calle' (obligatorio), 'Grupo' (dropdown con 'Pago efectivo' seleccionado), 'Comunidad' (dropdown con 'Aragón') y 'Código Postal' (dropdown con '44500 (Andorra)'). En la parte inferior hay dos botones: 'ACEPTAR' y 'CANCELAR'.

FIGURA G 34. Formulario del cliente.

Gestión de los proveedores

La pestaña de gestión de los proveedores (Figura G 35), al ser idéntica a la de los clientes, exceptuando que los proveedores no están asignados a ningún grupo, no cuenta con el filtro de grupos.

FIGURA G 35. BiomasaEUP - pestaña proveedores.

Gestión de las recepciones

La pestaña de recepciones permite gestionar las recepciones de materias primas. Dicha pestaña cuenta con dos tablas, una para las recepciones que hay almacenadas en el sistema y la otra para mostrar las materias primas de cada recepción.

Al pulsar en una recepción de la tabla de la izquierda se mostrarán las materias primas a la derecha. Si el estado de una recepción es “Aceptada” no se podrán añadir nuevas materias primas (el botón de añadir estará deshabilitado).

El formulario para añadir nuevas recepciones cuenta con un número de albarán, la fecha y hora de cuando llegó la recepción y el proveedor (Figura G 36).



Formulario de inserción de la recepción. El formulario tiene un encabezado azul con el título "Nueva Recepción". Debajo, hay tres campos de entrada: "Nº Albarán" (con un mensaje de error rojo que dice "El campo nº albarán es obligatorio."), "Fecha" (con el valor "06/09/2017" y un icono de calendario) y "Hora" (con el valor "15:39" y un icono de reloj). Debajo de estos, hay un campo de proveedor con el valor "Maderas Pepe, S.L." y un menú desplegable. En la parte inferior, hay dos botones: "ACEPTAR" y "CANCELAR".

FIGURA G 36. Formulario de inserción de la recepción.

Cuando se modifica una recepción aparecerá el *combobox* de estado para poder modificar su estado (Figura G 37), este campo no aparece cuando se añade una nueva recepción debido a que el estado de las nuevas recepciones es siempre “Disponible”.

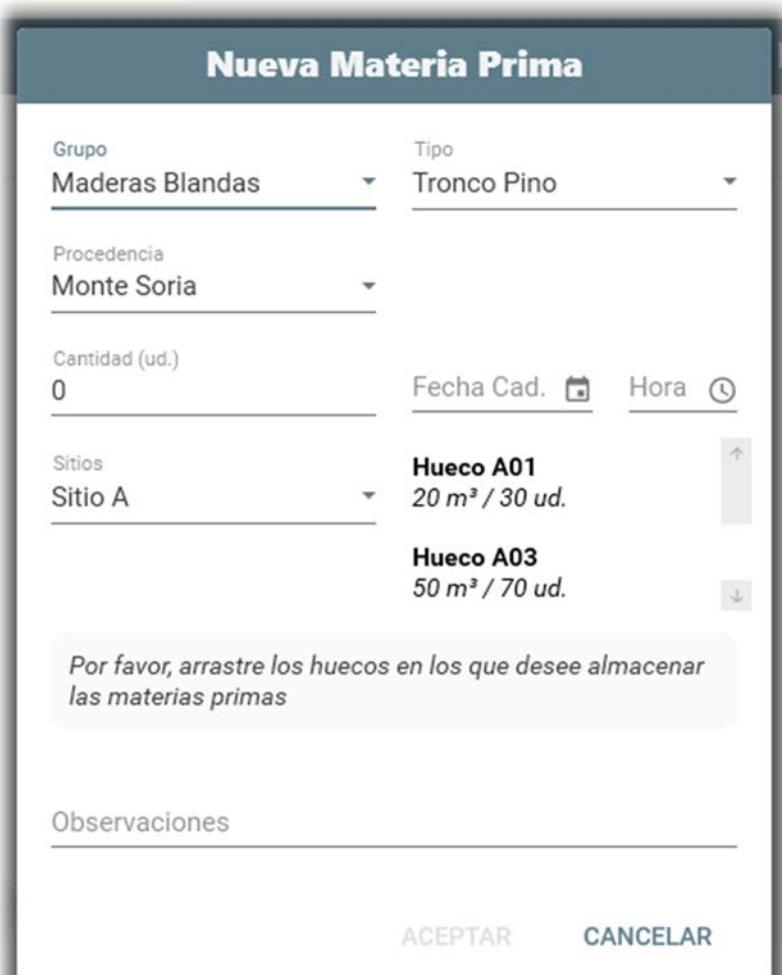


Formulario de modificación de la recepción. El formulario tiene un encabezado azul con el título "Editar Recepción". Debajo, hay tres campos de entrada: "Nº Albarán" (con el valor "A-0100B"), "Fecha" (con el valor "20/01/2017" y un icono de calendario) y "Hora" (con el valor "13:10" y un icono de reloj). Debajo de estos, hay un campo de proveedor con el valor "Maderas Pepe, S.L." y un menú desplegable, y un campo de estado con el valor "Aceptada" y un menú desplegable. En la parte inferior, hay dos botones: "ACEPTAR" y "CANCELAR".

FIGURA G 37. Formulario de modificación de la recepción.

En el formulario correspondiente a las materias primas (Figura G 39), dependiendo del tipo de materia prima que se seleccione el campo cantidad mostrará el tipo de unidades en el que se mide (ya sea en volumen con m³ o en unidades).

Para seleccionar los huecos de recepciones en los que se va a almacenar la cantidad especificada de dicha materia prima, es necesario ir arrastrando los huecos de la lista de la derecha central al panel rectangular que hay justo encima del campo observaciones.



Nueva Materia Prima

Grupo: **Maderas Blandas** Tipo: **Tronco Pino**

Procedencia: **Monte Soria**

Cantidad (ud.): **0** Fecha Cad. Hora

Sitios: **Sitio A**

Hueco A01
20 m³ / 30 ud.

Hueco A03
50 m³ / 70 ud.

Por favor, arrastre los huecos en los que desee almacenar las materias primas

Observaciones

ACEPTAR CANCELAR

FIGURA G 39. Formulario de la materia prima.

Es necesario especificar tantos huecos de recepción como sean necesarios para poder almacenar la cantidad especificada. El orden en el que se irá almacenado la materia prima será el orden en el que se fueron arrastrando los huecos (Figura G 40).



A05 60 de 60 ud. **B03** 20 de 50 ud.

FIGURA G 40. Huecos de recepción donde se almacenará la materia prima.

Cuando se modifica una materia prima, el formulario no dejará editar el tipo y los huecos si dicha materia ya ha sido usada para elaborar algún producto terminado (Figura G 41).

FIGURA G 41. Formulario de edición de una materia prima que ya se ha usado para elaborar.

FIGURA G 42. Formulario de edición de una materia prima sin usar.

Los dos iconos que hay a la derecha de cada una de las filas de la tabla de materias (Figura G 43) primas permiten:

- **Icono con el código de barras:** Generar un fichero PDF el cual contiene los códigos de barras que son necesarios para cada hueco de recepción. Este fichero será generado en una carpeta temporal por lo que si se desea guardar o imprimir deberá realizarse justo después de que se abra el PDF.
- **Icono con el símbolo del PDF:** Generar un fichero PDF con la trazabilidad de la materia prima. Se guardará en la carpeta donde se haya especificado en los ajustes (por defecto en la carpeta informes del escritorio).

PROCEDENCIA	F. BAJA/CADUCIDAD		
Monte Soria	S/D		

FIGURA G 43. Iconos que hay en las filas de la tabla de materias primas.

Procedencias

Monte Soria
Lado norte

Dehesa del Moncayo
Lado sur

Monte Teruel
Lado norte

Monte Galicia
Gran masa forestal

Grupos M. Primas

Maderas Blandas Árboles de un crecimiento más corto.	
Maderas Duras Árboles de un crecimiento lento.	
Varios Agrupación de diferentes tipos de materias primas.	

Tipos M. Primas

Tronco Pino
Madera de color amarillenta, fácil de trabajar.
Medido en unidades

Tronco Abeto
Madera de apariencia similar al pino, aunque más ligera y men
Medido en unidades

Tronco Cedro
Madera ligera, de tono rojizo y con buena resistencia.
Medido en unidades

Sitios Recepciones

Sitio A Este es el Sitio A	Sitio B Este es el Sitio B	Sitio C Este es el Sitio C	Sitio D Este es el Sitio D	Sitio E Este es el Sitio E
--------------------------------------	--------------------------------------	--------------------------------------	--------------------------------------	--------------------------------------

Huecos Recepciones

A01 (20 m³, 20 ud.)

VOLVER →

FIGURA G 44. *BiomasaEUP* - pestaña recepciones mostrando el contenido de "Más Opciones"

Cuando se haga clic en la flecha (**Más Opciones** →) ubicada en la esquina inferior derecha de la ventana de recepciones (Figura G 45) el contenido de la ventana cambiará para poder gestionar los sitios y huecos de recepciones, las procedencias, los grupos y tipos de las materias primas.

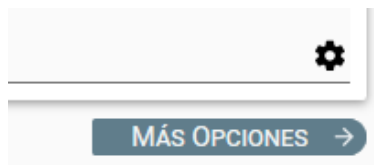


FIGURA G 45. Ubicación del botón "Más Opciones".

Todos los formularios de esta ventana tendrán los campos nombre y descripción. Además, el formulario para los tipos de materias primas tendrá un botón *switch* para poder elegir si la materia va a medirse en volumen o en unidades (Figura G 46).

Una imagen de un formulario de diálogo con el título "Nuevo Tipo M. Prima" en un encabezado azul. El formulario contiene dos campos de texto: "Nombre" y "Descripción". Debajo de cada campo hay un mensaje de error en rojo que dice "El campo nombre es obligatorio." y "El campo descripción es obligatorio." respectivamente. En la parte inferior del formulario, hay dos botones de selección: "Volumen" y "Unidades", ambos con un fondo gris. En la base del formulario, hay dos botones: "ACEPTAR" y "CANCELAR" en un color azul claro.

FIGURA G 46. Formulario del tipo de materia prima.

Gestión de la trazabilidad

En la pestaña de trazabilidad (Figura G 47) se puede obtener la trazabilidad de las recepciones o de las materias primas. Solo basta con introducir, en el campo de la esquina superior derecha, el número de albarán de la recepción o el código de barras de la materia prima que se quiera buscar.

Si el albarán o el código de barras existen en el sistema, se mostrará, a la izquierda, en forma de árbol todo el proceso que se ha llevado con dicha materia o con las materias que llegaron en dicha recepción.

Al hacer clic en un elemento del árbol se mostrará a la derecha información detallada del elemento.

Por otra parte, si se hace clic en el icono del PDF, ubicado en la esquina superior derecha, la aplicación generará un informe en formato PDF en la ruta predefinida en el menú de ajustes (por defecto en la carpeta informes del escritorio).

BiomasaEUPPT

Archivo

Opciones

Ayuda

PERMISOS

USUARIOS

CLIENTES

PROVEEDORES

RECEPCIONES

ELABORACIONES

VENTAS

TRAZABILIDAD

1000000005

Información

MATERIA PRIMA

Tipo: Corcho

Procedencia: Dehesa del Moncayo

Cantidad: 70 m³

Código: 1000000005

Fecha Baja: S/D

Observaciones: S/D

Trazabilidad Materia Prima

Forest Teruel, S.L.

B-00005

Corcho

D02

D01

Papel corcho

A04

B05

FIGURA G 47. BiomasaEUPPT - pestaña trazabilidad.

G21

Menú superior

La aplicación cuenta con un menú superior mediante el cual se puede cerrar la sesión o salir de la aplicación (si se escogió recordar sesión la próxima vez que se inicie el programa volverá a la ventana principal) desde el menú Archivo (Figura G 48).

Desde el menú Opciones (Figura G 49), se pueden acceder a los ajustes de la aplicación.

Por último, desde el menú de ayuda (Figura G 50) se puede acceder a la página del proyecto de GitHub en la cual se puede descargar la última versión del programa, así como ver su código fuente o si se desea reportar fallos.



FIGURA G 48. Menú Archivo desplegado.

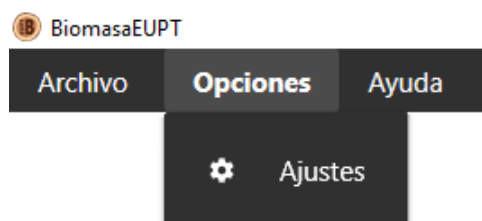


FIGURA G 49. Menú Opciones desplegado.

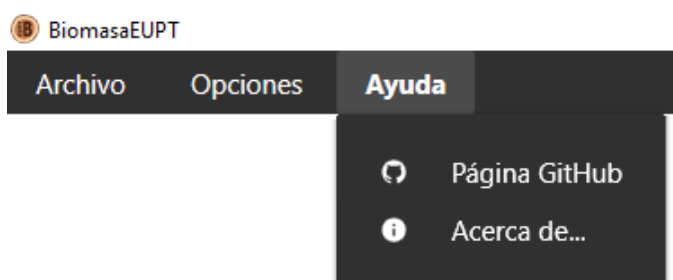


FIGURA G 50. Menú Ayuda desplegado.

Para poder seleccionar cada uno de los menús usando el teclado es necesario mantener pulsado la tecla Alt más la letra que aparecerá subrayada.

Ventana acerca de

Al acceder al menú Ayuda→Acerca de... se puede ver la ventana de acerca (Figura G 51).



FIGURA G 51. Ventana de Acerca de.

Ventana de ajustes

La ventana de ajustes, accediéndose desde Opciones→Ajustes, contiene una serie de pestañas mediante las cuales se pueden modificar distintas opciones de la aplicación.

Desde la pestaña “General” (Figura G 52) se pueden deshabilitar la búsqueda de actualizaciones al iniciar el programa (ventana *Splash* de carga) así como escoger el directorio donde se generarán los informes PDF.



FIGURA G 52. Pestaña de ajustes generales.

En la pestaña “Ventana” (Figura G 53) se puede recordar la última pestaña que se tenía seleccionada al cerrar la aplicación, iniciar la aplicación siempre en modo maximizado, recordar las posiciones de las pestañas (las pestañas de la ventana principal se pueden arrastrar) así como recordar el tamaño de la ventana.

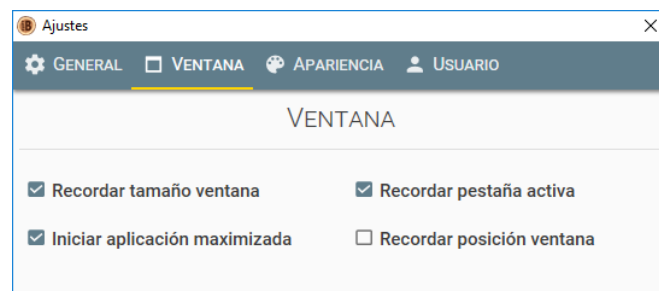


FIGURA G 53. Pestaña de ajustes de la ventana.

Por otra parte, se puede cambiar la contraseña del usuario que está usando la aplicación desde la pestaña “Usuario” (Figura G 54).

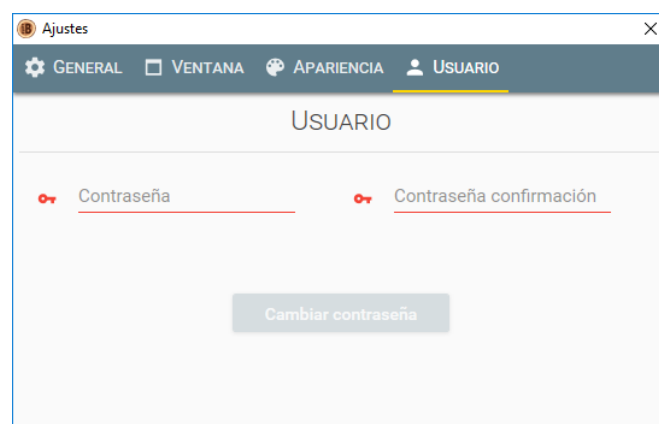


FIGURA G 54. Pestaña de ajustes del usuario.

Desde la pestaña “Apariencia” (Figura G 55) se puede cambiar el estilo visual de la aplicación. Permitiendo seleccionar tanto el color primario como el secundario. Además, marcando el *checkbox* se activará el modo nocturno (Figura G 56).



FIGURA G 55. Pestaña de ajustes de la apariencia.

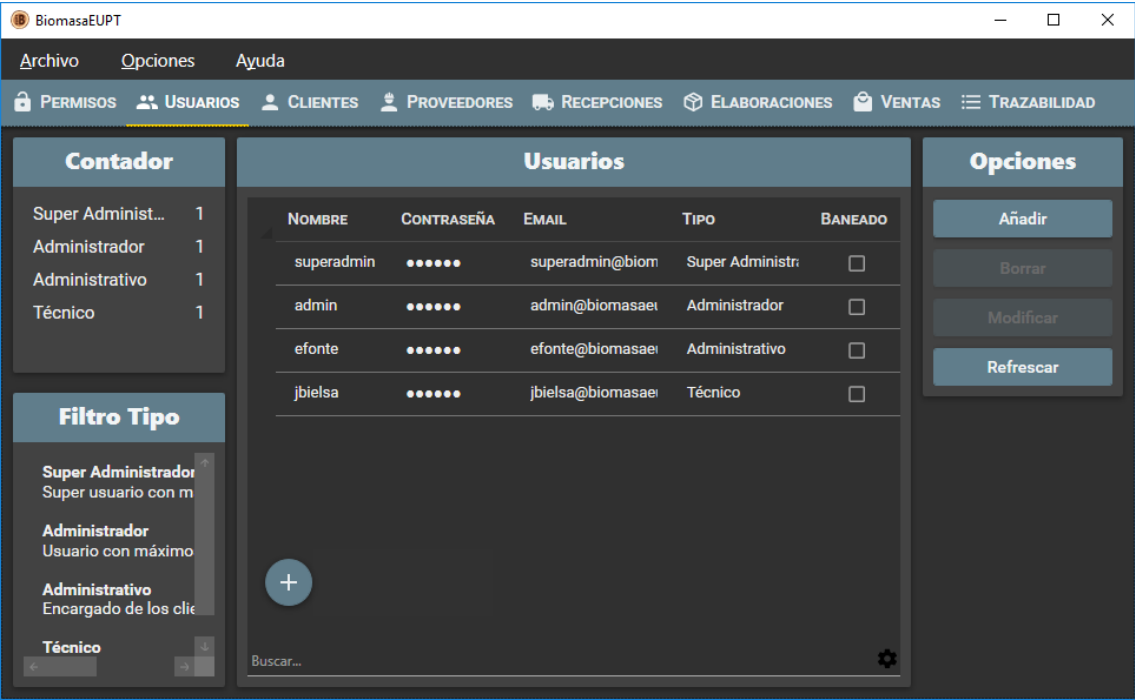


FIGURA G 56. Pestaña de gestión de usuarios con el modo nocturno activado.

