****

**Gestión de consumibles**

**Requisitos funcionales**

**Pedro Rodríguez Pérez**

Contenido

[1 Objetivo y finalidad. 3](#_Toc104978557)

[1.1 Características del proyecto. 3](#_Toc104978558)

[1.2 Detalle de Identidades de la Base de Datos 4](#_Toc104978559)

[1.2.1 vEmployees (Tabla de empleados) 4](#_Toc104978560)

[1.2.2 Providers 5](#_Toc104978561)

[1.2.3 ProductTypes 5](#_Toc104978562)

[1.2.4 Products 5](#_Toc104978563)

[1.2.5 ProductsProviders 5](#_Toc104978564)

[1.2.6 Stores 6](#_Toc104978565)

[1.2.7 StoreLocations 6](#_Toc104978566)

[1.2.8 Inventory 6](#_Toc104978567)

[1.2.9 InventoryProducts 7](#_Toc104978568)

[1.2.10 Regularizations 7](#_Toc104978569)

[1.2.11 RegularizationProducts 8](#_Toc104978570)

[1.3 Interfaces de Usuario. 9](#_Toc104978571)

[2.2.1 Pantalla Principal. 9](#_Toc104978572)

[2.2.2 Listado de almacenes 9](#_Toc104978573)

[2.2.3 Listado de productos y stock 10](#_Toc104978574)

[2.2.4 Listado de productos y proveedores 11](#_Toc104978575)

[2.2.5 Control de stock 12](#_Toc104978576)

[2.2.6 Inventario 13](#_Toc104978577)

[2.2.7 Regularización 14](#_Toc104978578)

[2.2.8 Traspaso entre almacenes 15](#_Toc104978579)

[2 Medios utilizados 16](#_Toc104978580)

[4 Requisitos funcionales de la aplicación 17](#_Toc104978581)

[4.1 Requisitos StoresManagement 21](#_Toc104978582)

[4.2 Requisitos ProductsManagement 23](#_Toc104978583)

[4.3 Requisitos TempProviders 26](#_Toc104978584)

[4.4 Requisitos StockControl 28](#_Toc104978585)

[5 Controladores 29](#_Toc104978586)

[5.1 Stores Management Controller 29](#_Toc104978587)

[5.2 Products Management Controller 35](#_Toc104978588)

[6 View Model y mapping 37](#_Toc104978589)

[7 Historial de mejoras del proyecto 38](#_Toc104978590)

[1.Main Menu 38](#_Toc104978591)

[2.Stores Management 38](#_Toc104978592)

[2.Products Management 43](#_Toc104978593)

[3.TempProviders 45](#_Toc104978594)

[4.Stock Control 45](#_Toc104978595)

# Objetivo y finalidad.

El objetivo de este desarrollo está en poder gestionar el control de consumibles en los almacenes que actualmente se encuentran en la empresa (Alter Technology), independientemente del área o lugar de trabajo en el que se encuentre dicho almacén.

Cuando se habla de almacenes, se trata de localizaciones concretas donde se guardan dichos consumibles, desde un armario, hasta una habitación o un cajón.

# Características del proyecto.

La aplicación que se pretende desarrollar constará de un formulario (Windows form) central donde se ubicarán todas las funcionalidades necesarias para el control de almacenes.

* Mantenimiento de almacenes.
* Mantenimiento de productos.
* Control de stock.
* Inventarios.
* Regularizaciones.
* Informes.

El funcionamiento general de la aplicación no seguirá un flujo concreto, sino que tendrá todas sus partes diferenciadas.

* Maestros: permitirá gestionar toda la información de las entidades maestras, como son fichas de productos y datos de almacenes.
  + Almacenes.
  + Productos.
  + Proveedores.
* Movimientos de almacén: permitirá gestionar toda la información relativa a los movimientos que se realizan en cada almacén como inventarios, salidas, entradas, revisión del stock, etc.
  + Control de stock.
  + Inventarios.
  + Regularizaciones.
  + Transferencia entre almacenes.

A continuación, se definen las entidades manejadas dentro de la aplicación, así como las interfaces de usuario de las que constará y los requisitos funcionales de cada uno de ellos.

Diagrama

Descripción generada automáticamente

## Detalle de Identidades de la Base de Datos

A continuación, se detallan cada una de las entidades, así como cada uno de sus campos.

Los campos que no admiten valores nulos se marcaran con cursiva.

Los campos que sean Primary Key se marcaran con negrita.

### vEmployees (Tabla de empleados)

Vista que maneja los empleados dentro de la gestión de consumibles.

Los campos manejados en esta entidad serán los siguientes:

* **EmployeeCode (PK):** corresponde al identificador de empleado.
* **EmployeeName:** Corresponde con el nombre del empleado.

### Providers

Entidad que manejas los proveedores dentro de la gestión de consumibles. En esta tabla tendremos los proveedores que tenemos registrados en SAP y también los proveedores que sean potenciales o posibles proveedores futuros.

Los campos manejados en esta entidad serán los siguientes:

* **ProviderCode (PK):** corresponde al identificador del proveedor.
* *ProviderName*: corresponde al nombre del proveedor.
* SAPProviderCode: corresponde al identificador en SAP.
* Address: corresponde con la dirección del proveedor.
* City*:* corresponde con la ciudad del proveedor.
* PostalZip*:* corresponde con el código postal del proveedor.
* CountryCode*:* corresponde con el código de país del proveedor.
* Telephone*:* corresponde con el teléfono del proveedor.
* Active*:* indica si el proveedor está activo o no.

### ProductTypes

Esta entidad contendrá los diferentes tipos de productos que se registrarán. Esta entidad no tendrá un mantenimiento “visual” sino que se mantendrá internamente desde la base de datos.

Los campos que serán necesarios para nuestro desarrollo serán los siguientes:

* **IdType (PK):** corresponde al identificador del tipo de producto.
* *TypeName*: corresponde al nombre del tipo de producto.

### Products

Entidad que maneja los productos o consumibles que se van a manejar dentro del laboratorio.

Los campos que serán necesarios en esta entidad serán los siguientes:

* **Reference (PK):** corresponde con el identificador del producto.
* *ProductName:* corresponde con el nombre del producto.
* IdType(FK): corresponde al identificador del tipo de producto. Se relaciona con la entidad ProductTypes por su campo IdType.
* Remarks: corresponde a un comentario del producto en general.
* UseExpirationDate: indica si el producto controla fechas de caducidad.
* UseLot: indica si el producto controla lotes.
* StockMin: corresponde al stock mínimo del producto.
* QuantityInUse: corresponde a la cantidad del producto que está en uso.
* WarningForMinStock: permite marcar un producto para aviso por stock mínimo.
* Dimensions: corresponde a la descripción de la dimensión del producto.

### ProductsProviders

Entidad que maneja los productos que están asociados a cada proveedor, así como información importante de dicha relación.

Los campos manejados en dicha entidad son los siguiente:

* ***Reference (PK)(FK):*** corresponde a la referencia del producto. Se relaciona con la entidad Products por su campo Reference.
* ***ProviderCode (PK)(FK):*** corresponde al código del proveedor. Se relaciona con la entidad Providers por su campo ProviderCode.
* Price: corresponde al precio del producto en ese proveedor.
* DeliveryDays: corresponde a los días de entrega de este producto por ese proveedor.
* DefaultProvider: indica si este proveedor es el proveedor predeterminado para este producto.

### Stores

Entidad que maneja los diferentes almacenes de los que se dispone para guardar los consumibles.

Los campos que serán necesarios en esta entidad serán los siguientes:

* **IdStore(PK):** corresponde al identificador que se le va a dar a cada almacén.
* StoreName: corresponde con el nombre del almacén.
* StoreDescription: corresponde con la descripción del almacén.
* *Owner(FK):* corresponde al empleado que es propietario o encargado del almacén. Se relaciona con la identidad vEmployees por su campo EmployeeCode.
* Area: corresponde al área donde se encuentra el almacén.
* StoreLocation: corresponde a la descripción del lugar donde se encuentra el almacén.
* Remarks: correspondiente con los comentarios asociados al almacén.

### StoreLocations

Entidad que maneja las diferentes ubicaciones o posiciones dentro de los almacenes de los que se dispone.

Los campos manejados en dicha entidad serán los siguientes:

* ***IdStore (PK)(FK):*** corresponde al identificador del almacén. Se relaciona con la entidad Stores por su campo IdStore.
* **LocationCode (PK):** corresponde al código o abreviatura de la ubicación dentro del almacén.
* LocationDescription: corresponde con una descripción de esta ubicación dentro del almacén.

### Inventory

Entidad que maneja los inventarios que se realizan sobre los diferentes almacenes.

Los campos manejados en dicha entidad son los siguientes:

* **IdInventory (PK):** corresponde al identificador del inventario.
* *IdStore (FK):* corresponde con el identificador del almacén en el que se hace el inventario. Se relaciona con la entidad Store por su campo IdStore.
* InventoryDate: corresponde a la fecha en la que se hace el inventario.
* *InventoryUser (FK):* corresponde al empleado que realiza el inventario. Se relaciona con la entidad vUsers por su campo UserId.
* Remarks: corresponde con un comentario asociado al inventario.

### InventoryProducts

Entidad que maneja los productos que se incluyen en un inventario.

Los campos manejados en dicha entidad son los siguientes:

* ***IdInventory (PK)(FK):*** corresponde con el identificador del inventario. Se relaciona con la entidad Inventory por su campo IdInventory.
* **Order (PK):** corresponde a la línea de producto dentro del inventario.
* *Reference (FK):* corresponde con la referencia del producto incluido en el inventario. Se relaciona con la entidad Products por su campo Reference.
* *IdStore (FK):* corresponde al almacén donde se está haciendo le inventario. Se relaciona con la entidad StoreLocation por su campo IdStore.
* *Location (FK):* corresponde a la ubicación dentro del almacén donde se ubica este producto. Se relaciona con la entidad StoreLocation por su campo LocationCode.
* Lot: corresponde al lote del producto incluido en la regularización. Este dato solo se incluirá en el caso de productos que controlen lotes.
* ExpiredDate: corresponde a la fecha de caducidad del producto incluido en la regularización. Este dato solo se informará en el caso de productos que controlen fecha de caducidad.
* Quantity: corresponde a la cantidad del producto que hay en el almacén.
* Remarks: corresponde con un comentario asociado al inventario.

### Regularizations

Entidad que maneja las regularizaciones (entradas o salidas) que se realizan sobre los diferentes almacenes.

Los campos manejados en dicha entidad son los siguientes:

* **IdRegularization (PK):** corresponde con el identificador del movimiento.
* *IdStore (FK):* corresponde con el identificador del almacén en el que se hace el inventario. Se relaciona con la entidad Store por su campo IdStore.
* TypeMovement: corresponde al tipo de movimiento que se hace:
  + R: regularización (Entrada y Salida de una vez).
  + T: traspaso entre almacenes (Salida de un almacén y entrada en otro).
* RegularizationDate: corresponde con la fecha en la que se hace la regularización.
* *RegularizationUser (FK):* corresponde al empleado que realiza la regularización. Se relaciona con la entidad vUsers por su campo UserId.
* Remarks: corresponde con un comentario asociado al inventario.

### RegularizationProducts

Entidad que maneja los productos que se incluyen en una regularización o movimiento de almacén.

Los campos manejados en dicha identidad son los siguientes:

* ***IdRegularization (PK)(FK):*** corresponde con el identificador de la regularización. Se relaciona con la entidad Regularizations por su campo IdRegularization.
* **Order (PK):** corresponde a la línea de producto dentro de la regularización.
* *Reference (FK):* corresponde con la referencia del producto incluido en la regularización. Se relaciona con la entidad Products por su campo Reference.
* *IdStore (FK):* corresponde al almacén donde se está haciendo la regularización. Se relaciona con la entidad StoreLocations por su campo IdStore.
* *Location (FK):* corresponde a la ubicación dentro del almacén donde se ubica este producto. Se relaciona con la entidad StoreLocation por su campo LocationCode.
* Lot: corresponde al lote del producto.
* ExpiredDate: corresponde a la fecha de caducidad del producto incluido en la regularización.
* QuantityIn: corresponde a la cantidad de producto que entra en el almacén, en la localización, y con la fecha y lote indicados.
* QuantityOut: corresponde a la cantidad de producto que entra en el almacén, en la ubicación, y con la fecha y lote indicados.
* UserForOut: corresponde al empleado al que se le entrega el producto, en el caso de que sea una salida de producto.
* Remarks: corresponde con un comentario asociado a la regularización.

## Interfaces de Usuario.

### 2.2.1 Pantalla Principal.

Forma

Descripción generada automáticamentePantalla principal: Nada más abrir la aplicación saldrá de la siguiente manera

En el menú lateral podremos clickar para obtener las siguientes funciones: [Menú Principal.](#_Características_del_proyecto.)

### 2.2.2 Listado de almacenes

En la primera opción del menú lateral tendremos Stores (Listado de almacenes).

Se ve de la siguiente manera.

Interfaz de usuario gráfica

Descripción generada automáticamente

Esta interfaz muestra al usuario un listado de los almacenes registrados en el sistema.

Un almacén puede ser desde un cajón hasta una habitación completa y dentro de un almacén puede haber infinitas ubicaciones, en función de cómo el responsable de dicho almacén quiere dividirlo física o virtualmente.

El usuario podrá filtrar el listado de almacenes que quiere consultar y seleccionar un almacén concreto para ver su información y las ubicaciones que tiene.

Desde esta interfaz el usuario podrá realizar las siguientes acciones:

* Filtrar el listado de almacenes.
* Editar la información de un almacén.
* Añadir almacenes nuevos.
* Eliminar un almacén (siempre que no haya información asociada)
* Añadir ubicaciones en un almacén.
* Eliminar ubicaciones de un almacén
* Ver el stock de un almacén concreto.
* Hacer un inventario nuevo.
* Hacer una regularización nueva.
* Hacer un nuevo traspaso entre almacenes

### 2.2.3 Listado de productos y stock

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEsta interfaz muestra al usuario un listado de los productos registrados en el sistema.

Desde aquí, el usuario podrá conocer el detalle de un producto y ver en detalle el stock de dicho producto en todos los almacenes de Alter.

Desde esta interfaz el usuario podrá realizar las siguientes acciones:

* Filtrar la lista de productos.
* Editar un producto
* Añadir un nuevo producto
* Eliminar un producto (si no tiene movimientos)
* Ver el stock de un producto en todos los almacenes

### 2.2.4 Listado de productos y proveedores

Esta interfaz muestra al usuario un listado de los productos registrados en el sistema.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Desde aquí, el usuario podrá conocer el detalle de un producto y ver en detalle el stock de dicho producto en todos los almacenes de Alter.

Desde esta interfaz el usuario podrá realizar las siguientes acciones:

* Filtrar la lista de productos.
* Editar un producto
* Añadir un nuevo producto
* Eliminar un producto (si no tiene movimientos)
* Ver los proveedores que suministran dicho producto
* Añadir una relación con un nuevo proveedor
* Editar la información de un proveedor para este producto
* Eliminar la relación del producto con el proveedor

### 2.2.5 Control de stock

Esta interfaz muestra al usuario el stock de un almacén.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para ello el usuario podrá seleccionar un almacén y el sistema mostrará el listado de productos disponibles en dicho almacén.

Por cada producto, el usuario podrá consultar las ubicaciones de dicho producto dentro del almacén, y podrá consultar la cantidad que hay en cada ubicación.

Desde esta pantalla el usuario también tendrá acceso directo para crear un nuevo inventario, una nueva regularización o un nuevo traspaso entre almacenes.

### 2.2.6 Inventario

Esta interfaz permitirá al usuario poder crear un nuevo inventario.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Un inventario no es más que una forma de inicializar las unidades de cada producto dentro de un almacén. Esto suele hacerse una vez al año como norma general, pero el responsable del almacén es quien decide cada cuanto tiempo se hace inventario en su almacén.

Al realizar un nuevo inventario, el usuario dispondrá de una serie de campos que identifican el inventario como son:

* Almacén sobre el que se realiza el inventario
* Fecha en la que se realiza
* Usuario que lo realiza
* Comentarios.

### 2.2.7 Regularización

Esta interfaz permitirá al usuario poder crear una nueva regularización.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una regularización es un movimiento en un almacén determinado. Este movimiento puede ser, salida de productos, entrada de producto, o darse ambas cosas.

Al realizar una nueva regularización, el usuario dispondrá de una serie de campos que la identifican como son:

* Almacén sobre el que se realiza
* Fecha en la que se realiza
* Usuario que lo realiza
* Comentarios.

### 2.2.8 Traspaso entre almacenes

Esta interfaz permitirá al usuario poder crear un nuevo traspaso entre dos almacenes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Un traspaso es un movimiento de entrada en un almacén determinado y un movimiento de salida en otro almacén.

Al realizar un nuevo traspaso, el usuario dispondrá de una serie de campos que lo identifican como son:

* Almacén sobre el que se realiza la salida
* Almacén sobre el que se realiza la entrada
* Fecha en la que se realiza
* Usuario que lo realiza

# Medios utilizados

Los medios utilizados para el proyecto son los siguientes:

* Software:
  + Entorno de desarrollo: Microsoft Visual Studio v.2022.
    - C1 component: aplicación de pago que añade nuevas características al cuadro de herramientas proporcionando al usuario más opciones de diseño.
    - Entity Framework: es un asignador de base de datos para .Net, permite al usuario interactuar con la base de datos como si de objetos se tratase.
    - Log4net: es una biblioteca de código abierto cuya función es permitir a .NET registrar la salida a una variedad de fuentes. Para mensajes.
    - AutoMapper: esta herramienta nos permite copiar datos de objeto a objeto.
    - LinqKit: para poder usar los predicados.
  + Base de datos: Microsoft SQL Server Management Studio: aplicación utilizada para la creación de la base de datos introducida posteriormente al proyecto en vsCommunity.
  + Lenguaje de programación: C#.
  + GreenShot: para hacer capturas de pantalla.
* Hardware: Pc (TOSHIBA).

# 4 Requisitos funcionales de la aplicación

Nada más abrir la aplicación el usuario dispondrá de diferentes opciones, las cuales son:

* Mantenimiento de almacenes.
* Mantenimiento de productos.
* Control de stock.
* Inventarios.
* Regularizaciones.
* Informes.

Antes de proseguir con los requisitos de cada apartado, para poder cargar los elementos de los formularios necesitamos una clase llamada “Extensions” cuya funcionalidad es poder cargar todos los elementos del formulario sin tener que repetir el código.

Hay que hacer esta clase ya que la herramienta ComponentOne no funciona como los elementos que vienen predeterminadamente en vsCommunity:

* Método dentro de la clase Extensions para poder cargar los combos (listas desplegables):Texto

  Descripción generada automáticamente

Estos dos métodos lo que hace es recorrer el formulario mirando cada elemento y cuando un elemento del formulario coincida con un C1Combo podrá mostrar los datos.

Si no se aplica estos dos métodos y estamos utilizando la herramienta component one al desplegar la lista aparecerán todos los datos de la tabla con los títulos de los campos incluidos.

* Método para limpiar todos los elementos: este método se utiliza para el botón Reset Filter (clear):Texto

  Descripción generada automáticamente

Texto

Descripción generada automáticamente

Lo que hace este método es recorrer el formulario entero y por cada tipo de control que haya en el formulario se limpiará depende del tipo de control que sea.

Una vez explicado la clase Extensions iremos por orden explicando las funcionalidades de cada apartado a lo largo del punto 4 dividiéndose en subpuntos por cada menú disponible que hay.

## 4.1 Requisitos StoresManagement

Al clickar el usuario en la primera opción titulada “Stores” dispondrá de la siguiente interfaz 🡪 [2.2.2 Listado de almacenes](#_2.2.2_Listado_de)

El usuario dispondrá en la parte superior los elementos necesarios para buscar un almacén, son textbox y comboBox en los que el usuario tendrá que meter los datos que quiere buscar. Cuando el usuario presione click en el botón Apply Filter (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos del almacén en la sección Master Test Records con sus datos en la parte derecha de esta misma sección. Si hay localizaciones en el almacén seleccionado aparecerán en la sección Locations for Selected Store.

Si el usuario presiona el botón Reset Filter se limpiarán todos los datos de la pantalla mostrados.

En la parte inferior de la interfaz están los siguientes botones:

* Edit: edita los datos del almacén.
  + Al presionar edit los demás botones estarán deshabilitados (incluidos el propio botón edit) y estará disponible la sección Selected Record Data (previamente deshabilitada, solo muestra los datos del almacén seleccionado) en el cual dispone de los datos del almacén y el usuario podrá cambiar los datos presionando el botón save o cancelar la edición presionando el botón cancel.
  + Al presionar el botón save o el botón cancel volverá a estar disponible el botón edit.
  + Texto

    Descripción generada automáticamenteCuando el usuario quiera editar o añadir un almacén el botón Code solo aceptará números ya que el código del almacén está integrado en la base de datos con tipo int.
* Add: añade un nuevo almacén.
  + El botón add estará disponible cuando el usuario no haya metido datos ni esté seleccionando un almacén.
  + Al presionarlo nos redirigirá a la sección Selected Record Data y permitirá al usuario meter los datos para así poder guardarlos o cancelarlos.
  + En este apartado los textbox en azul significa que son PK en la BD con lo cual es obligatorio rellenar esos campos si no saltará un mensaje de alerta indicando lo previamente mencionado.
  + Texto

    Descripción generada automáticamenteCuando el usuario quiera editar o añadir un almacén el botón Code solo aceptará números ya que el código del almacén está integrado en la base de datos con tipo int.
* Delete: elimina el almacén seleccionado.
  + El botón delete estará disponible cuando el usuario tenga seleccionado un almacén (solo los almacenes que no tengan localizaciones se podrán borrar, los demás no tendrán disponibles la opción de borrar).
  + Al presionar el botón delete el almacén seleccionado desaparecerá tanto en la lista como en la base de datos.
  + Hemos añadido una función en la que si tenemos seleccionado un almacén presionando la tecla supr del teclado directamente borra el almacén tanto de la bd como de la lista.
* Show Stock: muestra el stock
  + No implementado aún.
* New Inventory: añade un nuevo inventario.
  + No implementado aún.
* New Regularization: añade una nueva regularización.
  + No implementado aún.

## 4.2 Requisitos ProductsManagement

Al clickar el usuario en la segunda opción del menú lateral “Products” redirigirá al usuario a la siguiente interfaz 🡪 [2.2.3 Listado de productos y stock](#_2.2.3_Listado_de)

El usuario dispondrá en la parte superior los elementos necesarios para buscar un producto, elementos como textbox y comboBox en los que el usuario tendrá que introducir los datos que quiere buscar. Cuando el usuario presione el botón (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos del producto en la sección “Product Records” con sus datos en la sección titulada “Selected Product Data”. Si el producto tiene relación con algún proveedor aparecerá en la lista de la sección “Products Providers”.

Si el usuario presiona el botón Reset Filter se limpiarán todos los datos de la pantalla mostrados.

En la parte inferior de la interfaz están los siguientes botones:

* Edit: edita los datos del almacén.
  + Al presionar edit los demás botones estarán deshabilitados (incluidos el propio botón edit) y estará disponible la sección Selected Product Data (previamente deshabilitada, solo muestra los datos del producto seleccionado) en el cual dispone de los datos del producto y el usuario podrá cambiar los datos presionando el botón save o cancelar la edición presionando el botón cancel.
  + Al presionar el botón save o el botón cancel volverá a estar disponible el botón edit.
* Add: añade un nuevo producto.
  + El botón add estará disponible cuando el usuario no haya metido datos ni esté seleccionando un producto.
  + Al presionarlo nos redirigirá a la sección Selected Product Data y permitirá al usuario meter los datos para así poder guardarlos o cancelarlos.
  + En este apartado los textbox en azul significa que son PK en la BD con lo cual es obligatorio rellenar esos campos si no saltará un mensaje de alerta indicando lo previamente mencionado.
  + En el texto para añadir Stock el usuario podrá meter solo números a raíz de un método titulado OnlyFloats:Captura de pantalla de un celular

    Descripción generada automáticamente
* Delete: elimina el producto seleccionado.
  + El botón delete estará disponible cuando el usuario tenga seleccionado un producto.
  + Al presionar el botón delete el almacén seleccionado desaparecerá tanto en la lista como en la base de datos.
  + Hemos añadido una función en la que si tenemos seleccionado un producto presionando la tecla supr del teclado directamente borra el producto tanto de la bd como de la lista.
* Add Provider: la función de este botón es añadir un proveedor a un producto al clickar sobre el botón add de la pestaña Providers, redirigirá al usuario a otra ventana con la lista de los proveedores disponibles.Interfaz de usuario gráfica, Aplicación, Word

  Descripción generada automáticamente

La ventana de proveedores se ve de la siguiente manera:Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

## 4.3 Requisitos TempProviders

Al clickar el usuario en la segunda opción del menú lateral “Providers” redirigirá al usuario a la siguiente interfaz 🡪 [2.2.4 Listado de productos y proveedores](#_2.2.4_Listado_de)

El usuario dispondrá en la parte superior los elementos necesarios para buscar un proveedor, elementos como textbox y comboBox en los que el usuario tendrá que introducir los datos que quiere buscar. Cuando el usuario presione el botón (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos del proveedor en la sección “Provider Records” con sus datos en la sección titulada “Selected Provider Data”. Si el proveedor tiene relación con algún producto aparecerá en la lista de la sección “Product and Providers Records”.

Si el usuario presiona el botón Reset Filter se limpiarán todos los datos de la pantalla mostrados.

En la parte inferior de la interfaz están los siguientes botones:

* Edit: edita los datos del almacén.
  + Al presionar edit los demás botones estarán deshabilitados (incluidos el propio botón edit) y estará disponible la sección Selected Product Data (previamente deshabilitada, solo muestra los datos del proveedor seleccionado) en el cual dispone de los datos del proveedor y el usuario podrá cambiar los datos presionando el botón save o cancelar la edición presionando el botón cancel.
  + Al presionar el botón save o el botón cancel volverá a estar disponible el botón edit.
* Add: añade un nuevo proveedor.
  + El botón add estará disponible cuando el usuario no haya metido datos ni esté seleccionando un proveedor.
  + Al presionarlo nos redirigirá a la sección Selected Product Data y permitirá al usuario meter los datos para así poder guardarlos o cancelarlos.
  + En este apartado los textbox en azul significa que son PK en la BD con lo cual es obligatorio rellenar esos campos si no saltará un mensaje de alerta indicando lo previamente mencionado.
* Delete: elimina el proveedor seleccionado.
  + El botón delete estará disponible cuando el usuario tenga seleccionado un proveedor.
  + Al presionar el botón delete el almacén seleccionado desaparecerá tanto en la lista como en la base de datos.
  + Hemos añadido una función en la que si tenemos seleccionado un proveedor presionando la tecla supr del teclado directamente borra el proveedor tanto de la bd como de la lista.
* Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamenteAdd Products: La función de este botón es añadir un producto a un proveedor al clickar sobre el botón add de la pestaña Products, redirigirá al usuario a otra ventana con la lista de los productos disponibles.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteAl presionar el botón add redirigirá al usuario a la siguiente ventana indicando los productos que puede seleccionar:

## 4.4 Requisitos StockControl

Al clickar el usuario en la opción del menú lateral “Stock Control” redirigirá al usuario a la siguiente interfaz 🡪 [2.2.5 Control de stock](#_2.2.5_Control_de)

El usuario tendrá una lista desplegable en la que podrá elegir el almacén que desea seleccionar, al pulsar sobre buscar aparecerán en la lista de abajo los productos que están almacenados en ese almacén seleccionad y justo abajo en la otra lista aparecerán las localizaciones (si las hay) de esos almacenes.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

El usuario dispondrá en la parte superior los elementos necesarios para buscar un proveedor, elementos como textbox y comboBox en los que el usuario tendrá que introducir los datos que quiere buscar. Cuando el usuario presione el botón (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos del producto en la sección “Products”. En la lista de abajo deberán aparecer las localizaciones del producto.

Si el usuario presiona el botón Reset Filter se limpiarán todos los datos de la pantalla mostrados.

## 4.3 Requisitos Inventory

Al clickar el usuario en la opción del menú lateral “Inventory” redirigirá al usuario a la siguiente interfaz 🡪 [2.2.6 Inventario](#_2.2.6_Inventario)

El usuario dispondrá en la parte superior los elementos necesarios para buscar un inventario, elementos como textbox y comboBox en los que el usuario tendrá que introducir los datos que quiere buscar. Cuando el usuario presione el botón (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos del inventario en la lista “Inventory”. Si el inventario tiene productos aparecerán en la lista de abajo titulada “Products”.

Si el usuario presiona el botón Reset Filter se limpiarán todos los datos de la pantalla mostrados.

## 4.4 Requisitos Regularizations

Al clickar el usuario en la opción del menú lateral “Regularizations” redirigirá al usuario a la siguiente interfaz 🡪 [2.2.7 Regularización](#_2.2.7_Regularización)

El usuario dispondrá en la parte superior los elementos necesarios para buscar un inventario, elementos como textbox y comboBox, Date en los que el usuario tendrá que introducir los datos que quiere buscar. Cuando el usuario presione el botón (hemos añadido la opción de que al presionar la tecla enter pueda directamente aplicar la función del botón Apply Filter sin tener que buscar el botón con lo cual ahorra tiempo de búsqueda), aparecerán los datos de la regularización en la lista “Products”, al aparecer los productos del almacén (puede no aparecer ninguno) en la lista de productos elegimos el producto que queremos dentro o fuera del almacén.

# 5 Controladores

La aplicación necesita un intermediario entre vsCommunity y la base de datos, aquí es donde entra en juego los controladores creados a partir de la clase titulada StoreStockControlTool.Controllers.

La principal función de esta clase es conectarse a los modelos que a su vez tienen una infraestructura de clases para conectarse a la base de datos mediante un archivo .edmx.

## 5.1 Stores Management Controller

Esta clase es el controlador de la interfaz 🡪 [2.2.2 Listado de almacenes](#_2.2.2_Listado_de)

Para poder acceder a los controladores desde el formulario de StoresManagement.cs, utilizaremos el método FilterPredicate cuya función es comprobar si algo es cierto en un determinado x objeto. Este método está en el código del formulario StoresManagementTool.

Texto

Descripción generada automáticamentese utilizará para el botón Apply Filter y realizar la búsqueda comprobando que los datos que se buscan concuerdan con los datos que existen en la base de datos.

También se utiliza para cargar los comboBox del formulario que están en el Controller con el método GetAllOwners y pasando por parámetro el predicado.

Captura de pantalla de un celular con texto e imagen

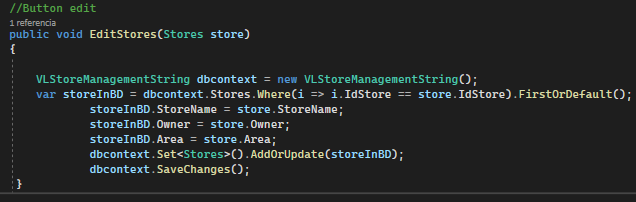
Descripción generada automáticamenteLo que hace es cargar todos los empleados de la base de datos, dependiendo de como lo queramos cargar se utilizará de una forma u otra, por ejemplo: si lo queremos cargar en un FlexGrid (en este proyecto utilizaremos C1FlexGrid por el componente de pago C1Component) utilizaremos la siguiente sentencia 🡪 FlexGrid.DataSource = Controller.GetAllOwners(predicate);

Métodos que hay dentro de StoresManagementController:

* GetAllAreas: la función de este método es cargar las áreas de la base de datos en el c1combo áreas de la interfaz.Texto

  Descripción generada automáticamente
* GetWithoutenter: la función de este método es cargar todos los almacenes cuando el usuario no haya metido ningún dato en el apartado de búsqueda.Texto

  Descripción generada automáticamente
* Texto

  Descripción generada automáticamenteGetStoreLocations: la función de este método es cargar los almacenes que tengan localizaciones (que estén relacionados en la base de datos), con esto lo que haremos posteriormente es mostrar dichas localizaciones en el FlexGrid de la parte inferior de la interfaz.
* EditStores: la función de este método es permitir al usuario editar un almacén existente. Guardando los datos que hay metido el usuario en una variable tipo var y después igualando dicha variable al parámetro del método, así el parámetro tiene el valor de la variable que ha introducido el usuario.

Captura de pantalla de un celular

Descripción generada automáticamenteLlamando al método desde el código del formulario quedaría así:

* Texto

  Descripción generada automáticamenteAddStores: la función de este método es permitir al usuario añadir un nuevo almacén introduciendo los datos, más simple que el método editar ya que solo introduce los datos y no los compara todos, solo se comparará la Pk de esta tabla(IdStore).

Texto

Descripción generada automáticamenteLlamando al método desde el código del formulario quedaría así:

Comparamos los controles que hay en la sección para editar y añadir almacenes y comprobamos que no estén vacíos, si cumplen los requisitos puede añadirse en la base de datos.

En este método utilizamos otro llamado CompareStores cuya función es comparar la Primary key de la tabla para que el usuario no pueda meter más de una vez el mismo código de almacén.

Procedemos a explicar el método CompareStores.

* CompareStores: comparamos las Primary keys y utilizamos una variable de tipo bool para saber si coinciden o no, en el caso en el que la variable var de null (significa que el código no está repetido en la base de datos). Pondremos la variable bool a false y damos permiso para utilizar AddOrUpdate(stores), si la variable var es distinto a null significa que ya existe la Primary key de la tabla con lo cual pondremos la variable de tipo bool a true. Retornamos la variable de tipo bool y según lo que devuelva pues el usuario podrá añadir el almacén o no.Texto

  Descripción generada automáticamente
* Captura de pantalla de computadora

  Descripción generada automáticamenteDeleteStores: la función de este método es borrar un almacén que esté seleccionado.

Texto

Descripción generada automáticamenteEn el código del formulario preguntamos al usuario si está seguro de borrar el almacén seleccionado y dependiendo de la respuesta del usuario lo borrará o no.

## 5.2 Products Management Controller

Esta clase es el controlador de la interfaz 🡪 [2.2.3 Listado de productos y stock](#_2.2.3_Listado_de)

Para poder acceder a los controladores desde el formulario de ProductsManagement.cs, utilizaremos el método FilterPredicate.

* Texto

  Descripción generada automáticamenteMétodos para cargar los comboBox (lista desplegable):
* Método para añadir productos:Texto

  Descripción generada automáticamente
* Método para editar productos:Texto

  Descripción generada automáticamente
* Método para eliminar productos:Texto

  Descripción generada automáticamente
* Método para añadir a la lista sin seleccionar nada del formulario: Texto

  Descripción generada automáticamente

## 5.3 Providers Controller

Esta clase es el controlador de la interfaz 🡪 [2.2.4 Listado de productos y proveedores](#_2.2.4_Listado_de)

Para poder acceder a los controladores desde el formulario de Provider.cs, utilizaremos el método FilterPredicate.

Tiene los mismos métodos que las clases anteriores. Pero editando su tipo de objeto.

## 5.4 Stock Control Controller

Esta clase es el controlador de la interfaz 🡪[2.2.5 Control de stock](#_2.2.5_Control_de)

Para acceder al controlador de esta clase es necesario el método Filter predicate como en las demás clases anteriores.

Tiene los mismos métodos que las clases anteriores. Pero editando su tipo de objeto.

Pero esta clase tiene un método extra que los demás no tienen es utilizado para poder aplicar las vistas de la base de datos ya que requiere una consulta de más de una tabla:

Texto

Descripción generada automáticamenteControlador para consultar los productos que hay en el almacén.

# 6 Vistas.

Para poder consultar algunos datos de distintas tablas de la base de datos necesitaremos el uso de vistas en la base de datos.

Ejemplo de vista en nuestra bd:Interfaz de usuario gráfica

Descripción generada automáticamente

Esta vista nos sirve para poder consultar los inventarios y sus productos.

# 7 Historial de mejoras del proyecto

## 1.Main Menu

En el formulario principal donde está el menú lateral con todas las opciones que se puede elegir, al reutilizar el menú con otra pantalla en primer plano no hacía nada. Solo hemos añadido el método focus justo después de detectar que el formulario ya se ha abierto.Texto

Descripción generada automáticamente

Al añadir la sentencia formChild.Focus(), al pulsar nuevamente el menú se pondrá en primer plano.

## 2.Stores Management

Lista de mejoras a realizar:

* ~~Los TextBox no tienes color predeterminado gris~~
  + Cambiando la propiedad Back color a 232; 234; 236
* ~~Owner no tiene color predeterminado gris~~
  + Cambiando la propiedad Back color a 232; 234; 236
* ~~Al presionar buscar, editar el almacén y al presionar cancelar no se limpia el cfgLocations (id store 2,3) el 1 si lo limpia. Al presionarlo 2 veces si lo limpia.~~
  + Cambiando el orden de los controles, cfgLocations tenia un nivel más de prioridad que cfgStores (cfgStores debe tener un nivel más de prioridad respecto a cfgLocations).
* ~~Ordenar los tabindex~~
  + Cambiando la propiedad tabindex según el orden en que se quiera tener las tabulaciones.
* ~~Area muestra el resultado en mayúscula~~
  + Cambiar la propiedad del comboBox (CharacterCasing) a normal.
* ~~Owner muestra el resultado en mayúscula~~
  + Cambiar la propiedad del comboBox (CharacterCasing) a normal.

~~Al iniciar el programa los botones de la parte inferior están habilitados, el único que lo tiene que estar es el botón add.~~

* + En el método InitializeFields() introducimos button.Enabled = false; para deshabilitarlos nada más iniciar la aplicación

Texto

Descripción generada automáticamente

* ~~Botón Delete deshabilitado cuando la lista de almacenes carga los almacenes.~~
  + En el método cfgStores\_RowColChange indicamos que el botón Delete esté habilitado (solo cuando se cuente + de 0 filas).

Texto

Descripción generada automáticamente

* ~~Si se mete el mismo pk en add salta error~~
  + Creando un nuevo controlador que pruebe la pk que se crea con las que hay en la base de datos.

Texto

Descripción generada automáticamente

* ~~Si no se rellena el combo área da error, aunque no sea pk~~
  + Añadiendo un if diciendo si es null que siga
* ~~En cfgStores si presionas supr te da el mensaje aún tenga Locations (quitar mensaje si tiene locations)~~
  + Texto

    Descripción generada automáticamenteEn el método cfgStores\_KeyUp\_1(object sender, KeyEventArgs e) decimos si el cfg tiene más de una fila no podemos borrar y en el else permitimos al botón hacer su función.
* ~~Al presionar el botón Reset Filter el botón delete no está deshabilitado.~~
  + Captura de pantalla de un celular

    Descripción generada automáticamenteEn el método del botón Reset Filter tipo click indicamos que el botón delete esté deshabilitado al pulsar click sobre el botón Reset Filter.

## 3.Products Management

Listas de mejoras a realizar:

* ~~Al añadir un producto nuevo no deja escribir en Reference.~~
  + Al presionar el botón editar Reference está con la propiedad ReadOnly true, para que no se pueda editar la pk. Con lo cual hay que poner esa propiedad a false cuando presionamos el botón add, ya que ahí el usuario tiene que introducir una pk que no sea repetida en la base de datos.
* ~~Poner error cuando las pk coinciden.~~
  + Creando método compareReference en el controller
* ~~Al presionar edit no deja escribir en Reference cuando queremos añadir un producto.~~
* ~~Al presionar save (editar), sale error productsInBD fue null, solo pasa en la parte reference. Porque es primary key~~
* ~~Al editar el producto en el campo Type (que es fk) si se pone un valor que no pertenece a la base de datos no sale un mensaje que indique al usuario que ese valor no es válido.~~
  + Interfaz de usuario gráfica, Texto

    Descripción generada automáticamenteCreamos un método que compare el idtype de la base de datos con la que mete el usuario en el campo type, si devuelve falso significa que no está en la bd con lo cual no existe y si da true se prosigue editando el producto.
  + Captura de pantalla de un celular

    Descripción generada automáticamenteMétodo para comparar los idtype.
* ~~Al añadir un producto y meter un Type que no existe da error.~~
  + Metiendo en orden los métodos para comparar los campos.
* ~~Al añadir un producto y meter el mismo reference lo deja añadir cuando no debería ya que es pk.~~
  + Metiendo en orden los métodos para comparar los campos.
* Al intentar relacionar un producto con un proveedor no lo hace. Es un tipo de relación nm (muchos a muchos). Hay que redireccionar un formulario con otro y a raíz de ese botón relacionar datos de 2 tablas a 1 que está en común con las pk.
* Al eliminar component one no deja averiguar por que

## 4.TempProviders

Listas de mejoras a realizar:

* ~~Cuando se presiona el botón Add hay que deshabilitar el panel superior izquierdo de la interfaz, para que no se puedan mezclar datos si el usuario manipula dicha parte.~~
  + Añadiendo una propiedad en el método click del botón Add, SubPanelLeft.Enabled = false; lo que hacemos es deshabilitarlo al pulsar el botón Add.
* Añadir tecla supr al listado de proveedores.
  + Texto

    Descripción generada automáticamenteEn el evento Keyup de la lista de proveedores añadimos las siguientes sentencias para que al presionar supr haga la función del botón Delete.

## 5.Stock Control

* ~~Los FlexGrid cargan al pulsar el botón Search los datos de la tabla RegularizationsProducts en vez de cargar la tabla products y la tabla locations.~~