Proyecto producción

TFG Lucian potcoava

Lucian | Potcoava | 02/06/2022

## Introducción

Mi proyecto es un ERP que facilita la fabricación y la gestión de productos. Controlar estas acciones suelen ser tareas muy laboriosas y que requieren una cantidad de papeles muy grandes, mientrás que una ERP agilizaría este proceso y a su vez, evitaria cantidades tan grandes de papel.

## Antecedentes del proyecto

Hoy en día hay muchas empresas que se dedican a la producción, algunas se han adaptado a los tiempos mientras que otras se han quedado con tecnologías obsoletas. Estas tecnologías les esta reduciendo la cantidad de ingresos, ralentizando el proceso de producción y además son mas propensas al fallo humano.

Las principales utilidades de mi proyecto son la siguientes:

* Permite poner a producir un producto, teniendo en cuenta todos los factores necesarios.
* Gestiona las materias primas dando la posibilidad de hacer pedidos.
* Mantenimiento de los articulos, incluyendo el tiempo que tienen de fabricación y las materias primas que lo componen.
* Guarda todo el historico de todas las producciones y compras de materia prima.

### ¿Estas funcionalidades que ventajas pueden ofrecer frente a una producción tradicional?

-Producción de un artículo: Gracias a esto le permite al empleado saber si un producto se puede o no fabricar. Por lo habitual debe de consultar la materia prima que compone al artículo y a su vez el stock del que se dispone de estos.

También tiene que verificar si hay algún pedido hecho de alguna de las materias primas, porque a lo mejor no hay stock en este momento, pero hay un pedido en camino. Un error humano aquí podría derivar en comprar más materia prima de la que se necesita.

-Guarda todos los registros: El empleado puede ver todas las fabricaciones que se han hecho y en qué fecha/hora. En caso de una incidencia esta información puede resultar muy útil para averiguar qué ha pasado.

También es útil para sacar informes, que producto ha sido el que más hemos fabricado, cual es la materia prima que más hemos necesitado etc.

A su vez, esto permite comprobar la eficiencia de las maquinas, ya que guarda el tiempo real de la producción y se puede comparar con el estimado. Por lo habitual es que sean distintos por un margen no muy grande, si vemos que la producción de un producto es mucho más grande que la estimada podemos deducir que hay un fallo.

Esto nos permite controlar el mantenimiento de las máquinas y detectar un problema con antelación evitando en que derive en algo más grande.

-Mantenimiento de los artículos y le das materias primas: Mi proyecto permite consultar con facilidad la composición de los artículos e información adicional sobre estos, como puede ser su stock, el precio, el tiempo estimado de fabricación, etc.

-Hacer pedidos de las materias primas: En caso de querer fabricar algo y no disponer de las materias primas suficientes o en caso de querer aumentar el stock de una materia prima, puedes hacer un pedido. Informa del stock real y del que está pedido que aún no ha llegado.

-Aceptar pedidos: si un pedido de materia prima llega, esta función registrará en el histórico el pedido como completado, modificará el stock de la materia prima correspondiente y dará el pedido como finalizado. Esto permite controlar el stock de manera inmediata y saber la información a tiempo real.

Podría darse el caso en donde esta inmediatez pueda ser crucial en la venta de un producto. De manera tradicional, si el que ha recibido el pedido no informa, informa tarde por carga de trabajo o no cuenta bien el stock recibido, las personas responsables de la venta verían que no hay stock suficiente para hacer un producto. Nos quitamos un posible fallo de comunicación entre la persona que recibe el pedido y el que manda la orden de fabricación.

Objetivos del proyecto

El objetivo de mi programa es automatizar procesos de producción, agilizarlos, tener un control más grande de las producciones, reducir la cantidad de papel, reducir y facilitar la mano de obra.

Tener un control del stock sin tecnología es una tarea laboriosa que involucra grandes cantidades de papel y con facilidad de fallo humano. La implementación de mi programa no significa que desaparezca el control manual del stock, es recomendable que como mínimo una vez al año se haga el control de manera manual.

Pero estaríamos reduciendo la cantidad del control manual de manera considerable. Además, nos ahorramos todo el control de los papeles del almacén, espacio para almacenar muchos papeles y evitar la tarea de buscar algo entre muchos cajones y muchas carpetas.

En caso de que se perdiese los papeles del almacén, tendríamos que invertir otra vez muchas horas en contar el almacén en vez de enfocarse en la producción. Mi programa usa un servidor de datos SQL que permite hacer copias de seguridad, se hacen copias a la noche, gracias a esto es mucho más complicado perder los datos comparándolo con el papel.

No olvidemos la importancia de reducir la cantidad de papel usado no solo para evitarse la gran cantidad de sitio que se necesita para almacenarla si no que para proteger el medio ambiente. Según un artículo publicado en “Nature” se cortan alrededor de 15 billones de árboles cada año, si todas las empresas empezasen a usar programas de este estilo, se reduciría esa cantidad de manera inmensa.

Recursos necesarios

El proyecto va está dividido en dos partes:

-Aplicación web:

La parte web utiliza el entorno de visual estudio para ser desarrollada. Es una aplicación web de ASP.NET con .NET framework. .NET nos proporciona un conjunto de herramientas que nos facilita la conexión con las bases de datos. Las peticiones se hacen con AJAX y se llama a servicios web .asmx.

¿Por qué usar tecnología AJAX?

La tecnología AJAX nos permite actualizar las páginas de manera asíncrona cambiando pequeñas partes de información con el servidor en paralelo al hilo principal. Gracias a esto nos permite actualizar parte de la página sin la necesidad de actualizar toda la página.

Las páginas web clásicas tenían que hacer una recarga cada vez que hacemos un cambio, esto es mucho menos ineficiente que las peticiones AJAX. No solo es más molesto para el cliente, sino que además es ineficiente ya que se usa más bando de ancha, aumenta el tráfico de la red y ralentiza la velocidad de las peticiones.

¿Por qué usar servicios web ASMX?

ASMX es una extensión de asp.net desarrollado por Microsoft al igual que el framework que estamos utilizando y el proyecto. Los servicios ASMX nos da la posibilidad de comunicación entre dos objetos de internet usando SOAP. El servicio web procesa las peticiones y nos da una respuesta.

SOAP permite a los desarrolladores invocar servicios web y recibir respuestas independientemente del lenguaje y las plataformas ya que se comunica utilizando un lenguaje de marcado.

Las peticiones de los servicios web se hacen a un servidor de datos SQL. SQL nos permite almacenar y dar información de las bases de datos. Es muy flexible, potente, rápida y asequible para las empresas. Tiene muchas utilidades, no solo almacena la información en tablas. En mi programa por ejemplo tienen mucho peso los procedimientos almacenados.

-Escritorio:

En la parte de escritorio también se usa .NET framework por la misma razón que he explicado antes para la aplicación web. En el escritorio tiene mucho peso lo que se hace a nivel de cliente para evitar llamadas innecesarias al servidor de datos. Se trabaja mucho a nivel local.

Hay muchas veces que se cargan Data Table por primera vez y luego la información se trata recorriendo con bucles las filas de ellos. En vez de poner un parámetro en la where y cada vez que ocurra un evento cambiar el valor del parámetro y hacer la petición. De esta manera reducimos tanto la cantidad de peticiones al servidor de datos tanto como las veces que abrimos y cerramos conexión.

En caso de que estos programas se ejecuten de muchas maquinas distintas, todos acceden a la misma base de datos, así evitamos que se sobrecargue el servidor de datos y mantenemos el servidor lo más ágil posible.

Fase de análisis

-Web: En la aplicación web, el trabajador debe de iniciar sesión, Una vez dentro, tiene dos opciones, encargarse del proceso de producción o el mantenimiento del producto.

El proceso de producción tendrá una lista con todos los productos, podrá verificar si el objeto se puede fabricar. Gracias a esta fabricación puede ver la materia prima que se va a emplear en la producción para evitar errores. Salen todas las materias primas que componen el producto y el stock del que se dispone de esas materias.

Por ejemplo: le damos a verificar en una herramienta como un pico, y nos sale que necesitamos madera y hierro. Nos sale un mensaje similar a este “madera: 2/99(12)” y” hierro: 3/50(0)”.

Como podemos ver nos sale las dos materias primas que lo componen en este caso, nos sale su nombre, cuanto necesitamos para la producción de un pico, cuanto tenemos en el stock y entre paréntesis el stock próximo.

Es decir, stock que aún no tenemos, pero tendremos, por ejemplo, en la madera entre paréntesis tenemos 12 lo que significa que hay un o varios pedidos que suman la cantidad de 12 pero esos pedidos aún no han llegado y no están en el almacén.

En la parte de mantenimiento, nos permite buscar por ID el producto, si lo encuentra nos da toda la información del producto. Tanto la información del producto como información de la categoría a la que pertenece, el tiempo que cuesta producirlo y las materias primas que lo componen.

En la aplicación de escritorio tenemos varias opciones, tenemos la posibilidad de hacer pedidos de las materias primas. Filtramos por categoría los productos y haciendo doble clic en el producto nos muestra las materias primas que lo componen.

Al hacer clic en el producto nos dice de que materias primas este compuesto. Al pulsar en la materia prima que queremos hacer el pedido, nos pedirá una cantidad y se agregará al carrito. Cuando ya tengamos en el carrito todo lo que queramos pedir realizará el pedido y estará en estado no finalizado, es decir está pedido, pero el stock aún no está en el almacén.

El siguiente botón nos permite ver todos los pedidos, tanto los que están finalizados como los que no lo están. Al pulsar en los pedidos nos sale información más detallada de estos. En caso de no estar finalizado nos da la posibilidad de finalizarlo. Esto le cambiara el estado y nos restara esas unidades al stock próximo y se las agrega al stock.

La siguiente opción nos permite modificar, borrar e insertar categorías. Permite al trabajador hacer un mantenimiento de las categorías de los productos. La siguiente opción es similar a la anterior, pero en vez de categorías es de materias primas.

Las siguientes dos opciones nos permite controlar los logs de las producciones y de las materias primas. Es importante tener registrado todo lo que sucede, para evitar situaciones que puedan ser prejudiciales a la empresa. Podemos detectar que algo no está funcionando bien con rapidez ya que el log nos informara de lo que ha sucedido realmente.

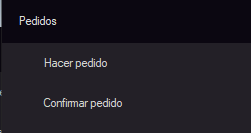
La primera opción dispone de los logs de las fabricaciones, al pulsar en uno podemos ver que producto se ha fabricado con más detalle y las materias primas que se han utilizado para el proceso. Además, abajo nos sale el tiempo que se ha utilizado para la fabricación. Si vemos que el tiempo real es mucho mayor que el estimado, podemos ver una anomalía y a lo mejor la máquina que lo produce necesita un mantenimiento.

Fase de diseño

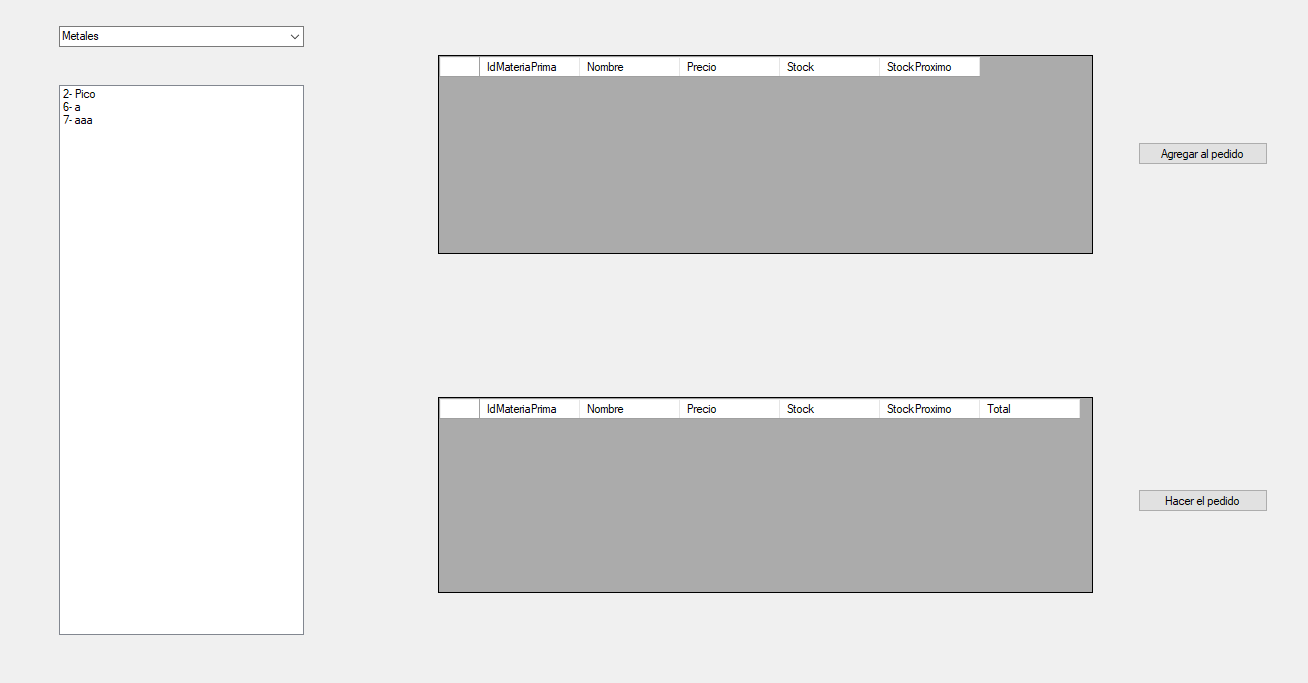
-Escritorio:



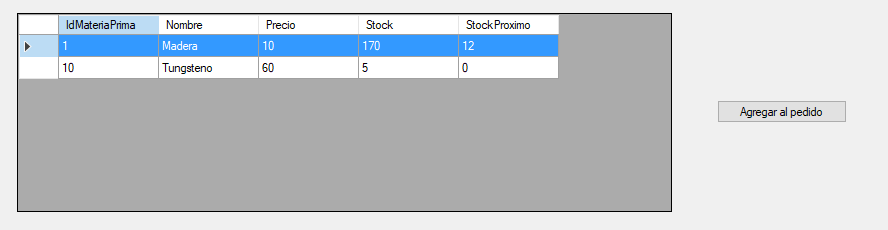
Tenemos un menú lateral en la izquierda, si pulsamos sobre cualquier opción se nos desplegara un menú con mas opciones.



En pedidos tenemos dos opciones, hacer pedido o confirmar un pedido. Hacer el pedido no agrega directamente la cantidad al stock, lo pedido se agregará a un campo que se llama stock próximo y hasta que no se confirme el pedido no se agregará al inventario. A la hora de la producción el stock próximo no se tiene en cuenta, es informativo.

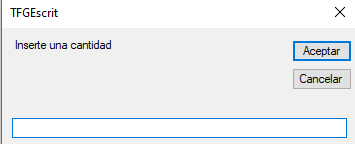


En esta pestaña podemos ver varias cosas, vamos a empezar por la izquierda. En la parte superior izquierda tenemos un combo box con las categorías de los productos. Podemos filtrar los productos que salen debajo en la lista, si pulsamos sobre un producto nos dirá de que materias primas esta compuesto.

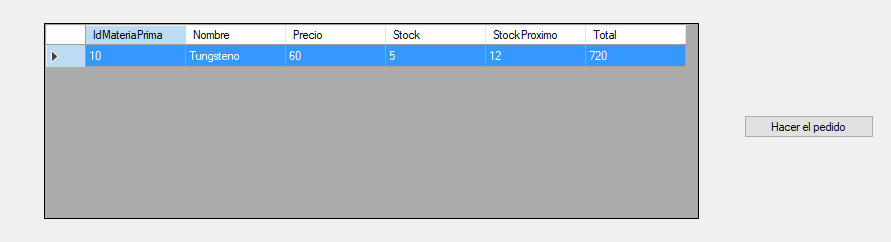


Por ejemplo al pulsar sobre pico, se carga el grid de arriba con esta información. Podemos ver que ese producto de la categoría de metales se compone de madera y tungsteno. En caso de querer pedir alguna materia prima de las que se nos presenta, tenemos que seleccionar la materia prima y nos saldrá con ese color azul la fila como se ve en la captura.

Posteriormente le daremos al botón de la derecha del grid que pone “agregar pedido”.



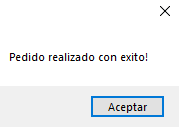
Nos saldrá este cuadro solicitando una cantidad. Introducimos la cantidad que queremos pedir y le damos a aceptar.



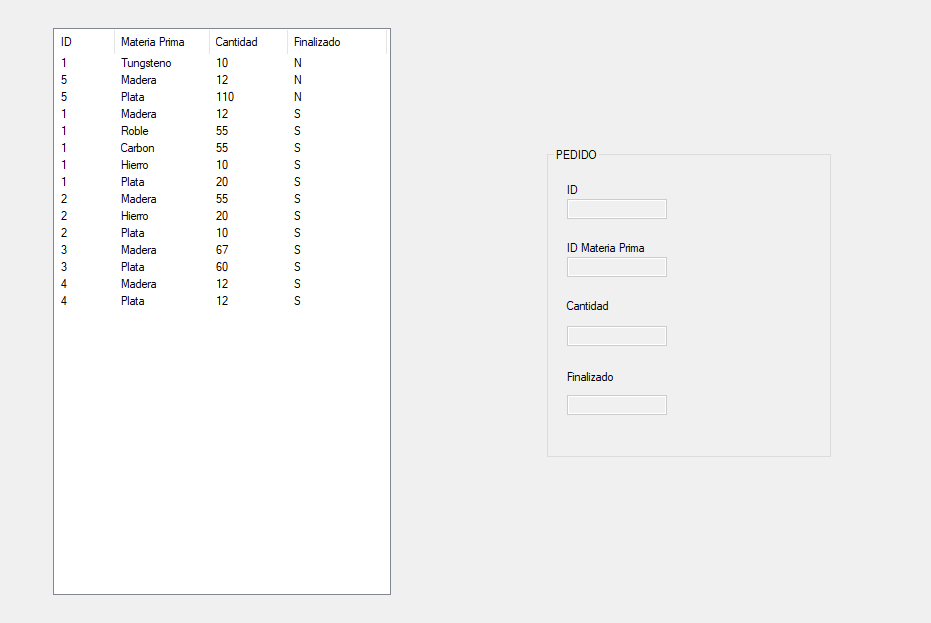
Ahora en el grid de abajo (la cesta) nos saldrá la materia prima que hemos seleccionado, en este caso el tungsteno. En stock próximo podemos ver que se ha agregado la cantidad que hemos puesto.

En caso de volver a seleccionar el tungsteno la cantidad se agregará a la fila del tungsteno que ya tenemos en la cesta, no hará una fila nueva. Si queremos que se hagan dos pedidos en vez de uno, podemos hacer el pedido y luego realizar otro. De esta manera habrá dos pedidos distintos de la misma materia y no se sumará.

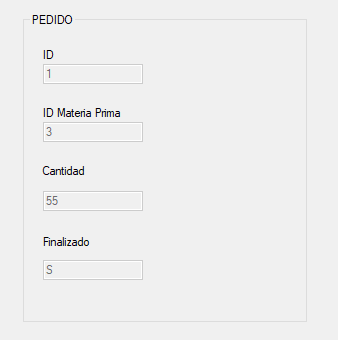
Una vez agregado todos los productos que queremos le damos al botón “Hacer el pedido” que se encuentra a la derecha del grid.



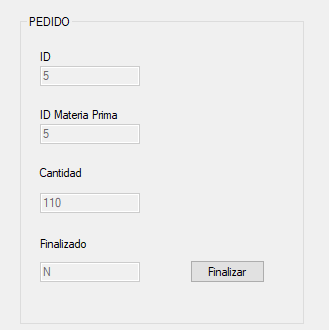
Se notificará al usuario que el pedido se ha realizado con éxito.



Nos sale a la izquierda todos los pedidos ordenados por su estado. De esta manera nos sale los primeros los que no están finalizados. Si pulsamos sobre uno se rellenarán los datos con los detalles de pedido en la parte izquierda.

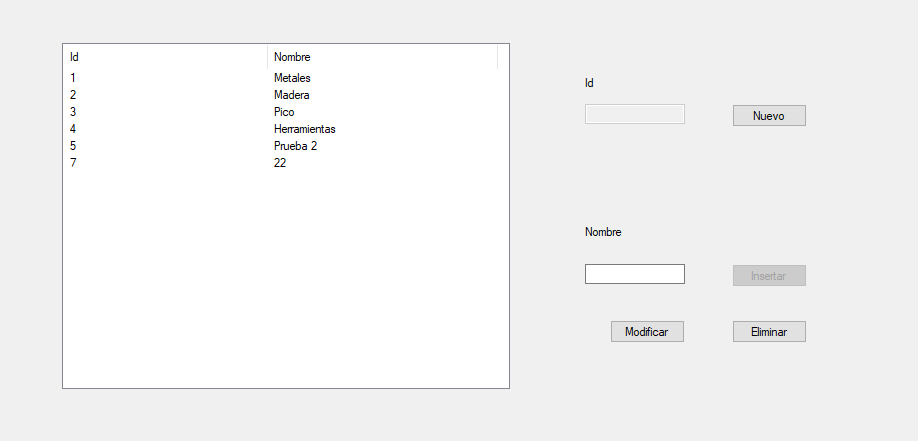


Aquí por ejemplo he puesto los detalles de un pedido que ya esta finalizado. En caso de tener el estado finalizado a S solo podremos ver la información del pedido. Podemos ver el ID del pedido, el ID de la materia prima la cantidad y su estado.

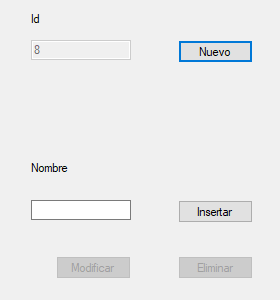


En este caso podemos ver que el pedido que he traído tiene el estado de finalizado a N, es decir no ha llegado aún. En caso de que el pedido llegase, el trabajador le daría al botón de finalizar que está a la derecha del estado.

De esta manera daría por finalizado el estado, le cambiaria el estado a S, quitaría las unidades del stock próximo que corresponden, se las agregaría al stock y todo esto estaría reflejado en los logs de las materias primas.

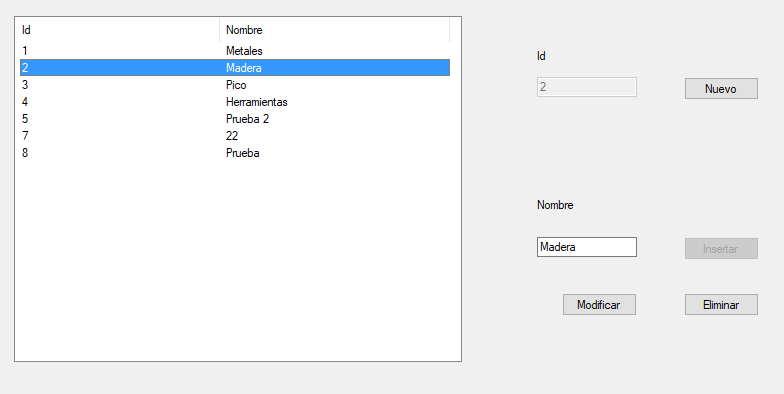


El siguiente botón es para hacer un mantenimiento de las categorías, si le das a nuevo se autogenera la siguiente primary key que este disponible. Esta primary key se guarda en una variable para no hacer la llamada cada vez que le demos a nuevo.



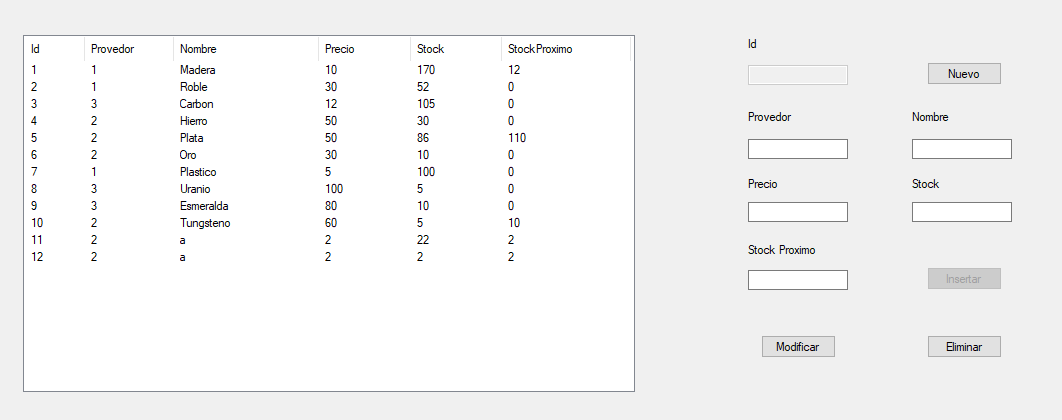
En este caso se nos genera el ID 8 y tenemos la posibilidad de ponerle un nombre a la categoría y se nos ha desbloqueado el botón de insertar mientras que el resto se deshabilitan.

Una vez introducido el nombre le daríamos a insertar y se mandaría a SQL y a su vez se agregaría de manera local a la lista donde nos salía todas las categorías.



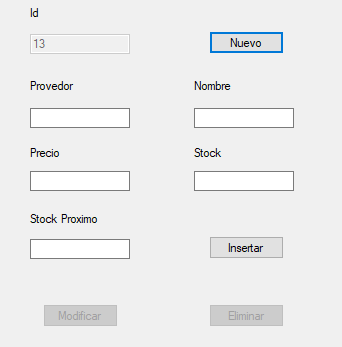
Si pulsamos dos veces sobre una categoría se nos rellenaran los campos de la derecha con sus datos. Solo podremos modificar el nombre, la clave primaria estará protegida. Como podemos ver en la imagen el botón de insertar se bloquea mientras que el de modificar y eliminar están disponibles.

El botón de modificar modificara tanto en SQL como en la lista y el botón de eliminar eliminara de ambos lados también.



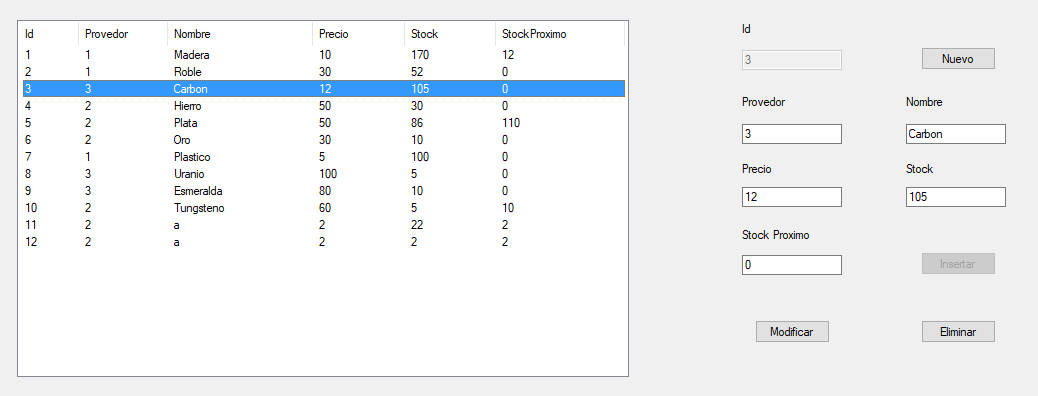
El siguiente botón es para hacer un mantenimiento de las materias primas, si le das a nuevo se autogenera la siguiente primary key que esté disponible. Esta primary key se guarda en una variable para no hacer la llamada cada vez que le demos a nuevo.

Tiene un funcionamiento muy similar al anterior pero el mantenimiento en vez de ser de categorías es de materias primas.



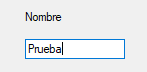
En este caso se nos genera el ID 13 y tenemos la posibilidad de ponerle un nombre a la categoría y se nos ha desbloqueado el botón de insertar mientras que el resto se deshabilitan.

Una vez introducido el nombre le daríamos a insertar y se mandaría a SQL y a su vez se agregaría de manera local a la lista donde nos salía todas las categorías.

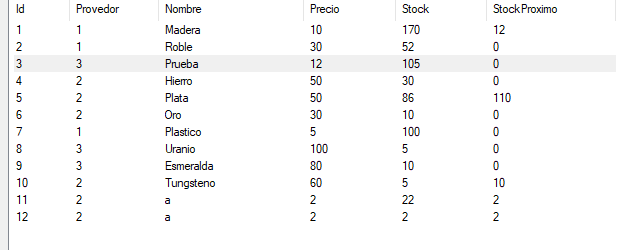


Si pulsamos dos veces sobre una materia prima se nos rellenaran los campos de la derecha con sus datos. Solo podremos modificar el nombre, la clave primaria estará protegida. Como podemos ver en la imagen el botón de insertar se bloquea mientras que el de modificar y eliminar están disponibles.

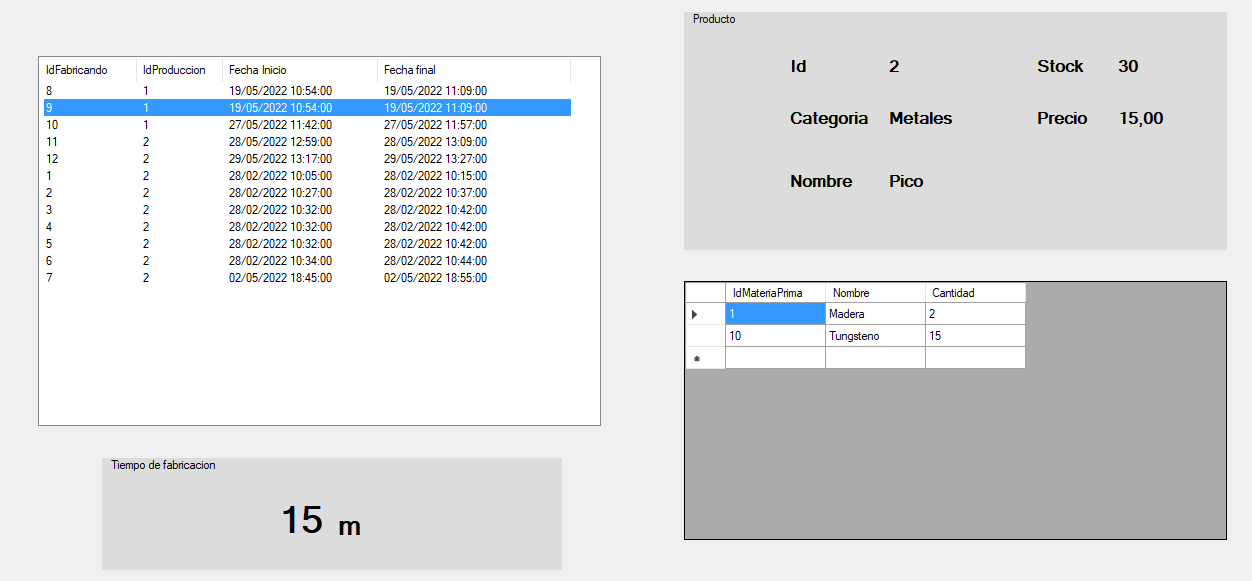
El botón de modificar modificara tanto en SQL como en la lista y el botón de eliminar eliminara de ambos lados también.



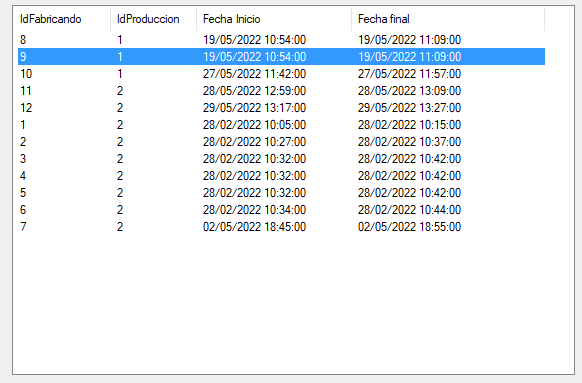
Le cambiamos el nombre para probar y le damos a modificar.



Como podemos ver el nombre ya no es el anterior (carbón), ahora se ha modificado con éxito tanto en SQL como en la lista y se llama prueba. El cambio de la lista se hace de manera local, no se hace otra llamada a SQL para refrescar.



En el siguiente apartado tenemos los logs de las producciones, se divide en 4 partes y voy a explicarlos uno por uno.



En la parte de la izquierda podemos ver todo el historial de producciones. Te dice el ID del movimiento, la cabecera de la producción que se ha hecho, la fecha de inicio real y la fecha de fin real. De esta manera podemos ver si difiere mucho la fecha estimada con la real para controlar la efectividad de las maquinas o de las personas de producción.

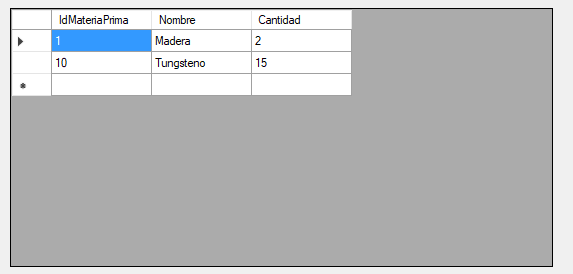
El ID de la producción no nos da demasiada información, es por eso que las otras pestañas nos ayudan a ver a que hace producto hace referencia realmente.



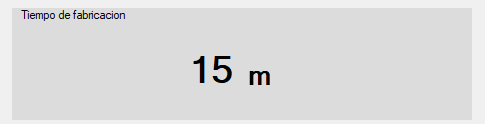
Si pulsamos dos veces sobre un log de producción se nos llenara el panel de la derecha que pone producto con la información sobre el producto que se ha fabricado.

De esta manera no solo vemos el ID de producción al que hace referencia si no que además vemos información detallada del producto que se ha fabricado como el nombre de la categoría al que pertenece, su id, su nombre, el stock que tiene y el precio.

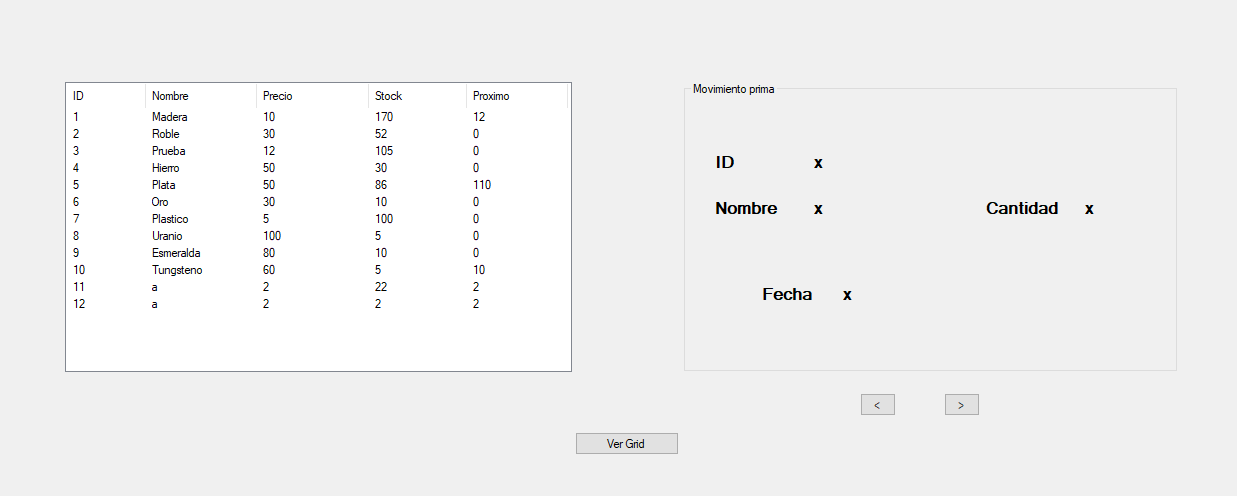
El stock que sale no es el que tenia en el momento de la producción, es el stock actual de ese producto en el inventario. A parte de esta información es importante saber que materias primas se han empleado y la cantidad de ellas. Serían las líneas de la producción.



Aquí podemos ver que en este caso el pico este hecho por dos materias primas, necesita dos de madera y 15 de tungsteno. Esto facilita al trabajador a tener un control más fácil e inmediato de todo lo que implica la producción.

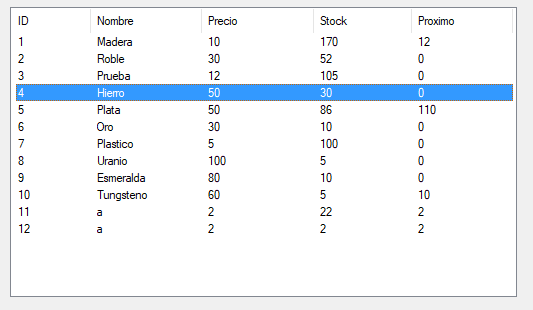


Por ultimo se muestra el tiempo real invertido en la producción, es la diferencia entre la hora final y la inicial. De esta manera el trabajador no tiene que estar haciendo cálculos para sacar el tiempo, es sencillo y rápido de consultar.



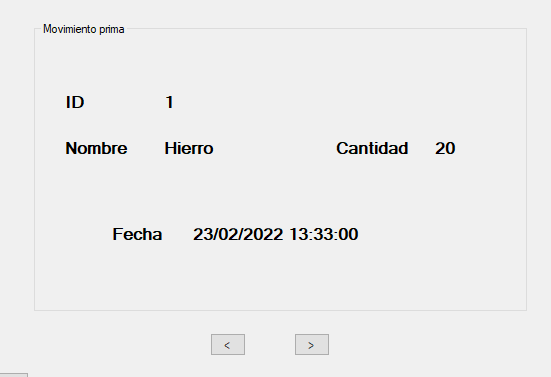
La siguiente opción es para el log de las materias primas, podemos ver que la pantalla esta dividida en tres partes, una lista a la izquierda con todas las materias primas, un formulario a la derecha y un botón abajo.

Voy a ir explicando paso por paso la manera de funcionar de la pantalla empezando por la parte izquierda.



A qui podemos ver la lista de las materias primas disponibles, las materias primas componen productos y son importantes en la producción. Podemos ver los campos básicos como el id, el nombre, el precio, el stock y el stock próximo (stock que este pedido, pero todavía no ha llegado).

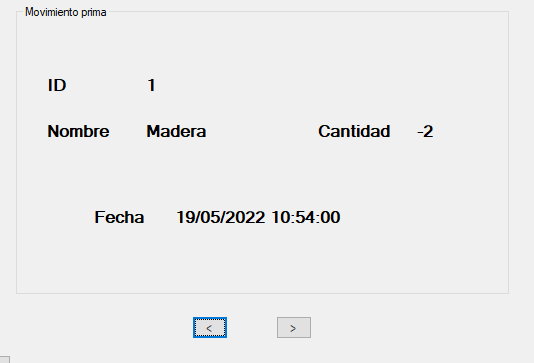
En este apartado lo único que podemos hacer es pulsar sobre la materia prima de la que queremos ver el log.



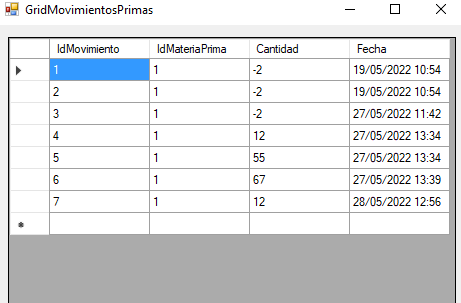
Aquí podemos ver que se ha cargado con información el formulario de la derecha. Aquí nos muestra los logs y podemos avanzar o retroceder en la lista con los botones que tenemos en la parte inferior.

El ID que se nos muestra no es el de la materia prima, es el del movimiento. Se nos muestra el id del movimiento, el nombre de la materia prima, la cantidad del movimiento y la fecha.

En este caso la cantidad está en positivo, la cantidad entra en el stock por un pedido, por ejemplo. Esta cantidad no siempre va a ser positiva. En la próxima captura muestro una materia prima con una cantidad negativa.

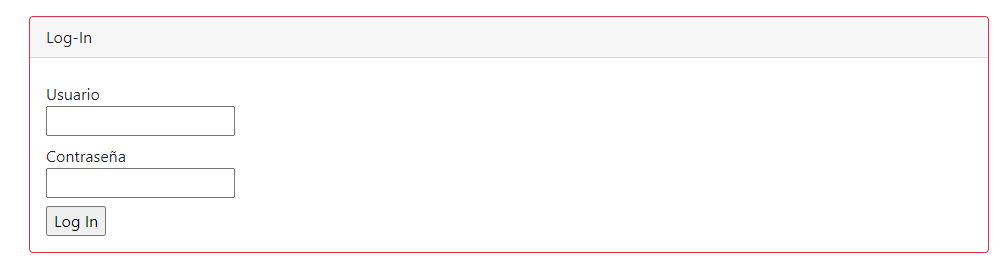


En este por ejemplo podemos ver que la madera el 19/05/2022 a las 10:54 tiene una cantidad negativa en el movimiento. Esto quiere decir que hubo una producción, esa materia prima la perdemos del stock.



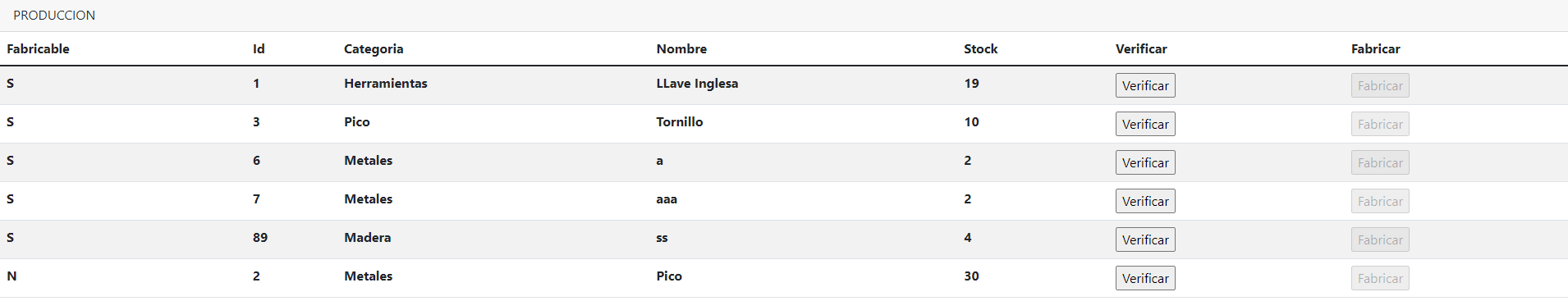
Por último, si le damos al botón de ver la lista, podemos ver la lista de todos los logs de ese material sin tener que ir viéndolos de uno en uno. Esto nos permite llevar un control mucho más ágil y tener una visualización mucho mas amplia.

-Aplicación web:



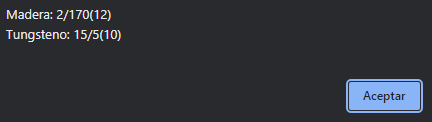
Se nos muestra un formulario donde introducir el usuario y la contraseña. Comprueba los valores y mira si existe algún registro en SQL con esas credenciales. En caso de existir deja avanzar y en caso de no hacerlo te muestra que las credenciales son erróneas.

Si te registras con éxito crea una variable de sesión con el nombre de usuario, si intentas acceder directamente a la parte de mantenimiento o de producción sin haberte registrado antes detectará que no está la variable de sesión creada y te redirigirá al log-in.



Si vamos al apartado de producción podemos ver una lista con todos los productos. Muestra el nombre de la categoría a la que pertenece, el id del producto, su stock, si es fabricable o no (tenemos las suficientes materias primas en stock como para producirlo o no) y dos botones que ahora voy a especificar.

El primer botón te da un mensaje como este:

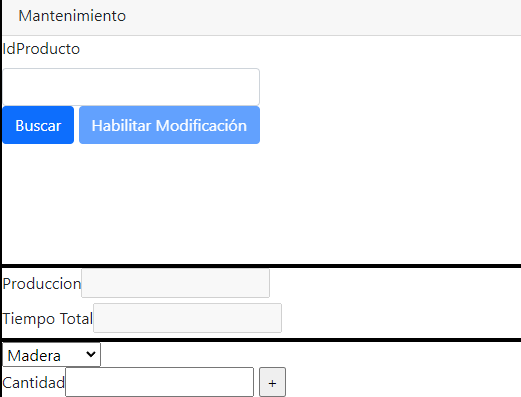


En el que te informa la cantidad que requiere para la fabricación el stock del que se dispone y el stock que este pedido, pero todavía no ha llegado.

A pesar que en la tabla se vea si es fabricable o no, gracias a este botón es una medida más de seguridad. Ya que el trabajador tiene que primero verificar las materias primas que se van a emplear en la producción y ver que es lo que quiere y que todo está correcto.

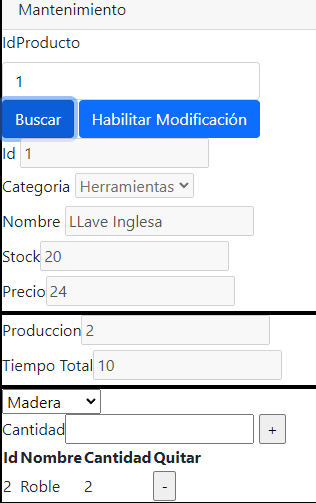
En caso de que se tenga la materia prima suficiente como para fabricar el producto el botón de fabricar se habilitara. El botón de fabricar lo que hace es poner a fabricar un producto, se encarga de poner el log de fabricación como encargarse del cambio en el stock de las materias primas y el cambio en el stock del producto.

La otra opción que tenemos en la página web es un mantenimiento más extenso de los productos. No solo deja modificar los campos de los productos si no que además relaciona varias tablas como pueden ser las líneas de producción, la cabecera de producción, las materias primas y las categorías.



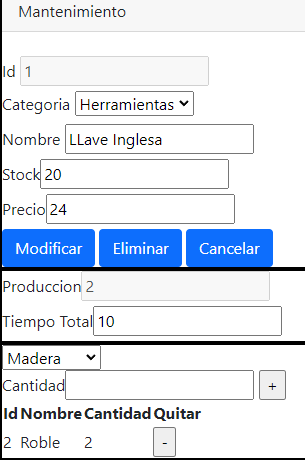
Aquí podemos ver como podemos insertar un id de producto en la caja de texto superior. Si introducimos uno existente en la base de datos y le damos a buscar se nos rellenaran el resto de los campos con la información de ese producto.

Además, se nos habilitara el botón de habilitar modificación. En caso de no encontrarlo nos permitirá dar de alta un0 nuevo.



En este caso el ID uno existe y es una llave inglesa. Nos llena con toda la información sobre el producto y el nombre de la categoría a la que pertenece. Después de la raya negra podemos ver información relacionada con la cabecera de la producción, el id de la producción y el tiempo estimado.

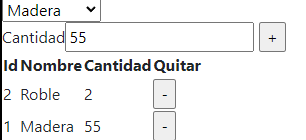
Por último, debajo podemos ver las líneas que componen el producto. También tiene una funcionalidad de modificar las líneas que voy a especificar a continuación.



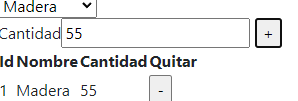
Si le damos al botón de modificar, se nos habilitar los campos para poder modificarlos. Podemos cambiarle la categoría, en el combo que hay nos sale todas las categorías que hay y por defecto viene seleccionada la que pertenece al producto.

También se puede modificar la cabecera de la producción por si queremos cambiar el tiempo estimado. El ID sigue protegido porque no admite modificaciones de las claves primarias.

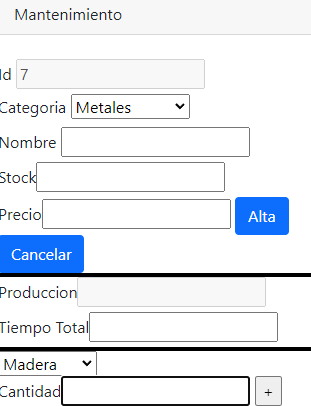
Por último, tenemos la posibilidad de modificar las líneas de las materias primas que componen el producto.



Por ejemplo, aquí he agregado una línea donde digo que también hace falta 55 de madera para fabricar el producto.



Dándole al botón de menos se quita la línea y ahora solo hace falta madera, ya el roble no hace falta.



Si el ID no existe se habilitan todos los textos para rellenarlos con información del producto nuevo. Una vez rellenado le damos a Alta y se insertara en SQL el nuevo producto.

En caso de buscarlo otra vez en el buscador se encontrara el producto, se rellenaran todos los campos con su información y se habilitará la posibilidad de modificar y eliminar.

Funciones:

SERVICIO WEB:

public void HacerPedido(int idMateriaPrima, int cantidad)

Llama a un procedimiento almacenado creado en SQL llamado produccion.HacerPedido y le pasa como valor a los parámetros necesarios del procedimiento los parámetros recibidos para hacer la llamada.

public void FinalizarPedido(int idPedido, int IdMateriaPrima)

El anterior método hace el pedido, pero lo deja en no finalizado, es decir, el stock se agrega al campo de stock próximo y hasta que no se finaliza no se agrega al campo de stock.

Finalizar pedido llama a un procedimiento almacenado de SQL llamado Produccion.FinalizarPedido y le pasa como parámetros los parámetros recibidos en la llamada. Este procedimiento finaliza el pedido y reduce las cantidades del stock próximo al stock

public void ListaProductos(int opc)

Recibimos como parámetro un int que puede ser 0,1 o 2 y depende del valor de este parámetro el JSON que devuelve varia. En los tres casos devuelve un JSON de productos.

Si la opción es 0, devuelve todos los productos sin ningún filtro con el nombre de las categorías.

Si la opción es 1, llama a una vista creada en SQL que tiene unas subconsultas con el objetivo de que los productos que devuelvan se puedan fabricar, es decir, tenemos la suficiente cantidad de materias primas que necesita las líneas del producto.

Si la opción es 2, llama a una vista creada en SQL que tiene unas subconsultas con el objetivo de que los productos que devuelvan no se puedan fabricar, es decir, no tenemos la suficiente cantidad de materias primas que necesita las líneas del producto. Con que haya una materia prima que no tengamos lo suficiente el producto ya no es fabricable.

public void ProductoDisponible(int idArticulo)

Llamamos a un procedimiento almacenado de SQL llamado Produccion.Disponibilidad donde necesita un parámetro que es el ID del articulo que lo recibimos en el parámetro de llamada a esta función.

El método devuelve 1 si el articulo se puede fabricar y 0 si no se puede.

public void Verificar(int idProducto)

Verifica devuelve los recursos necesarios para hacer un producto. Le pasamos el ID del producto que queremos verificar para agregarlo al filtro de la consulta.

public void FabricarArticulo(int idArticulo)

Llamamos a un procedimiento almacenado de SQL llamado Produccion.Producir. Le pasamos el ID del articulo al procedimiento, id recibido como parámetro en la petición de la función.

Sumara una al stock de ese producto, restara las materias primas que se hayan utilizado para esta producción, además de poner tanto el log de los movimientos de las materias primas como el log de la producción

public void CargarCategorias()

Hace una petición a SQL para recibir el ID y el nombre de todas las categorías. Las devuelve en formato JSON.

public void AltaProducto(int idArticulo, string categoria, string nombre, int stock, float precio)

Inserta un producto nuevo, para esta petición le tenemos que pasar el ID del nuevo producto que queremos crear, el ID de la categoría, el nombre que le queremos poner al producto, el stock del que disponemos y el precio del producto.

public void GetProducto(int IdArticulo)

Le pasamos como parámetro el ID del articulo que queremos buscar. Nos devuelve en formato JSON información adicional de ese producto, su ID, el ID de la categoría, el nombre, el stock, el precio, el ID de la cabecera de producción relacionada con este producto y el tiempo de producción.

public void ActualizarProducto(int idArticulo, int idCategoria, string nombre, int stock, float precio)

Actualiza un producto, para esta petición le tenemos que pasar el ID del nuevo producto que queremos modificar, el ID de la categoría, el nombre que le queremos poner al producto, el stock del que disponemos y el precio del producto.

public void GetLineasDeProducto(int IdProducto)

Le pasamos el ID del producto del que queremos recibir las líneas de producción. Las líneas de producción nos dicen de que materia prima están hechos los productos.

Nos devuelve información sobre cada línea y información adicional sobre cada materia prima.

public void BorrarProducto(int idArticulo)

Hace un comando DELETE en SQL, para ello le pasamos el ID del articulo que queremos eliminar para ponerlo en la cláusula where.

public void GetMateriasPrima()

Devuelve información sobre todas las materias primas que están dadas de alta en el servidor de datos.

Nos devuelve la información en formato JSON y nos devuelve su ID, el ID del proveedor, el nombre de la materia prima, el precio, el stock y el stock próximo.

public void InsertarLineaProduccion(int idProducto, int idMateriaPrima, int cantidad)

Inserta una línea de producción de la cabecera de producción que le corresponde el id del producto que le pasamos como parámetro.

Le pasamos el ID de la materia prima que le queremos agregar a la composición al ID del producto que le pasamos y la cantidad de esa materia prima necesaria para hacer el producto.

public void BorrarPrimaLinea(int IdProducto, int IdMateriaPrima)

Le pasamos el ID del producto al que le queremos borrar una línea. El producto esta relacionado con las cabeceras de producción y a su vez estas cabeceras están relacionadas con sus líneas.

También le tenemos que pasar el ID de la materia prima que queremos eliminar de las líneas ya que los productos pueden estar compuestos por varias líneas.

public void InsertarProduccion(int idProducto, float tiempoTotal)

Inserta una cabecera de producción para el producto que le pasamos como parámetro y el tiempo total estimado de la producción de ese producto.

public void ModificarCantidadProduccion(int idProduccion, float tiempoTotal)

Modifica el tiempo estimado para la producción del producto que le pasamos como parámetro y el tiempo total nuevo que le queremos asignar.

ESCRITORIO:

-Pedidos

private void Pedidos\_Load(object sender, EventArgs e)

Inicializa muchos comandos y parámetros de SQL. Carga los esquemas de las tablas de SQL en todos los grid , llena el combo box con los nombres y los valores de las categorías y llena la lista con el nombre de los productos.

La primera carga viene filtrada por la primera categoría, es decir, los productos de la primera carga en la lista pertenecen a la primera categoría de la base de datos.

private void cmbCategorias\_SelectionChangeCommitted(object sender, EventArgs e)

Cada vez que se cambia el valor seleccionado en el combo se cargan los valores de la lista con los productos que corresponden a ese filtro.

private void btnPedido\_Click(object sender, EventArgs e)

Coge la fila seleccionada de las materias primas y pide al usuario una cantidad. Se pasa esta información al grid de debajo.

private void lstProductos\_DoubleClick(object sender, EventArgs e)

Carga con información sobre las materias primas que componen el producto que has clicado. De esta manera luego puedes seleccionar una materia prima y darle al botón de pedido.

private void btnHacerPedido\_Click(object sender, EventArgs e)

Llama a un procedimiento almacenado para hacer el pedido pasándole todas las líneas que están agregadas en el grid para hacer esa petición. Por cada línea de pedido de materia prima se hace una llamada al procedimiento almacenado.

-Aceptar pedidos

private void AceptarPedido\_Load(object sender, EventArgs e)

Carga la lista de la izquierda con todos los pedidos ordenados por su estado, de esta manera salen primero los que no están finalizados.

También inicializa muchos comandos y parámetros.

private void lstPedidos\_MouseDoubleClick(object sender, MouseEventArgs e)

Llena con valores los campos de la derecha, es el evento que responde al doble click de un elemento de la lista. Pone con mas detalle información del pedido en el formulario. Además si el estado del pedido es no finalizado se activara un botón que permita finalizar el pedido.

private void btnFinalizar\_Click(object sender, EventArgs e)

Finaliza el pedido, al hacer click en el botón le cambia el estado y cambia la cantidad del stock próximo relacionada con la cantidad del pedido que acabas de confirmar y se agrega al stock real.

-Mantenimiento de categorías

private void MantenimientoCategoria\_Load(object sender, EventArgs e)

Carga en una lista todas las categorías disponibles en la base de datos. Aparte inicializa numerosas conexiones de los comandos y configura los parámetros de los comandos para posteriormente usarlos solo dándole valor.

private void lstCategorias\_MouseDoubleClick(object sender, MouseEventArgs e)

Al hacer doble click en una fila de la lista, se rellena con información adicional el formulario de la derecha.

private void btnNuevo\_Click(object sender, EventArgs e)

Al darle a nuevo en caso de no estar inicializada la variable ID calculara automáticamente el ID próximo para una categoría nueva. Si no hay ninguna categoría pondrá el número 1 y en caso de haber cogerá el ID más alto de las categorías y le sumará uno.

private void btnInsertar\_Click(object sender, EventArgs e)

Después de rellenar el formulario le podemos dar al botón de insertar y se insertara tanto en SQL como a nivel local en la lista de las categorías.

La lista no hace ninguna llamada adicional para refrescarse de esta manera nos ahorramos aperturas innecesarias de conexiones y peticiones al servidor de datos.

private void btnModificar\_Click(object sender, EventArgs e)

Despues de pulsar en la lista, al rellenarse los campos con la información de la línea seleccionada, podemos modificar los valores que queramos menos la clave primaria.

Una vez modificado, le podemos dar al botón de modificar y de esta manera modificaremos los valores en SQL y también en la lista. Recorremos la lista y buscamos por la clave primaria para modificarla a nivel de cliente.

private void btnEliminar\_Click(object sender, EventArgs e)

Despues de pulsar en la lista, al rellenarse los campos con la información de la línea seleccionada, eliminar el registro.

Le podemos dar al botón de eliminar y de esta manera eliminaremos el valor en SQL y también en la lista. Recorremos la lista y buscamos por la clave primaria para eliminarla a nivel de cliente de manera simultánea.

-Mantenimiento Primas

private void MantenimientoMateriaPrima\_Load(object sender, EventArgs e)

Carga en una lista todas las materias primas disponibles en la base de datos. Aparte inicializa numerosas conexiones de los comandos y configura los parámetros de los comandos para posteriormente usarlos solo dándole valor.

private void lstMateriaPrima\_MouseDoubleClick(object sender, MouseEventArgs e)

Al hacer doble click en una fila de la lista, se rellena con información adicional el formulario de la derecha.

private void btnNuevo\_Click(object sender, EventArgs e)

Al darle a nuevo en caso de no estar inicializada la variable ID calculara automáticamente el ID próximo para una categoría nueva. Si no hay ninguna materia prima pondrá el número 1 y en caso de haber cogerá el ID más alto de las materias primas y le sumará uno.

private void btnInsertar\_Click(object sender, EventArgs e)

Después de rellenar el formulario le podemos dar al botón de insertar y se insertara tanto en SQL como a nivel local en la lista de las categorías.

La lista no hace ninguna llamada adicional para refrescarse de esta manera nos ahorramos aperturas innecesarias de conexiones y peticiones al servidor de datos.

private void btnModificar\_Click(object sender, EventArgs e)ç

Despues de pulsar en la lista, al rellenarse los campos con la información de la línea seleccionada, podemos modificar los valores que queramos menos la clave primaria.

Una vez modificado, le podemos dar al botón de modificar y de esta manera modificaremos los valores en SQL y también en la lista. Recorremos la lista y buscamos por la clave primaria para modificarla a nivel de cliente.

private void btnEliminar\_Click(object sender, EventArgs e)

Despues de pulsar en la lista, al rellenarse los campos con la información de la línea seleccionada, eliminar el registro.

Le podemos dar al botón de eliminar y de esta manera eliminaremos el valor en SQL y también en la lista. Recorremos la lista y buscamos por la clave primaria para eliminarla a nivel de cliente de manera simultánea.

-Log producciones

private void LogFabricando\_Load(object sender, EventArgs e)

Llena la lista con todos los logs de las fabricaciones que ha habido. Nos muestra el ID del log a que hace referencia y la fecha.

También inicializa la conexión de los comandos y configura los parámetros para su posterior uso.

private void lstDatos\_MouseDoubleClick(object sender, MouseEventArgs e)

Al hacer doble click en un registro el formulario de la derecha se llena con información adicional del producto al que hace referencia. Además, el grid se llena con las materias primas que componen ese producto.

Por último, coge las dos fechas y resta la fecha final con la de inicio para sacar la duración real de la producción en minutos.

-Log materias primas

private void MovimientoMateriaPrima\_Load(object sender, EventArgs e)

Carga en una lista todas las materias primas disponibles en la base de datos. Aparte inicializa numerosas conexiones de los comandos y configura los parámetros de los comandos para posteriormente usarlos solo dándole valor.

private void lstPrima\_MouseDoubleClick(object sender, MouseEventArgs e)

Carga un data table con los logs de la materia prima seleccionada. Ese data table se usa para mostrar la información de los logs en el formulario de la derecha.

private void btnAtras\_Click(object sender, EventArgs e)

Retrocede una posición en los logs que se están mostrando de ese producto. Avanza hasta que llega al primero, una vez llegado al último no retrocede más.

private void btnDelante\_Click(object sender, EventArgs e)

Avanza una posición en los logs que se están mostrando de ese producto. Avanza hasta que llega al último, una vez llegado al último no avanza más.

private void button1\_Click(object sender, EventArgs e)

Se abre un grid donde se muestran todos los logs del producto seleccionado. En el grid podemos ver la información más compacta sin tener que estar avanzando de uno en uno.