

|  |
| --- |
| **Universidad Internacional de La Rioja (UNIR)**  **Escuela de Ingeniería**  **Grado en Ingeniería Informática** |
|  |
| Aplicación Web para la gestión de máquinas Vending y administración |

**Ubicación del código fuente:**http://…

**Trabajo Fin de Grado**

**presentado por:**Moreno Palacios, Carlos

**Director/a:**Soltero Domingo, Francisco

Ciudad: Madrid

Fecha:

# Resumen

En la situación actual la gestión monetaria y de productos de las máquinas *vending* se realizan de manera manual y sin ningún tipo de control autónomo. Se ha desarrollado una aplicación que permite conocer la distribución de las máquinas, su stock de productos, el estado actual de los recursos monetarios, la gestión del personal y su gestión de manera automatizada

**Palabras Clave:** Vending, ventas, automatización, stock.

# Abstract

In the current situation, the monetary and product management of vending machine are done by manually form and without any type of control.

A web application has been developed that can control all machine information that allow to know the machines distribution, their products stock, the current state of the monetary resources and finally, the management of the workers.

**Keywords:** Vending, sales, automation, stock

**1**.**Objetivo**

El objetivo del presente trabajo es analizar los distintos procesos que conforman las operaciones referentes a sistema de gestión y mantenimiento de máquinas de tipo *vending*[[1]](#footnote-2).

Para ello se desarrollará una aplicación web que administre y gestione éstas con el fin de mantener toda la información disponible de manera ordenada y automatizada, y con la inclusión de servicios de alertas automáticas que informen al usuario. Dado que no existe ningún software que realice de manera conjunta la gestión de usuarios y la de las máquinas vending, se ve la necesidad de desarrollar un nuevo software que administre éstos de manera eficiente, accesible y de licencia *open source*.

Para que la funcionalidad de la aplicación fuera completa es necesario adaptar las máquinas existentes para que sean compatibles con la herramienta. Es decir, habría que desarrollar un firmware que controlase los eventos de cada máquina y que tuviera una conexión telemática con la base de datos que utiliza la aplicación. Dada la complejidad y la dificultad de acceso a información restringida desde fuera del entorno de desarrollo de estas máquinas, la idea será crear una api web, la cual podrán implementar los fabricantes para completar el funcionamiento de la aplicación.

Debido a la gran diversidad de hardware y posibles combinaciones de software que controlan y gestionan este tipo de máquinas, y dado que el fundamento de esta aplicación es crear un software libre y genérico, no será objeto de este proyecto el desarrollo centralizado sobre ninguna máquina del mercado.

**2**. **Motivación**

En una organización es necesario minimizar costes, al igual que huella digital[[2]](#footnote-3) generada al realizar cualquier gestión.

El desarrollo de una aplicación permitirá, a través de cualquier dispositivo Smartphone o pc, conectarse, gestionar y consultar en el acto, la situación actual de del stock de los productos, el nivel de monedas, ubicación, estadísticas y demás.

Al basarse la aplicación en las nuevas tecnologías, se permitirá que se desarrolle de manera continuada y que esta sea cien por cien compatible con el software actual, gracias a que el desarrollo estará basado en metodologías ágiles y tendrá una arquitectura flexible.

**3.1 INTRODUCCIÓN**

Hoy en día prima la eficiencia de cualquier sistema de gestión optimizando los beneficios económicos y sociales y minimizando el coste.

Las grandes y pequeñas organizaciones no tienen un sistema centralizado de gestión telemática de las máquinas que poseen, de modo que no están utilizando toda la capacidad de gestión que proporciona la tecnología actual.

Todo beneficio que se pueda generar siempre es conveniente, de manera que al disponer de la información necesaria, una empresa puede decidir las rutas más favorables, obtener datos sobre la viabilidad de una máquina, y en general, cualquier dato que se desee para alcanzar los objetivos de eficiencia.

Estamos por lo tanto, ante una oportunidad de negocio que no solo beneficie a la institución de manera económica, sino que también brindará otros beneficios como la reducción de la huella ecológica, entre otros.

Para conseguir estos objetivos es necesaria la realización de una aplicación que contenga las siguientes funcionalidades:

* Gestión de las máquinas que permitan: localización, modelo, marca, productos incluidos, monedas de las que dispone, administrador/usuario de dicha máquina.
* Gestión de los usuarios: datos de usuario, email y máquinas que administra.
* Gestión de la economía de la máquina.
* Mensajería entre los usuarios de la plataforma
* Estadística de ventas de cada máquina y producto.

Este trabajo estudiará el valor añadido que supone la utilización de las nuevas tecnologías en la gestión de máquinas vending, permitiendo a su vez un ahorro significativo de costes y optimización en lo relativo al tiempo necesario para su gestión.

El presente trabajo se dividirá en distintos capítulos:

* El capítulo 2 versará sobre la situación actual de las máquinas vending, de la arquitectura y los patrones que se utilizarán para el desarrollo de una aplicación que mejore la situación actual y de la metodología que se seguirá para que esta sea un éxito.
* En el capítulo 3 se hablará sobre los distintos requisitos funcionales y no funcionales que describen la funcionalidad del proyecto, de los distintos casos de uso de la aplicación, de las herramientas que se van a utilizar para desarrollar la aplicación y de los roles que se asignarán siguiendo la metodología asignada.
* En el capítulo 4 se tratará todo aquello relacionado con la descripción de alto nivel del trabajo, en que se describirán, incluyendo imágenes, todos los apartados funcionales de la aplicación.
* En el capítulo 5 se realizará una descripción de bajo nivel de toda la estructura de la aplicación, realizándose una explicación sobre su funcionamiento.
* En el capítulo 6 se comprobará la validez y viabilidad del proyecto que se ha desarrollado.
* En el capítulo 7 se comentarán las conclusiones sobre el proyecto y se propondrán líneas de futuro con la propuesta de nuevas funcionalidades.

**3.2 CONTEXTO Y ESTUDIO PRELIMINAR**

En este capítulo se hablará de la historia y evolución de las máquinas vending, de los patrones de diseño utilizados, de la arquitectura que utiliza la aplicación, la metodología aplicada durante el diseño y el desarrollo del trabajo y de los requisitos funcionales y no funcionales que definen la aplicación. Además también se definirán los casos de uso según los distintos roles así como las herramientas que se necesitan y se han utilizado para su implementación.

**3.2.1. MÁQUINAS DE VENTA AUTOMÁTICA**

Una máquina vending, o también llamada máquina expendedora o de autoventa (35), tiene su origen en los Estados Unidos y en el Reino Unido a finales del siglo XIX (35). Por aquella época y dependiendo de su ubicación, vendían únicamente un producto (26)

Una de las primeras máquinas sitas en territorio español vendía cerillas (35), y desde entonces estas máquinas han tenido una gran evolución (32) hasta lo que son hoy en día, aunque existen noticias (31) de que la primera máquina dispensaba, que no vendía, agua bendita.

Además de una evolución tecnológica en su sistema de disposición de productos, las máquinas han evolucionado también en su sistema de pago, entre otros (32).

Estas máquinas se hicieron realmente famosas porque se utilizaron como vendedores de productos prohibidos de manera anónima, en varias ocasiones. Hablamos sobre todo de libros (2) prohibidos que fueron vendidos entre los años veinte y cincuenta en países como los Estados Unidos y el Reino Unido, en cambio, en España no llegaron hasta varios años posteriores (35).

Las expendedoras de libros siguen existiendo aún en la actualidad, permitiendo mantener el negocio de venta de algunos locales las 24 horas sin necesidad de personal, por lo que son elementos que se rentabilizan de por sí (34).

En el año 1920 (33) se empezaron a incorporar productos refrigerados en estas máquinas, como refrescos, agua y otros productos, lo que propició un nuevo mercado y un impulso para estas máquinas así como para los productos vendidos.

Todas ellas poseen un funcionamiento genérico que se diferencias entre sí por los productos que venden y por los sistemas de entrega, pago y forma.

* En general, todas las máquinas constan de (36, 4): Sistema de gestión monetaria con un depósito de monedas clasificadas y un depósito de monedas introducidas así como detector de cantidad introducida y cambio restante.
* Panel de selección de artículo.
* Dispensador automatizado con sistema de entrega automático o de caída.
* Refrigerador en la mayoría de los casos e iluminación.
* Maquina procesadora y controladora de lo anteriormente comentado.
* *Slots*[[3]](#footnote-4) en la que generalmente poseen el mismo producto y tiene una capacidad equivalente a todos los *slots* de una misma máquina.

**3.2.2. SITUACIÓN ACTUAL**

Tal y como se ha comentado en el apartado anterior, las máquinas más modernas ya incluyen sistemas telemáticos sobre todo para la gestión de pago telemático mediante tecnología RFID o de pago con tarjeta. Últimamente se están desarrollando nuevas funcionalidades para dotar de inteligencia a estas máquinas mediante la aplicación de técnicas de telemetría mediante hardware específico [38].

Las máquinas suelen disponer de interfaces de conexión que pueden servir precisamente para obtener la información pertinente que nutra la base de datos de la información necesaria para la funcionalidad de la aplicación.

De entre todas, se pueden citar las siguientes interfaces que pueden contener las máquinas [39]:

* Interfaces de puerto serie: Son las más modernas y se suelen utilizar para comunicar los distintos módulos esclavos con el módulo actuador o master.
* Interfaces electromecánicos: Son las más antiguas y actúan directamente sobre las `partes de las máquinas sin ningún tipo de procesamiento.

Partiendo del primer tipo de interfaces que son los que suelen utilizar, existen los siguientes [39]:

* Protocolo A: Propiedad de Mars Electronic International . Este protocolo es utilizado para los procesos relacionados con la gestión de pago y de cambio de las máquinas. La alimentación de los componentes van por otras conexiones, ya que este protocolo no permite la alimentación.
* BDV: Creado por Bdv Association. Este tipo de protocolo se usa únicamente en territorio Germano. Como ventaja con respecto al anterior, tiene que tanto la comunicación como la alimentación van de la misma mano, siguiendo el protocolo BDV001.
* MDB/ICP: protocolo creado por Coinco. Este protocolo, aun muy utilizado por la industria, permite conectar todos los dispositivos a un controlador maestro. Aunque en su origen fue diseñado para el manejo de la parte económica de las máquinas, hoy en dia se puede usar el mismo protocolo para otras funciones.
* MicroMech: Inicialmente fue usado también para controlar el pago y la gestión económica. Actualmente se utiliza sobre todo en las máquinas que disponen de sistema de pago por tarjeta de crédito y sobre todo en territorio norte Americano.
* Simple III/ L+: Fue desarrollado por NRI for simple III y por Coinco. Esto es una extensión de la interfaz anterior que permitía la gestión de más repositorios de monedas.
* Simplex 0: Desarrollado por NRI. Esta interfaz es una de las primeras que incluye controladores para detectar la cantidad de monedas, gestionar el pago con tarjeta, controlador del motor.
* VCSS: Es una interfaz controladora propiedad de las máquinas de la firma de Coca-Cola. Lo destacable de esta interfaz es que posee alimentación por la propia interfaz y un sistema de comunicación de dos vías.
* HII: Desarrollado por MEI. Lo más importante es que este sistema, el cual fue desarrollado en los años 90s, se construye siguiendo el protocolo de modelo OSI.

Estas interfaces, sobre todo en aquellas máquinas que no dispongan un puerto o interfaz de comunicación que permitiesen la auditoría, permitirán mediante un hardware específico controlar las ventas puenteando los sensores y así sabiendo en qué momento actúa.

Las máquinas más modernas empiezan a incorporar otras interfaces de comunicación que sirven tanto para la escritura de dados en la máquina, o para la lectura de información.

A continuación se exponen los tipos de interfaces de tipo auditoría más relevantes:

* EVA-DTS: anteriormente llamado BDV002, define el estándar de la transferencia de datos sobre todo en territorio europeo. Este estándar es el más común de todos y será sobre el que se basará el trabajo.
* DATA-FORMAT: Este protocolo funciona conjuntamente con el anterior. La unidad central de procesamiento de la máquina se conecta a todos los módulos y les dota de una identificación alfa numérica, como por ejempo DT111.
* DATA-CONNECTION Dex/Ucs: Se usa principalmente en las máquinas de los Estados Unidos y la interfaz es un simple Jack que sigue el protocolo de comunicación RS232. Los reportes son transferencias de alto nivel con formato ASCII.
* DATA-CONNECTION óptica: principalmente usa la conexión por infrarojos mediante una comunicación de bajo nivel siguiendo una interfaz basada en el protocolo IrDa.
* DATA-CONNECTION RS232: puerto de comunicación que proporciona datos sobre el registro de ventas. Las máquinas más modernas ya incluyen este puerto.
* DATA-CONNECTION CCTALk :Es otro de los protocolos más modernos que tienen la ventaja de poder utilizar la tecnología de low cost UART.

Existen ya soluciones para descifrar los datos directamente de los buses de conexión, como por ejemplo el MDB [43 - 47], pero también existen soluciones en el mercado que brindan esta característica, como por ejemplo el siguiente que se conecta por RS232:



O por usb con conexiones gsm o wifi:



Vending Machine Controllers

http://www.digitalmediavending.com/vending-machine-controllers/

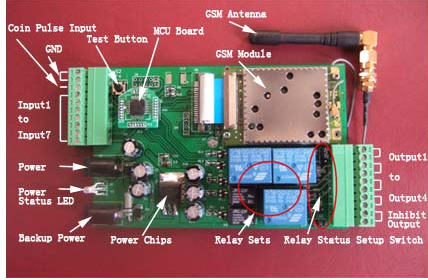
o incluso convertidores de interfaces MDB a placas como RaspberryPi, por lo que supone infinidad de posibilidades.



MDB Converter

http://www.abrantix.com/MDBConverter\_en.html

Existen además otras soluciones que también permiten controlar las máquinas a distancia mediante placas con puertos de entrada y salida, como el siguiente:



Aquellos productos más sofisticados incluyen un servicio cerrado de captura de información, pero esta solo controla los eventos de cuando se produce una venta y el precio obtenido. Además, estos servicios son generalmente de coste elevado y suponen el pago de una cuota mensual. Por otro lado, estos sistemas no son compatibles con las máquinas más antiguas ni tampoco permiten la introducción manual de datos, ni el clasificado de productos, ni tampoco una administración remota.

Esta aplicación al tener una licencia de software de tipo *open source,* permitirá a las medianas y pequeñas empresas acceder a una tecnología moderna y eficiente de manera gratuita y con gran compatibilidad.

Por otro lado, tal y como se verá en los siguientes apartados, el software objeto de este proyecto presentará distintas ventajas sobre las soluciones comerciales actuales.

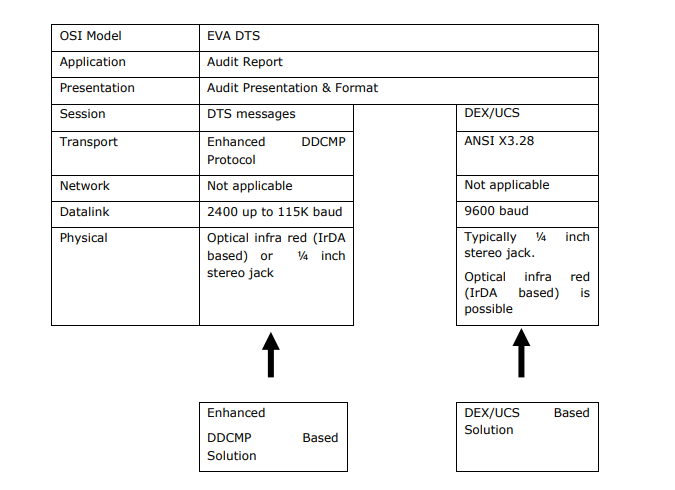
Una de las muchas ventajas que posee frente a la competencia es que está desarrollado con las tecnologías más actuales por lo que brinda eficiencia, durabilidad, escalabilidad y sobre todo movilidad para la persona, dado que es compatible con cualquier dispositivo con conexión a internet con independencia de su ubicación.

Pero la verdadera ventaja es que esta aplicación permite conocer en tiempo real el stock de los productos, de las monedas, de los usuarios y de las ventas que se realicen.

Todo esto permite la gestión eficiente de las máquinas vending, con una interfaz de uso muy sencilla de rápido aprendizaje.

**3.2.2.1 INTERFAZ EVA DTS**

Esta interfaz de auditoría es una de las más modernas y utilizadas en Europa. Sigue el modelo OSI, tal y como se puede ver en la siguiente imagen:



Eva Dts Manual 6.1.2

A partir de los datos recibidos es posible obtener toda la información necesaria para el funcionamiento correcto de la aplicación. Los datos del reporte de dividen en segmentos, siendo cada uno de estos información relevante.

Por ejemplo, un reporte parcial sería este:

VA1\*VA101\*VA102\*VA103\*VA104\*PA1\*1\*25crlf

Cada segmento es dividido del fragmento por un asterisco, por lo que por ejemplo un segmento sería:

PA1\*1\*25crlf

Este último, nos reporta información sobre el precio de un producto. Es decir, el producto situado en la sección registrada como PA101 tiene un precio de 25.

Mediante el puerto se puede obtener información incluso de la unidad de control de la máquina vending, obteniendo datos como número de serie, fabricante, fecha de fabricación y muchos otros.

En cuanto a los datos, estos se pueden obtener desde la ultima inicialización de la auditoría o desde el último reset, todo accesible según los comandos introducidos.

La información que a la aplicación le interesa, son los siguientes:

* Precio aplicado por sección.
* Identificación de evento, que incluye el id de la transacción y el id de la sección de una venta
* Fecha del evento
* Cantidad de producto actual
* Cantidad de monedas actual
* Identificación del producto

Una de las opciones más interesantes es que nos permite poner a la máquina en un estado que en cuanto se emita un evento y se lea, se elimine del registro, siempre y cuando se haya transmitido correctamente. Como punto negativo, para que el funcionamiento sea correcto es necesario que la transferencia de datos siga también el protocolo DEX/UCS o el eDDCMP.

**3.2.3. APLICACIÓN WEB**

Una aplicación web es una herramienta que permite a los usuarios acceder a un conjunto de funcionalidades que se expresan de manera visual a través de la interfaz del navegador desde internet o una intranet.

Dicho de otra forma, es un programa que se ejecuta en el servidor y que permite actuar al navegador web como cliente ligero y que tras una ejecución por parte del usuario, se ejecutan ciertas instrucciones en la máquina obteniendo unos resultados concretos.

La ventaja más importante de estas aplicaciones es su capacidad de ejecutarse en cualquier dispositivo a través de su navegador independientemente del sistema operativo que tenga fuera embebido o no. Además, dada la tecnología actual, siempre existe la posibilidad de la ejecución en cualquier momento y lugar en distintos dispositivos como por ejemplo los *Smartphones*.

Además de lo expuesto, el uso de las aplicaciones permite también el ahorro económico y de tiempo así como la total disponibilidad de los datos de un servidor.

Algunas de las máquinas vending más modernas y actualizadas en cuanto a la tecnología que poseen, permiten por ejemplo el realizar pagos con tecnología RFID[[4]](#footnote-5)o con tarjeta, o controles telemáticos de la temperatura para gestionar adecuadamente la cadena de frío y la seguridad y calidad de los alimentos (37). Existen algunas empresas que están empezando a implementar sistemas de control mediante telemetría que permiten la consulta de datos, sin permitir su modificación in situ ni tampoco a distancia. Además, estas tienen un sistema de control mediante una aplicación un una web basada en estándar como Asp.net.

A pesar de que en la actualidad existen algunas plataformas que implementen ciertas funcionalidades que se desean para este trabajo, no incluyen funcionalidades como la gestión de usuarios, o no están basadas en nuevos estándares como core 2 de Microsoft, el cual permite la ejecución en cualquier servidor con independencia de su sistema operativo.

Al implementar de manera conjunta el control de las máquinas así como al control de los usuarios, permite crear distintas configuraciones de acceso según roles y esto abarca un sinfín de posibles nuevas funcionalidades.

Por último comentar que otras plataformas al no disponen de sistema de gestión del personal no tendrá comunicación entre los usuarios ni tampoco dispondrá de emails de alertas personalizado a cada uno de ellos, funcionalidad que si dispondrá la nueva aplicación.**3.2.4. PATRONES DE DISEÑO**

Para el desarrollo de la aplicación se han utilizado patrones de diseño que permitan facilitar el mantenimiento y actualización del software.

Éstos son soluciones que permiten resolver de manera ordenada y fácil problemas que ocurren de manera reiterada (29). Es decir, ante un problema se plantea una solución de manera que facilita su mantenimiento, reusabilidad y entendimiento.

Los patrones se pueden clasificar en tres categorías diferentes (28):

* Patrón de creación (27): los cuales y de manera generalizada se utilizan para la creación de manera abstracta de los objetos en un lenguaje orientado a los objetos. Estos objetos se crearan de manera dinámica reduciendo la toma de decisión de su creación del objeto solo en el momento de que se establezca la necesidad.
* Patrón estructural: Es utilizado cuando se necesitan estructuras complejas y es menester crear ciertas interconexiones entre los distintos objetos y que esto no sea un problema en el momento que surjan cambios de requisitos en el diseño así como nuevas características.
* Patrones de comportamiento (30): Este tipo de patrón es utilizado para especificar el comportamiento de los objetos en sí de manera individual y el comportamiento entre los distintos objetos que componen el software.

En este proyecto se han utilizado dos patrones de diseño según la necesidad, pero todo el código está basado en el patrón *MVC*[[5]](#footnote-6). Estos patrones que se han utilizado son:

* *Singleton* (20): Usado para controlar el acceso único a la aplicación desde un usuario dado de alta, evitar accesos a rutas de la web sin haber realizado el proceso de *login* y para el proceso de cierre de sesión.
* *MVC* (19): Patrón utilizado por el propio *framework*[[6]](#footnote-7)que separa la lógica, de la interfaz[[7]](#footnote-8) y del modelo de datos.

### 3.2.4.1 MVC

Este tipo de patrón es de tipo estructural el cual separa los datos, de la lógica y de la vista. Es decir, por un lado tenemos la información y los datos que se utilizan, por otro lado tenemos los métodos que controlan los eventos y la lógica de funcionamiento del programa, y por otro lado, tenemos la vista que es lo que se le muestra al usuario .

Tal y como se ha comentado a lo largo del presente proyecto, la aplicación estará basada principalmente en este patrón estructural, de manera que se dividirá la estructura en estos tres roles.

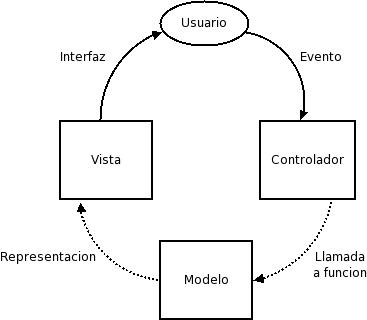
La capa de la vista será la que nos presente la información a través de la interfaz gráfica de la aplicación, esta capa recibirá la información a mostrar a través de los controladores. No obstante, existe la posibilidad de que la capa de la vista pueda trabajar también con la capa de modelo de datos, pero en este trabajo se limitará esta funcionalidad para que exista una definición clara de la estructura.

El usuario como es lógico interactúa con la vista, de manera que esta genera eventos que será capturada por el controlador. El controlador capturará este evento y si es necesario interactuará con la capa del modelo, transformará la información y volverá a comunicarse con la capa de vista para que esta muestre la información nueva

La capa del modelo será aquella que defina la estructura de información disponible. En este caso, la capa de modelo interactuará únicamente con la capa de los controladores. Esta capa define la arquitectura de los datos y esta interactuará directamente con la base de datos, proveyendo la información necesaria para que funcione adecuadamente la aplicación.

La capa de controladores es la capa que interactuará conjuntamente con la capa de datos y con la capa de la vista. Esta es la capa principal ya que será la que obtenga y transforme la información de manera adecuada para que la capa de la vista la muestre. Es decir, es la lógica principal de la aplicación.

En definitiva, al separar el software en estos tres componentes, permite que un cambio en uno de los componentes no afecte al resto, y por otro lado, el código se mantiene bien clasificado para nuevos desarrolladores que deban continuar con el desarrollo del proyecto.



### PATRONES DE DISEÑO Y LA PROGRAMACIÓN ARTÍSTICA -- EN BLACKSHELL

Blackshell.usebox.net. (2018). *Patrones de diseño y la programación artística -- en blackshell*. [online] http://blackshell.usebox.net/archive/patrones-de-diseno-y-la-programacion-artistica.html

**3.2.5. ARQUITECTURA**

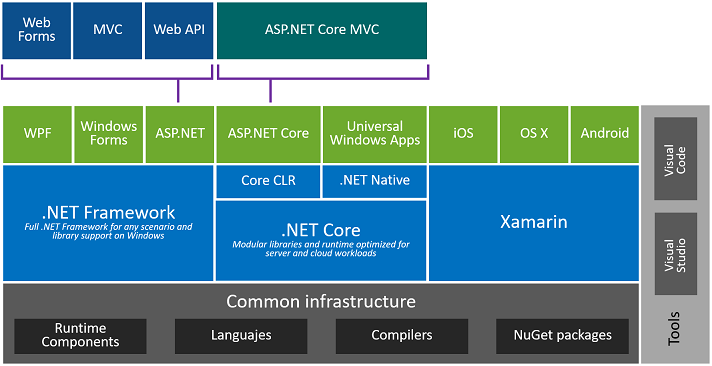
Para el diseño de una aplicación hay infinidad de soluciones aplicables de las que dependen sobre todo de la aplicabilidad que se requiera, sistema operativo del servidor sobre el que se va a alojar y ejecutar y también la compatibilidad con otros sistemas.

Entre todas las arquitecturas disponibles, se ha utilizado la de .Net Core[[8]](#footnote-9) (19), tecnología de Microsoft, la cual es de uso general, modular, multiplataforma y de código abierto.

Entre otras cosas, contiene numerosas APIs[[9]](#footnote-10) de .Net componentes de entorno en tiempo de ejecución, marcos de trabajo, y compilador y prácticamente todas las herramientas para generar una aplicación funcional.Otra arquitectura compatible con los requisitos de diseño del proyecto seleccionado sería la tecnología de Asp.Net, tecnología también patente de Microsoft.

Existe una clara diferencia entre ambas tecnologías, la cual ha inclinado la balanza hacia .Net Core 2. La diferencia es que Asp[[10]](#footnote-11) .net exige definir la arquitectura del sistema operativo del servidor sobre el que se va a hospedar, así como que todos los componentes instalados y utilizados sean compatibles.

En cambio, .Net Core 2 es compatible con la mayoría de los sistemas operativos, por lo que el software se puede migrar entre distintos servidores con distintas arquitecturas sin necesidad de realizar ninguna operación adicional.

****

### backend webpages

(Unitec.azurewebsites.net, 2018)

Unitec.azurewebsites.net. (2018). *Backend*. [online] https://unitec.azurewebsites.net/Backend .

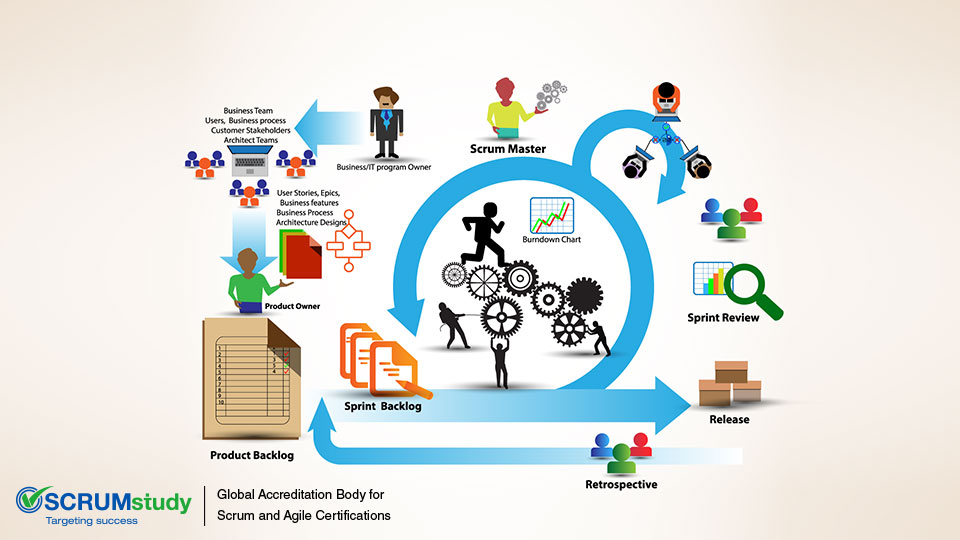
**3.2.6 METODOLOGÍA**

**3.2.6.1. SCRUM**

El uso de tecnologías punteras como .net *core* 2 (6) en un lenguaje moderno orientado a objetos como el C# (12) y *frameworks* utilizados como .net o como *EntityFramework* (11), demandan un tipo de desarrollo que permita adaptarse a las necesidades cambiantes de los requisitos que puedan surgir durante el desarrollo del presente proyecto.

Para ello, se ha decidido utilizar un tipo de metodología de desarrollo ágil, cuya principal característica es el poder adaptarse a estos posibles cambios de manera continua sin interrumpir el desarrollo y permitir así un diseño estructurado y bien determinado.

A pesar de que el proyecto se realiza solamente por una persona, se ha decidido por la metodología *Scrum[[11]](#footnote-12)*(23, 14), la cual permitirá dividir las tareas que conforman en proyecto en tareas y poder clasificarlo de manera que los hitos entregables encajen con los plazos de entrega preestablecidos.



Kaizenia Institute. (2018). *Roles Centrales de Scrum ¡Conforma exitosamente a tu equipo!*.

http://www.kzi.mx/roles-centrales-de-la-metodologia-scrum/

Siguiendo esta metodología, el primer paso, obviando aquellos pasos que impliquen reuniones de equipo como la toma de decisiones grupales o *brainstorming[[12]](#footnote-13)*, es el de organizar las tareas mínimas que conforme viabilidad al proyecto y una estructura adecuada tanto que sirva como soporte al diseño inicial así como aquellas características que se decidieran incorporar en un futuro próximo.

A parte de seguir la metodología básica de *scrum*, también se han seguido los Principios del manifiesto del desarrollo ágil de *scrum*.

**3.3 IDENTIFICACIÓN DE REQUISITOS**

Siguiendo los datos obtenidos en el estudio preliminar, es necesario identificar aquellos requisitos que son base para el desarrollo de la aplicación. Los requisitos se pueden dividir en requisitos funcionales y requisitos no funcionales.

La diferencia principal entre ambos tipos sin entrar en detalles radica en que los primeros son aquellos que especifican una funcionalidad principal o característica requerida y los segundos atienden más a comportamientos, lenguajes o temas relacionados con el sistema.

**3.3.1 REQUISITOS FUNCIONALES**

En este aparatado se identificarán los requisitos funcionales con los siguientes elementos:

* Un código descriptivo
* Nombre
* La fecha de creación
* El autor
* La prioridad

|  |  |
| --- | --- |
| CÓDIGO REQUISITO FUNCIONAL | RQF-1 |
| NOMBRE | Login |
| FECHA CREACIÓN | 07/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-2** |
| NOMBRE | Recuperación |
| FECHA CREACIÓN | 07/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-3** |
| NOMBRE | Error Datos Login |
| FECHA CREACIÓN | 07/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | RQF-5 |
| NOMBRE | Alta de usuario |
| FECHA CREACIÓN | 07/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-6** |
| NOMBRE | Baja de usuario |
| FECHA CREACIÓN | 07/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-7** |
| NOMBRE | Información stock y máquinas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-8** |
| NOMBRE | Usuarios registrados |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-9** |
| NOMBRE | Asignación de máquinas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-10** |
| NOMBRE | Registro de conexión |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-11** |
| NOMBRE | Cierre de sesión |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | RQF-12 |
| NOMBRE | Ayuda |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-13** |
| NOMBRE | Alta máquina |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-14** |
| NOMBRE | Modificación de máquinas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-15** |
| NOMBRE | Borrado máquinas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-16** |
| NOMBRE | Ubicación máquinas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-17** |
| NOMBRE | Items de máquina |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | RQF-18 |
| NOMBRE | Modificación de productos |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-19** |
| NOMBRE | Stock de productos |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-20** |
| NOMBRE | Modificación de stock |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-21** |
| NOMBRE | Alta producto |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-22** |
| NOMBRE | Modificación de producto |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-23** |
| NOMBRE | Borrado de producto |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-24** |
| NOMBRE | Economía |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | RQF-25 |
| NOMBRE | Modificación de monedas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-26** |
| NOMBRE | Mensajería |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-27** |
| NOMBRE | Estadística productos |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-28** |
| NOMBRE | Alertas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-29** |
| NOMBRE | Acceso |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO FUNCIONAL | **RQF-30** |
| NOMBRE | Estadística máquinas |
| FECHA CREACIÓN | 12/05/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |

Tras un análisis de los requisitos definidos y las funcionalidades existentes de las distintas aplicaciones que tengan aspectos similares a las de este trabajo, se consideran que aportan ventajas aquellas que tienen el código de identificación RQF {7,8,9,10,13,14,15,17,18,19,20,21,22,23,26,28}

**3.3.2 REQUISITOS NO FUNCIONALES**

En este aparatado se identificarán los requisitos no funcionales con los siguientes *elementos*:

* Un código descriptivo
* Nombre
* La fecha de creación
* El autor
* La prioridad.

|  |  |
| --- | --- |
| CÓDIGO REQUISITO NO FUNCIONAL | RQNF-1 |
| NOMBRE | Usuario |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-2** |
| NOMBRE | Visualización |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-3** |
| NOMBRE | Búsqueda |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-4** |
| NOMBRE | Seguridad |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | RQNF-5 |
| NOMBRE | Integridad Alta usuario |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-6** |
| NOMBRE | Alertas |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-7** |
| NOMBRE | Comunicación base de datos |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-8** |
| NOMBRE | Patrones |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-9** |
| NOMBRE | Aprendizaje y usabilidad |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Media |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-10** |
| NOMBRE | Información |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-11** |
| NOMBRE | Interfaz gráfica |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Baja |
| CÓDIGO REQUISITO NO FUNCIONAL | RQNF-12 |
| NOMBRE | Lenguaje |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-13** |
| NOMBRE | Seguridad acceso |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-14** |
| NOMBRE | Acceso autorizado |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-13** |
| NOMBRE | Integridad |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |
| CÓDIGO REQUISITO NO FUNCIONAL | **RQNF-14** |
| NOMBRE | Confirmación |
| FECHA CREACIÓN | 08/04/2018 |
| AUTOR | Carlos Moreno Palacios |
| PRIORIDAD | Alta |

Tras un análisis de los requisitos definidos y las funcionalidades existentes de las distintas aplicaciones que tengan aspectos similares a las de este trabajo, se consideran que aportan ventajas aquellas que tienen el código de identificación RQNF {2,4,6,7,11,12}

**3.3.4.1. ROLES**

Los personajes que actúan o participan en scrum se dividen en diferentes roles:

* *Scrum master*: es el principal responsable de asegurar el entendimiento del funcionamiento de scrum a todos los participantes del equipo así como de organizar las reuniones.
* -*Product Owner*: se trata el interesado en el producto o cliente final, el cual deberá definir los requisitos de usuario los cuales se convertirán en tareas para completar el desarrollo de la aplicación.
* -*Dev team*:se considera a todo el equipo de programadores que está implicado en desarrollo de la aplicación.

Todos estos roles serán asumidos por la misma persona en este proyecto, a diferencia que en su ejecución se procederá a simular los requisitos de cada rol de la manera más aproximada posible.

**3.3.4.1. REUNIONES**

Una de las ventajas que proporciona la metodología de desarrollo ágil como *scrum* son las reuniones que se realizan, ya que permiten un seguimiento de todas las tareas.

Los requisitos funcionales y no funcionales derivan en tareas, las cuales pueden dividirse a su vez en subtareas. Estas se organizan en una estructura denominada *product backlog* que actúa como “reserva” donde quedan las tareas pendientes por realizar.

Todo el desarrollo se divide en unas iteraciones denominadas *sprint,* la cual se compone de unas tareas que están valoradas cada una por unos puntos de historia equivalente a dificultad y tiempo necesario para desarrollarlas. Estos puntos de historia son asignadas a cada tarea por parte de todo el equipo.

Cada día el equipo se reúne durante un corto periodo para comentar las dificultades a las que se enfrentan y posibles soluciones. Es decir, en general es una reunión para comentar la situación del día a día del desarrollo.

Cuando se ha finalizado el tiempo establecido para cada *sprint*, es menester la realización de un *sprint review* en el que se muestre mediante una demo las nuevas funcionalidades incorporadas al software. Esta reunión estará formada por todo el equipo de desarrollo, el *scrum master* y por el *product owner*.

Por último, también se realizarán unas reuniones llamadas reuniones de retrospectiva, en las cuales se reúnen el equipo de desarrollo y el *scrum master*, donde se pone en situación aquellos problemas que han surgido en la aplicación de esta metodología, propuestas de mejora, etc.

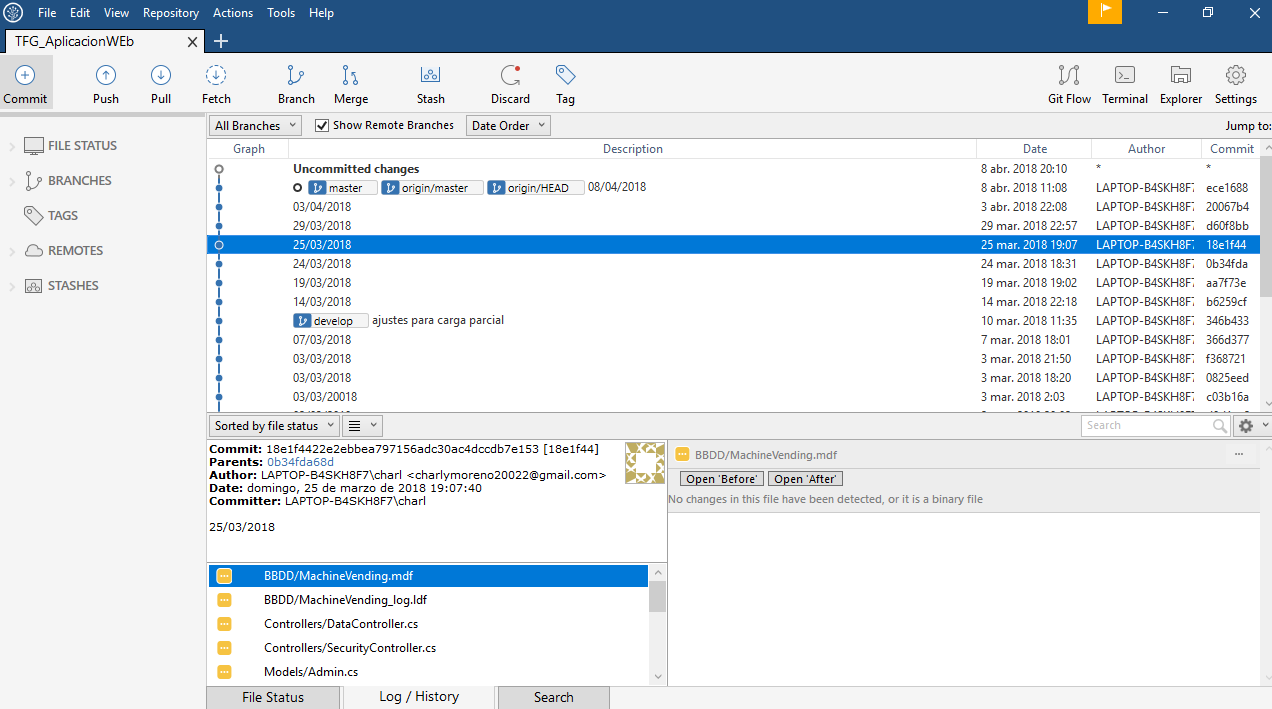
**3.3.5 HERRAMIENTAS**

Las herramientas utilizadas para desarrollar la aplicación son las siguientes:

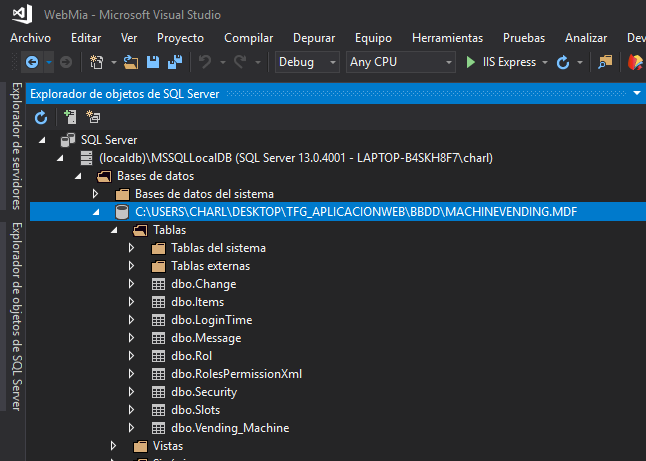
* IDE de desarrollo: Microsoft Visual Studio 2017



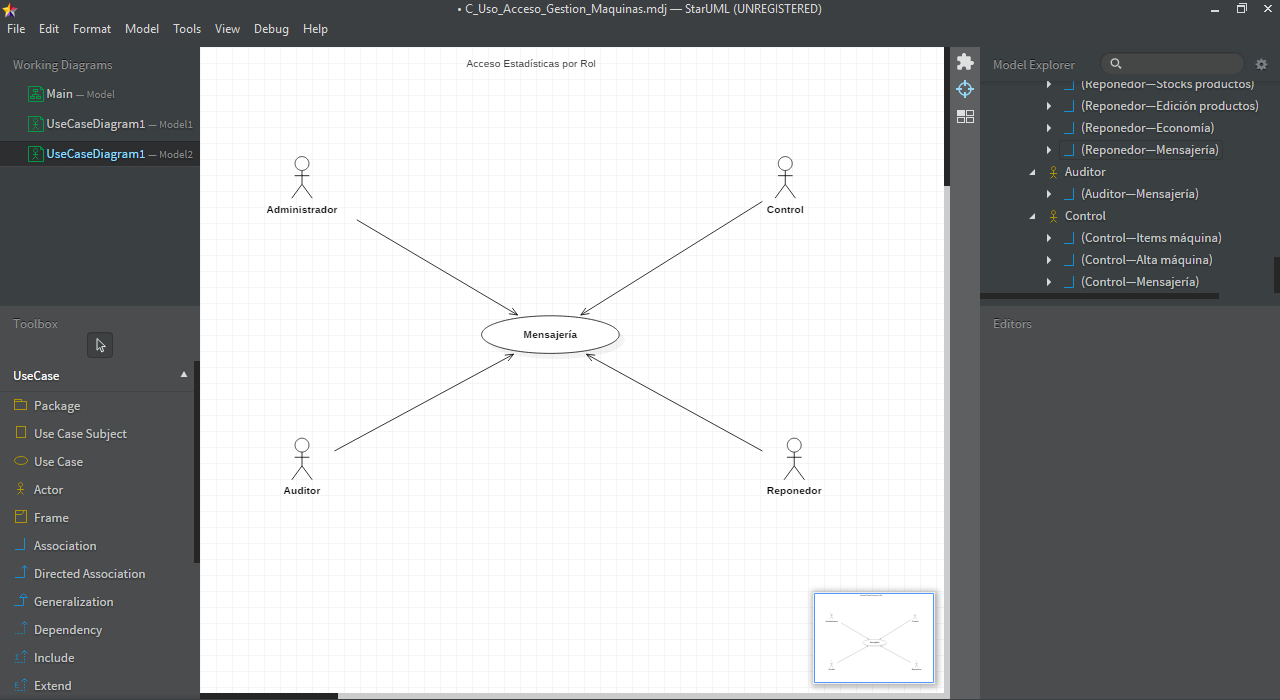
* Control de Versiones: Git con la gestión de Atalassian Source Tree



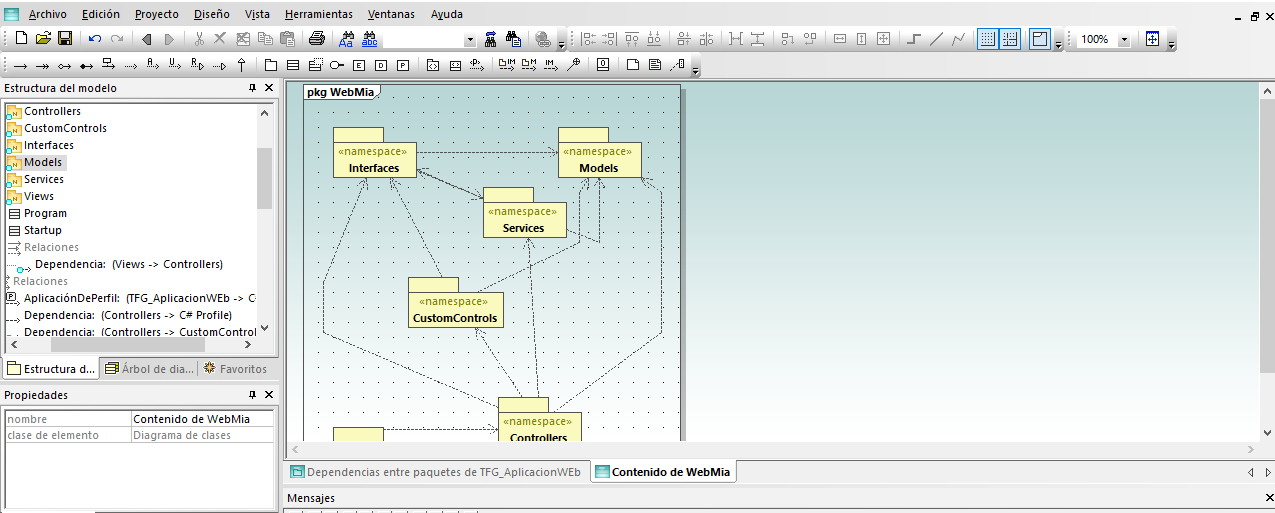
* Base de Datos: Explorador objetos Sql Microsoft Visual Studio 2017



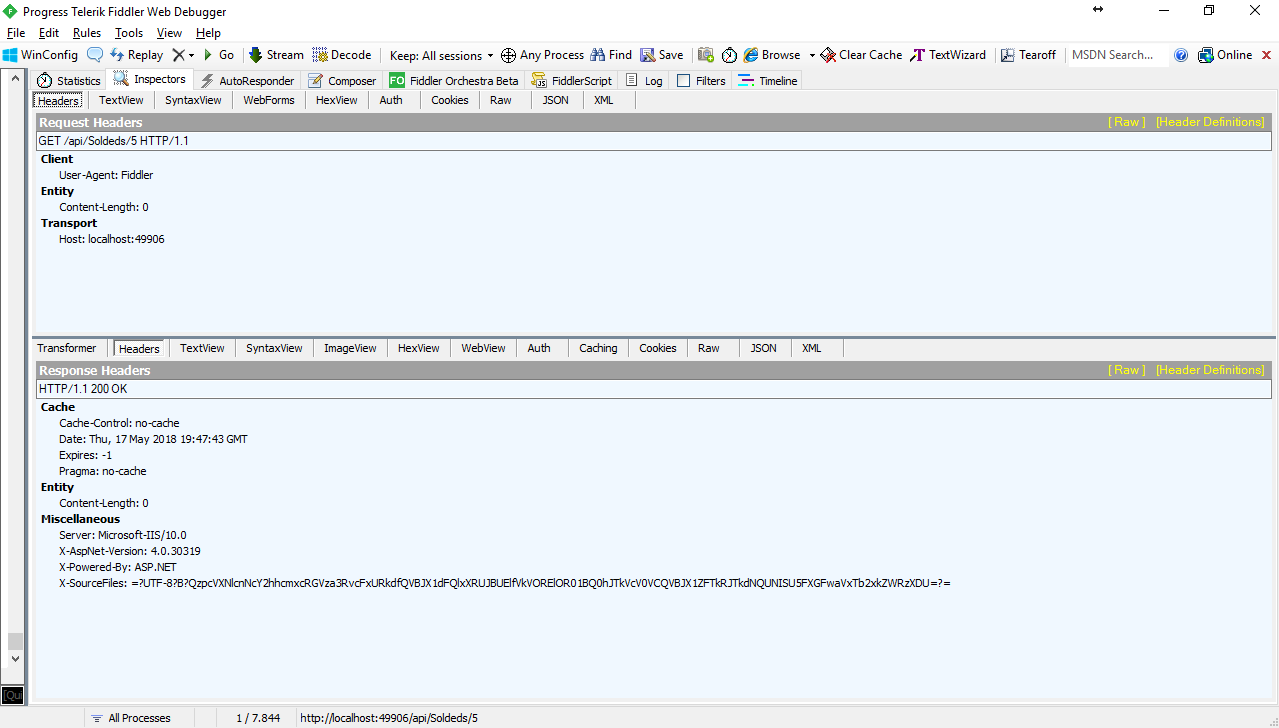
* Diseño UML: Star UML



* Diseño UML: Altova Umodel

****

* Verificación funcionamiento Api Web: Telerik Fiddler Wwb

****

**3.4 DESCRIPCIÓN DEL PROYECTO**

El objetivo del trabajo es crear una aplicación que permita la gestión de las máquinas vending de manera que se obtenga toda la información que facilite el seguimiento de cada máquina así como de los beneficios obtenidos, productos y efectivo del que disponen. No obstante, también se tiene como objetivo la gestión del personal que administran dichas máquinas y por lo tanto es necesario crear un apartado que controle esto.

Para que la aplicación sea completa y funcional, esta debe tener los siguientes apartados divididos en submódulos.

**3.4.1 ANÁLISIS**

**3.4.1.1 PROTOTIPOS DE INTERFAZ**

**3.4.1.1.1 LOGIN, CIERRE DE SESIÓN, INDEX, RECUPERACIÓN DE CONTRASEÑA Y AYUDA**

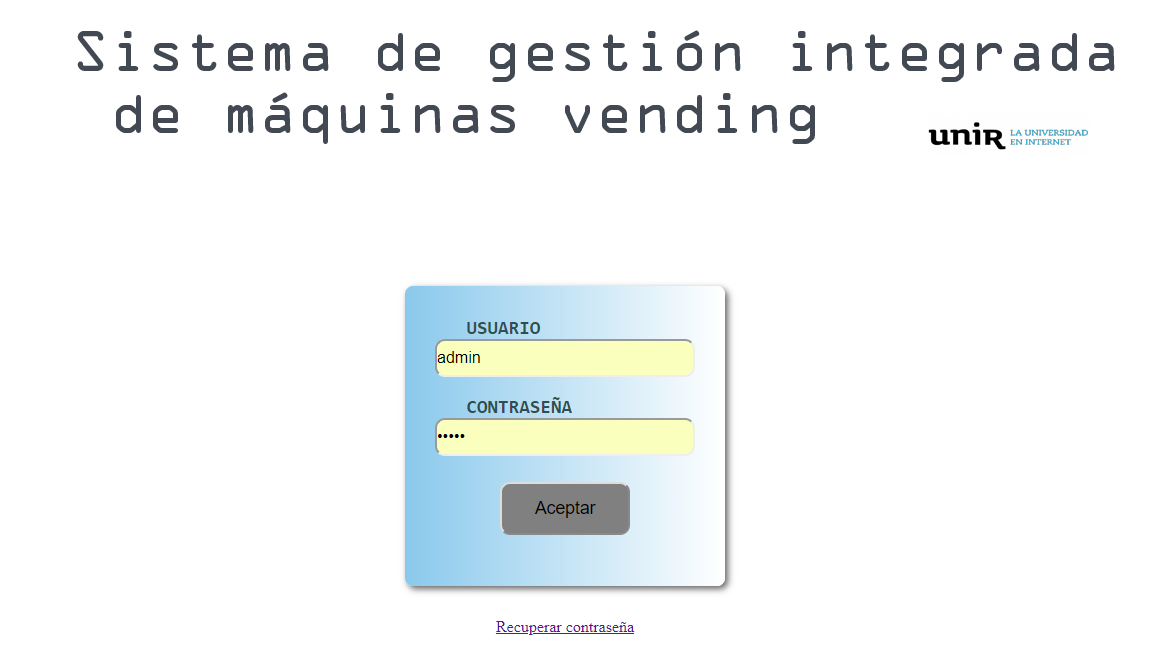
Estas funcionalidades están basadas en controlar el acceso a la aplicación impidiendo el acceso no autorizado. De igual modo también es necesario que las personas puedan cerrar sesión por si se quieren conectar con otro perfil de los disponibles.

Una vez logueado, el sistema mostrará de manera automática aquellos productos y monedas que se encuentran por debajo de un determinado umbral.

En el proceso de login, si el usuario no recordase su contraseña, el sistema le permitirá recuperar la contraseña a través del usuario y su email. La aplicación le enviará una nueva contraseña generada automáticamente al email registrado.

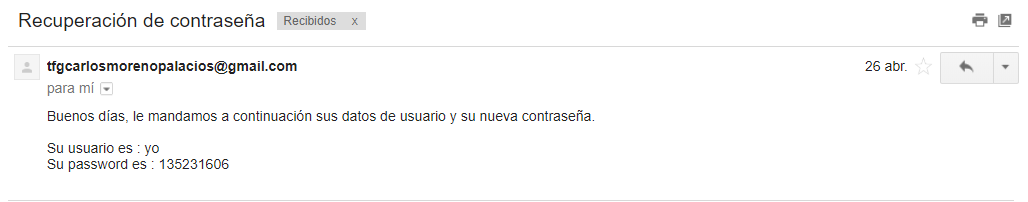
Por último, también existirá un apartado de ayuda en el que se mostrarán datos esenciales de comunicación con el administrador de la aplicación.

* Proceso de login en la aplicación

****

* Proceso de recuperación de contraseña

****

****

* Inicio tras login con alertas

****

* Ayuda



**3.4.1.1.2 GESTIÓN DE MÁQUINAS**

Este apartado se corresponde con la gestión de las máquinas. Es decir, el usuario aquí podrá gestionar todas las máquinas y todo lo relacionado con ellas.

**3.4.1.1.2.1 ALTA DE MÁQUINAS**

En este apartado se podrá dar de alta una nueva máquina permitiendo así gestionarla posteriormente. Una vez concluida la transacción el sistema se actualizará de manera automática sin necesidad de acciones posteriores.



**3.4.1.1.2.2 MODIFICACIÓN- BAJA**

Aquí el usuario podrá realizar cambios en aquellas máquinas registradas. Estos cambios pueden ser el modelo de la máquina así como los datos referentes a la ubicación en la que se encuentre situada.

De igual forma, el usuario también podrá dar de baja aquellas máquinas que seleccione.



**3.4.1.1.2.3 UBICACIÓN**

Este apartado es muy similar al anterior en cuanto a los datos mostrados de cada máquina, a excepción de que no son editables. Por otro lado, el punto fuerte de este apartado es la posibilidad de situar la máquina de manera automática en los planos de la ciudad en la que se encuentra pulsando en un icono.

La situación se realiza a través de la plataforma de Google maps.



****

**3.4.1.1.2.4 ITEMS MÁQUINA**

De todas las funcionalidades de la aplicación, este apartado es uno de los más interesantes e importante.

Aquí el usuario podrá consultar pulsando sobre el enlace de cada máquina todos los productos que contiene dicha máquina.

Además, el contenido que nos muestra esta máquina es editable, pudiéndose elegir en contenido de cada máquina entre los productos registrados en el sistema, su precio y la cantidad de productos que dispone.

Esto es útil sobre todo en los casos en los que el encargado se encuentre manipulando la máquina y cambie el producto que está en venta, o esté reponiendo un producto o por ejemplo que desee cambiar el precio de venta.

Además también se podrá realizar todas estas operaciones a distancia desde cualquier dispositivo con conexión a internet.



**3.4.1.1.2.5 STOCKS PRODUCTOS**

Aquí, el usuario podrá seleccionar un producto y el sistema le mostrará todas aquellas máquinas que venden dicho producto.

Además mostrará el stock de estos productos en cada máquina pudiendo ser editada la cantidad de producto.

Esto es adecuado sobre todo para aquellas ocasiones en que la empresa obtenga de su proveedor una cantidad determinada de producto y pueda así decidir en qué máquina es necesario reponerlo y de qué manera es mejor repartirlos. También es adecuado para saber qué momento es adecuado para solicitar nuevos suministros al proveedor.



**3.4.1.1.2.6 EDICIÓN PRODUCTOS**

Debido a que cada día salen nuevos a la venta nuevos productos y que el mercado cambia de manera constante, se hace necesario añadir, editar o eliminar los productos que tiene la plataforma registrada.

Por ello, este apartado de la aplicación contiene todo aquello necesario para la gestión de los productos.

****

**3.4.1.1.2.7 ECONOMÍA**

Otro factor importante de las máquinas es el tema de las monedas de las que dispone para ofrecer el cambio, ya que un cliente podrá optar por no realizar una compra si la máquina no dispone de cambio.

Este apartado el usuario podrá consultar la cantidad de monedas de cada tipo de la que dispone cada máquina.

Al igual que los demás apartados, el usuario podrá realizar estas actualizaciones in situ o a distancia según le convenga.



**3.4.1.1.3 GESTIÓN DE USUARIOS**

En este apartado, se gestionará todo aquello necesario para el control del personal que gestionan las máquinas de la empresa. Se permitirá asociar a cada usuario una o más máquinas y se registrará su acceso a la plataforma.

**3.4.1.1.3.1 ALTA DE USUARIO**

Aquí el usuario podrá dar de alta a un nuevo usuario de la plataforma. Justo en el momento del registro del nuevo usuario se le asignará un rol de acceso que no se podrá cambiar en ningún momento, por lo que si se cometiera un error o en algún momento posterior se deseara cambiar de rol, será necesario eliminar y volver a darle de alta en la plataforma por temas de seguridad.



**3.4.1.1.3.2 BAJA DE USUARIO**

Aquí podrá el usuario autorizado a eliminar a otro usuario que estén registrado. Dado que es un apartado de vital importancia, este estará restringido a aquellos usuarios con más privilegios.



**3.4.1.1.3.3 CAMBIAR CONTRASEÑA**

El administrador será el que tenga la potestad de dar el alta a un nuevo usuario generando una contraseña a voluntad.

Cuando el usuario no recuerde su contraseña, se le generará otra nueva de manera aleatoria y automática.

Por todo esto, es necesario que el sistema permita modificar las contraseñas , por lo que el sistema generará una nueva de manera automática y aleatoria.



**3.4.1.1.3.4 USUARIOS EN ALTA**

En este apartado el usuario habilitado podrá consultar todos los usuarios que estén registrados en la plataforma.



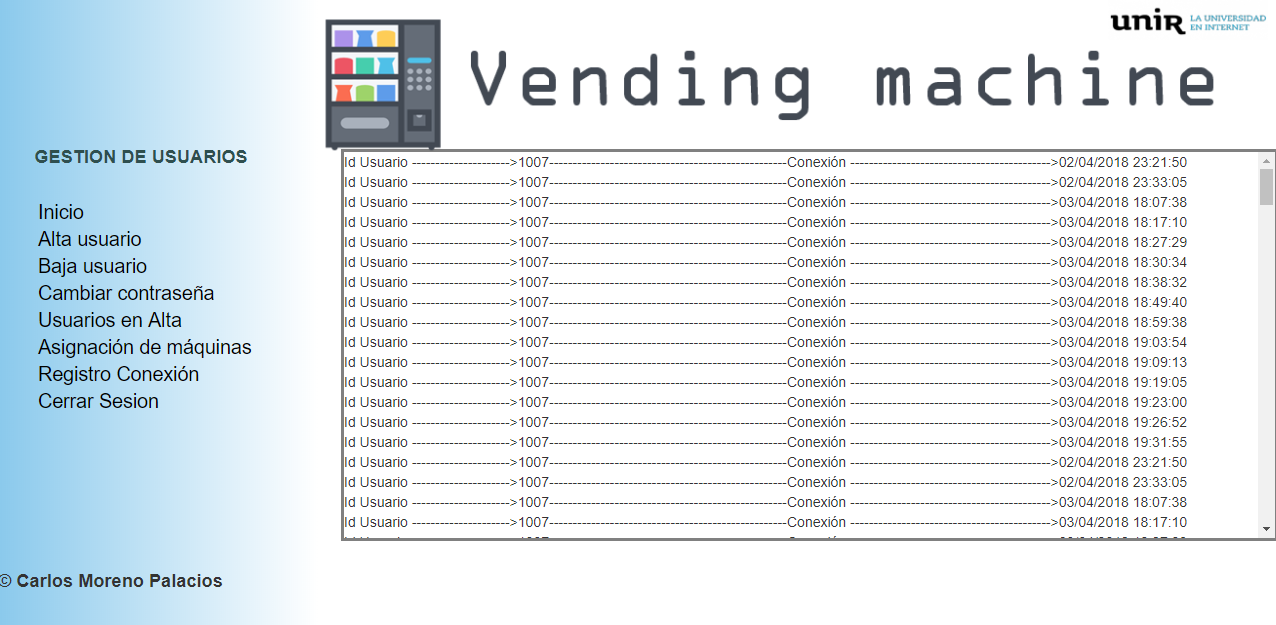
**3.4.1.1.3.5 ASIGNACIÓN DE MÁQUINAS**

Aquí el administrador y/o autorizado, podrá asignar a cada una de las máquinas un usuario, de esta manera será este el encargado de gestionar la nueva máquina.



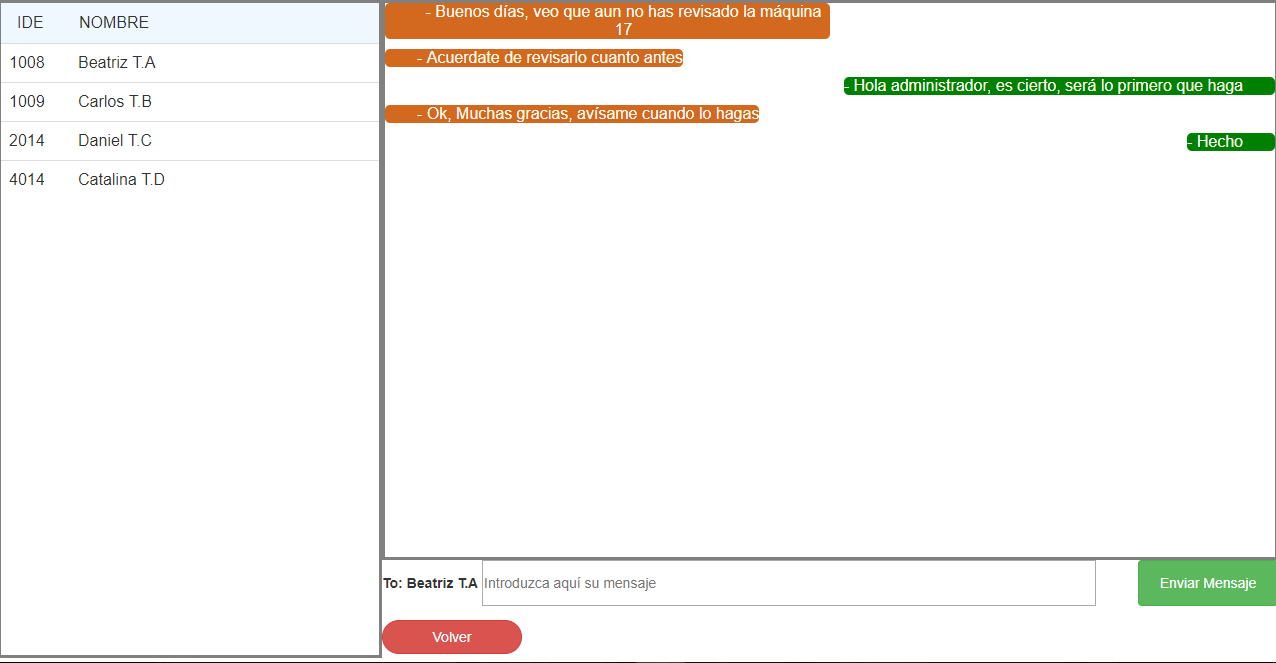
**3.4.1.1.3.6 REGISTRO CONEXIÓN**

En este apartado se podrá consultar todo el registro de los usuarios que han accedido a la plataforma, conociendo el día y la hora a la que han accedido.



**3.4.1.1.4 MENSAJERÍA**

Dado que a veces es necesaria la sincronización entre los empleados de una misma empresa, se ha habilitado un apartado de mensajería interna entre todos los usuarios. Por ello, aquí un usuario podrá contactar con otro y quedará registrada toda la conversación.



**3.4.1.1.5 ESTADÍSTICAS**

En este apartado se mostrará mediante gráficas la evolución de ventas de los productos y el las ventas totales de una determinada máquina.

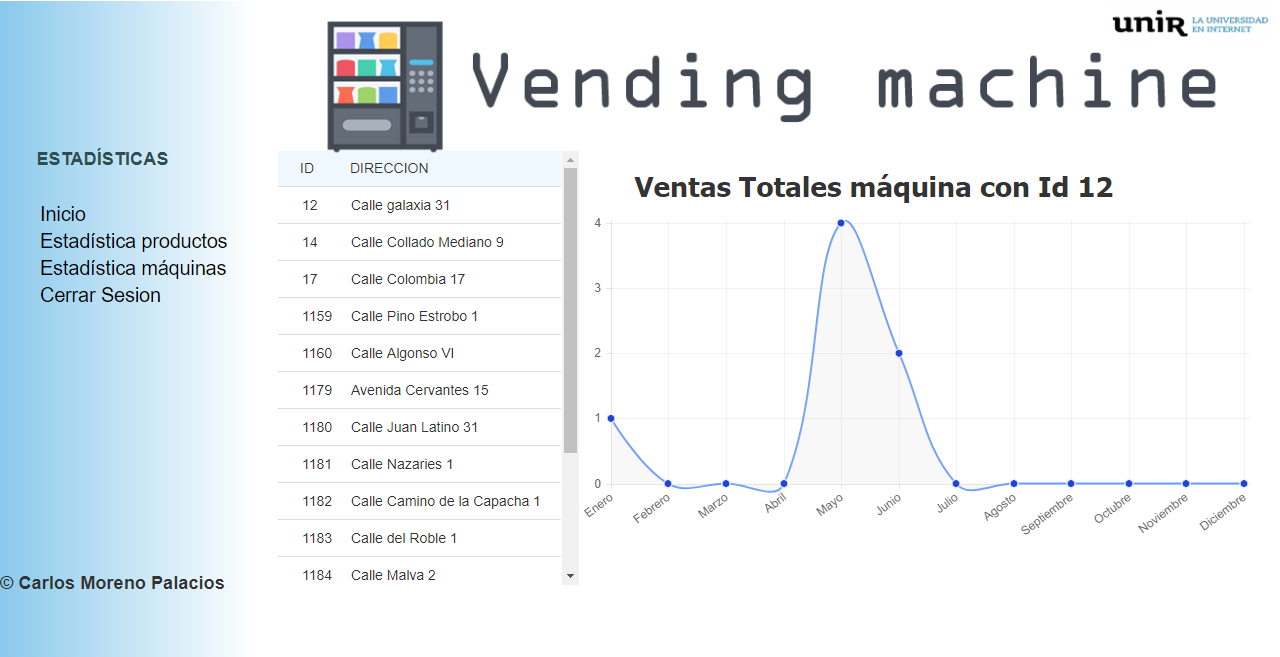
**3.4.1.1.5.1 ESTADÍSTICA PRODUCTOS**

En este apartado el usuario podrá visualizar mediante la selección de un producto la evolución de ventas de dicho producto de manera que se podrá comprobar la acogida que tiene dicho producto en el mercado.



**3.4.1.1.5.2 ESTADÍSTICA MÁQUINAS**

Aquí el usuario podrá consultar mediante la selección de una máquina el número de ventas totales de todos los productos que posee, de manera que así podrá obtener la información necesaria para determinar la viabilidad de la máquina en dicha ubicación.



**3.4.1.1.6 ALERTAS POR EMAIL**

El sistema enviará cada doce horas de manera periódica una alerta a cada email registrado con la información sobre niveles bajos de productos y monedas de cada máquina asignada a dicho usuario.

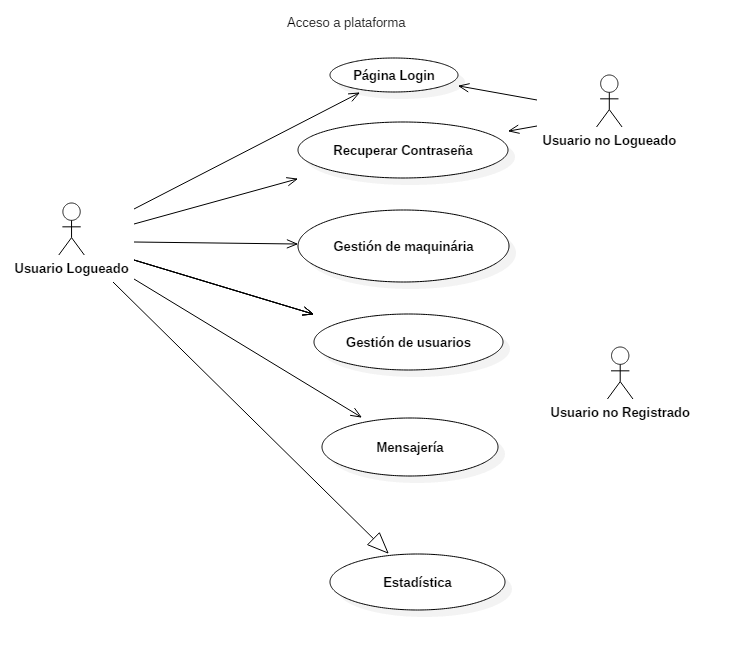


**3.4.1.2 DIAGRAMA DE CASOS DE USO**

En los siguientes apartados se mostraran los distintos diagramas de caso de uso de acceso de las funcionalidades según los usuarios y los roles disponible.

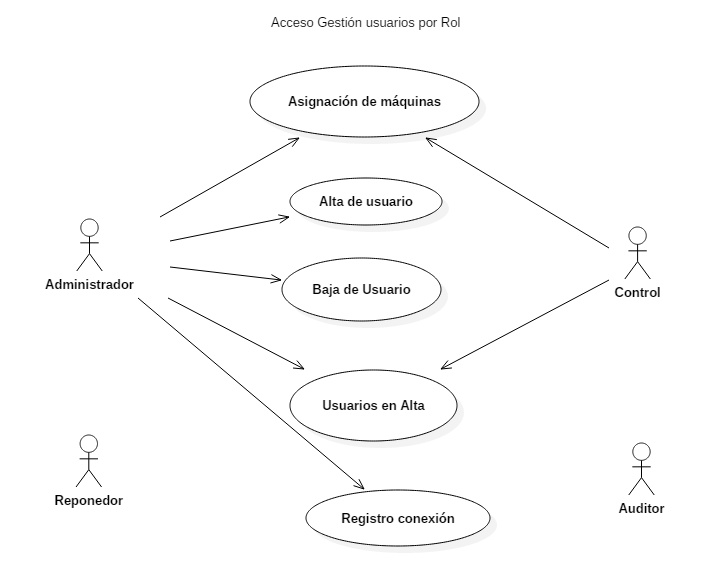
**3.4.1.2.1 ACCESO A LA PLATAFORMA**

El siguiente diagrama se corresponde al caso de uso del *login* a la plataforma por los distintos tipos de usuarios:



**3.4.1.2.2 ACCESO GESTIÓN USUARIOS SEGÚN ROL**

En este caso de uso se muestra el acceso a los módulos de la gestión de usuarios según el rol que tenga asignado un usuario en particular.



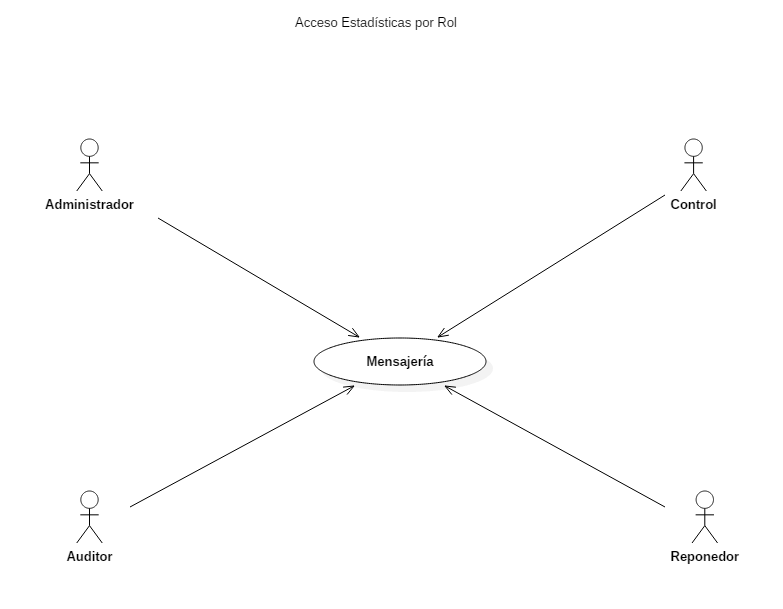
**3.4.1.2.3 ACCESO GESTIÓN MÁQUINAS SEGÚN ROL**

En este apartado se muestra el caso de uso de acceso al apartado de la gestión de máquinas según el rol asignado al usuario actual.

****

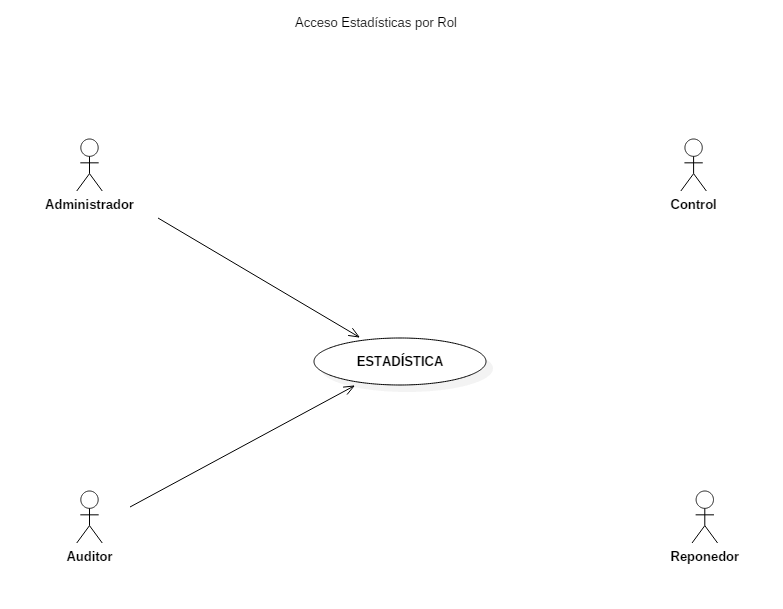
**3.4.1.2.4 ACCESO MENSAJERIA SEGÚN ROL**

En este caso de uso de muestra el acceso al apartado de mensajería según el rol asignado al usuario actual.

****

**3.4.1.2.5 ACCESO ESTADÍSTICA SEGÚN ROL**

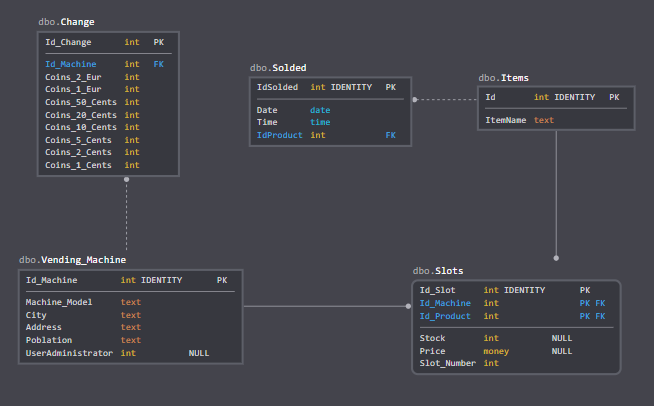
En este apartado se muestra el caso de uso de acceso al módulo de estadística según el rol asignado al usuario actual.



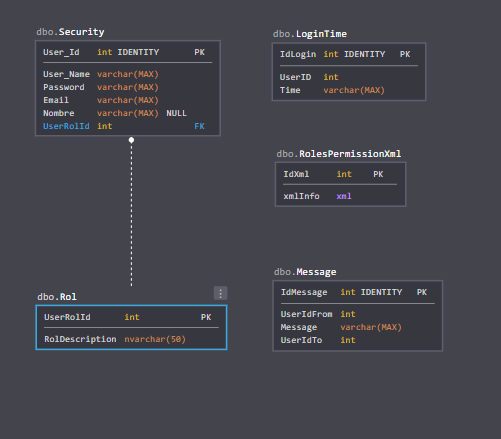
**3.4.1.3 MODELO DE BASE DE DATOS**

Aquí se muestra el diagrama de la base de datos que se utiliza para la aplicación, incluyendo las relaciones entre ellos y su tipo.

* La tabla dbo.Vending\_Machine contiene toda la información sobre cada una de las máquinas, incluyendo datos referentes a su ubicación y al usuario que la administra.
* La tabla dbo.Change contiene toda la información de las monedas de una máquina, por ello tiene una clave foránea a la clave primaria de la tabla de dbo.Vending\_Machine.
* La tabla dbo.Slots contiene información de cada uno de los slots de cada máquina. En cada slot existirá un determinado producto. Por ambas, esta tabla tiene clave foránea a la clave primária de la tabla dbo.Vending\_Machine y a la clave primaria de la tabla dbo.Items.
* La tabla dbo.Items contiene la descripción de cada uno de los productos dado de alta en la base de datos.
* La tabla dbo.Solded contiene toda la información referente a los productos vendidos de cada máquina, por ello tiene clave foráneas a las claves primárias de la tabla dbo.Vending\_Machine y a la tabla dbo.Items.

****

* La tabla dbo.Security contiene toda la información referente a los usuarios incluyendo los datos relevantes para el uso de la plataforma como por ejemplo el tipo de rol asignado. Por esto último, se hace referencia a la clave primaria mediante una clave foránea a la tabla dbo.Rol.
* La tabla dbo.Rol contiene los tipos de rol de uso permitidos en la aplicación.
* La tabla dbo.LoginTime contiene la información del registro de login de cada usuario en la aplicación.
* La tabla dbo.RolesPermissionXml contiene un xml con la información de acceso de cada rol a cada punto distinto de la aplicación.
* La tabla dbo.Message contiene todos los mensajes enviados entre los usuarios de la aplicación.

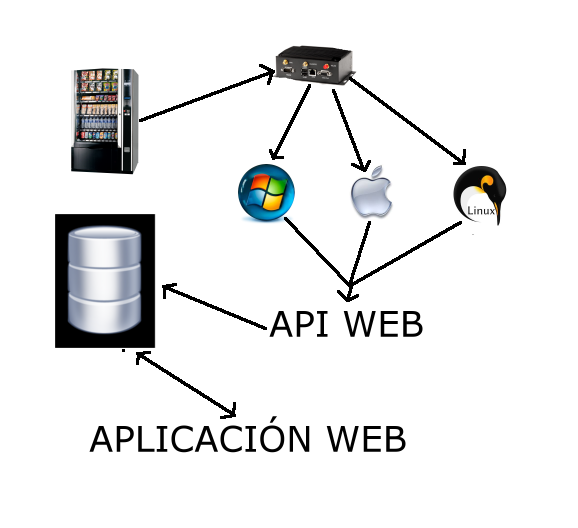
****

**3.4.2 DISEÑO DE LA SOLUCIÓN**

Este apartado versará sobre el diseño de alto nivel de la aplicación y de todo lo relacionado para que esta sea funcional.

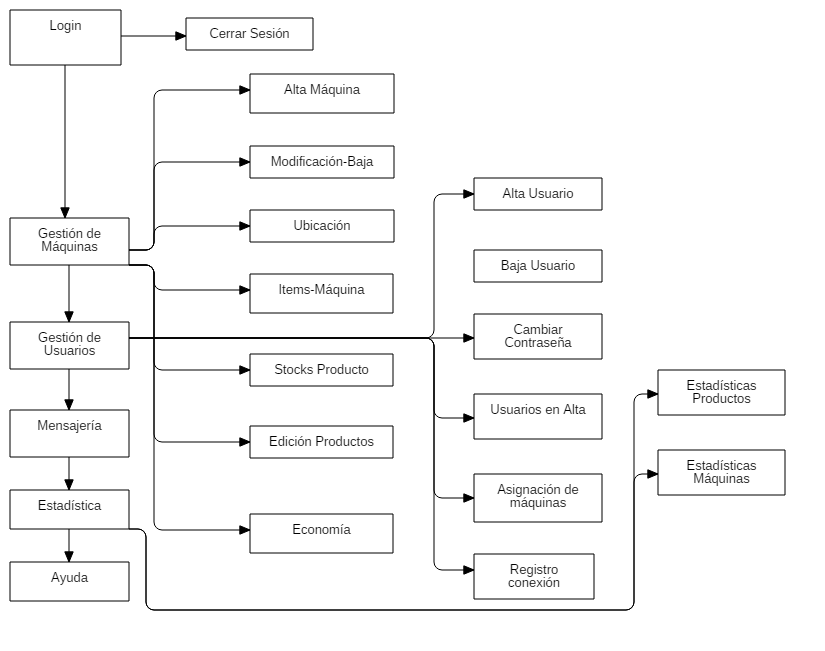
**3.4.2.1 ARQUITECTURA**

Tal y como se pudo observar en apartados anteriores, existe infinidad de dispositivos hardware que permiten la obtención de los datos mediante las interfaces de conexión que disponen. Pero esto solo es posible en las máquinas más modernas y más caras. Debido a esto, se creará una API WEB sobre la cual podrá trabajar cualquier dispositivo que se implemente en una máquina. Como ejemplo se muestra una de los productos que se conectan a las interfaces de la máquinas, pero su puede sustituir por cualquier producto de mercado, incluido las conocidas placas de programación como Arduino o RaspberryPi.

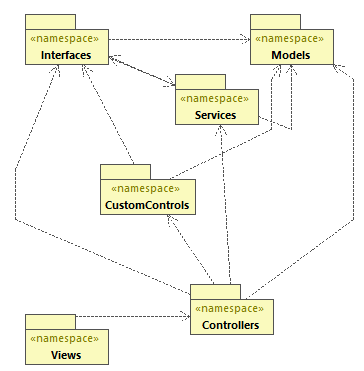


**3.4.2.2 PROTOTIPO DE LA APLICACIÓN**

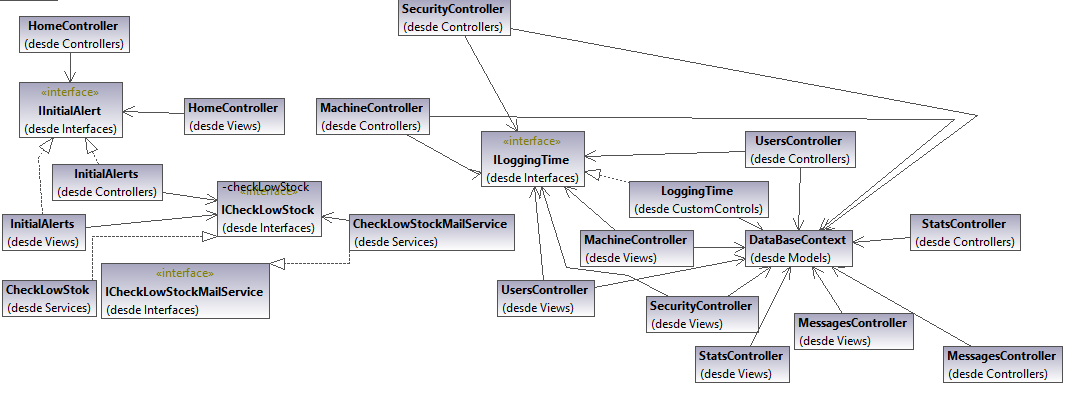
El esquema general de la aplicación seguirá el siguiente prototipo:

****

**3.4.2.3 DISEÑO DE LA SOLUCIÓN**

****

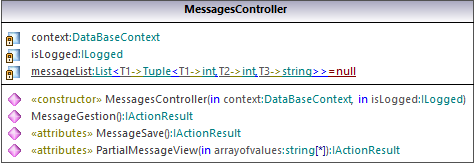
* Diagrama Uml Completo

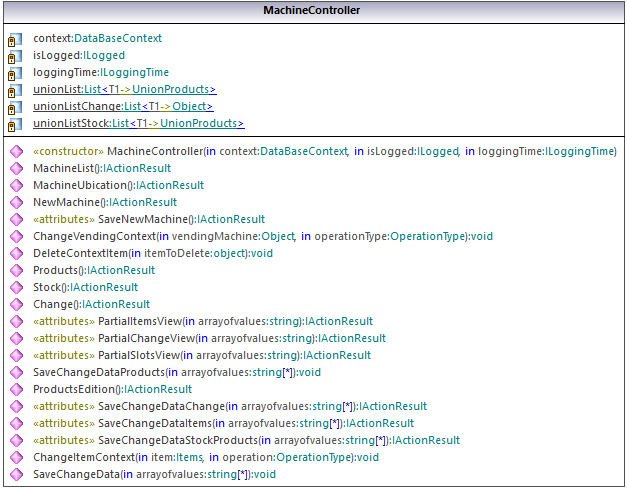
****

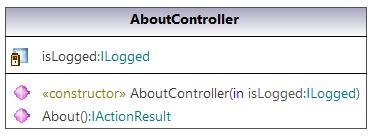
Nota: En la construcción de este diagrama se ha obviado la clase UserLogged y la interfaz ILogged, debido a que todas las clases del diagrama están relacionadas con estas y dificultaría la comprensión.

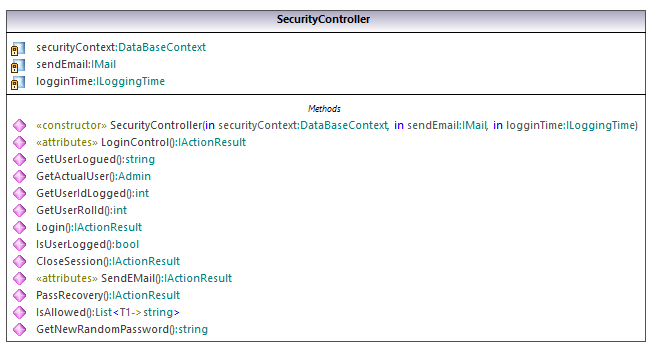
**3.4.3 IMPLEMENTACIÓN**

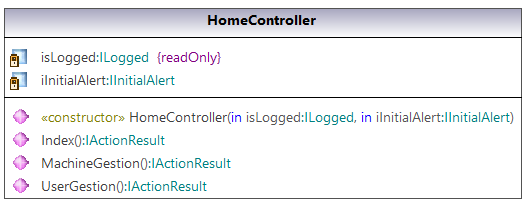
**3.4.3.1 CONTROLADORES MVC**

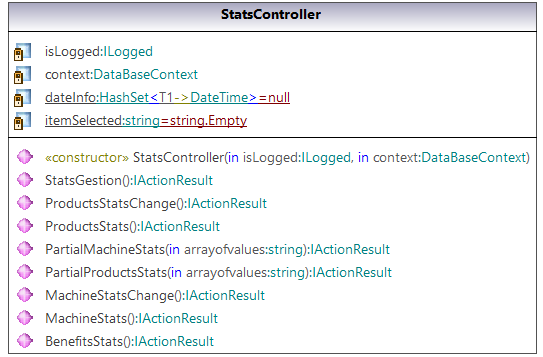


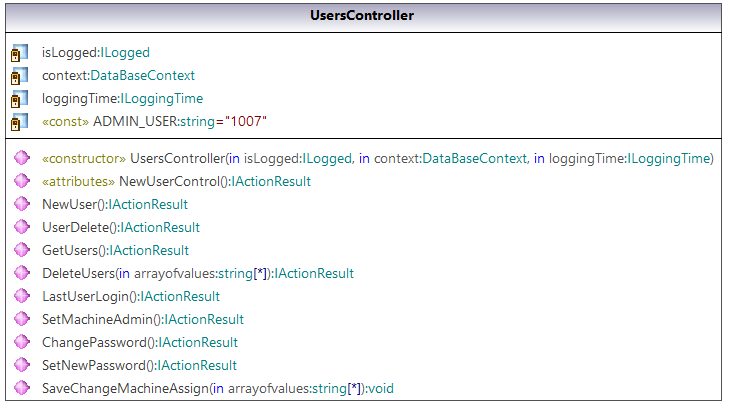






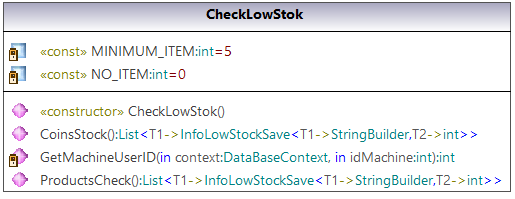


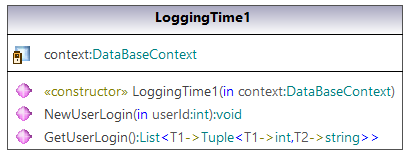


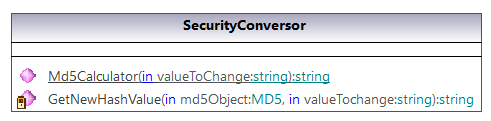


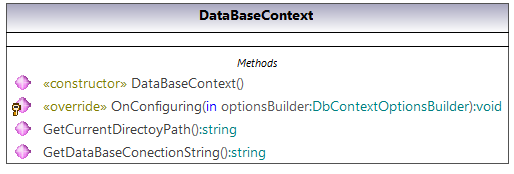
**3.4.3.2 CONTROLADORES PERSONALIZADOS**

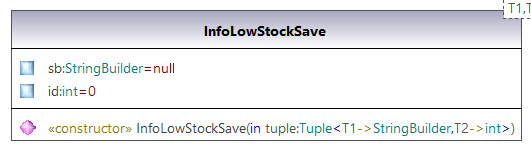


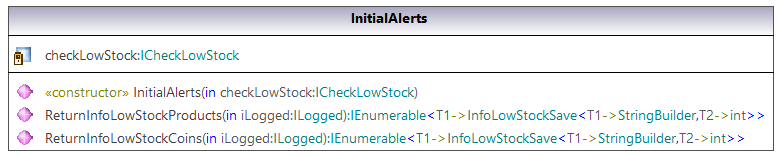


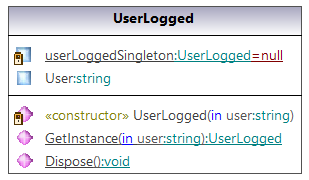




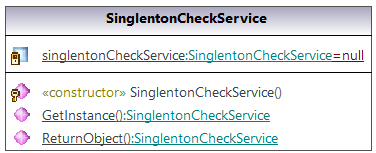


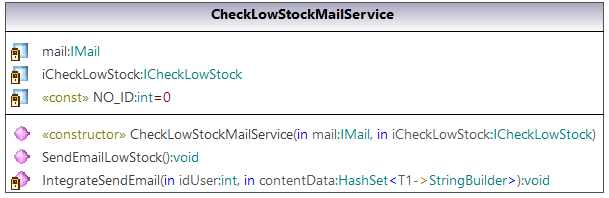




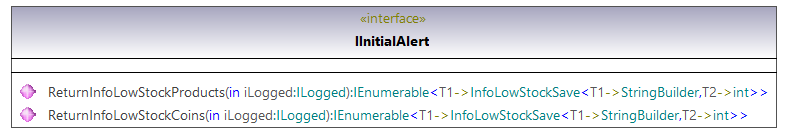


**3.4.3.3 SERVICIOS**

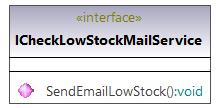




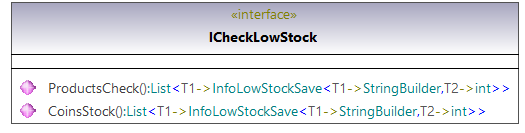
**3.4.3.4 INTERFACES**

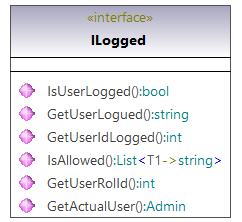












**3.5 DESCRIPCIÓN TÉCNICA**

**3.5.1 FRAMEWORKS UTILIZADOS**

* Net framework 4.6.1: Framework esencial para la compilación y ejecución de la solución basada en core 2 de Microsoft.
* Entity Framework 6.2.0-Beta: Es el framework utilizado como intermediario en la comunicación con una base de datos mediante la creación de modelos que contenga las columnas de cada tabla. Este framework elude la necesidad de crear conexiones a la base de datos y de su gestión así como el ahorro de crear querys específicas para la consulta sobre la base de datos.

**3.5.2 CLASES**

**3.5.3 MÉTODOS**

**3.5.4 MODELOS**

**3.5.5 CONTROLADORES**

**3.5.6 CONTROLES PARTICULARES**

**3.5.7 VISTAS**

**3.5.8 SERVICIOS**

**3.5.9 DEPENDENCIAS**

**3.5.10 INTERFACES**

**3.5.11 BASE DE DATOS**

**3.5.12 IMPLEMENTACIÓN MVC**

**ANEXO 1. DESCRIPCIÓN REQUISITOS FUNCIONALES**

|  |  |
| --- | --- |
| CÓDIGO REQUISITO FUNCIONAL | DESCRIPCIÓN |
| RQF-1 | El sistema deberá controlar el acceso a la plataforma mediante un sistema de control de usuario y contraseña. |
| RQF-2 | El sistema deberá permitir acceder a los usuarios que hayan olvidado la contraseña mediante la generación de una nueva contraseña cuando coincidan email y la identificación de usuario mediante el envío de un email. |
| RQF-3 | El sistema deberá detectar e informar al usuario si los datos en el proceso de login son incorrectos o si el formato introducido no es válido. |
| RQF-5 | El sistema deberá permitir dar de alta a un nuevo usuario/trabajador en la plataforma, indicando nombre completo, rol, usuario, contraseña y email. |
| RQF-6 | El sistema deberá permitir dar de baja un usuario existente de la plataforma eliminando todos los datos asociados. |
| RQF-7 | El sistema deberá informar al realizarse el login del stock actual y de las máquinas asociadas al usuario logueado. |
| RQF-8 | El sistema deberá dar información sobre los usuarios datos de alta en la plataforma y su rol asociado. |
| RQF-9 | El sistema deberá permitir asociar una máquina determinada a cada usuario mostrando el rol específico de cada usuario. |
| RQF-10 | El sistema deberá registrar y mostrar los datos de conexión y login a la plataforma sin exclusión. |
| RQF-11 | El sistema deberá permitir cerrar sesión del acceso a la plataforma. |
| RQF-12 | El sistema deberá mostrar información de ayuda sobre la plataforma y acceso a información de comunicación con el administrador. |
| RQF-13 | El sistema deberá permitir dar de alta una nueva máquina indicando modelo, ciudad, dirección, población y número de slots. |
| RQF-14 | El sistema deberá permitir modificar datos de cada máquina, incluyendo modelo, ciudad, población y dirección. |
| RQF-15 | El sistema deberá permitir eliminar por completo las máquinas seleccionadas y sus datos asociados. |
| RQF-16 | El sistema deberá mostrar la ubicación de la máquina posicionándola en google maps |
| RQF-17 | El sistema deberá mostrar los ítems que tiene cada máquina incluyendo el slot en el que se encuentra, la cantidad y el precio. |
| RQF-18 | El sistema deberá permitir modificar los productos de los slots de cada máquina, su cantidad actual y el precio. |
| RQF-19 | El sistema deberá dar información sobre la distribución de un productos entre todas las máquinas registradas y su stock actual |
| RQF-20 | El sistema deberá permitir modificar el stock de un producto concreto entre todas las máquinas |
| RQF-21 | El sistema deberá permitir dar de alta un nuevo producto. |
| RQF-22 | El sistema deberá permitir modificar un producto existente por otro |
| RQF-23 | El sistema deberá permitir borrar un producto existente. |
| RQF-24 | El sistema deberá mostrar las monedas que posee cada máquina, incluyendo las monedas de curso legal del estado de España. |
| RQF-25 | El sistema deberá permitir modificar la cantidad de monedas de cada máquina. |
| RQF-26 | El sistema deberá permitir la comunicación mediante mensajería interna. |
| RQF-27 | El sistema deberá dar información estadística sobre las ventas de productos de cada máquina |
| RQF-28 | El sistema deberá enviar un email al usuario asignado cuando una máquina tenga un producto con un stock por debajo de umbral establecido. |
| RQF-29 | El sistema deberá permitir el acceso y visualización de los enlaces solo a aquellos que tengan autorización para ello |
| RQF-30 | El sistema deberá mostrar datos estadísticos sobre las ventas totales de cada máquina |
|  |  |

**ANEXO 2. DESCRIPCIÓN REQUISITOS NO FUNCIONALES**

|  |  |
| --- | --- |
| RQNF-1 | El sistema deberá almacenar los datos de usuario como id usuario y contraseña mediante cifrado MD5 |
| RQNF-2 | El sistema debe visualizarse en cualquier navegador y funcionar en cualquier tipo de servidor independientemente de su sistema operativo |
| RQNF-3 | El sistema no deberá tardar más de 5 segundos en mostrar los datos al realizar una selección o búsqueda. |
| RQNF-4 | El sistema deberá permitir acceder a un usuario a los datos que tenga acceso sin perjudicar otros datos. |
| RQFN-5 | El sistema deberá validar si el usuario introducido existe y si ambas contraseñas introducidas coinciden para evitar duplicados en la base de datos. |
| RQNF-6 | El sistema deberá emitir informes por email cada 12 horas desde que se inicie la aplicación. |
| RQNF-7 | El sistema deberá permitir la comunicación de la base de datos desde las máquinas de vending. |
| RQNF-8 | El sistema deberá utilizar patrones de creación para el login de usuarios. |
| RQNF-9 | El tiempo de aprendizaje de la plataforma deberá ser inferior a 2 horas en tiempo medio. | |
| RQNF-10 | El sistema deberá informar adecuadamente de los errores de manera comprensible por el usuario medio. | |
| RQNF-11 | El sistema deberá tener una interfaz gráfica bien definida y adaptada. | |
| RQNF-12 | El sistema deberá basarse en el patrón de comportamiento MVC y usar core2 .net framework con lenguaje c# y razor. | |
| RQNF-13 | El sistema deberá impedir el acceso no autorizado a los datos protegidos | |
| RQNF-14 | El sistema solo permitirá el acceso a las vistas solo cuando el usuario se haya logueado correctamente. | |
| RQNF-15 | El sistema no deberá mostrar datos una vez sean eliminados por el mismo o por otro usuario. | |
| RQNF-16 | El sistema deberá confirmar los cambios introducidos antes de modificar la base de datos. | |

**X REFERENCIAS BIBLIOGRÁFICAS**

### lA Historia de Máquinas vendedoras DE LIBROS [1]

Pico, R. La historia de las máquinas expendedoras de libros http://www.libropatas.com/listas/maquinas-expendedoras-libros

### EXPENDEDORA DE LIBROS. TIENDAS ABIERTAS 24 HORAS [2]

Eravending.es. Expendedora de Libros. Tiendas abiertas 24 horas. http://www.eravending.es/maquinas/ver/61/expendedora-de-libros

### CONOCE LA HISTORIA DE LAS MÁQUINAS EXPENDEDORAS [3]

Autosnack.com.co. Conoce la historia de las máquinas expendedoras. http://www.autosnack.com.co/historia-de-las-maquinas

### HOW VENDING MACHINE IS MADE - MATERIAL, MANUFACTURE, HISTORY, USED, PARTS, COMPONENTS, STRUCTURE, PRODUCT, INDUSTRY, HISTORY [4]

Madehow.com. How vending machine is made - material, manufacture, history, used, parts, components, structure, product, industry, History.

http://www.madehow.com/Volume-7/Vending-Machine.html

### VENDING | ARTICLES | VENDING TIMES [5]

Vendingtimes.com. Vending | Articles | Vending Times.

https://www.vendingtimes.com/articles?taxonomy=VendingFeatures

### MICROSOFT BLOG [6]

Cesar de la Torre [Microsoft] - BLOG.

https://blogs.msdn.microsoft.com/cesardelatorre/2016/06/27/net-core-1-0-net-framework-xamarin-the-whatand-when-to-use-it/

### MICROSOFT ASP.NET [7]

Unitec.azurewebsites.net. Backend.

https://unitec.azurewebsites.net/Backend

### PATRONES DE DISEÑO Y LA PROGRAMACIÓN ARTÍSTICA -- EN BLACKSHELL [8]

Blackshell.usebox.net. Patrones de diseño y la programación artística -- en blackshell

http://blackshell.usebox.net/archive/patrones-de-diseno-y-la-programacion-artistica.html

### NOVEDADES PARA EL DESARROLLO WEB EN .NET CORE 2.0 – BIT [9]

Novedades para el desarrollo Web en .NET Core 2.0 – Bit.

https://www.bit.es/knowledge-center/novedades-para-el-desarrollo-web-en-net-core-2-0/

### INFORMACIÓN GENERAL SOBRE ENTITY FRAMEWORK [10]

Docs.microsoft.com. Información general sobre Entity Framework

https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/ef/

### ENTITY FRAMEWORK [11]

Msdn.microsoft.com. Tutorial rápido (Entity Framework)

https://msdn.microsoft.com/es-es/library/bb399182(v=vs.100).aspx

### gUÍA PROGRAMACIÓN CSHARP[12]

Docs.microsoft.com.  Guía de programación de C#.

https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/

### INGENIERÍA DE SOFTWARE [13]

Software engineering

Sommerville, I. Software engineering. Wokingham, Eng: Addison-Wesley.

### NEW ERA FOR INSULATION [14]

Babyak, Richard J. "New Era for Insulation (Change is in the Wind for Blowing Agents)." Appliance Manufacturer 41, no. 8 (August 1993): 47-48.

### STARUML [15]

Staruml.io.  StarUML

http://staruml.io/support

### INTRODUCCIÓN A LAS CONSULTAS LINQ (C#) [16]

Docs.microsoft.com.  Introducción a las consultas LINQ (C#)

https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries

### cURSO UML [17]

La web del programador, curso Uml

https://www.lawebdelprogramador.com/cursos/UML/index1.html

### TUTORIALES DE SQL SERVER [18]

Docs.microsoft.com. Tutoriales de SQL Server

https://docs.microsoft.com/es-es/sql/sql-server/tutorials-for-sql-server-2016

### INTRODUCCIÓN A ASP.NET CORE MVC Y VISUAL STUDIO [19]

Docs.microsoft.com. Introducción a ASP.NET Core MVC y Visual Studio

https://docs.microsoft.com/es-es/aspnet/core/tutorials/first-mvc-app/start-mvc?tabs=aspnetcore2x

### EL PATRÓN SINGLETOn [20]

Msdn.microsoft.com. El Patrón Singleton

https://msdn.microsoft.com/es-es/library/bb972272.aspx

### iNSERCIÓN DE DEPENDENCIAS EN ASP.NET CORE [21]

Docs.microsoft.com. Inserción de dependencias en ASP.NET Core. https://docs.microsoft.com/es-es/aspnet/core/fundamentals/dependency-injection

### METODOLOGÍAS DE DESARROLLO DE SOFTWARE [22]

Metodologías de desarrollo de software, Juntaex.es

http://www.juntaex.es/filescms/con01/uploaded\_files/dgaeti/IngenieriaSoftwareGobex.pdf

### SCRUM [23]

Proyectos Ágiles.  Qué es SCRUM.

https://proyectosagiles.org/que-es-scrum/

### SCRUM 2 [24]

Kaizenia Institute.  Roles Centrales de Scrum ¡Conforma exitosamente a tu equipo!.

http://www.kzi.mx/roles-centrales-de-la-metodologia-scrum/

### PRINCIPIOS DEL MANIFIESTO ÁGIL [25]

Agilemanifesto.org.  Principios del Manifiesto Ágil

http://agilemanifesto.org/iso/es/principles.html

**Historia del vending ¿cómo eran las primeras máquinas expendedoras? [26]**

Vendival

https://www.vendival.com/historia-del-vending-de-los-egipcios-a-la-coca-cola/

**PRINCIPIOS DEL CÓDIGO LIMPIO [27]**

Renze, M. Pluralsight

https://app.pluralsight.com/library/courses/clean-architecture-patterns-practices-principles/table-of-contents

**PATRONES Y ANTIPATRONES [28]**

Msdn Microsoft

Msdnhttps://msdn.microsoft.com/es-es/library/bb972242.aspx

**PATRONES Y DE DISEÑO DE SOFTWARE [29]**

Leiva, A. DevExperto

https://devexperto.com/patrones-de-diseno-software/

**PATRONES PLURALSIGHT [30]**

Lagunas, B. Pluralsight

https://app.pluralsight.com/library/courses/patterns-library/table-of-contents

**MAQUINAS VENDING [31]**

Hostel Vending

https://www.hostelvending.com/noticias/noticias.php?n=8193

**EVOLUCIÓN MÁQUINAS VENDING [32]**

Grupo Fractalia Systems

http://www.fractaliasystems.com/evolucion-de-los-sistemas-vending/

**HISTORIA MÁQUINA VENDING [33]**

Era Venta

http://www.eraventa.com/grupo/historia-del-vending

**TIENDAS AUTOMÁTICAS [34]**

Era Vending

http://www.eravending.es/tiendas-automaticas

**MÁQUINA EXPENDEDORA DE LIBROS [35]**

Campos Pico, R. Librópatas

http://www.libropatas.com/listas/maquinas-expendedoras-libros/

**MÁQUINAs VENDING [36]**

Vending Machines Take a Beating

**PRIMERAS VENDING [37]**

Suárez, L. Máquinas vending

http://lasnuevastecnologias.com/aplicaciones-practicas-tecnologia-maquinas-vending/

**Nuevas TEcnologías [38]**

Nuevas tecnologías

https://www.hostelvending.com/noticias/noticias.php?n=1201

**Interfaces vending machine [39]**

European Vending Association aisbl

www.vending-europe.eu

**Interfaces COMUNICACIÓN vending machine [40]**

Vending and Vending Machines

http://www.bonusdata.net/vending/AuditData/vending.htm

**PROTOCOLOS DEX Y MDB VENDING MACHINE [40]**

Automatic vending

http://www.vendingmarketwatch.com/article/10272928/dex-and-mdb-a-primer-for-vendors

**LEyendo eva-dts con dex/ucs [41]**

Bonus Data Reading data Vending Machine

http://bonusdata.ch/Download/DEX-UCS/dexucs\_english.htm

**Vendsoft [42]**

Mdb Protocol for Simplifying your Vending Business

<https://www.vendsoft.com/mdb-vending-machine-protocol>

**Vending machine bus [43]**

Sniffing Vending Machine Bus

https://hackaday.com/tag/multidrop-bus/

**MARGINALLY CLEVER ROBOTS [44]**

Sniffing Code MDB Machine bus

https://www.marginallyclever.com/2014/04/making-sense-of-vending-machine-multidrop-bus-communications/

**EUROTECH M2M [45]**

Data Collection

https://www.eurotech.com/DLA/Library/as/Datacollection\_vendingmachine\_M2M\_gateways\_as.pdf

**CCTALK TuTORIAL ARDUINO [46]**

Cctalk tutorial arduino

https://cctalktutorial.wordpress.com/

**DMVI DIGITAL MEDIA VENDIGN [47]**

Vending Machine Controllers

http://www.digitalmediavending.com/vending-machine-controllers/

**MAYA VENDING [48]**

Thinkchip controladores vending

<https://mayavending.com/vending-controllers/>

**Lizard Software [49]**

VDM Comunnicator

<http://lizard-sw.com/en/rs232.html>

**ABRANTIX PAYMENT SOFTWARE ENGNEERING [50]**

MDB Converter

<http://www.abrantix.com/MDBConverter_en.html>

**ERa vending telemetry [51]**

Telemetría a distancia máquinas vending

<http://www.eravending.es/maquinas/ver/70/telemetria-a-distancia-para-maquinas-expendedoras>

**Wafer expand the word [52]**

Wf-800-GSM

<http://www.waferstar.com/en/GSM-control.html>

**EVA DTS MANUAL [53]**

Eva Dts Manual 6.1.2

Eva Dts Manual 6.1.2

1. Máquina de venta automática de productos [↑](#footnote-ref-2)
2. Señal o rastro que queda de una cosa o de un suceso [↑](#footnote-ref-3)
3. Franja o cajón que alberga productos del mismo tipo [↑](#footnote-ref-4)
4. Identificación por radiofrecuencia [↑](#footnote-ref-5)
5. Patrón de arquitectura modelo, vista y controlador [↑](#footnote-ref-6)
6. Conjunto de conceptos y prácticas que enfocan una problemática particular [↑](#footnote-ref-7)
7. Conexión funcional visible por el usuario [↑](#footnote-ref-8)
8. Framework creado por Microsoft [↑](#footnote-ref-9)
9. Interfaz de programación de aplicaciones [↑](#footnote-ref-10)
10. Tecnología propiedad y distribuida de Microsoft [↑](#footnote-ref-11)
11. Metodología de desarrollo ágil [↑](#footnote-ref-12)
12. Técnica de búsqueda de ideas grupal [↑](#footnote-ref-13)