****

**Universidad EAFIT**

**Escuela de Ingenierías**

**Departamento de Informática y Sistemas**

**Especialización en Desarrollo de Software**

**Marcos Y Patrones**

**PROYECTO INTEGRADOR – PRIMERA PARTE**

**Presentado a**

*Santiago Hurtado Gutiérrez.*

**Presentado por**

*Hans Linderman Parra Padilla*

*Juan David Cuartas León*

Medellín, Marzo 02 de 2013

**TABLA DE CONTENIDO**

[Contexto 3](#_Toc350087374)

[Problema 3](#_Toc350087375)

[Descripción de la solución 3](#_Toc350087376)

[Requerimientos Funcionales 4](#_Toc350087377)

[Diagrama Casos de Uso 7](#_Toc350087378)

[Modelo Conceptual 8](#_Toc350087379)

[Procesos Relevantes del Negocio 10](#_Toc350087380)

[Aplicación de patrones en el diseño de la solución 11](#_Toc350087381)

[Definir y justificar un patrón de interacción con la capa de presentación 11](#_Toc350087382)

[Definir y justificar un patrón de integración con sistemas externos o componentes de utilidad o infraestructura 13](#_Toc350087383)

[Identificar y enunciar cuatro patrones de diseño que se pueden utilizar, en que componente y su funcionalidad 15](#_Toc350087384)

[Justificar dos patrones de diseño 16](#_Toc350087385)

[Bibliografía 19](#_Toc350087386)

# Contexto

* Existen soluciones en la nube, y soluciones móviles para administrar portafolios de inversión en la bolsa.
* Estas soluciones se conectan con los principales mercados financieros para mostrar información en tiempo real.

# Problema

En el mercado de la bolsa de valores solo existen aplicaciones como Google Finance, YAHOO! Finance y Stocks App para el control de acciones, algunos de los problemas más comunes son:

* Las aplicaciones sólo muestran información de los principales mercados financieros: NASDAQ, NYSE.
* Ninguna de estas aplicaciones se conecta con el mercado colombiano (BVC) para mostrar información de su estado en tiempo real.
* No existen aplicaciones sobre la BVC en la App Store o en Google Play.

# Descripción de la solución

Desarrollar una aplicación móvil que permita administrar portafolios de inversión en la BVC, obteniendo información de las acciones en tiempo real.

La aplicación debe contener las siguientes características:

* Multiplataforma: La aplicación debe estar disponible en los principales sistemas operativos móviles: Android, iOS, Windows Phone, Blackberry.
* Puede almacenar la información tanto localmente, como en un repositorio de datos en la nube.

La aplicación al ser de uso personal incluirá las siguientes funciones:

* Administración de cuentas.
* Órdenes de compra y venta.
* Depósitos y retiros de capital.
* Cálculo de dividendos.
* Operaciones con repos.
* Balance del portafolio en tiempo real.

# Requerimientos Funcionales

|  |  |
| --- | --- |
| **FRQ-0011** | **Administración de maestros** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0003] Administrar comisionistas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0003)   [[UC-0001] Administrar mercados](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0001)   [[UC-0002] Administrar acciones](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0002) |
| **Descripción** | El sistema deberá permitir gestionar a través de una interfaz web todos los maestros de información (mercados, acciones, comisionistas, etc.) |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0002** | **Creación de portafolios** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0004] Crear portafolio](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0004) |
| **Descripción** | El sistema deberá permitir la creación de uno o varios portafolios de inversión a cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007). |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0003** | **Administración de**[**capital**](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0009) |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0006] Retirar efectivo](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0006)   [[UC-0005] Depositar efectivo](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0005)   [[UC-0015] Actualizar saldo portafolio](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0015) |
| **Descripción** | El sistema deberá permitir administrar a los inversionistas el [capital](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0009) de cada [portafolio](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0006) a través de depósitos y retiros. |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0004** | **Órdenes de compra** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0007] Colocar orden de compra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0007)   [[UC-0017] Procesar orden de compra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0017)   [[UC-0018] Procesar orden de venta](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0018) |
| **Descripción** | El sistema deberá permitir a cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007) realizar ordenes de compra de acciones en cada uno de sus portafolios. |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0005** | **Órdenes de venta** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0014] Colocar orden de venta](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0014) |
| **Descripción** | El sistema deberá permitir a cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007) realizar ordenes de venta de acciones en cada uno de sus portafolios. |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0006** | **Administración de repos** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0009] Enrepar acciones](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0009) |
| **Descripción** | El sistema deberá permitir a cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007) realizar transacciones [repo](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0002) en cada uno de sus portafolios. |
| **Importancia** | quedaría bien |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0007** | **Administración de dividendos** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0010] Agregar dividendos](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0010) |
| **Descripción** | El sistema deberá permitir a cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007) especificar los dividendos a los que tienen derecho las acciones matriculadas en sus portafolios. |
| **Importancia** | quedaría bien |
| **Estado** | validado |
| **Estabilidad** | alta |

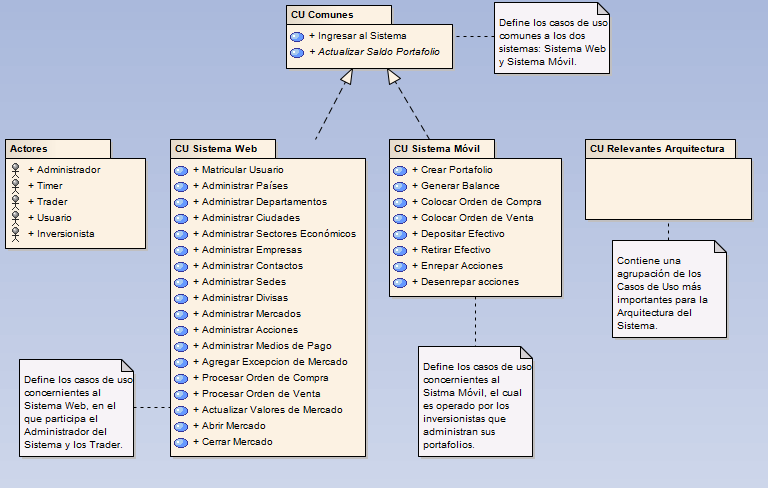
|  |  |
| --- | --- |
| **FRQ-0008** | **Creación del**[**estado financiero**](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0015) |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0011] Generar balance](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0011) |
| **Descripción** | El sistema deberá calcular el [estado financiero](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0015) de cada [portafolio](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0006) basado en los depósitos y retiros de [capital](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0009), las compras y ventas de acciones, los repos y dividendos de los mismos. |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

|  |  |
| --- | --- |
| **FRQ-0009** | **Sincronización de información** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** |  [[UC-0012] Sincronización de información](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#UC-0012) |
| **Descripción** | El sistema deberá sincronizar la información generada en los dispositivos móviles hacia la nube, en la cual se almacenará los datos de los usuarios, mercados, precios de las acciones, comisionistas y portafolios de cada [inversionista](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0007). |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

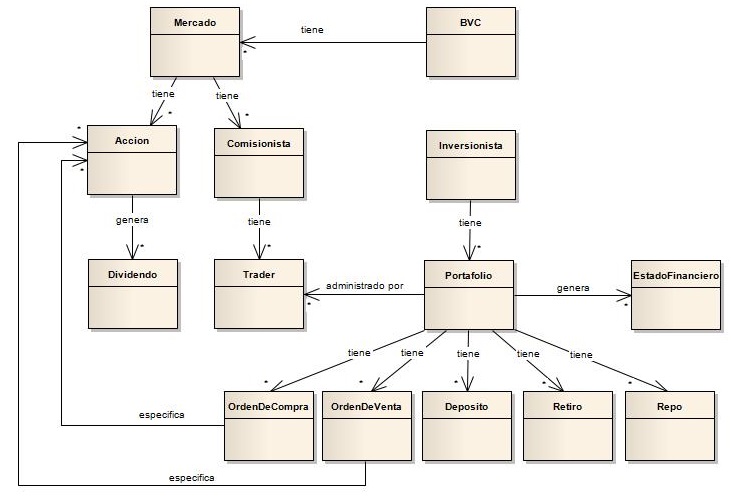
|  |  |
| --- | --- |
| **FRQ-0010** | **Actualización del precio de las acciones** |
| **Versión** | 1.0 ( 13/09/2012 ) |
| **Autores** | * [Hans Parra](file:///E:\\Universidad\\Postgrado\\Especializacion\\Proyecto%20integrador\\1%20-%20Requisitos\\Requisitos_REM_Default_Spanish_CRS.html" \l "STK-0001) * [Juan Cuartas](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#STK-0002) |
| **Dependencias** | Ninguno |
| **Descripción** | El sistema deberá actualizar el valor de cada [acción](file:///E:\Universidad\Postgrado\Especializacion\Proyecto%20integrador\1%20-%20Requisitos\Requisitos_REM_Default_Spanish_CRS.html#GLO-0014) en tiempo real de acuerdo a los valores dados por el mercado. |
| **Importancia** | vital |
| **Estado** | validado |
| **Estabilidad** | alta |

# Diagrama Casos de Uso

Los casos de uso se podrán encontrar especificados anexos a este trabajo.



# Modelo Conceptual



**Acción**: Una acción es la parte alícuota del capital social de una sociedad anónima. Cabe resaltar que poseer acciones de una compañía confiere legitimidad al accionista para exigir sus derechos, pero a su vez para cumplir con sus obligaciones.

**BVC**: Abreviatura de la Bolsa de Valores de Colombia. Es el único mercado de acciones y otros valores de Colombia, organizado a través de la estructura de bolsa.

**Comisionista**: Son los únicos entes autorizados por la BVC para vender o comprar en el mercado accionario. Cuando un inversionista realiza una compra, realmente es la firma comisionista la que hace la compra y luego le sede estos activos a través de una transacción.

**Deposito**: Es una operación en la que un inversionista le hace un ingreso de efectivo a un portafolio.

**Dividendo**: El dividendo activo, es la parte del beneficio obtenido por las sociedades mercantiles cuyos órganos sociales acuerdan que sea repartido entre los socios de las mismas. Es decir, una vez acordado su reparto, es un crédito del socio frente a la sociedad. De otro lado, el dividendo pasivo es el crédito que ostenta la sociedad mercantil frente al socio, por la parte del capital social que suscribió y que se comprometió a desembolsar. La diferencia entre las aportaciones y el desembolso inicial de los accionistas.

**Estado financiero**: Es el saldo actual en capital que tiene un inversionista en un portafolio y la cantidad de acciones que posee, los cuales son reconocidos por la firma comisionista que administra el portafolio.

**Inversionista**: Es aquella persona natural (persona física) o jurídica que es propietaria de acciones de los distintos tipos de sociedades anónimas o comanditarias que pueden existir en el marco jurídico de cada país. En el caso de nuestro sistema, es el usuario que compra o vende acciones.

**Orden de compra**: Es una operación en la que un inversionista le ordena a un trader la compra de una cantidad determinada de acciones a un cierto valor y dentro de un determinado rango de fechas establecido por el inversionista.

**Orden de Venta**: Es una operación en la que un inversionista le ordena a un trader la venta de una cantidad determinada de acciones a un cierto valor y dentro de un determinado rango de fechas establecido por el inversionista.

**Portafolio**: La cartera de inversiones o portafolio de inversiones, es el conjunto de activos financieros en los cuales se invierte, para el caso de nuestro sistema, estos activos pueden ser acciones o capital.

**Repo**: Es una operación con pacto de recompra, es decir, una entidad financiera vende un activo con un pacto de recompra por un precio determinado dentro de un tiempo determinado. Esta operación también se conoce como reporto, Repurchase Agreement o Sale and Repurchase Agreement.

**Retiro**: Es una operación en la que un inversionista le hace un retiro de efectivo a un portafolio.

**Trader**: Persona que realiza la compra, venta de acciones en la Bolsa de Valores en nombre del inversionista que pone la orden.

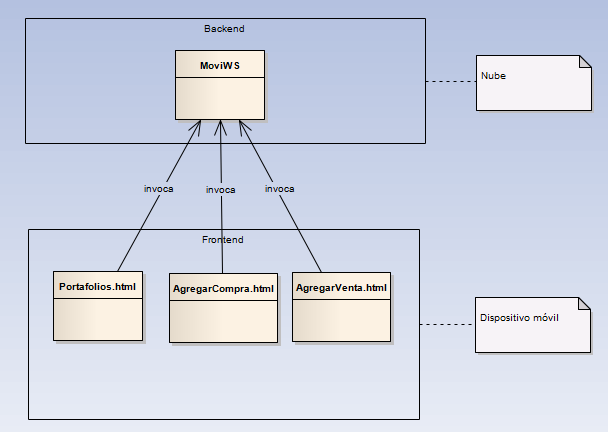
# Procesos Relevantes del Negocio

* **Orden de compra:** Es una operación en la que un inversionista le ordena a un trader la compra de una cantidad determinada de acciones a un cierto valor y dentro de un determinado rango de fechas establecido por el inversionista.
* **Orden de Venta:** Es una operación en la que un inversionista le ordena a un trader la venta de una cantidad determinada de acciones a un cierto valor y dentro de un determinado rango de fechas establecido por el inversionista.
* **Actualizar valores del mercado:** El sistema alimentara la información en los dispositivos de acuerdo a los intercambios en los instrumentos financieros y la definición de sus precios en el mercado.
* **Procesamiento de órdenes:** Al realizar una orden de compra o venta un trader realiza un análisis en el mercado para ejecutar la transacción.

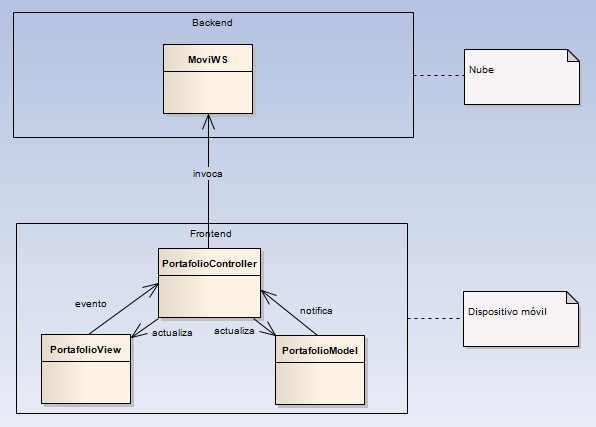
# Aplicación de patrones en el diseño de la solución

## Definir y justificar un patrón de interacción con la capa de presentación

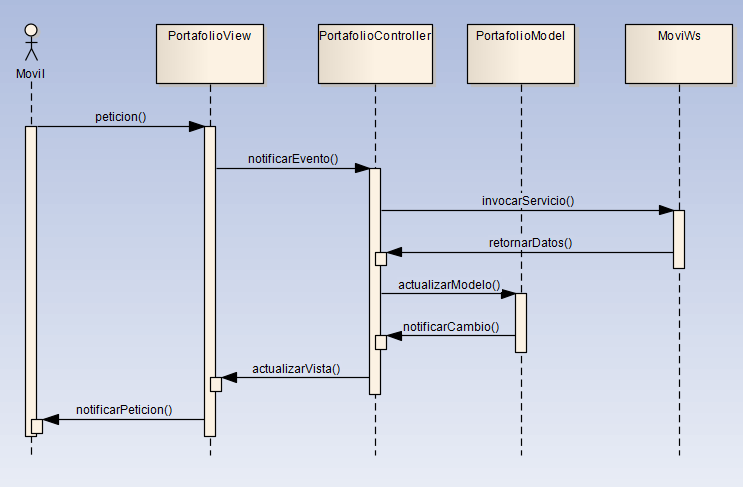
* **MVC:**
  + Diagrama de clase de la estructura sin patrón



* + Diagrama de clases de la estructura utilizando el patrón de diseño definido



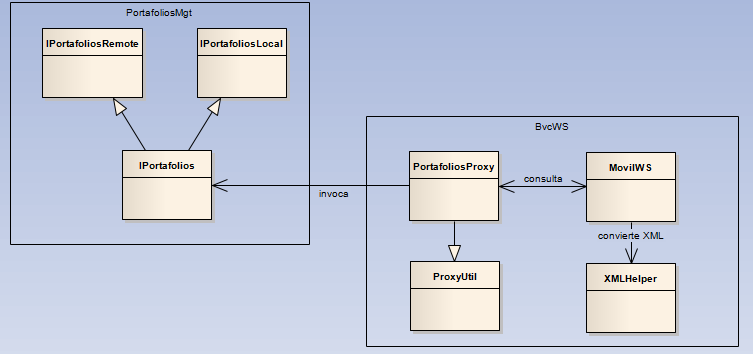
* + Interacción de los diferentes elementos en el patrón



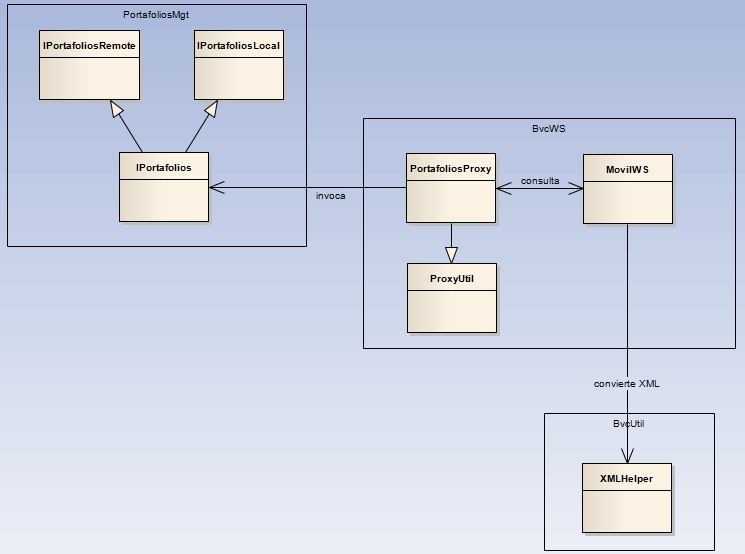
* + Consecuencias: Sin la aplicación del patrón, como se puede ver en el diagrama de clases de de la estructura sin utilizar el patrón, las vistas están invocando directamente el servicio Web lo cual crea código más complejo, largo y dificil de mantener. Aplicando MVC en nuestro frontend, como se puede ver en el diagrama de clases aplicando el patrón, podremos dividir nuestra lógica de negocio y nuestra presentación de modo que la aplicación sea más sencilla de mantener, también ayudará en la reutilización de componentes, y porporcionará mayor escalabilidad a nuestro sistema.

## Definir y justificar un patrón de integración con sistemas externos o componentes de utilidad o infraestructura

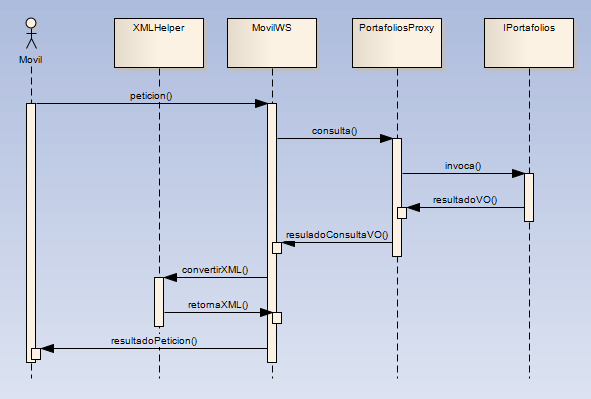
* **Utility abstraction**
  + Diagrama de clase de la estructura sin patrón



* + Diagrama de clases de la estructura utilizando el patrón de diseño definido



* + Interacción de los diferentes elementos en el patrón



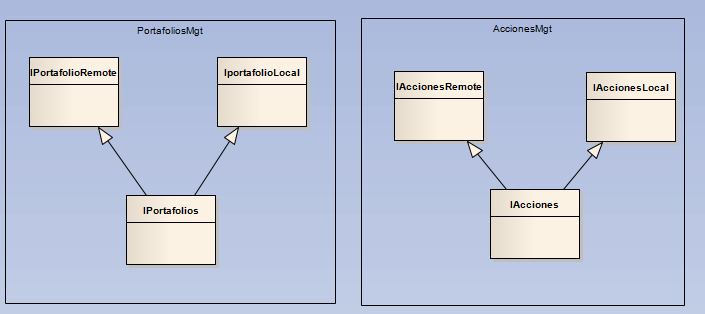
* + Consecuencias: Se creara un componente adicional que implica más tiempo y coste en el desarrollo del producto, pero a favor tendremos grandes oportunidades de reutilización de código por lo que en cada componente al realizar la conversión a XML solo debemos solicitar el método de este servicio, si este tipo de conversión requiere cambios futuros será necesaria realizarla únicamente en el componente de utilidad, por lo que se ahorraría tiempo en el desarrollo siendo altamente escalable.

## Identificar y enunciar cuatro patrones de diseño que se pueden utilizar, en que componente y su funcionalidad

* **Proxy**: El patrón proxy lo usaremos para controlar el acceso a los método públicos incluidos en los servicios de los componentes (PortafolioMgt, UsuarioMgt, DivisaMgt, AccionMgt, EmpresaMgt, LugarMgt), se implementara en los Web Services:
  + WebWS (ProxyUtil)
  + MovilWS(ProxyUtil)
* **Singleton**: Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto, se implementara en el Componente:
  + BvcUtil: Se implementara para la carga de la configuración de la codificación (UTF -8 o ISO-8859-15) de acuerdo al tipo del sistema operativo y en las clases de tipo Helper que generan el formato de la transmisión de datos entre componentes, estas clases se encargaran de convertir a XML los VO’s(Value Objects).
* **DAO (Data Access Object)**: Este patrón se utiliza como interfaz entre los componentes y las entidades del negocio, se implementara en un nuevo componente de tipo patron Entity Abstraction o en cada componente ya implementado (aún por decidir); que dará acceso a los métodos CRUD (Create, Read, Update and Delete) que interactuaran con la base de datos; se implementara en:
  + PortafolioMgt, UsuarioMgt, DivisaMgt, AccionMgt, EmpresaMgt, LugarMgt; creando un paquete DAO en cada componente o EntityAbstractionMgt; creando un DAO accedido por los demás componentes del sistema aprovechando la reutilización del código.
* **Null Object**: El patrón será utilizado para realizar la validación en la conversión de formatos a XML, si el objeto viene nullable no entrara a realizar la conversión; adicionalmente lo utilizamos para validar si los objetos se deben iterar y no arrojar fallos en el sistema; esta especificación se implementara en los componentes:
  + PortafolioMgt, UsuarioMgt, DivisaMgt, AccionMgt, EmpresaMgt, LugarMgt.
* **Template Method**: Al crear una súper clase o interfaz podemos implementarla o extenderla por subclases redefiniendo algunos o la totalidad de sus métodos definiendo unas interfaces que sirvan de guía para la utilización de la misma, los EJB de tipo local o remoto son un claro ejemplo ya que se sub-implementan por una interfaz q redefine la especificación de sus métodos, por ende se utilizaría en todos los componentes del sistema:
  + PortafolioMgt, UsuarioMgt, DivisaMgt, AccionMgt, EmpresaMgt, LugarMgt.
  + EntityAbstractionService (opcional)

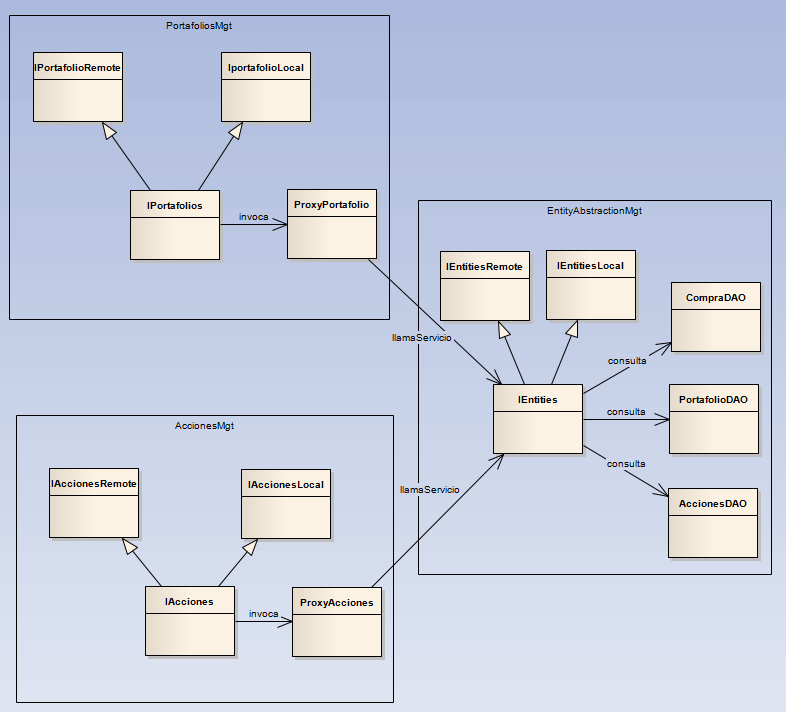
## Justificar dos patrones de diseño

* **DAO (Data Access Object)**
  + Diagrama de clase de la estructura sin patrón

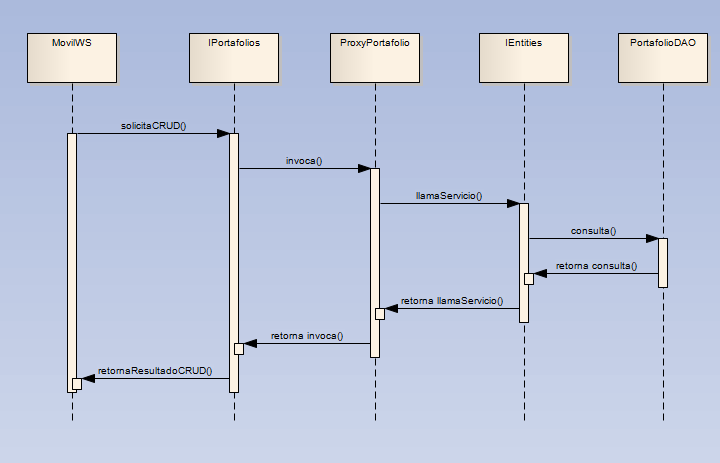


Los objetos IPortafolio y IAcciones tendrán la lógica de llamado de la base de datos, ejecutando las operaciones CRUD desde las mismas clases.

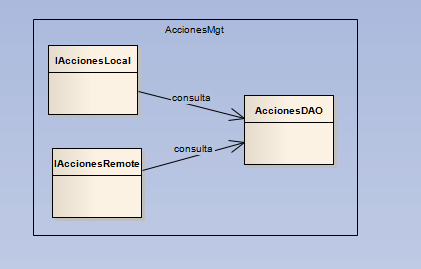
* + Diagrama de clases de la estructura utilizando el patrón de diseño definido



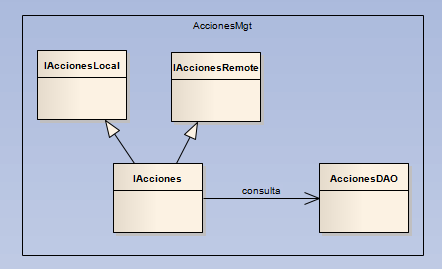
* + Interacción de los diferentes elementos en el patrón



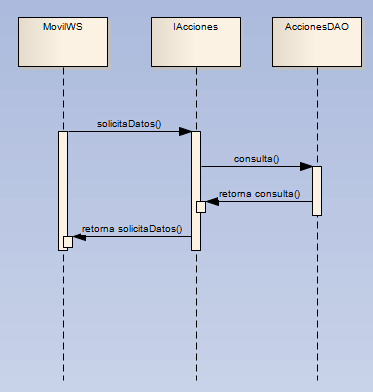
* + Consecuencias: Al implementar este patrón convertimos las entidades de la base de datos en objetos por lo que es más sencilla su reutilización, centralizando los objetos de control de la información los cuales nos ayudaran a mantener una estabilidad futura en cuanto a las modificaciones, involucrando un solo componente y no un set evitando fallos y posibles inconvenientes; esto nos ayudaría a realizar migraciones a nuevas versiones y diferentes tecnologías siendo menos costoso y complejo, las desventajas que nos impone este patrón es una mayor cantidad de clases que hagan posible la reutilización de los métodos por lo que influye en el momento de desarrollo en más tiempo y esfuerzo, además de una planificación previa para los objetos de servicio.
* **Template Method**
  + Diagrama de clase de la estructura sin patrón



* + Diagrama de clases de la estructura utilizando el patrón de diseño definido



* + Interacción de los diferentes elementos en el patrón



* + Consecuencias: Al tener centralizadas las interfaces podemos realizar un refactor de código más eficiente permitiéndonos editar métodos de una manera más ordenada sin realizar generalización (redundancia) en el código, en nuestro caso una doble invocación al método accionesDAO, permitiéndonos una mayor reutilización y un mayor control en la lógica del objeto.

# Bibliografía

* <http://msdn.microsoft.com/es-es/library/bb972272.aspx>; El Patrón Singleton
* <http://patronesdediseno.blogspot.com/2009/05/patron-proxy.html>; Patron Proxy
* <http://es.wikipedia.org/wiki/Data_Access_Object>; Data Access Object
* <http://programacionsolida.com.ar/2012/06/patrones-de-diseno-de-comportamiento_14.html>; Patrones de diseño de comportamiento: Null object
* <http://www.oodesign.com/template-method-pattern.html>; Template Method