

Proyecto

Diseño: documento de arquitectura del sistema

Grupo

Andrés Aldao

Alejandro Fontes

Mauricio Rodríguez

Tutor

Martin Rodríguez de los Santos

Facultad de Ingeniería – Consejo de Educación Técnico Profesional

Tecnólogo Informático

Montevideo – Uruguay

Índice

# Introducción

En este documento se describe la arquitectura del sistema a desarrollar, utilizando un enfoque basado en “vistas”, que representan al sistema bajo diferentes perspectivas y con diferente énfasis en un determinado aspecto.

Éste enfoque se denomina “4+1 View Model of Architecture” y presenta al sistema bajo 5 vistas: la Vista de Casos de Uso, la Vista Lógica, la Vista de Distribución, la vista de Implementación, y la vista de Procesos.

Los diagramas presentados en el presente documento son presentados en lenguaje UML en su versión 2.0.

# Modelo de dominio

Como resultado de la etapa de análisis se llegó a un modelo conceptual que presenta los principales conceptos del dominio del Marketplace a implementar. El mismo se presenta con el fin principal de dar un panorama general de la aplicación y a su vez complementar la siguiente sección correspondiente a la Vista de Casos de Uso.

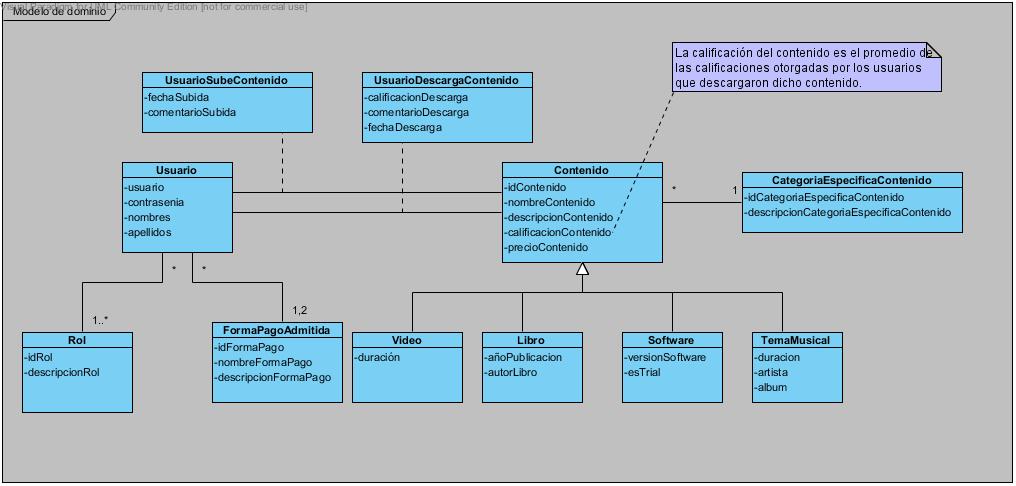


Imagen : Modelo de dominio de la aplicación

# Vista de Casos de Uso

La Vista de Casos de uso tiene como objetivo presentar un subconjunto del total de los casos de uso definidos para el sistema al cual llamaremos “casos de uso críticos para la arquitectura del sistema”. Éstos casos de uso críticos se eligieron utilizando criterios de cobertura (que accionen el mayor número de componentes de la arquitectura), disparidad (que no hayan dos casos de uso similares), prioridad (que sean prioritarios dentro de todos los casos de uso definidos), complejidad (que posean lógica compleja) y riesgo tecnológico (que tengan una alta dependencia de módulos propios o de terceros que puedan representar riesgo).

Los casos de uso críticos para la arquitectura del sistema serán prototipados en una etapa temprana del desarrollo del sistema con el fin de validar la arquitectura.

## Casos de uso críticos

### Módulo Web

|  |  |
| --- | --- |
| **Identificador** | LOGIN\_WEB |
| **Nombre** | Login web |
| **Descripción** | Un usuario anónimo se autentica en el sistema a través de la página web. Para ello ingresa su usuario y contraseña. |

|  |  |
| --- | --- |
| **Identificador** | BC\_WEB |
| **Nombre** | Buscar contenidos web |
| **Descripción** | Un usuario anónimo o autenticado en el sistema busca un contenido a través de la interfaz web, con la posibilidad de seleccionar la categoría del contenido entre las disponibles o ingresar el nombre del mismo. |

|  |  |
| --- | --- |
| **Identificador** | LOGOUT\_WEB |
| **Nombre** | Logout web |
| **Descripción** | Un usuario previamente autenticado en el sistema hace clic en el botón “salir” de la interfaz web del sistema quedando automáticamente como un usuario anónimo. |

### Módulo Móvil

|  |  |
| --- | --- |
| **Identificador** | LOGIN\_MOVIL |
| **Nombre** | Login a través de dispositivo móvil |
| **Descripción** | Un usuario anónimo se autentica en el sistema a través de la aplicación móvil del portal instalada en su dispositivo móvil Android. Para ello ingresa su usuario y contraseña. |

|  |  |
| --- | --- |
| **Identificador** | BC\_MOVIL |
| **Nombre** | Buscar contenidos desde dispositivo móvil |
| **Descripción** | Un usuario anónimo o autenticado en el sistema busca un contenido a través de la aplicación móvil, con la posibilidad de seleccionar la categoría del contenido entre las disponibles o ingresar el nombre del mismo. |

|  |  |
| --- | --- |
| **Identificador** | LOGOUT\_MOVIL |
| **Nombre** | Desincronización del dispositivo móvil |
| **Descripción** | Un usuario cuyo dispositivo móvil estaba sincronizado con la tienda hace clic en el botón desvincular quedando instantáneamente sin conectividad con la cuenta del usuario. |

## Diagrama de casos de uso críticos

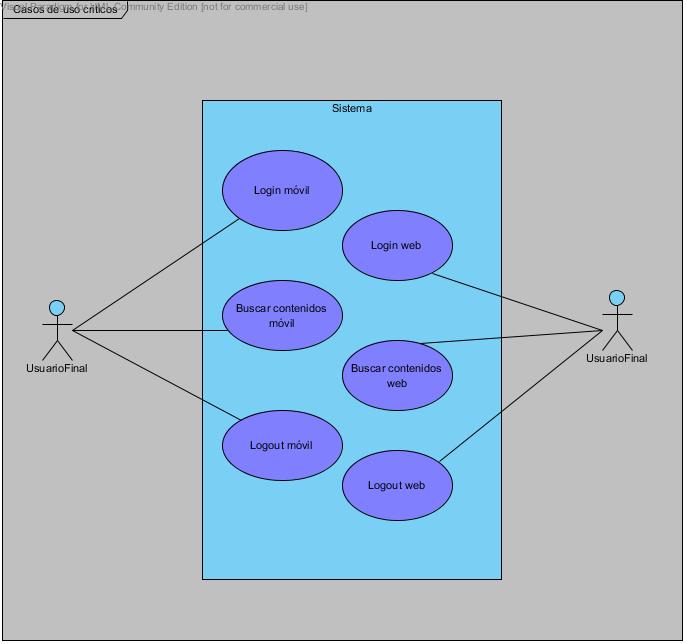


Imagen : Diagrama de casos de uso críticos

# Vista Lógica

La Vista Logica permite describir el sistema en base a abstracciones fundamentales del diseño orientado a objetos para dar soporte a los requerimientos funcionales establecidos. Adoptando un enfoque *top-down*descompondremos al sistema en un conjunto de subsistemas, como pueden ser las capas lógicas, y a través de sucesivos refinamientos se llegará a las unidades lógicas mas pequeñas.

## Estilo Arquitectónico

El estilo arquitectónico adoptado para el desarrollo de la solución es el de Capas Estrictas, donde cada capa consume interfaces (tiene visibilidad) de su inmediata inferior. En la siguiente imagen se presenta el diagrama de componentes de la arquitectura previamente mencionada, en la cual se identifican cuatro subsistemas: Capa de Persistencia, Capa de Negocio, Capa de Servicios y Capa de Presentacion estando las dos ultimas a un mismo nivel ya que ambos dependen de la Capa de negocio y no tienen dependencias entre sí.

### Subsistemas de la arquitectura

En esta sección se dará una breve introducción cada Subsistema identificado para la arquitectura adoptada, asi como una reseña de las unidades lógicas y funcionalidades que contienen.

#### Capa de Presentación

La capa de presentacion contiene los componentes de la aplicación que conforman la interfaz de usuario y manejan las interacciones con los mismos.

Dado que la aplicación a desarrollar constará tanto de un modulo web como un modulo movil, vale destacar algunas particulariades. En el caso de la aplicación web la Capa de presentación estará compuesta por paginas html (JSF o Java Server Faces) asi como Manages Beans siendo los ultimos las clases de logica de presentación del lado del servidor. Por otra parte la aplicación móvil será conformada

En la Figura N se ilustran algunos de los componentes logicos que conforman la Capa de Presentacion (se optó por analizar la interfaz web, el caso de la aplicación movil es análogo por lo que se obvia). Se puede apreciar que la WebMarketplace ofrece a los usuarios las funcionalidades definidas para el sistema, utilizando las clases correspondientes a la logica de presentacion o Managed Beans (como son UsuarioMB y ContenidoMB). Los Managed Beans delegan la tarea los componentes de Capa de Negocio a traves de interfaces bien definidas para así realizar los casos de uso.

#### Capa de Servicios

La Capa de Servicios tiene como finalidad ofrecer un subconjunto de la logica dde negocio a clientes remotos, como pueden ser paginas html a traves de pedidos ajax o la aplicación movil para la recarga de contenidos. Dichos servicios serán expuestos a traves de Servicios Web SOAP asi como Servicios Web REST utilizando las tecnologias JAX-WS y JAX-RS respectivamente.

Los clientes remotos serán tanto la aplicación web como la aplicación movil que consumiran dicha logica con el fin de recargar contenido, llevar a cabo operaciones como pueden ser compra y descarga, calificacion de contenidos, entre otros.

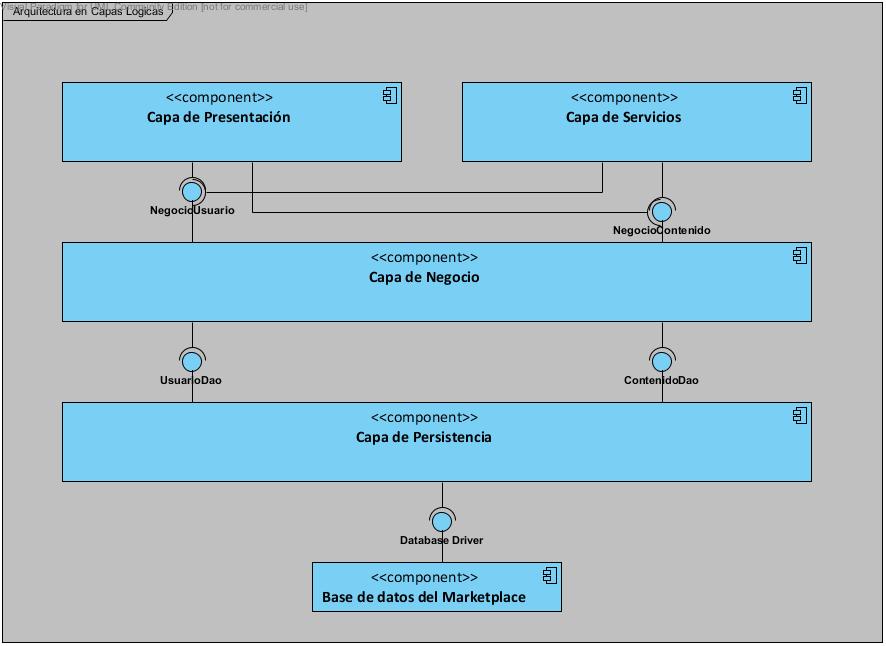


Imagen : Arquitectura en Capas Lógicas

#### Capa de Negocio

Este subsistema encapsula los componentes del sistema que realizar las operaciones que implementan las funcionalidades propiamente dichas. Tambien conocida como Capa Logica o Capa de dominio ésta capa contiene las clases de logica que constituyen la fachada de la aplicación, ofreciendo una interfaz simplificada de las operaciones ofrecidas para cada elemeto del dominio.

La fachada de la aplicación se implementa como un conjunto de clases (como son por ejemplo NegocioUsuario y NegocioContenido) que siguen el patron *Session Facade*. Estas clases se implementan como *Statless Beans* de java e implementan las operaciones de casos de uso relacionados entre sí.

En la Figura N se ilusta el diseño de la Capa de Negocio o Capa Logica.

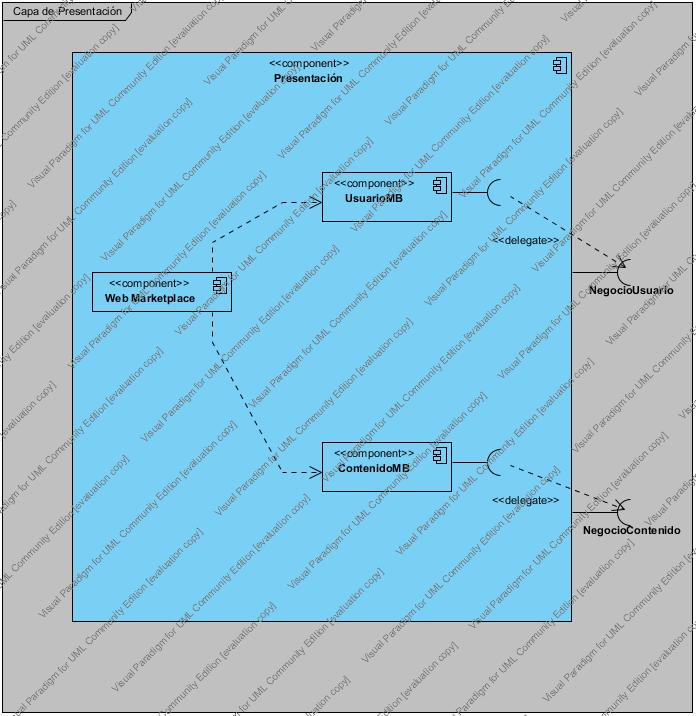


Imagen : Capa de Presentación

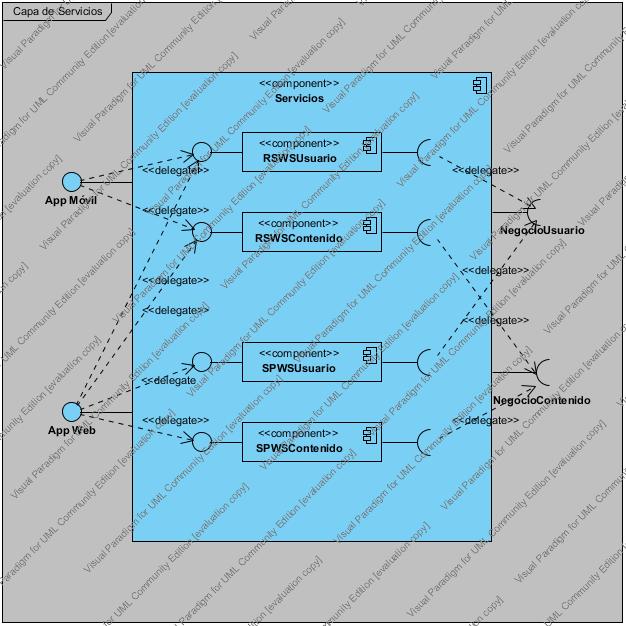


Imagen : Capa de Servicios

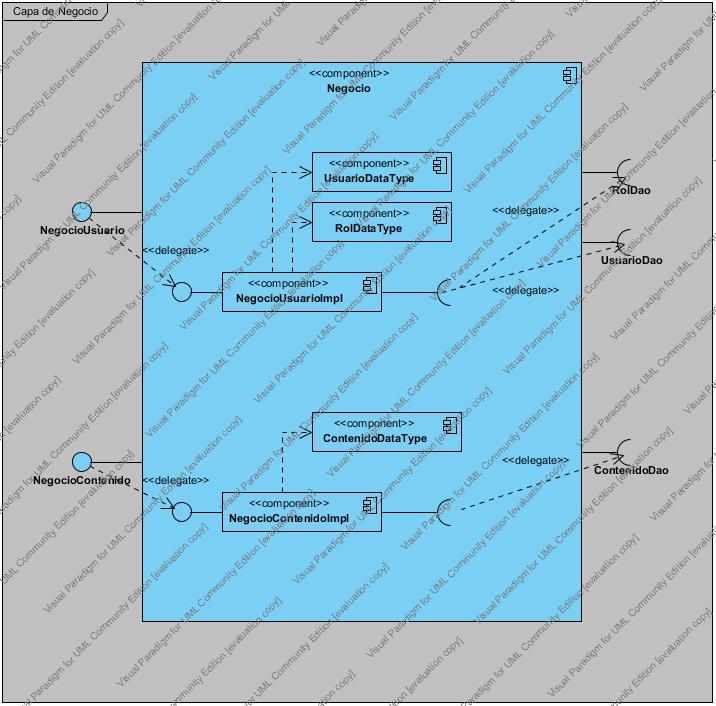


Imagen : Capa de Negocio

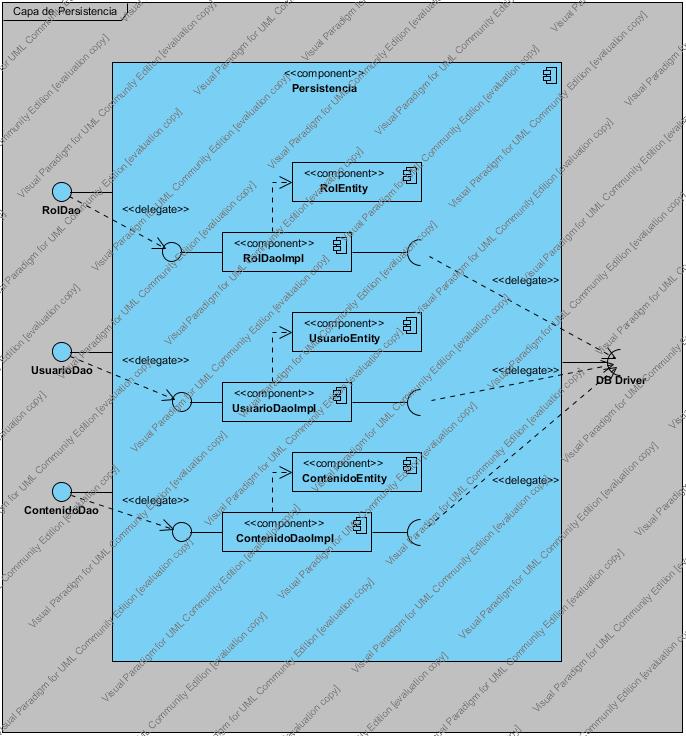


Imagen : Capa de Persistencia

### Diagramas de Interacción

En la presente seccion se presentan las interacciones entre los diferentes componentes logicos que conforman la aplicación para la realizacion de los casos de uso críticos para la arquitectura del sistema.

Para representar las interacciones entre dichos componentes se utilizarán los Diagramas de Secuencia del Sistema.

# Vista de Deployment

La Vista de Deployment o Vista de Distribución del sistema permite visualizar los posibles escenarios de asignacion de componentes logicos de la aplicación en diferentes nodos fisicos o tiers.

En el caso del Marketplace a implementar podemos identificar los siguientes nodos fisicos que conformarán el diagrama de distribución:

* **Browser:** un navegador web estandar instalado en el cliente. Ejemplos de éste tier pueden ser Google Chrome, Mozilla Firefox, Internet Explorer, entre otros.