



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Escuela de Ingeniería Informática



Trabajo Fin de Grado

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

## **Aplicación Web 2.0 para la Gestión de Dibujos**

Jorge Adrián Delgado de la Rosa

Las Palmas de Gran Canaria

Diciembre 2013



Trabajo Fin de Grado realizado en la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria, para la consecución del título de Ingeniero Informático.

**Título:** Aplicación Web 2.0 para la Gestión de Dibujos

**Alumno:** Jorge Adrián Delgado de la Rosa

**Tutor:** Javier Sánchez Pérez

**Fecha:** Diciembre de 2013



*A mi familia*



## **Agradecimientos**

Quiero dar mi agradecimiento a mi tutor Javier Sánchez Pérez, por aportar la idea del proyecto y haberme ayudado en la realización del mismo. También al resto de profesores y compañeros de la facultad que me han permitido llegar hasta aquí.

Y finalmente a mi familia y amigos que siempre me han apoyado.





## Resumen

El TFG (Trabajo Fin de Grado) de esta aplicación consiste en una Web 2.0 de gestión de dibujos, destacando la participación de cada usuario para gestionar y a su vez comentar, votar o etiquetar dibujos.

La Web 2.0 promueve la participación de los usuarios y el intercambio de opiniones e información, colaborando los usuarios entre sí como creadores de contenido. De esta manera se le conoce como “web social”, proporcionando un medio para gestionar información con el fin de compartir los resultados. Por tanto, la diferencia entre la Web 2.0 frente a una web normal, es que los usuarios no se limitan a observar pasivamente los contenidos que se han creado para ellos.

Una de las características principales de la aplicación es la herramienta para crear y editar dibujos, implementado en un Applet. Un Applet es un componente de una aplicación que está desarrollado en Java y se puede ejecutar desde una página web. La herramienta en sí te permite insertar figuras, dibujar a mano alzada y elegir el color que quieras. Al ser desarrollado en Java, el formato que tiene la aplicación web es JSP (Java Server Page). La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP. Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios.

## **Abstract**

The TFG of this application consists of a Web 2.0 of drawings management, highlighting the involvement of each user to manage and in turn to comment, vote or tag drawings.

Web 2.0 encourages user participation and exchange of views and information, collaborating the users with each other as content creators. Thus it is known as "social web", providing a means to manage information in order to share the results. Therefore, the difference between Web 2.0 opposite a normal web, is that users are not limited to passively observe the contents that have been created for them.

One of the main features of the application is the tool to create and edit drawings, implemented in an Applet. An Applet is a component of an application that is developed in Java and can be run from a web page. The tool itself allows you to insert figures, draw freehand and choose the color you want. Being developed in Java, the format that has the web application is JSP (Java Server Page). The main advantage of JSP over other languages is that Java is a general purpose language that exceeds the web world and is suitable for creating classes that handle business logic and data access in a neat way. This allows separate tiered web applications, leaving the party responsible for generating the HTML document in the JSP file. Another advantage is that the JSP inherits Java portability, and it is possible to run applications on multiple platforms without change.



## Índice general

Prefacio .....	v
1 Introducción .....	1
1.1 Motivación y objetivos .....	1
1.2 Aportaciones .....	2
1.3 Organización del documento .....	2
2 Estado actual del arte.....	5
2.1 Aplicaciones de escritorio .....	5
Paint .....	5
Photoshop .....	5
Gimp .....	6
2.2 Aplicaciones web.....	7
Sketchpad.....	7
SVGedit.....	8
Kleki .....	8
2.3 Aplicaciones web 2.0.....	8
Imgur .....	8
deviantART .....	8
oekaki .....	9
3 Recursos utilizados.....	11
3.1 Recursos software .....	11
3.2 Recursos hardware.....	12
4 Planificación del trabajo.....	13
4.1 Metodología de desarrollo.....	13
4.2 Planificación y temporización .....	16
4.3 Presupuesto.....	17
5 Desarrollo del trabajo.....	19
5.1 Requisitos del sistema.....	19
5.1.1 Modelo del dominio .....	19
5.1.2 Lista de características .....	21
5.2 Requisitos del software .....	23
5.2.1 Actores .....	23

5.2.2	Modelo de casos de uso.....	24
5.2.3	Especificación de casos de uso.....	27
5.2.4	Prototipo de interfaz de usuario .....	33
5.3	Modelo de análisis .....	38
5.3.1	Organización del modelo análisis.....	38
5.3.2	Diagramas de clases .....	38
5.3.3	Diagramas de colaboración .....	42
5.4	Modelo de diseño.....	45
5.4.1	Arquitectura del sistema .....	46
5.4.2	Organización del modelo de diseño .....	47
5.4.3	Diagramas de clases .....	47
5.4.4	Diagramas de secuencia .....	55
5.5	Implementación .....	59
6	Conclusiones y trabajo futuro .....	65
	Anexo I: Competencias.....	67
	Anexo II: Legislación vigente .....	77
	Anexo III: Manual de usuario .....	79
	Anexo IV: [ENTORNO DE DESARROLLO].....	87
	Bibliografía .....	89





## Prefacio

Web 2.0 hace referencia a una serie de herramientas online creadas con la intención de fomentar la participación de los usuarios y el libre intercambio de opiniones e información. Por esta razón recibe el nombre de “web social”. Proporciona a los usuarios un medio para generar, optimizar y gestionar información con el fin de aunar esfuerzos y compartir los resultados. Algunas de las herramientas más conocidas de la Web 2.0:

- Blogs: permiten publicación de textos, vídeos, imágenes, etc., de manera periódica y los lectores pueden hacer comentarios en cada una de las entradas.
- Wiki: recoge artículos generados por sus propios usuarios quienes pueden crear, modificar o vetar informaciones en función de los protocolos y jerarquías establecidas. Es una herramienta orientada para comunidades profesionales o personas interesadas en un mismo campo.
- RSS: permite la suscripción de los usuarios a los contenidos de un blog o de una página web. Se pueden leer desde el navegador o mediante un programa como Google Reader sin necesidad de acceder diariamente a las páginas que visitamos con más frecuencia. Cuando una web actualiza su información, automáticamente se descarga en programas como Google Reader.
- Videos compartidos: compartir videos entre los usuarios. Permiten subir videos, comentarlos, votarlos, crear canales y suscribirse a ellos, etiquetar, etc.
- Imágenes compartidas: almacenar, gestionar y compartir imágenes. Cada usuario puede crear su propia galería, etiquetar y compartirla o no según desee.
- Presentaciones compartidas: SlideShare es el más conocido donde se comparten de manera pública o privada presentaciones en PowerPoint y otros formatos.
- Redes sociales: para poner en contacto a personas de intereses similares. El usuario crea su perfil personal con información sobre sus gustos y adicciones. Permite compartir fotos, videos, comentarios, etc. Facebook, Tuenti, Twitter o LinkedIn como las más conocidas.
- Herramientas de trabajo colaborativo: como por ejemplo Google Docs que permite almacenar en un servidor web documentos, presentaciones y hojas de cálculos pudiendo compartirlas con otros usuarios.

En este proyecto el objetivo es desarrollar una aplicación web de gestión de dibujos permitiendo a los usuarios crear, editar, almacenar, compartir y comentar dibujos.

La metodología de desarrollo empleada es PUD (Proceso Unificado de Desarrollo de Software) enfocada en detectar los posibles riesgos dentro del ciclo de vida del proyecto, es una metodología iterativa e incremental dirigida por casos de uso. Haciendo uso de la herramienta UML se han podido realizar los diferentes diagramas de casos de uso de la aplicación.

Java es el lenguaje de programación elegido para el desarrollo de la aplicación haciendo uso de la API JDBC para poder operar con la base de datos MySQL. El editor de dibujos se ha



desarrollado en JApplet para ser incrustada después en la página web. Al ser desarrollada en java y siendo una aplicación web, las páginas generadas son en jsp pues todo se ejecuta desde el servidor. También se ha utilizado Javascript para dar más facilidad a algunas funciones de la aplicación.

# 1 Introducción

Este proyecto se centra en una aplicación Web 2.0 de gestión de dibujos, dando una mayor participación e importancia a los usuarios.

Cualquier usuario (anónimo o no) podrá ver los dibujos que están en la web. Sin embargo, un usuario registrado puede subir los dibujos ya realizados que tenga en disco, compartirlos o no con otros usuarios dependiendo de la privacidad del dibujo, eligiendo una categoría (si es un dibujo abstracto, o compuesto de figuras, etc.) y etiquetarlos para facilitar la búsqueda de dibujos. También pueden crear dibujos usando la herramienta de dibujo implementada en la aplicación, subirlos al servidor dando también las mismas opciones de privacidad, categoría y etiquetas. Posteriormente un dibujo puede ser editado o borrado.

Otras opciones que tiene la aplicación para ser realmente una Web 2.0 es poder comentar o votar (tanto positiva como negativamente) cualquier dibujo, además de compartir enlace del dibujo. La aplicación cuenta además con una barra de búsqueda para encontrar un determinado dibujo (sabiendo el título del dibujo) o varios dibujos según la categoría.

Lo que se espera de esta aplicación es que pueda servir a todo tipo de usuarios que les guste dibujar (ya sean principiantes o expertos en crear y editar dibujos) o simplemente compartir dibujos ya creados con otros usuarios.

## 1.1 Motivación y objetivos

La razón principal de crear este proyecto es la gestión de dibujos que el usuario podrá realizar, enfocado a una Web 2.0 para permitir a los usuarios una mayor participación y les motive a usar la aplicación.

Luego entonces, se desarrollará una aplicación Web 2.0 que permita compartir los dibujos con otros usuarios para posteriormente ser vistas, votadas y/o comentadas.

En cuanto a los objetivos tenemos los siguientes:

Objetivo principal:

- Realizar una aplicación web para gestionar sus dibujos y ser compartidas con otros usuarios.

Objetivos del proyecto:

1. Creación y edición de dibujos. La aplicación permitirá al usuario crear y editar dibujos.
2. Subir dibujos. El usuario podrá subir dibujos que tenga en su PC a la aplicación web.
3. Eliminar dibujos. Podrá eliminar cualquiera de sus dibujos ya sea creado o subido.

4. Visualizar dibujos. Cualquier dibujo será mostrado a cualquier usuario (anónimo o registrado) a no ser que sea un dibujo privado, en cuyo caso solo lo podrá ver el usuario propietario.

Objetivos secundarios:

1. Etiquetar dibujo. Tanto al crear como al subir dibujo el usuario podrá etiquetar el dibujo. Posteriormente se podrá ver las etiquetas asignadas cuando el usuario visualice el dibujo
2. Votar dibujo. Los usuarios podrán dar voto positivo o negativo al dibujo.
3. Comentar dibujo. Los usuarios podrán dejar comentarios relacionados con el dibujo.

## 1.2 Aportaciones

La aplicación propuesta como TFG pretende no solo compartir dibujos sino que además puedan ser gestionados, comentados, etiquetados. El usuario podrá optar por no compartir sus dibujos y sean privados.

Como aplicación Web 2.0 que contaría con la participación de muchos usuarios, se podría realizar concursos como por ejemplo la creación de los mejores dibujos originales y ser votados por la comunidad.

## 1.3 Organización del documento

En este documento se describirá todo el proceso necesario para poder llevar a cabo el desarrollo de una aplicación, habiendo un ciclo de investigación, análisis, diseño e implementación dividida en etapas. El proyecto está centrado en seis capítulos donde se especifica las tareas involucradas del trabajo realizado.

El **primer capítulo** es una introducción del proyecto describiendo en qué consiste la aplicación del proyecto e indicando sus objetivos principales.

El **segundo capítulo** se centra en la actualidad del tema, indicando las aplicaciones existentes que tengan o pueden tener relación con la aplicación de este proyecto.

En el **tercer capítulo** se detallan los recursos de hardware y software necesarios para llevar a cabo este proyecto.

Continuando con el **cuarto capítulo**, englobará la planificación del trabajo especificándose la metodología de desarrollo y una estimación económica para obtener un presupuesto.

En el **capítulo cinco** comienza con el modelo de negocio para continuar con el resto de etapas del desarrollo del proyecto siguiendo la metodología PUD (Proceso Unificado Desarrollo del Software). Se especifican los requisitos del software, modelo de análisis, diseño e

implementación indicando algunos aspectos concretos para dar forma a la aplicación del proyecto. Concluye con las pruebas realizadas durante el desarrollo de la aplicación.

El **último capítulo** contiene conclusiones tras finalizar el proyecto y se cierra con los apéndices técnicos y la bibliografía con fuentes de ayuda que han servido para poder realizar el proyecto.

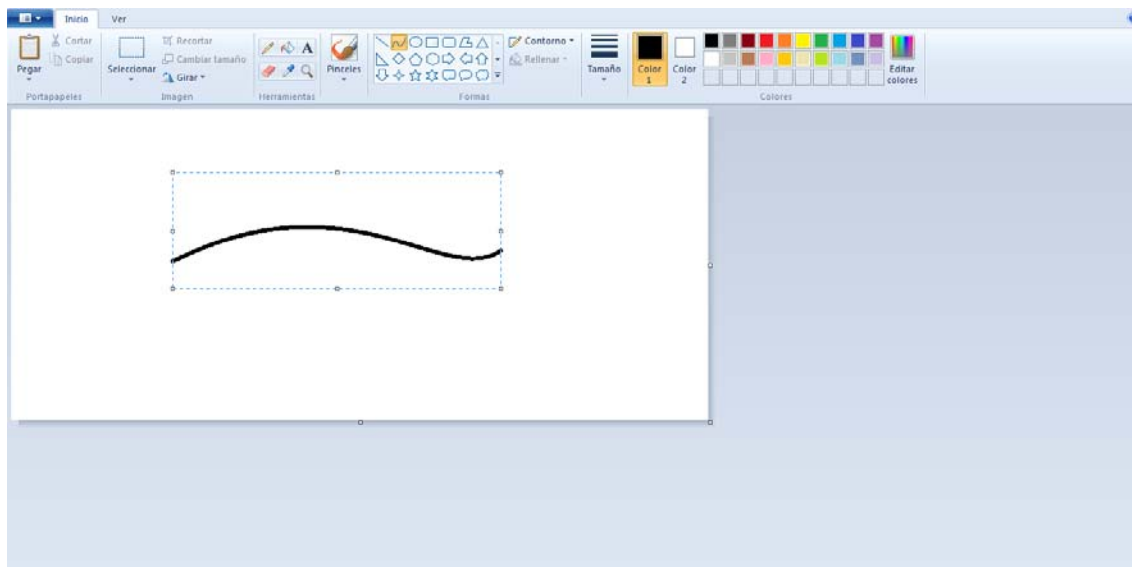


## 2 Estado actual del arte

Actualmente existen varias aplicaciones escritorio y aplicaciones Web dedicadas al dibujo. Las aplicaciones de escritorio como Paint permiten realizar y editar dibujos, siendo guardados en disco. Con las aplicaciones Web y más concretamente las web 2.0 se pretende ir más lejos; no solo se crean, editan y guardan los dibujos sino también se comparten, comentan, votan, etc.

### 2.1 Aplicaciones de escritorio

**Paint** es una característica de Windows que puede usar para dibujar, aplicar color y editar imágenes. Está orientado a cualquier tipo de usuario y puedes usar Paint como un bloc de dibujo digital para crear imágenes sencillas y proyectos creativos, o para agregar texto y diseños a otras imágenes, como las capturadas con una cámara digital. Tiene una colección de herramientas de dibujo para crear dibujos a mano alzada y agregar diversas formas a las imágenes como cuadrados, círculos, triángulos, flechas (entre otras cosas) y también permite crear una forma personalizada si usa la forma Polígono. Algo destacable es que también permite cambiar de tamaño libremente hasta que se *rastericen* (pasa de formato gráfico vectorial a un conjunto de píxeles) cuando se selecciona otra herramienta.



**Photoshop** es un programa de Adobe de licencia comercial y es algo más que una herramienta de dibujos ya que también permite la edición de imágenes digitales orientada especialmente a fotógrafos, diseñadores, profesionales web y profesionales de vídeo. Entre las funciones más destacadas se encuentra:

El enfoque suavizado que analiza las imágenes para maximizar la claridad y minimizar el ruido y los halos permitiendo realizar ajustes para mejorar la calidad.

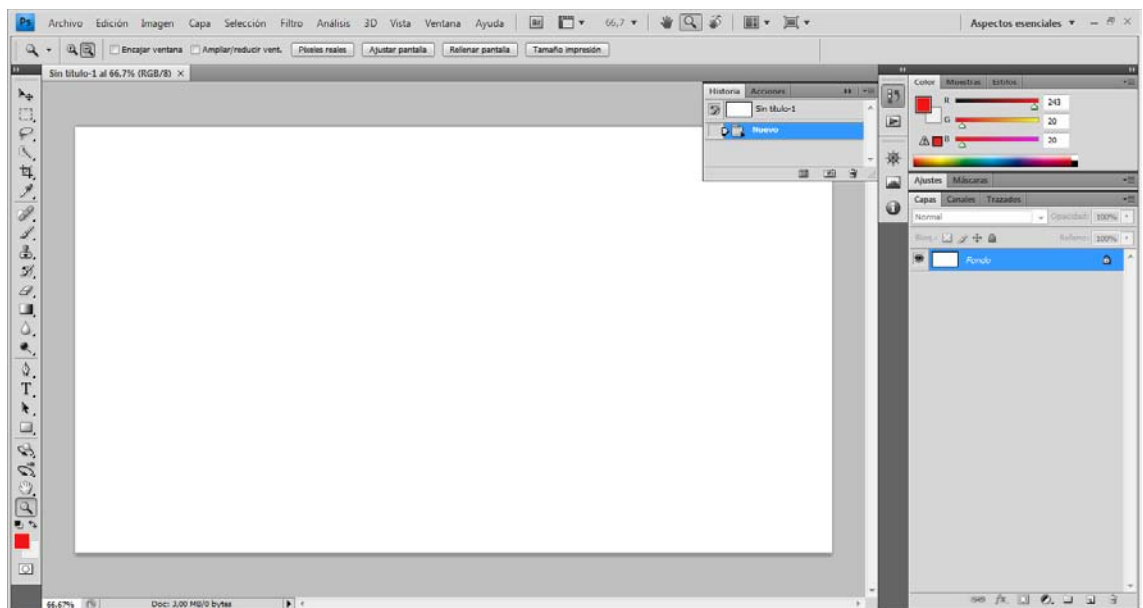
Aumento de resolución inteligente, agrandando las imágenes con baja resolución para que tengan buen aspecto para imprimir a tamaño de folio, cartel o póster conservando los detalles y la nitidez sin introducir ruido.

Estabilizador de imágenes de cámara, intenta recuperar instantáneas que pueden darse por perdidas debido al movimiento de la cámara.

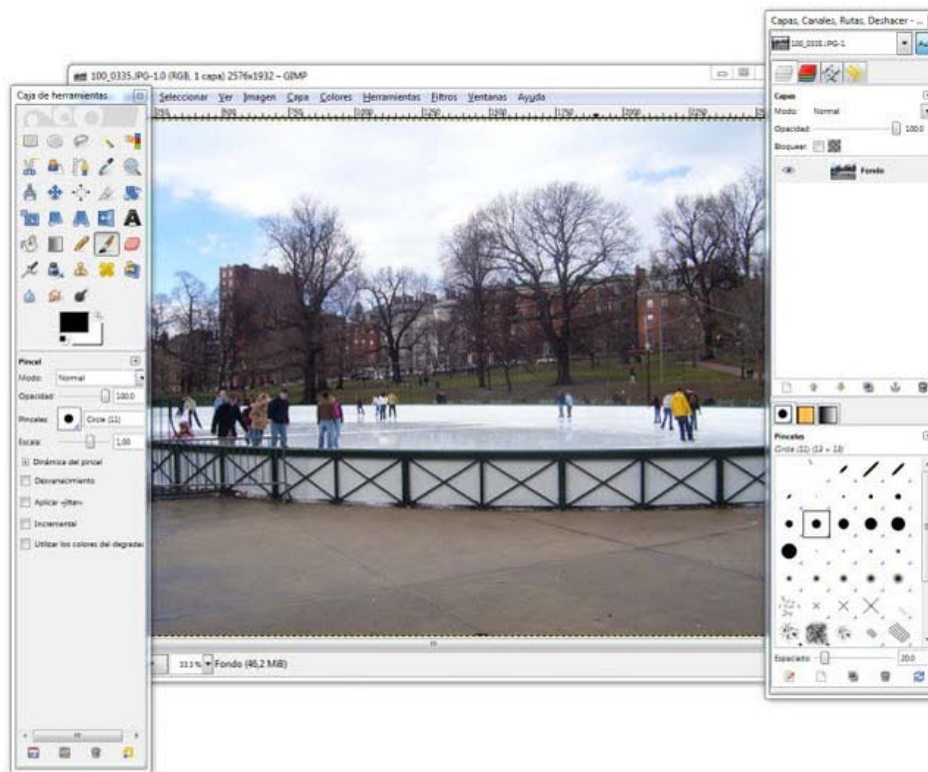
Selección de varias formas y trazados.

Compatibilidad con CSS importando muestras de color directamente desde HTML, CSS o SVG para conseguir el color exacto de esquemas web existentes. Genera código CSS para colores y otros elementos de diseño.

Hay muchas más funciones pero se centra especialmente en aplicar una serie de efectos, textos, marcas, reconoce múltiples formatos de imágenes y tratamientos en un espacio de trabajo multicapas.

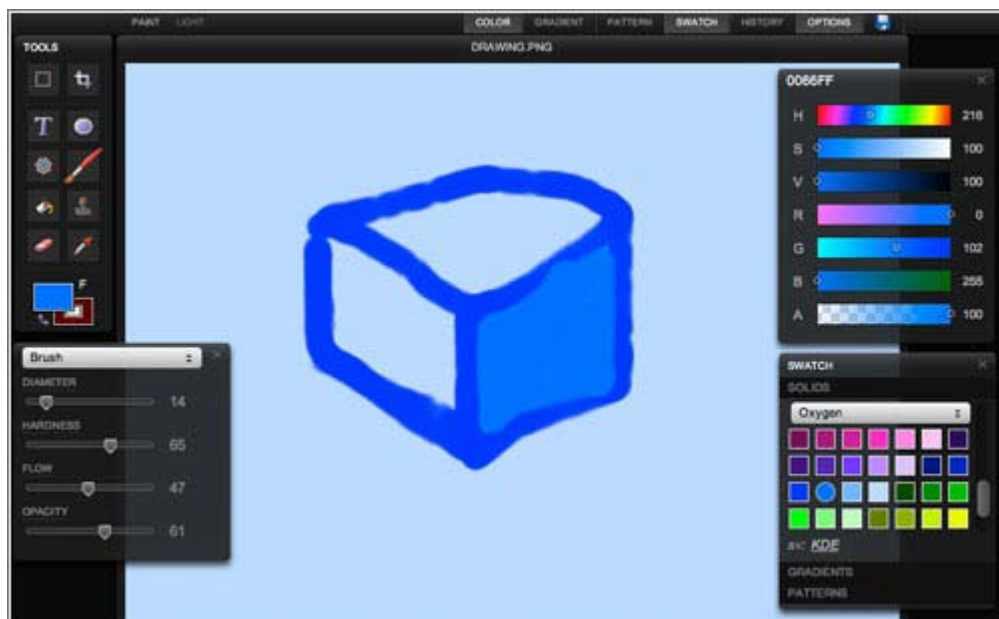


**Gimp** podría considerarse la alternativa libre de Photoshop ya que es una aplicación de tratamiento de imágenes ideal para retocar fotografías, crear imágenes especialmente para web, trabaja con capas, reconoce múltiples formatos de imágenes, lee algunos formatos gráficos vectoriales como PostScript o SVG y permite la automatización de muchos procesos mediante macros o secuencias de comandos.



## 2.2 Aplicaciones web

**Sketchpad** es similar a otras aplicaciones de dibujos existentes que funciona directamente en el navegador. Está desarrollado en HTML5 y no recurre a Flash, que requieren la instalación de componentes adicionales en el ordenador.



\* <http://mudcu.be/sketchpad/>



**SVGedit** un editor online de gráficos vectoriales que utiliza JavaScript, HTML5, CSS y SVG sin funcionalidad del lado del servidor y trabaja directamente en el navegador. Está pensado para usuarios que necesitan hacer ediciones rápidas de imágenes SVG existentes.

\* <http://svg-edit.googlecode.com/svn-history/r1771/trunk/editor/svg-editor.html>

**Kleki** es una herramienta de dibujo, pintura, y edición de imagen que funciona directamente desde el navegador, siendo compatible con Google Chrome, Firefox, Safari, Opera y Safari Mobile (iPad). Funciona con HTML5 y permite crear una imagen, pintarla y modificarla o arrastrar una imagen desde nuestro PC hasta Kleki. Se puede optar a dibujar o editar una simple imagen o jugar con una colección de filtros en diversas capas. Permite además subir los dibujos en Imgur para poder ser compartido en redes sociales, o subirlos a Dropbox, Picassa, Google Drive entre otros. Cualquier usuario puede dar uso de esta herramienta pero es más recomendada a aquellos que les guste dibujar.

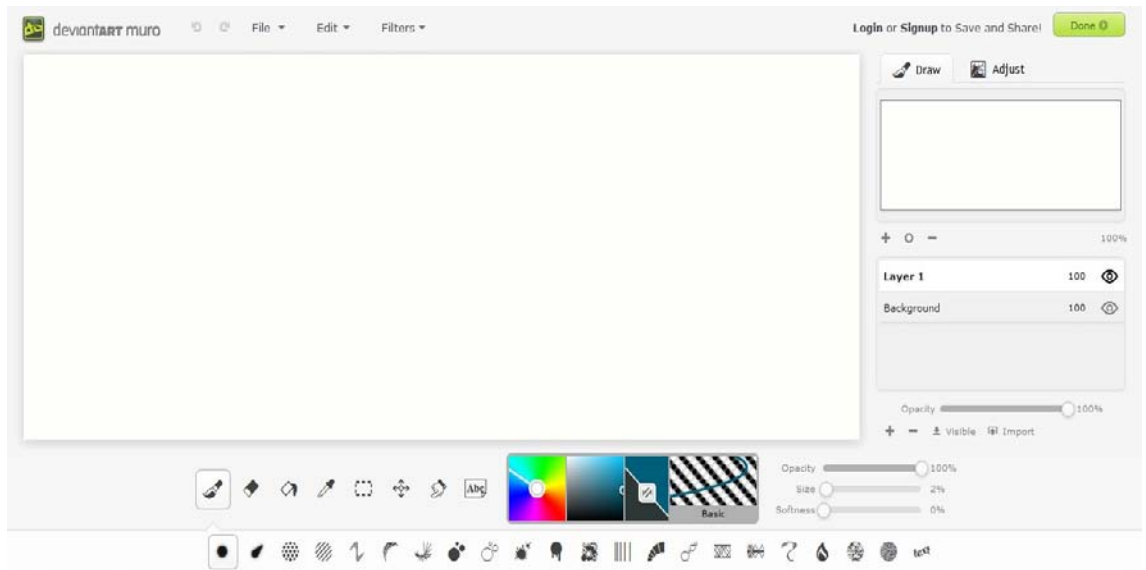
\* <http://kleki.com/>

## 2.3 Aplicaciones web 2.0

**Imgur** es un servicio gratuito para alojar y compartir todo tipo de imágenes. Permite subir imágenes que tengamos en nuestro PC o cargar fotos directamente desde la web para luego compartirlas con otros usuarios. No permite crear o editar dibujos, lo más parecido es crear un meme que es añadiendo texto a una imagen.

\* <http://imgur.com/>

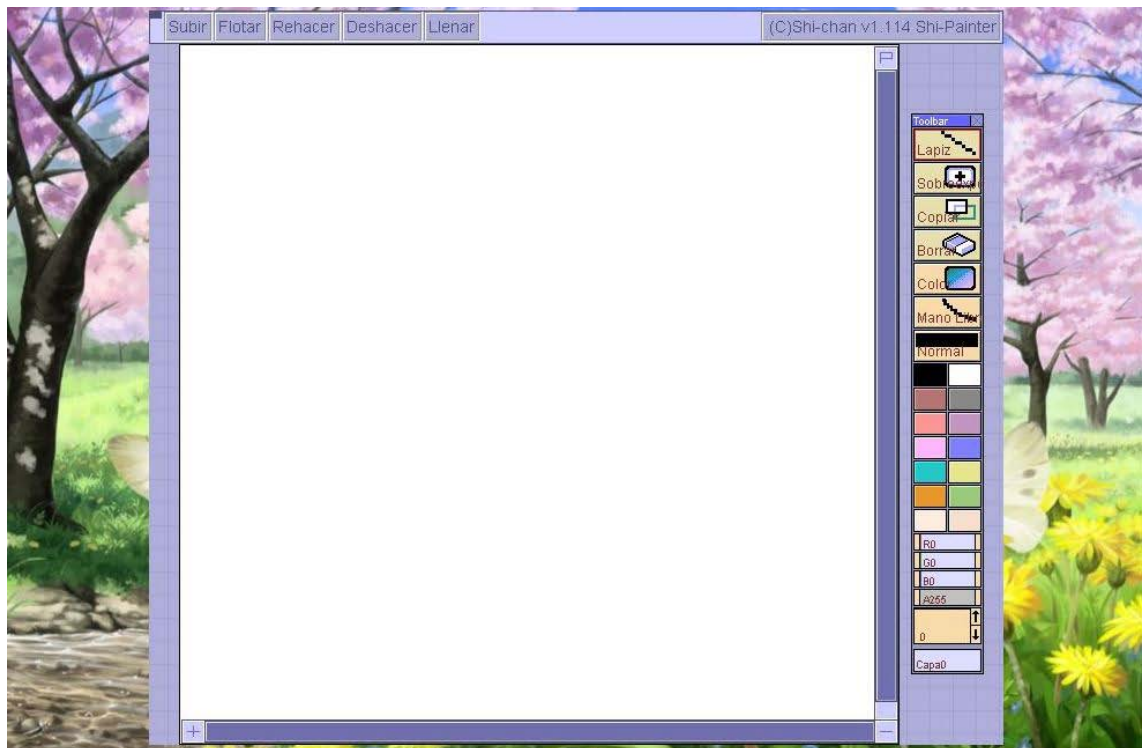
**deviantART** es la aplicación web más cercana a la aplicación de este proyecto. Como Web 2.0 cuenta con la participación de usuarios que quieren mostrar sus dibujos y se sometan a comentarios y críticas de otros usuarios. Cuenta además con un editor de dibujo **Sta.sh** que permite al usuario crear sus dibujos guardándolos automáticamente como bocetos pudiéndolo compartir mediante un link, hasta que el usuario finalmente de por finalizado el dibujo y decida subirlo a devianART. En un principio era sólo para usuarios Premium pero ahora esta accesible a todos los usuarios en general, orientado principalmente a los relacionados con las artes visuales y gráficas, como fotógrafos, diseñadores gráficos y pintores. Cuenta con una amplia lista de categorías que sirve para clasificar las obras que son subidas, organizadas primeramente en distintas ramas del arte, según el medio con el que fueron organizadas, subdivididas a su vez en subcategorías más específicas.



\* <http://www.deviantart.com/>

\* <http://sta.sh/muro/>

**oekaki** es otra página Web 2.0 para compartir dibujos entre personas que les apasiona dibujar. La herramienta está en Applet y para acceder hay que registrarse. También se hace concursos cada “x” tiempo para motivar a los usuarios.



\* <http://oekaki.newillusion.tk>



## 3 Recursos utilizados

En este capítulo se indicará los recursos de software y hardware que han sido necesarios para el desarrollo del proyecto.

### 3.1 Recursos software

**Microsoft Windows** concretamente Windows 7 como sistema operativo en que se ha desarrollado el proyecto.

**XAMPP** es una forma fácil de instalar la distribución Apache que contiene MySQL y los intérpretes para lenguajes de script: PHP y Perl.

**Apache** es un servidor web HTTP de código abierto y multiplataforma que implementa el protocolo HTTP.

**MySQL y phpMyAdmin** como gestor de base de datos.

**NetBeans** como entorno de desarrollo integrado (IDE).

**Java** como lenguaje de programación, mediante la plataforma JEE (Java Enterprise Edition) permitiendo utilizar arquitecturas de N capas.

**Javascript** permite realizar un cierto conjunto de operaciones en las páginas web desde el lado del usuario tales como: interacción con el usuario, control del navegador, comunicación asíncrona y posibilidad de modificar el documento web una vez mostrado.

**CSS** Cascading Style Sheets (Hojas de Estilo en Cascada). Se utilizan para dar formato a los elementos del documento web, dejando la estructura para el HTML o XHTML.

**JSP** JavaServerPages ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML o XML entre otros tipos de documentos. JSP es similar a PHP pero usa el lenguaje de programación Java.

**Microsoft Word** es un software destinado al procesamiento de textos. Fue creado por la empresa Microsoft, y actualmente viene integrado en la suite ofimática Microsoft Office. La redacción de la memoria y los manuales de usuario será confeccionado mediante este potente procesador de texto.

**StarUML** es un proyecto que intenta crear un UML lo más dinámico posible sin quitar utilidades.

**Pencil Evolus** ayuda a realizar mockups.

## 3.2 Recursos hardware

El proyecto se ha desarrollado en un portátil ASUS K52J con las siguientes características:

Procesador Intel Core i3 a 2,53 GHz.

Memoria RAM 4GB DDR3.

Tarjeta gráfica AMD Radeon HD 6370M.

Pantalla de 15,6 pulgadas.

Conexión Internet vía wifi.

El mismo portátil ha actuado de cliente servidor y se han realizado varias copias de seguridad en un disco duro externo para evitar imprevistos.

Para poder ejecutar la aplicación los requisitos mínimos son los siguientes:

- Máquina Virtual de Java (MVJ) 1.7 o superior
- Sistemas Operativos:
  - Windows NT, 2000, XP, 2003, Vista, Windows 7 y Windows 8
  - GNU/Linux
  - Apple MAC OS X compatibles con java 1.7
- Disco duro: espacio mínimo de 1 GB para instalación inicial y posterior almacenamiento.
- Memoria RAM: 512 MB.
- Resolución pantalla: 1024x768ppp. (Se recomienda 1366x768).
- Es necesario tener acceso a Internet.

## 4 Planificación del trabajo

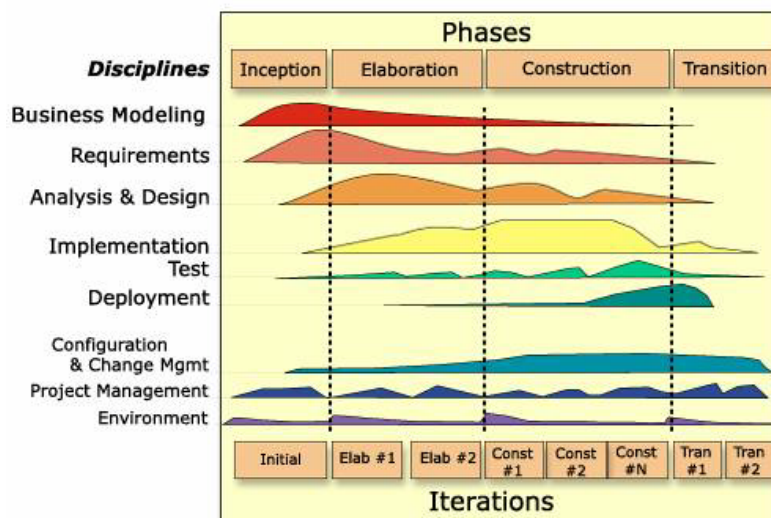
En este capítulo se describe la metodología aplicada para la realización del proyecto, se desglosarán las tareas a realizar y se elaborará una estimación temporal del desarrollo del mismo. También se hace una estimación de costes de mano de obra y material necesarios para realizar este proyecto.

### 4.1 Metodología de desarrollo

La metodología PUD (Proceso Unificado Desarrollo del Software) es la que se ha optado para la realización de este proyecto y puede definirse como el conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software. El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

El Proceso de Unificado tiene dos dimensiones:

- Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso a lo largo de su desenvolvimiento.
- Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo a su naturaleza.



La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos (milestones).

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos. Las más importantes son: Requisitos, Análisis, Diseño, Implementación y Pruebas. Estas disciplinas se realizan para cada una de las cuatro fases y cada ciclo constituye una versión del sistema.

**Fase de inicio** es la fase donde se desarrolla una descripción del producto final y se presenta al análisis de negocio. En esta fase se identifican y priorizan los riesgos más importantes. El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles y diferentes arquitecturas posibles. La fase de inicio finaliza con el Hito de Objetivos del Ciclo de Vida.

**Fase de elaboración** donde se especifican con detalle la mayoría de los casos de uso del producto y se diseña la arquitectura. Se construye el cuerpo básico del software en la forma de un prototipo arquitectural, los casos de prueba, la mayoría de los casos de uso que describen la funcionalidad del sistema y un plan detallado para las siguientes iteraciones. Finaliza con el hito de la Arquitectura del Ciclo de Vida.

**Fase de construcción** se crea el producto. La línea de la arquitectura crece hasta convertirse en el sistema completo. Al final de esta fase el producto contiene todos los casos de uso implementados, sin embargo puede que no esté libre de defectos. Finaliza con el hito de Capacidad Operativa Inicial.

**Fase de transición** cubre el período donde el producto se convierte en la versión beta. Se continúan agregando características al software que el usuario se encuentra utilizando activamente. Finaliza con el hito de Lanzamiento de Producto, y es cuando se han alcanzado los objetivos fijados en la fase de inicio y el usuario está satisfecho.

El Proceso Unificado se basa en componentes (component-based), lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas (well-defined interfaces) por lo que tenemos componentes reutilizables.

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave: dirigido por casos de uso (use-case driven), centrado en la arquitectura (architecture-centric), iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

### **Dirigido por casos de uso**

Un sistema de software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios prospectos.

El término usuario se refiere no solamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término usuario representa algo o alguien que interactúa con el sistema por desarrollar.

Un *caso de uso* es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el *modelo de casos de uso* el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Una especificación funcional tradicional se concentra en responder la pregunta: ¿Qué se supone que el sistema debe hacer? La estrategia de casos de uso puede ser definida agregando tres palabras al final de la pregunta: ¿por cada usuario? Estas tres palabras tienen una implicación importante, nos fuerzan a pensar en términos del valor a los usuarios y no solamente en términos de las funciones que sería bueno que tuviera. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo.

Aún y cuando los casos de uso dirigen el proceso, no son elegidos de manera aislada. Son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida.

### **Centrado en la arquitectura**

El papel del arquitecto de sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. El edificio se mira desde diferentes puntos de vista: estructura, servicios, plomería, electricidad, etc. Esto le permite al constructor ver una radiografía completa antes de empezar a construir. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido.

El concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros stakeholders, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales (ej. desempeño, confiabilidad). La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. Ya que lo importante depende en parte del criterio, el cual a su vez viene con la experiencia, el valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo, el proceso ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad (understandability), flexibilidad en los cambios futuros (resilience) y reúso.

¿Cómo se relacionan los casos de uso con la arquitectura? Cada producto tiene función y forma. Uno sólo de los dos no es suficiente. Estas dos fuerzas deben estar balanceadas para obtener un producto exitoso. En este caso función corresponde a los casos de uso y forma a la arquitectura. Existe la necesidad de intercalar entre casos de uso y arquitectura. Por una parte, los casos de uso deben, cuando son realizados, acomodarse en la arquitectura y la arquitectura debe proveer espacio para la realización de todos los casos de uso, hoy y en el futuro. En la realidad, arquitectura y casos de uso deben evolucionar en paralelo.



## Iterativo e incremental

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada.

Los desarrolladores basan su selección de qué van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

## 4.2 Planificación y temporización

En este apartado se ilustran las diferentes secciones y tareas de las que se compone este proyecto y que han sido necesarios cumplimentar para llegar a la conclusión del mismo. Las siguientes tablas muestran las diferentes tareas y su coste temporal que han sido necesarias para llevar a cabo el proyecto, desglosadas por las distintas secciones que componen el proyecto:

	Nombre de la tarea	Horas	Comienzo	Fin
<b>1</b>	<b>TFG</b>	<b>381</b>	<b>20/10/2012</b>	<b>22/11/2013</b>
<b>2</b>	<b>CUESTIONES PREVIAS A LA REALIZACIÓN DEL TFG</b>	<b>28</b>	<b>26/10/2012</b>	<b>28/10/2012</b>
<b>3</b>	Estudio metodología PUD	20	26/10/2012	28/10/2012
<b>4</b>	Aprender utilizar JSP	8	28/10/2012	28/10/2012
<b>5</b>	<b>ANÁLISIS DE REQUISITOS</b>	<b>40</b>	<b>20/10/2012</b>	<b>16/01/2013</b>
<b>6</b>	Definir casos de uso y actores	5	20/10/2012	20/10/2012
<b>7</b>	Lista de características	5	12/11/2012	16/01/2013
<b>8</b>	<b>DISEÑO DE LA APLICACIÓN</b>	<b>50</b>	<b>28/10/2012</b>	<b>06/11/2013</b>
<b>9</b>	Creación panel de usuario	2	28/10/2012	28/10/2012
<b>10</b>	Diseño de clases	18	28/10/2012	22/10/2013
<b>11</b>	Mejorar aspecto interfaz	30	25/06/2012	06/11/2013
<b>12</b>	<b>BASE DE DATOS</b>	<b>30</b>	<b>16/10/2012</b>	<b>19/11/2013</b>
<b>13</b>	Crear tablas y entidad relación	5	16/10/2012	16/10/2012
<b>14</b>	Configurar JDBC	5	20/06/2013	20/06/2013
<b>15</b>	Gestionar usuarios y dibujos	20	20/06/2013	19/11/2013

16	<b>IMPLEMENTACIÓN</b>	<b>150</b>	<b>28/10/2012</b>	<b>06/11/2013</b>
17	Desarrollo requisitos de usuario	75	28/10/2012	06/11/2013
18	Desarrollo requisitos servidor	75	28/10/2012	06/11/2013
19	<b>PRUEBAS</b>	<b>40</b>	<b>15/11/2013</b>	<b>19/11/2013</b>
20	Crear banco de pruebas	15	16/11/2013	17/11/2013
21	Verificación del proyecto	15	17/11/2013	18/11/2013
22	Conclusiones de las pruebas	10	18/11/2013	19/11/2013
23	<b>REALIZACIÓN DOCUMENTACIÓN</b>	<b>43</b>	<b>04/11/2013</b>	<b>17/11/2013</b>
24	Introducción	2	04/11/2013	04/11/2013
25	Estado actual del arte	4	12/11/2013	12/11/2013
26	Recursos utilizados	1	04/11/2013	04/11/2013
27	Planificación del trabajo	8	12/11/2013	12/11/2013
28	Desarrollo del trabajo	16	13/11/2013	14/11/2013
29	Conclusiones y trabajo futuro	3	14/11/2013	14/11/2013
30	Competencias	3	15/11/2013	15/11/2013
31	Documentación legislación vigente	2	16/11/2013	16/11/2013
32	Creación manual usuarios	2	17/11/2013	17/11/2013
33	Bibliografía	2	04/11/2013	17/11/2013

\* Nota: Hay tareas que distan su fecha de comienzo con la fecha de finalización y no emplean las horas de forma continua, sino más bien, están repartidas en días salteados.

### 4.3 Presupuesto

El presupuesto necesario para la realización de este proyecto se descompone en cuatro partes, detallándose el coste de la misma y al final se expone el coste completo de la realización del proyecto.

#### Coste de personal

Primero hay que determinar el coste por hora trabajada. En este caso se han determinado 15 euros cada hora trabajada por el estudiante, siendo una media de 20 días laborales por mes y 4 horas por día, nos da 80 horas al mes y multiplicando por los 15 euros cada hora trabajada podemos suponer unos 1200 euros brutos mensuales. Se consideran 30 euros por cada hora trabajada por el profesor siendo unos 2400 euros brutos mensuales.

Teniendo en cuenta los datos anteriores tendríamos los siguientes costes laborales:

- Total tutor: 24 horas x 30€ = 720€
- Total alumno: 381 horas x 15€ = 5715€

Concepto	Cantidad	Precio por unidad	Precio total
<b>Coste Laboral Tutor</b>	24	30€	720€
<b>Coste Laboral Alumno</b>	381	15€	5715€

### Costes inventariables

El material necesario para este proyecto es de un ordenador portátil. En cuanto al software necesario no supondrá ningún coste debido al uso de software libre. El coste del portátil es de 700 euros.

Concepto	Cantidad	Precio por unidad	Precio total
Equipo portátil	1	700€	700€

### Costes fungibles

Para la entrega del proyecto es necesario realizar una copia de la memoria y cinco copias en CDs siendo una para la administración, otra para la biblioteca y los tres restantes para los miembros del tribunal evaluador.

Precio aproximado de la memoria y encuadernado unos 13 euros.

Precio aproximado pack cinco CDs en 1,50 euros.

Concepto	Cantidad	Precio por unidad	Precio total
Memoria encuadernada	1	13€	13€
CDs	5	0,3€	1,5€

### Costes indirectos

Son aquellos que no dependen directamente del proyecto de la realización del proyecto, esto es: trabajo del personal de la administración, gastos en los servicios de alumbrado y de red del edificio de la EII, etc. Son costes difíciles de calcular por su complejidad y variedad, por lo que se ha presupuestado un 5% del total del presupuesto para simplificar.

$(\text{Coste de personal} + \text{Costes inventariables} + \text{Costes fungibles}) * 5\% = 7149,5 * 5\% = 357,50 \text{ €}$

Concepto	Total	Porcentaje	Precio total
CP+CI+CF	7149,5	5%	357,5€

### Total del presupuesto

A continuación se muestra una tabla resumen con los costes asociados al proyecto:

Concepto	Costes
Costes laborales del alumno	5715€
Costes laborales del tutor	720€
Costes materiales	700€
Costes de documentación	14,5€
Costes indirectos	357,5€
<b>PRESUPUESTO TOTAL</b>	<b>7507€</b>

## 5 Desarrollo del trabajo

En esta sección seguiremos las directrices que marca la metodología de desarrollo. Se ilustrarán varios de los diferentes artefactos que se han realizado hasta llegar a obtener el producto final, desde la recopilación de requisitos hasta la etapa de diseño del software.

Primero se identifican los requisitos del sistema y los requisitos del software que servirá como base para definir y delimitar el ámbito del proyecto, se utilizará el modelo del dominio, la lista de características y los casos de uso. Después se define cualquier característica que fuese deseable incluir en la aplicación y por último se perfila de forma más concreta las funciones que se esperan del mismo.

Al utilizar la metodología PUD se sigue el siguiente esquema:

### 5.1 Requisitos del sistema

Los requisitos del sistema para poder desplegar la aplicación del proyecto vienen siendo:

- Requisitos mínimos de hardware descrito en el apartado de Recursos hardware.
- Máquina Virtual de Java en el servidor de despliegue para conectar con la base de datos.

Se describirán dos artefactos muy importantes a la hora de establecer el ámbito en el que moverá la aplicación, siendo el modelo del dominio y la lista de características ayudando a los desarrolladores a comprender del contexto del sistema y poder recopilar requisitos funcionales y no funcionales.

#### 5.1.1 Modelo del dominio

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, prestando como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino e la propia realidad física.

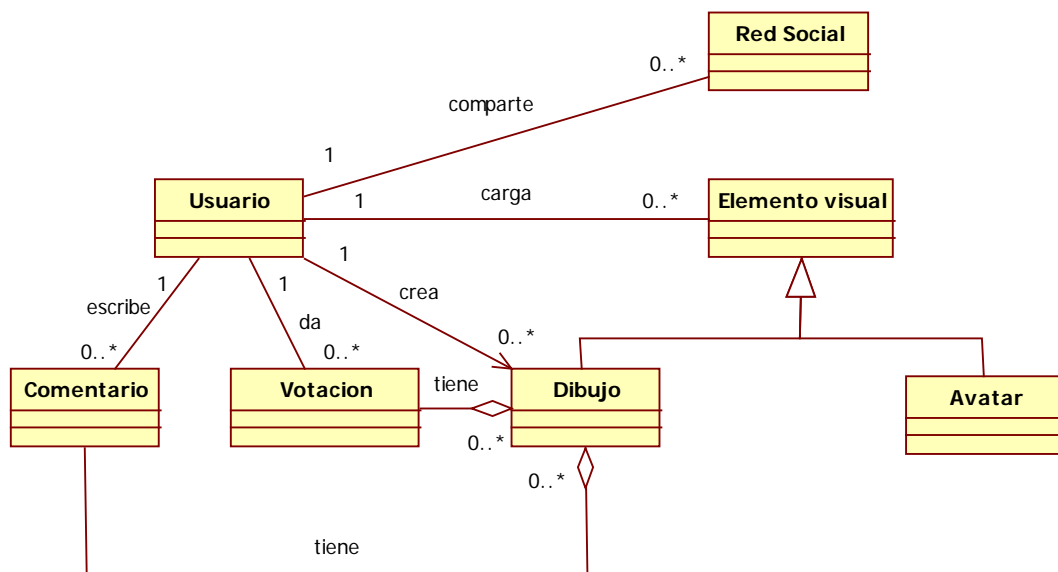
Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

El modelo de dominio puede ser tomado como el punto de partida para el diseño del sistema. Esto es así ya que cuando se realiza la programación orientada a objetos, se supone que el funcionamiento interno del software constituye una primera versión del sistema.

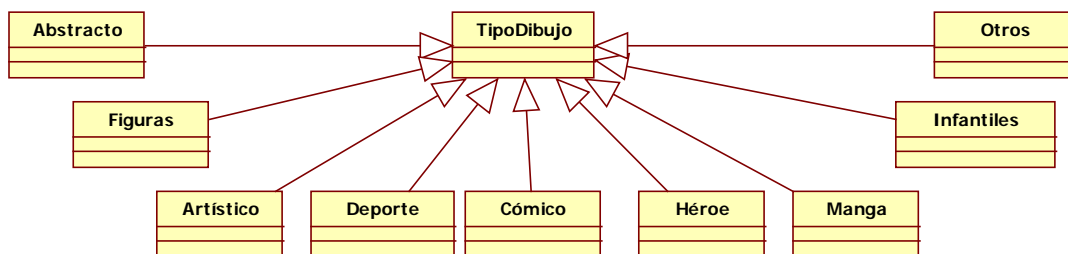
El uso del modelo del dominio y el glosario de términos se utilizan en el desarrollo de los modelos de los casos de uso y de análisis para:

- Describir los casos de uso y diseñar la interfaz de usuario.
- Sugerir clases internas al sistema en desarrollo durante el análisis.

El siguiente diagrama refleja lo que el usuario puede hacer en la aplicación Web 2.0. Por un lado puedes compartir



En este otro diagrama se aprecia los diferentes tipos de dibujo que el usuario puede subir o crear.



### 5.1.2 Lista de características

La lista de características es un artefacto que se obtiene después de aplicar la tarea de "Enumerar los requisitos candidatos" que se propone en la metodología PUD (Proceso Unificado de Desarrollo), para la captura de requisitos. Por lo tanto, se encuentra englobada dentro de la fase de inicio. Esta lista sirve para contener las ideas de clientes, usuarios, analistas y desarrolladores a modo de fichas sobre los posibles aspectos que se podrían incluir en la aplicación, y que, posteriormente, se podrían traducir en requisitos del software. Estas ideas se consideran requisitos candidatos que se podrían desarrollar en la versión actual del sistema o se podrían postergar a versiones futuras. Este artefacto sirve para gestionar el proyecto y sólo se utiliza para la planificación del trabajo. Podría ir variando a medida que avance el proyecto, pudiéndose añadir y modificar las características que se crean oportunas, en cualquier momento del desarrollo.

La tabla de la lista de características se compone de:

- ID: Identificador de la característica para dividir las características en varios grupos (Gestión de usuario, Gestión de dibujo, etc.).
- Nombre: el nombre de la característica.
- Descripción: una breve descripción de la funcionalidad de la característica.
- Prioridad: orden de desarrollo de la característica.
- Estado: indica cómo va evolucionando el desarrollo de la característica. Los posibles valores son: Aprobado, Pendiente y Finalizado.

ID	NOMBRE	DESCRIPCION	PRIORIDAD	ESTADO
LC-GD-1	Abrir documento	Abre un archivo con extensión compatible con la App.	Muy alta	Finalizado
LC-GD-2	Guardar documento	Guarda el documento con extensión compatible con la App.	Muy alta	Finalizado
LC-GD-3	Descargar dibujo	Descarga documento que está abierto.	Alta	Pendiente
LC-GD-4	Borrar documento	Elige y borra un documento subido al servidor.	Muy alta	Finalizado
LC-GD-5	Subir documento	Sube el dibujo al servidor.	Muy alta	Finalizado
LC-GD-6	Importar dibujo	Elige e importa un dibujo subido del servidor.	Muy alta	Aprobado
LC-GD-7	Control usuarios documento	Sube un dibujo previamente importado, con otro nombre.	Alta	Aprobado
LC-GD-8	Renombrar documento	Cambia el nombre del dibujo.	Alta	Aprobado
LC-GD-9	Elegir tamaño documento.	Modifica el tamaño del dibujo.	Media	Pendiente
LC-GD-10	Añadir hipervínculo documento.	Permite mostrar y copiar hipervínculo del dibujo.	Alta	Finalizado
LC-GD-11	Compartir	Comparte documento del dibujo para	Alta	Pendiente

	documento	poder realizar modificación simultánea con otros usuarios.		
LC-GD-12	Compartir documento con redes sociales	Poder compartir el documento de dibujo con redes sociales (facebook, Twitter, etc.)	Alta	Aprobado
LC-GD-13	Comentar dibujo	Permite dejar un comentario al dibujo.	Muy alta	Finalizado
LC-GD-14	Votar dibujo	Da un voto positivo o negativo al dibujo.	Muy alta	Finalizado
LC-D-1	Insertar figura	Inserta una figura simple o geométrica en el documento.	Muy alta	Aprobado
LC-D-2	Borrar figura	Borra una figura insertada.	Alta	Aprobado
LC-D-3	Mover figura	Mueve una figura insertada.	Alta	Aprobado
LC-D-4	Cambiar color contorno figura	Cambia el color del contorno de la figura.	Alta	Aprobado
LC-D-5	Cambiar grosor contorno figura	Cambia el grosor de la figura.	Alta	Aprobado
LC-D-6	Colorear figura	Rellena la figura con un color.	Alta	Aprobado
LC-D-7	Copiar figura	Realiza una copia de la figura.	Alta	Aprobado
LC-D-8	Cortar figura	Corta la figura para moverla/pegarla en otra parte o en otro documento de dibujo	Alta	Aprobado
LC-D-9	Pegar	Pega una figura.	Alta	Aprobado
LC-D-10	Insertar texto	Añade texto al documento dibujo.	Media	Pendiente
LC-D-11	Cambiar tamaño figura	Modifica tamaño de la figura.	Alta	Aprobado
LC-D-12	Girar figura	Realiza una rotación a la figura.	Media	Pendiente
LC-D-13	Escalar figura	Escala la figura.	Media	Pendiente
LC-D-14	Comprobar color	Obtener color como si fuera la función cuenta gotas.	Media	Pendiente
LC-D-15	Borrar parte de la figura	Usar una goma que borre solo parte del dibujo.	Alta	Pendiente
LC-D-16	Insertar trazo libre	Manejar pincel a mano alzada.	Muy alta	Finalizado
LC-D-17	Hacer Zoom in	Acerca el zoom del dibujo.	Alta	Pendiente
LC-D-18	Hacer Zoom out	Aleja el zoom del dibujo.	Alta	Pendiente
LC-D-19	Deshacer	Deshace una acción.	Alta	Finalizado
LC-D-20	Rehacer	Rehace una acción.	Alta	Aprobado
LC-D-21	Aplicar efectos artísticos	Brillo, contraste, ...	Media	Pendiente
LC-D-22	Elegir color pincel	Permite al usuario elegir un color	Muy alta	Finalizado
LC-D-23	Elegir figura simple	El usuario elige una línea, cuadrado, rectángulo, círculo o triángulo.	Muy alta	Aprobado
LC-D-24	Elegir figura geométrica	El usuario elige un polígono de 5 lados o más.	Alta	Aprobado
LC-GU-1	Registrar usuario	El visitante de la página podrá registrarse.	Muy alta	Finalizado
LC-GU-2	Iniciar sesión	Da acceso a privilegios de usuario.	Muy alta	Finalizado
LC-GU-3	Modificar contraseña	El usuario puede modificar sus datos.	Alta	Aprobado
LC-GU-4	Recordar contraseña	Mediante preguntas cuyas respuestas solo conozca el usuario para recuperar contraseña.	Alta	Aprobado
LC-GU-5	Modificar foto perfil	Actualiza la foto perfil del usuario.	Muy alta	Finalizado
LC-GU-6	Cancelar cuenta	Permite dar de baja a un usuario.	Alta	Aprobado

## 5.2 Requisitos del software

Describen el comportamiento de la aplicación que se va a desarrollar. Incluye un conjunto de casos de uso que describe las interacciones que tendrá el usuario con la aplicación, deducidos a partir de la información obtenida con el modelo de dominio y la lista de características.

Para cada usuario se crean casos de usos que podrá realizar cada uno, definiéndose los pasos concretos.

### 5.2.1 Actores

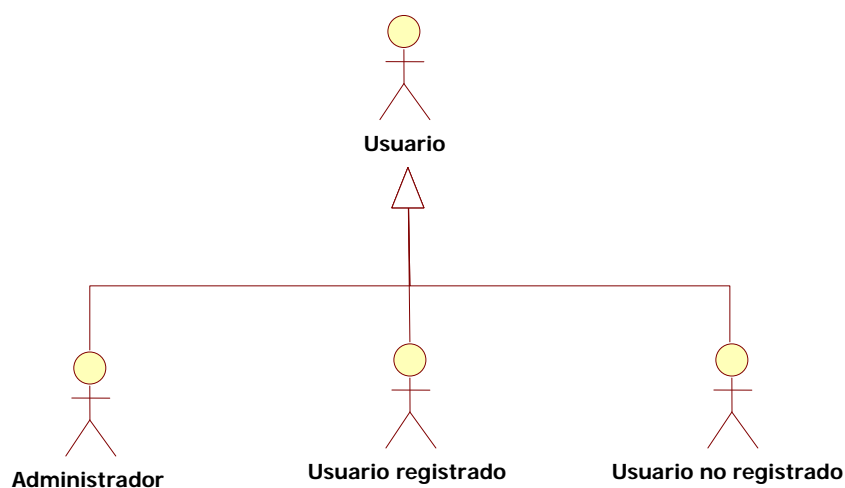
Los actores son los que utilizan el sistema interactuando con los casos de uso. Se identifican los usuarios representados en actores, habiendo actores que representen sistemas externos y actores para el mantenimiento del sistema.

Para la aplicación se han identificado los siguientes actores:

**Usuario no registrado** siendo aquel usuario que no está identificado en el sistema. No podrá acceder a ciertas partes de la aplicación.

**Usuario registrado** es el actor principal ya que la mayor parte de la funcionalidad del sistema se encuentra en los casos de uso de este actor.

**Administrador** se encarga del mantenimiento y se responsabiliza del buen funcionamiento del sistema.





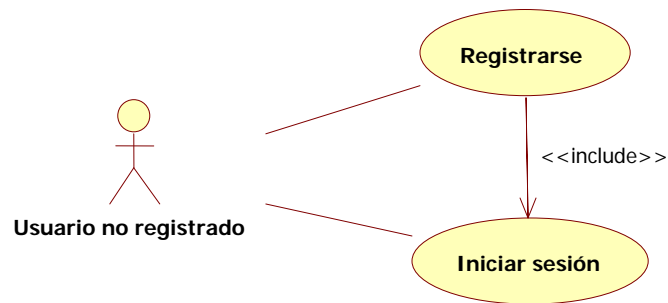
## 5.2.2 Modelo de casos de uso

Para seguir un orden y no dificulte entender los casos de uso, se dividirán en varios grupos:

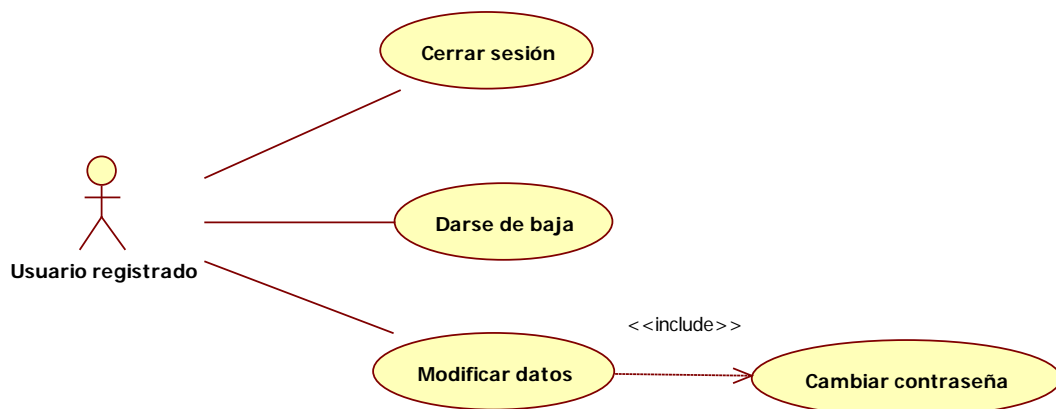
- Casos de uso Gestión de usuario
- Casos de uso Gestión de dibujo
- Casos de uso Dibujo

Empezando con los casos de uso Gestión de usuario, son las acciones que puede realizar un **Usuario no registrado**, tales como Visualizar dibujo, Registrarse o Iniciar sesión.

Al principio el **Usuario no registrado** podrá registrarse para pasar a ser **Usuario registrado** ejecutando automáticamente Iniciar sesión para identificarlo dentro de la aplicación.

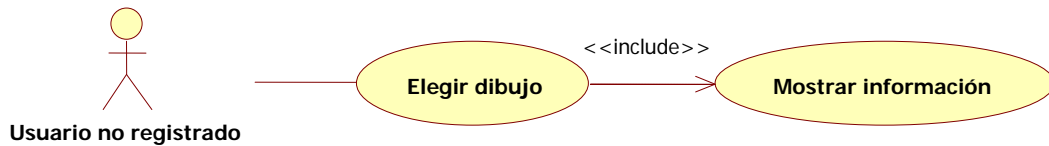


El **Usuario** cuando esté dentro del sistema podrá cerrar sesión, cerrar su cuenta o cambiar datos de su cuenta.



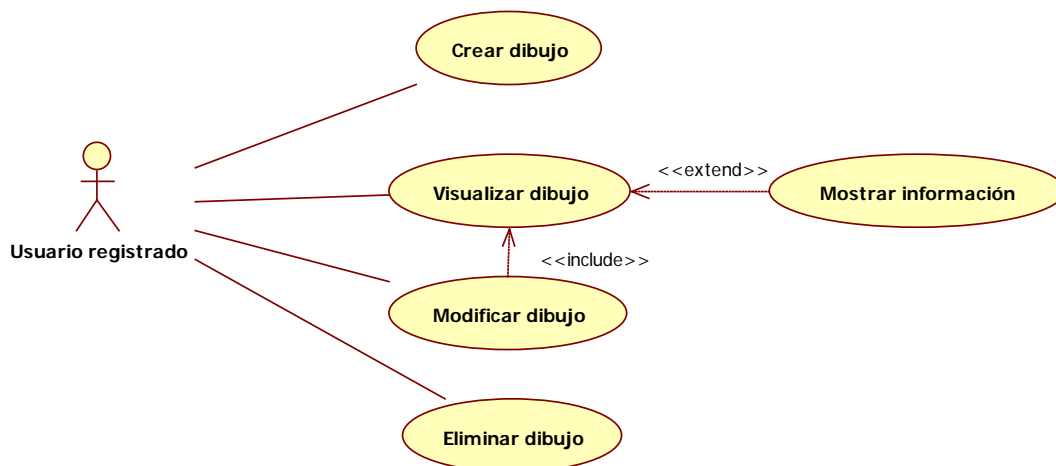
En casos de uso de Gestión de dibujo, continuamos con el **Usuario no registrado**.

Otra acción que puede hacer el **Usuario no registrado** es elegir un dibujo que esté subido en la aplicación y mostrar su información, que viene siendo el título del dibujo, la fecha, los comentarios, etc.

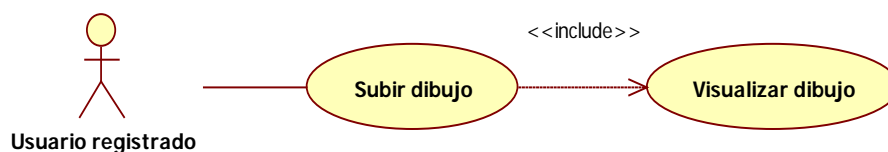


A continuación, se destaca las diferentes acciones que puede realizar el **Usuario** sobre un dibujo siguiendo la filosofía CRUD (Create-Read-Update-Delete).

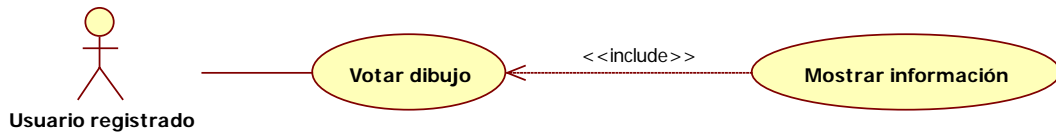
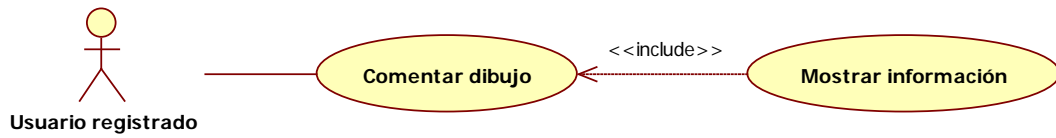
Cuando el usuario crea o modifica un dibujo, posteriormente se muestra en el perfil de Usuario para ver los resultados y puede optar a ver la información del mismo. Para poder modificar el dibujo, previamente se debe visualizar.



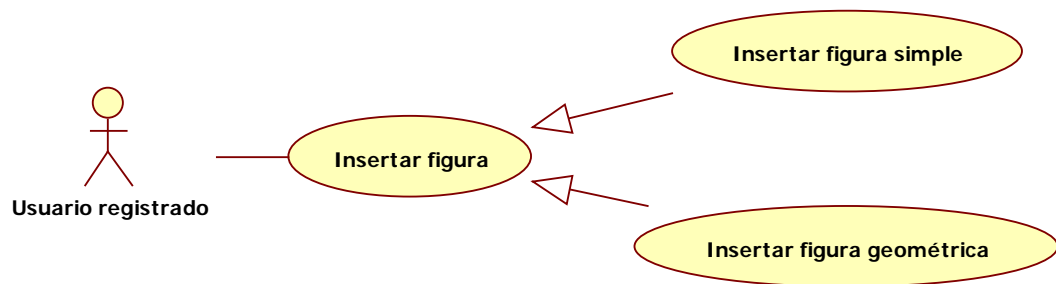
Subir dibujo podría estar en el anterior diagrama pero para no añadir más acciones diferentes a CRUD se opta por representarlo en otro diagrama. Para subir dibujo, el **Usuario** elige un dibujo que tendrá guardado en su equipo y será mostrado en una vista previa.



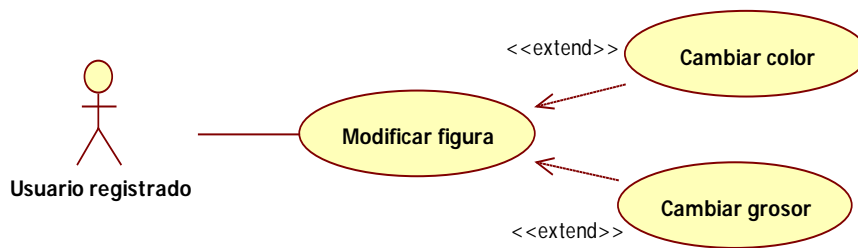
Finalmente el **Usuario** puede comentar y/o votar dibujo una vez haya elegido y mostrado la información del dibujo.



Y por último en casos de uso de Dibujo se muestra las acciones que puede hacer el Usuario cuando crea dibujo, al insertar figura.



Posteriormente se puede modificar las figuras insertadas en el dibujo, optando por cambiar de color o grosor.



### 5.2.3 Especificación de casos de uso

Continuando con los casos de uso se muestra en una tabla la siguiente información:

- Actor que realiza la acción.
- Descripción breve del caso de uso.
- Precondición y poscondición.
- Camino normal de cómo se ejecutaría el caso de uso.
- Camino alternativo cuando surgen condiciones que puedan afectar en su ejecución.

CASO DE USO: Registrarse
<b>Actor:</b> Usuario no registrado.
<b>Descripción:</b> Paso previo para ser un Usuario registrado de la aplicación.
<b>Precondición:</b> <b>Poscondición:</b> Usuario registrado, sesión iniciada
<b>Camino normal:</b> <ol style="list-style-type: none"><li>1. El usuario accede al formulario de registrar usuario.</li><li>2. El usuario proporciona sus datos:<ol style="list-style-type: none"><li>1. Nombre</li><li>2. Contraseña</li><li>3. Repetir contraseña</li><li>4. Correo electrónico</li></ol></li><li>3. El usuario hace clic en aceptar del formulario.</li><li>4. La aplicación verifica que no esté el mismo nombre registrado y comprueba las dos contraseñas para comprobar que no se ha equivocado.</li><li>5. Si las comprobaciones han ido bien, el usuario quedará registrado en la base de datos.</li><li>6. El usuario se autentifica.</li><li>7. Se muestra en la página principal como sesión iniciada al usuario.</li></ol>
<b>Camino alternativo:</b> <ol style="list-style-type: none"><li>4. Algún campo está vacío.</li><li>4. Nombre del usuario está registrado.</li><li>4. Las contraseñas no coinciden</li></ol>

<b>CASO DE USO: Registrarse</b>
<b>Actor:</b> Usuario no registrado.
<b>Descripción:</b> Iniciar sesión como Usuario.
<b>Precondición:</b> <b>Poscondición:</b> Sesión inicia
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario accede al formulario de iniciar sesión.</li> <li>2. El usuario proporciona sus datos: <ol style="list-style-type: none"> <li>1. Nombre</li> <li>2. Contraseña</li> </ol> </li> <li>3. El usuario hace clic en aceptar del formulario.</li> <li>4. La aplicación verifica que está registrado en la base de datos.</li> <li>5. El usuario se autentifica.</li> <li>6. Se muestra en la página principal como sesión iniciada al usuario.</li> </ol>
<b>Camino alternativo:</b> <ol style="list-style-type: none"> <li>4. Algún campo está vacío.</li> <li>4. Nombre del usuario no está registrado.</li> <li>4. Contraseña mal puesta.</li> </ol>

<b>CASO DE USO: Cerrar sesión</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Finalizar la sesión de usuario.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Fin de sesión
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario hace clic en cerrar sesión.</li> <li>2. La aplicación "destruye" la sesión.</li> </ol>

<b>CASO DE USO: Elegir dibujo</b>
<b>Actor:</b> Usuario no registrado.
<b>Descripción:</b> Elige un dibujo que quiera ver.
<b>Precondición:</b> <b>Poscondición:</b>
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. Elige un dibujo que está subido en la aplicación.</li> <li>2. Se muestra la información del dibujo.</li> </ol>

La especificación del caso de uso de Gestión de dibujo (CRUD) se dividirá en tres partes

<b>CASO DE USO: Crear dibujo</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Gestión de dibujo para crear dibujo siguiendo la filosofía CRUD.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Dibujo creado subido al servidor
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario hace clic a crear dibujo.</li> <li>2. Termina el dibujo y continúa con el formulario de subir dibujo para: <ol style="list-style-type: none"> <li>1. Insertar título (obligado).</li> <li>2. Una breve descripción.</li> <li>3. Elegir público o privado.</li> <li>4. Etiquetar dibujo.</li> </ol> </li> <li>3. Se verifica los datos.</li> <li>4. Si todo ha ido bien se registra en la base de datos.</li> <li>5. Se sube el dibujo al servidor.</li> <li>6. Se muestra el dibujo en el perfil de usuario.</li> </ol>
<b>Camino alternativo:</b> <ol style="list-style-type: none"> <li>3. Título del dibujo vacío.</li> </ol>

<b>CASO DE USO: Modificar dibujo</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Gestión de dibujo para modificar dibujo siguiendo la filosofía CRUD.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Dibujo modificado subido al servidor
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario hace clic a modificar dibujo.</li> <li>2. Se muestra el dibujo que se quiere modificar.</li> <li>3. Se termina modificación del dibujo y se guardan al servidor.</li> </ol>

<b>CASO DE USO: Eliminar dibujo</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Gestión de dibujo para eliminar dibujo siguiendo la filosofía CRUD.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Dibujo eliminado del servidor
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario hace clic a eliminar dibujo.</li> <li>2. Se verifica los datos del dibujo.</li> <li>3. Se elimina el dibujo de la base de datos y del servidor.</li> </ol>

<b>CASO DE USO: Subir dibujo</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Gestión de dibujo para subir dibujo siguiendo la filosofía CRUD.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Dibujo cargado
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>1. El usuario hace clic a subir dibujo.</li> <li>2. Elige un dibujo que tiene el usuario en su PC, mostrándose en la vista previa.</li> <li>3. Continúa con el formulario de subir dibujo para: <ol style="list-style-type: none"> <li>1. Insertar título (obligado).</li> <li>2. Una breve descripción.</li> <li>3. Elegir público o privado.</li> <li>4. Etiquetar dibujo.</li> </ol> </li> <li>4. Se verifica los datos.</li> <li>5. Si todo ha ido bien se registra en la base de datos.</li> <li>6. Se sube el dibujo al servidor.</li> <li>7. Se muestra el dibujo en el perfil de usuario.</li> </ol>
<b>Camino alternativo:</b> <ol style="list-style-type: none"> <li>3. Título del dibujo vacío.</li> </ol>

<b>CASO DE USO: Comentar dibujo</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Dejar comentario en el dibujo.
<b>Precondición:</b> Sesión iniciada <b>Poscondición:</b> Comentario visible al usuario



**Camino normal:**

1. El usuario elige un dibujo y se muestra información del mismo.
2. Escribe un comentario y envía al servidor.
3. El comentario queda registrado en la base de datos.
4. Se visualiza por pantalla como más reciente.

**CASO DE USO: Votar dibujo****Actor:** Usuario registrado.**Descripción:** Dar voto al dibujo.**Precondición:** Sesión iniciada**Poscondición:** Incrementa el voto del dibujo**Camino normal:**

1. El usuario elige un dibujo y se muestra información del mismo.
2. Hace clic en voto positivo o negativo.
3. Se envía el voto a la base de datos para incrementar el número.
4. Se visualiza por pantalla la votación.

**CASO DE USO: Insertar figura****Actor:** Usuario registrado.**Descripción:** Insertar figura al dibujo.**Precondición:** Sesión iniciada y estar en editor de dibujo**Poscondición:****Camino normal:**

1. El usuario inserta una figura simple o geométrica para dar forma al dibujo
2. Se muestra por pantalla

<b>CASO DE USO: Modificar figura</b>
<b>Actor:</b> Usuario registrado.
<b>Descripción:</b> Modificar figura del dibujo.
<b>Precondición:</b> Sesión iniciada y estar en editor de dibujo <b>Poscondición:</b>
<b>Camino normal:</b> <ol style="list-style-type: none"> <li>3. El usuario puede cambiar el color o el grosor de la figura que posteriormente va ser insertada.</li> <li>4. Se muestra por pantalla</li> </ol>

#### 5.2.4 Prototipo de interfaz de usuario

Otro elemento a tener en cuenta en la etapa de captura de requisitos es el prototipado de la interfaz de usuario, que ayuda a comprender como interactuaría el usuario con la aplicación. Este proceso se realiza partiendo de los casos de uso para cada usuario del sistema.

Se dispondrá de un conjunto de esquemas y prototipos de interfaces de usuario que describen la apariencia de la futura aplicación. Se debe asegurar que cada caso de uso es accesible a través de los elementos de la interfaz y que el conjunto de casos de uso accesibles para cada usuario tenga una interfaz de usuario bien entregada.

Hay que diferenciar entre diseño lógico del prototipo conteniendo los objetos necesarios para la interacción y el diseño físico que es el equivalente en forma de ventanas.

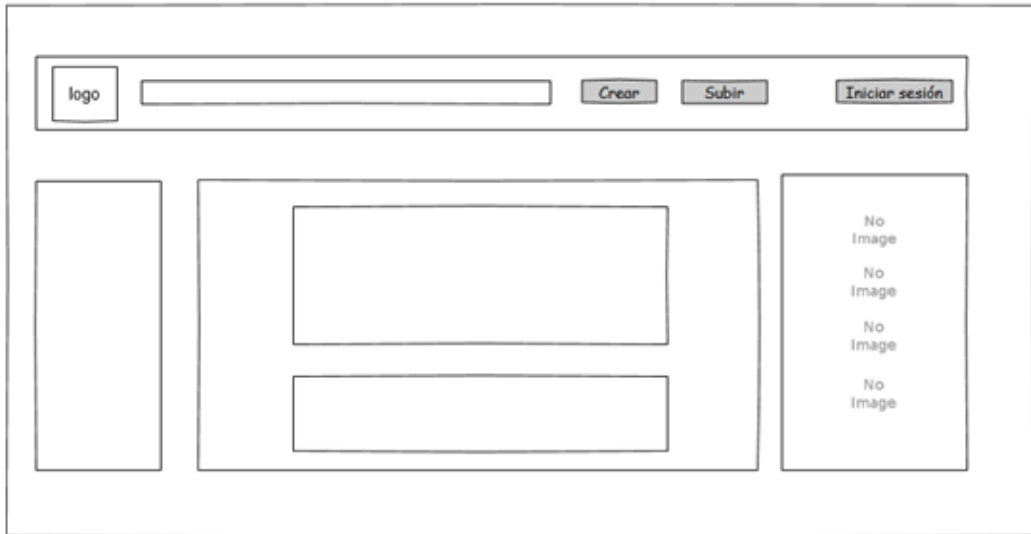
##### **Página principal**

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Dibujo: conjunto matriz de pixeles que representan en el caso de formato JPG el RGB o en el caso de formato GIF que puede contener entre 2 y 256 colores.

Diseño físico. La página está dividida en cuatro partes:

Primero se muestra una cabecera donde aparece el logo de la aplicación, un buscador y los botones crear/subir dibujo e iniciar sesión. En la parte central se mostrarán los dibujos subidos más recientemente y los diferentes tipos de dibujo. En la parte izquierda se muestra información de búsqueda por canales, más populares, etc. Y en la parte derecha los dibujos más votados.

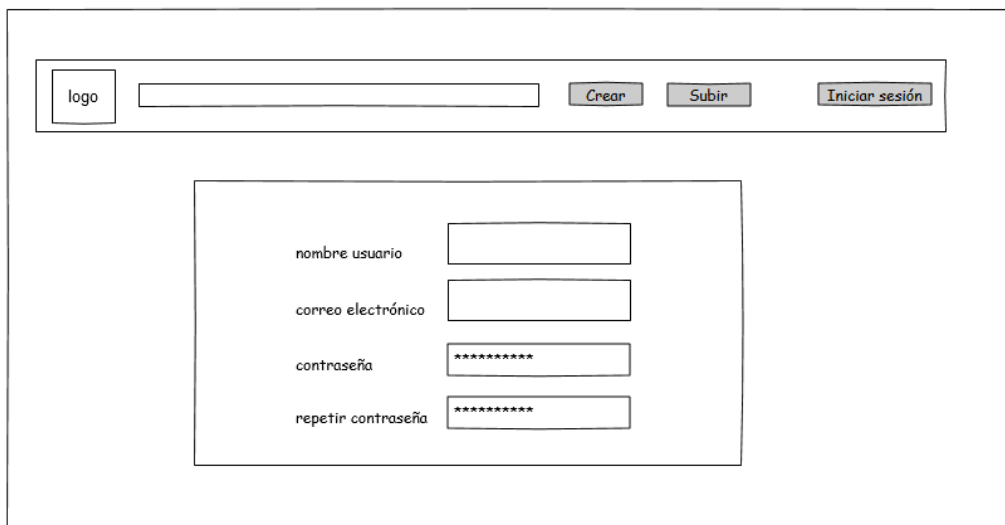


### Página registro de usuario

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Nombre de usuario: nombre del usuario para registrarse en la aplicación (String).
- Correo electrónico: correo electrónico para activar cuenta de usuario (String).
- Contraseña: contraseña que servirá para autenticarse en la aplicación (String).
- Repetir contraseña: debe ser igual a la primera contraseña (String).

Diseño físico. Se mantiene la cabecera y la parte central aparece el formulario a rellenar por el usuario para tener una cuenta y tener privilegios de usuario registrado.



## Página iniciar sesión

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Nombre de usuario: nombre del usuario registrado en la aplicación (String).
- Contraseña: contraseña que servirá para autenticarse en la aplicación (String).

El diagrama muestra un formulario de inicio de sesión con los siguientes elementos:

- Un recuadro superior que contiene un campo de texto etiquetado como "logo", un botón "Crear", un botón "Subir" y un botón "Regístrate".
- Un recuadro central que contiene dos campos de entrada: "nombre usuario" y "contraseña". El campo de "contraseña" muestra caracteres ocultos por asteriscos.

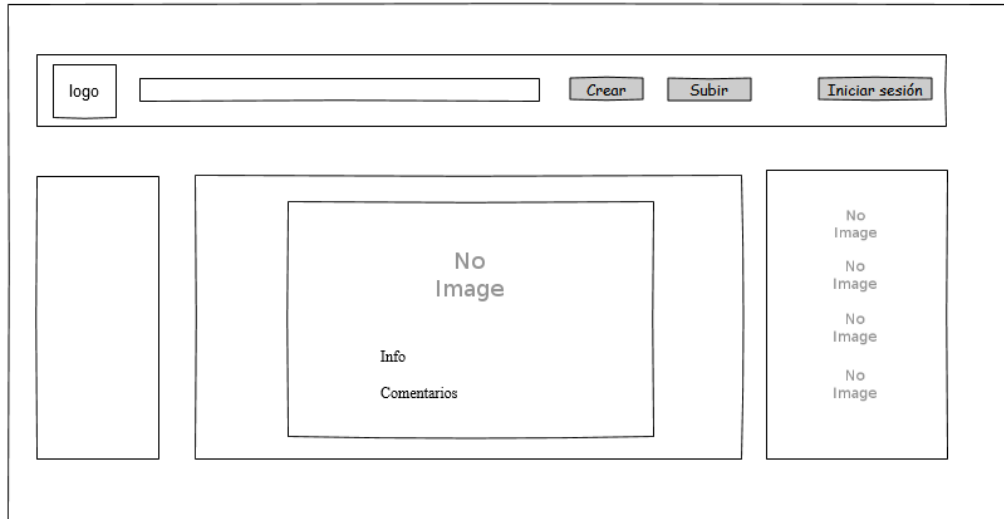
Diseño físico. Idéntico a la página de registro de usuario cambiando el botón de iniciar sesión por crear una cuenta y luego el formulario que solamente nos pide el nombre de usuario y la contraseña para autenticarse.

## Página información de dibujo

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Dibujo: conjunto matriz de pixeles que representan en el caso de formato JPG el RGB o en el caso de formato GIF que puede contener entre 2 y 256 colores.
- Comentario: comentario del usuario que quedará relacionada con el dibujo (String).
- Votación: es el del usuario al dibujo (Integer).

Diseño físico. Muy parecida a la página principal donde en el panel central se muestra el dibujo que se quiere visualizar y la información del dibujo junto a los comentarios que se hayan escrito. El panel de la derecha son los dibujos relacionados con el dibujo mostrado.

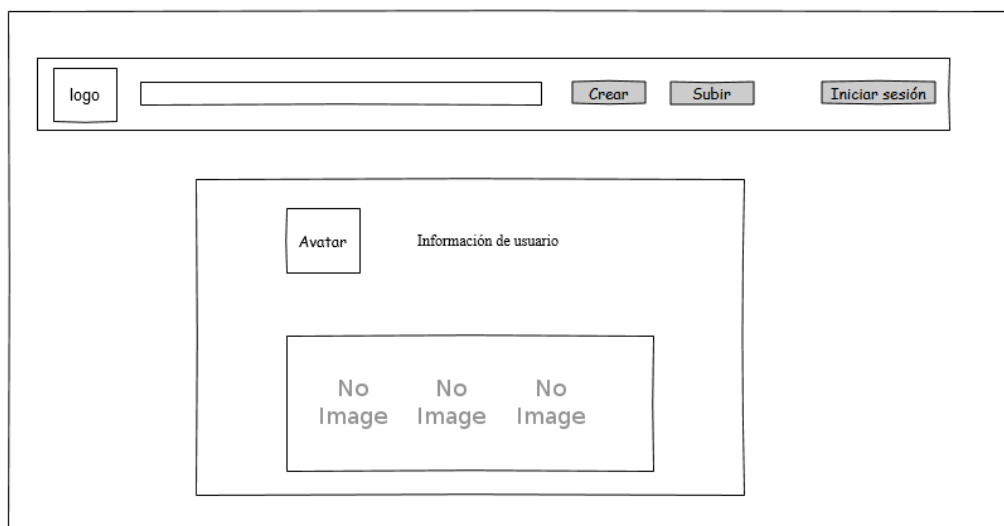


### Página perfil de usuario

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Dibujo: conjunto matriz de pixeles que representan en el caso de formato JPG el RGB o en el caso de formato GIF que puede contener entre 2 y 256 colores.

Diseño físico. Muestra en el panel central la información del usuario junto con una foto perfil. Al final de la página aparecen de forma paginada los dibujos que el usuario haya subido o creado.

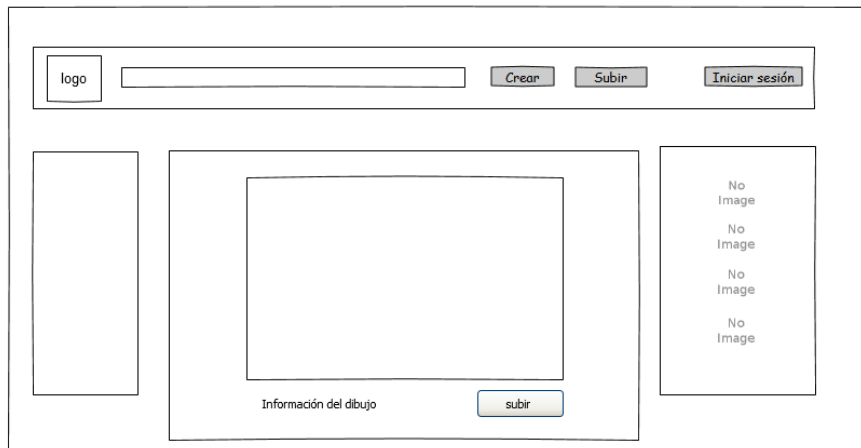


## Página subir dibujo

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Dibujo: conjunto matriz de pixeles que representan en el caso de formato JPG el RGB o en el caso de formato GIF que puede contener entre 2 y 256 colores (String).
- Título: título que el usuario debe poner al dibujo (String).

Diseño físico. Se muestra una vista provisional del dibujo que se quiere subir.

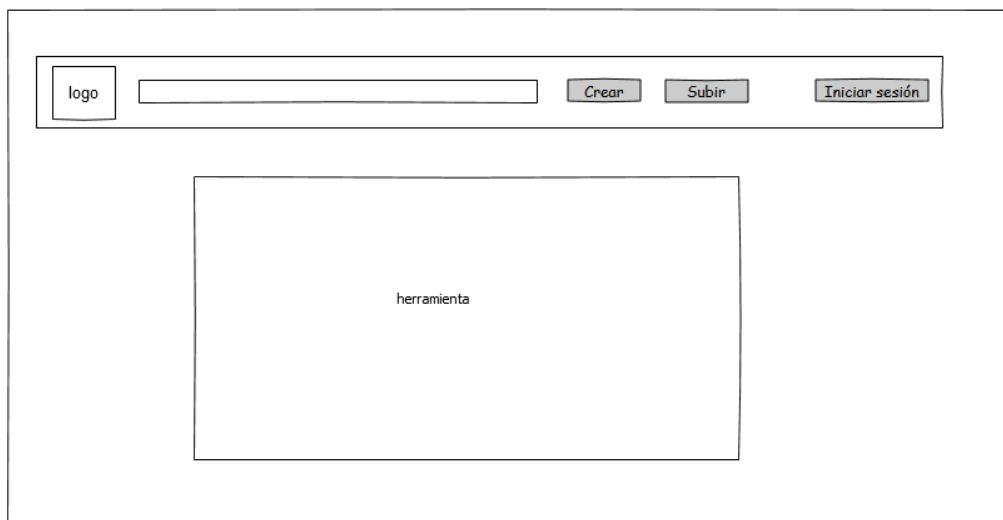


## Página crear dibujo

Diseño lógico, los elementos de la interfaz de usuario necesarios son:

- Dibujo: conjunto matriz de pixeles que representan en el caso de formato JPG el RGB o en el caso de formato GIF que puede contener entre 2 y 256 colores (String).

Diseño físico. En la parte central aparece el editor de dibujo.



## 5.3 Modelo de análisis

El modelo de análisis debe centrarse en los requisitos visibles dentro del problema o dominio de negocio. Se debe minimizar el acoplamiento de todo el sistema, debe tener la seguridad de que el modelo de análisis proporciona valor a todos los interesados y mantenerse tan simple como sea posible.

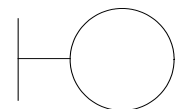
### 5.3.1 Organización del modelo análisis

Se realizarán los diagramas de clase para identificar los elementos estructurales que intervienen en el caso de uso especificado. Posteriormente se obtendrá el diagrama de colaboración, en el que quedará reflejado cómo interactúan estas entidades, tanto en la forma de comunicarse (mensajes que se envían), como en la temporización de la interacción (orden que se realiza cada paso de la comunicación).

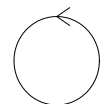
### 5.3.2 Diagramas de clases

En un diagrama de clases de modelo de análisis se utilizan tres tipos de clase:

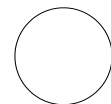
- Interfaces: clases que actúan de intermediario entre usuario y sistema, o entre sistemas.
- Control: son las clases que realizan la lógica del modelo de negocio. Sirven para coordinar y comunicar clases (interfaces, datos y otras clases de control).
- Datos: representan un objeto de datos estáticos, normalmente una entidad del modelo de negocio. Generalmente son manipuladas por clases de control.



**Interfaz**



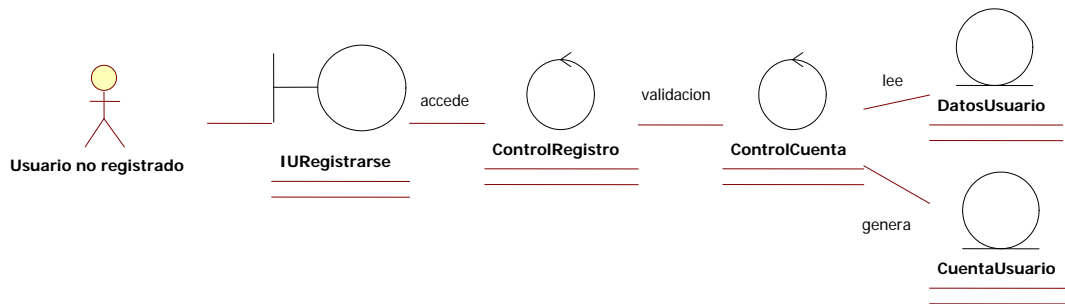
**Control**



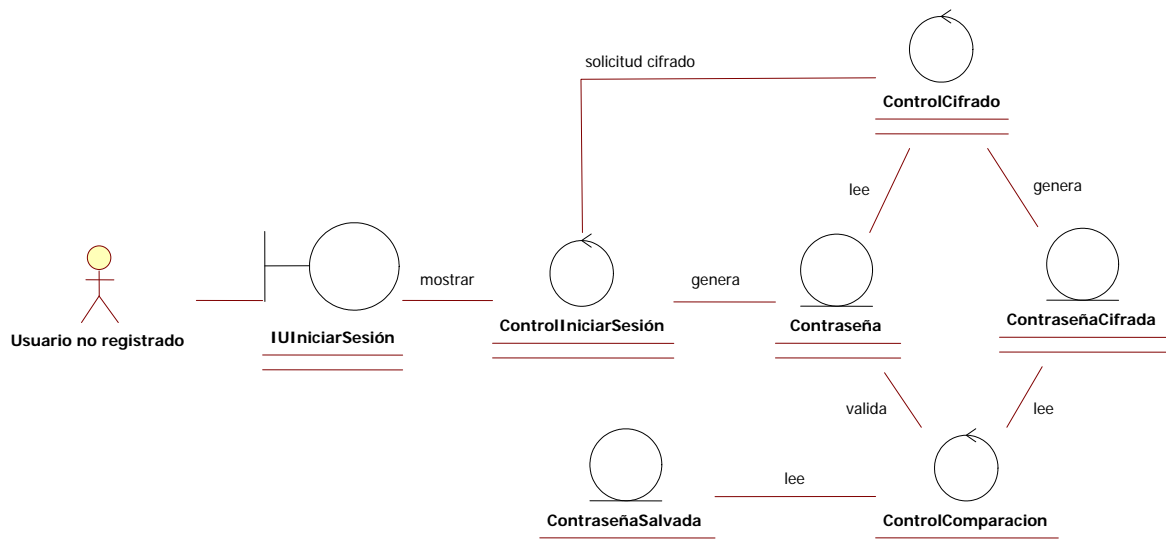
**Datos**

## Gestión de usuario:

Registrarse: en el caso de existir el usuario, ControlCuenta no validará el registro.



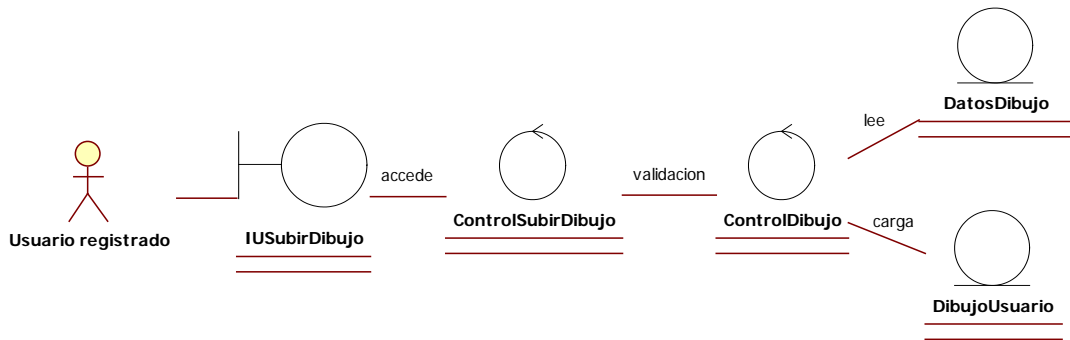
Iniciar sesión: A partir de la contraseña que introduce el usuario, ControlCifrado genera ContraseñaCifrada para posteriormente ser comparada con la que está registrada en la base de datos. Si son iguales el usuario iniciará sesión.



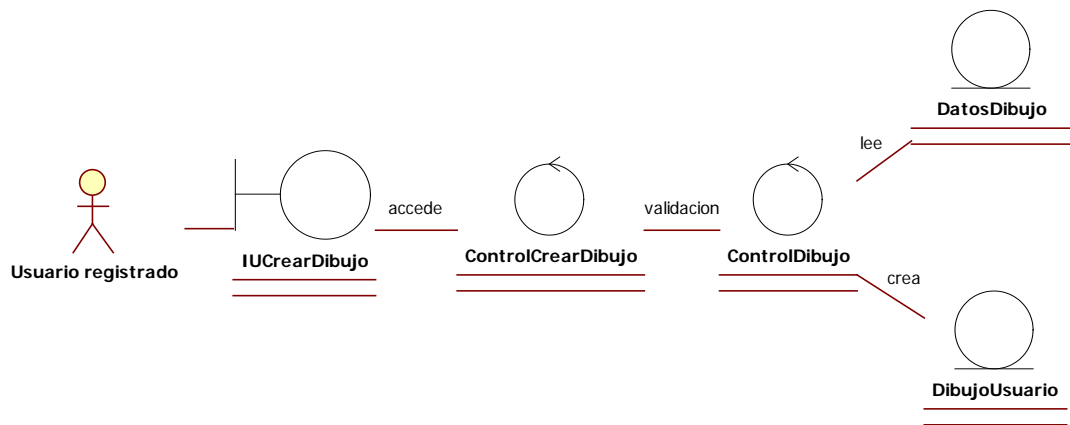


**Gestión de dibujo:** ControlDibujo hará una acción según el caso de

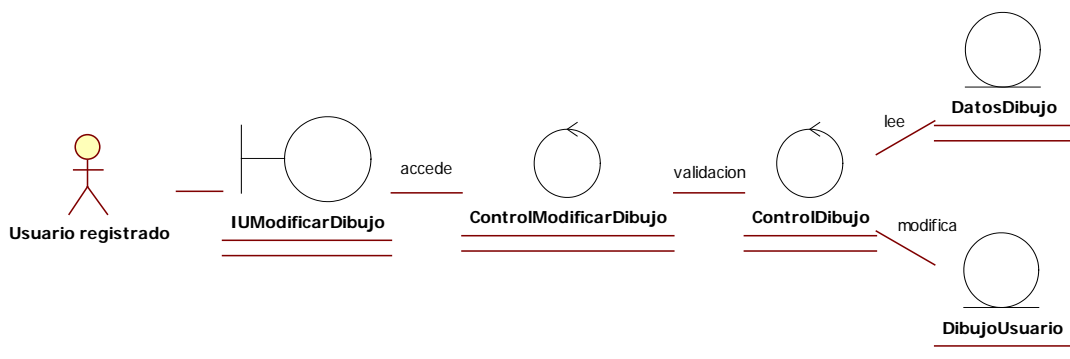
Subir dibujo



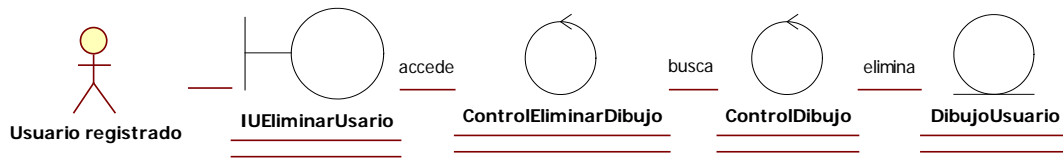
Crear dibujo



Modificar dibujo

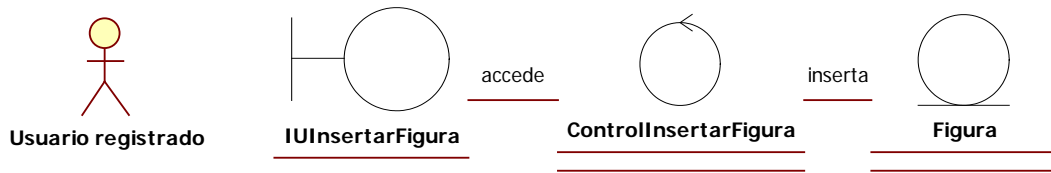


## Eliminar dibujo

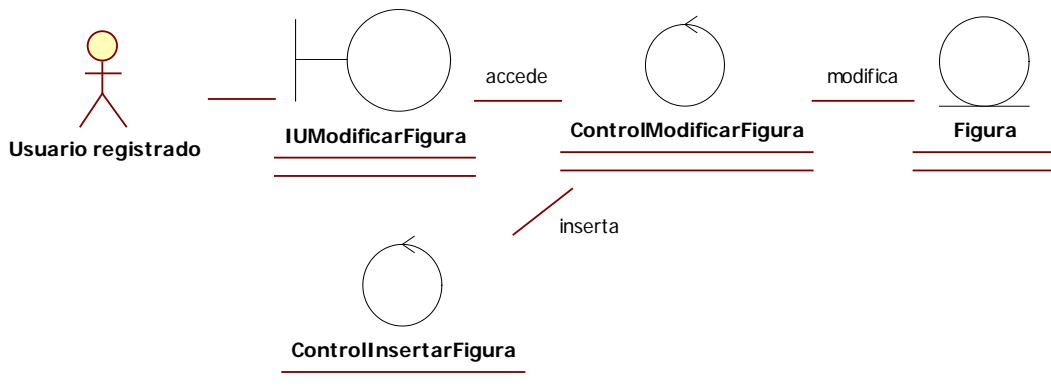


## Dibujo:

### Insertar figura



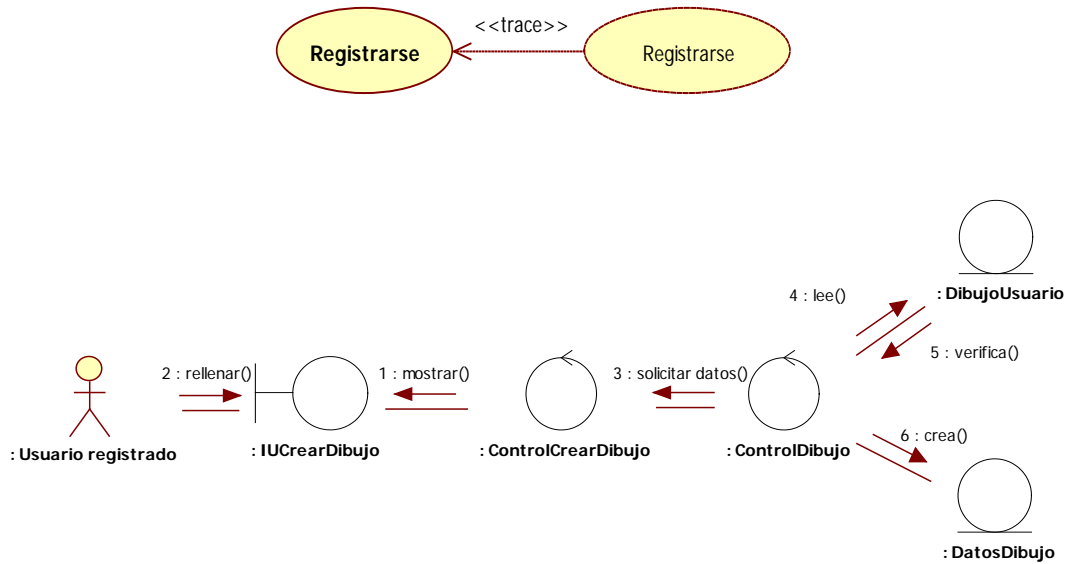
Modificar figura: llama a ControlInsertarFigura para insertar la figura modificada.



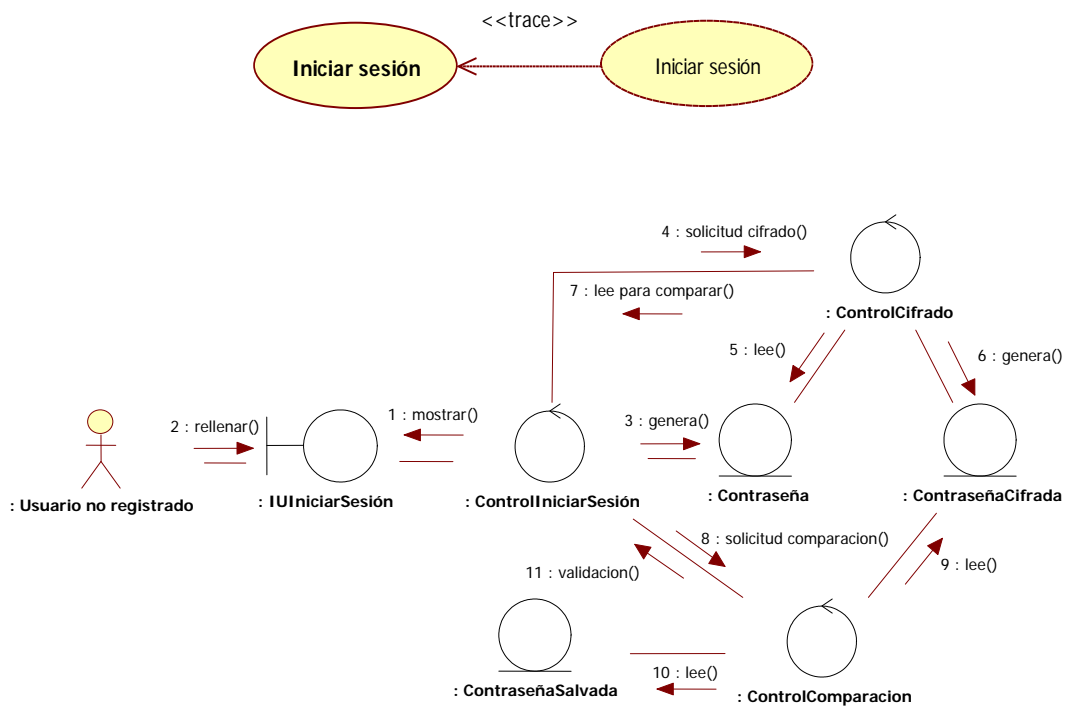
### 5.3.3 Diagramas de colaboración

#### Gestión de usuario:

#### Registrarse

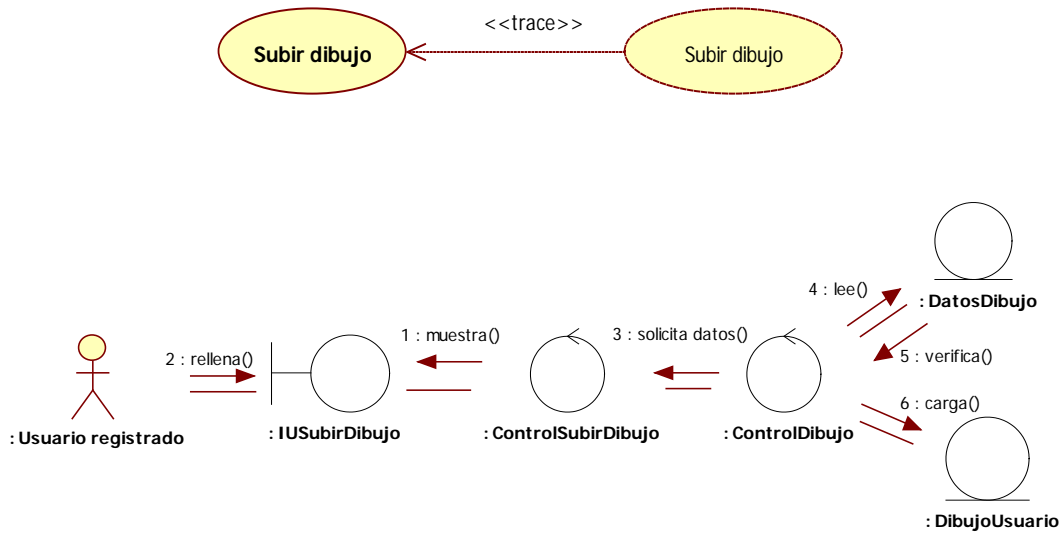


#### Iniciar sesión

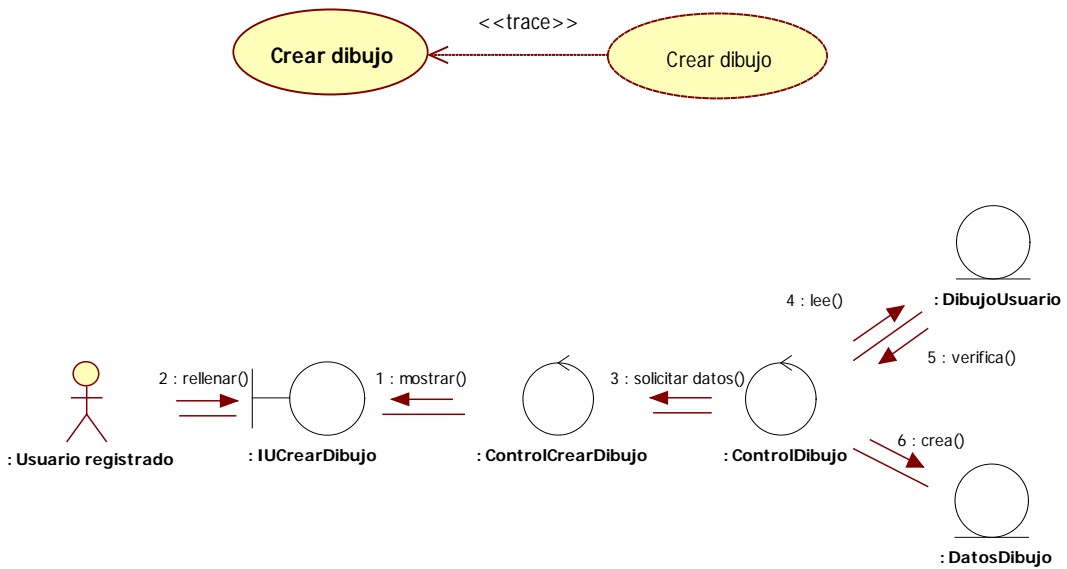


## Gestión de dibujo

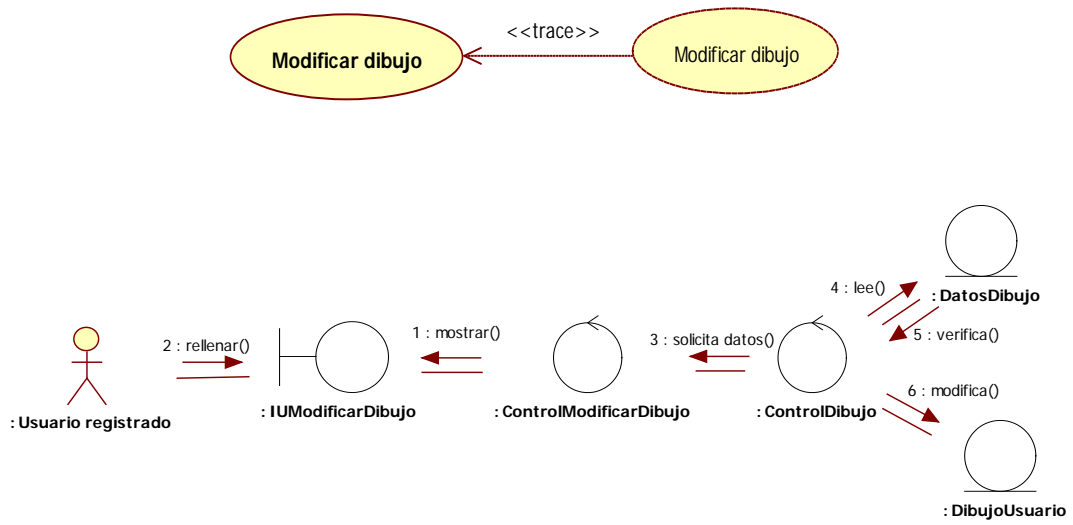
### Subir dibujo



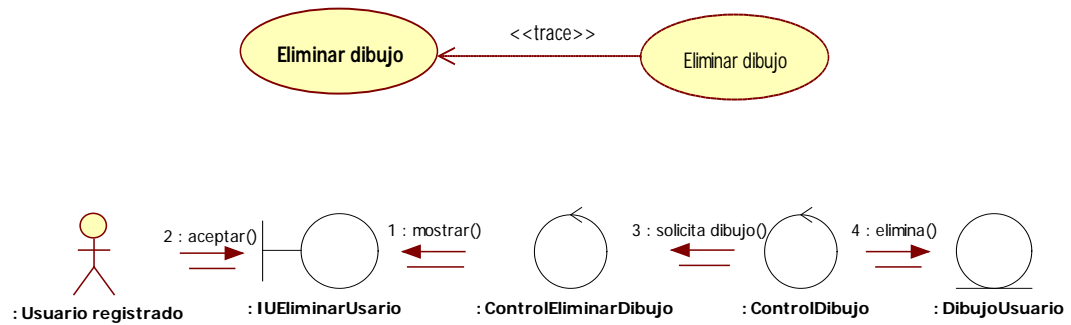
### Crear dibujo



## Modificar dibujo

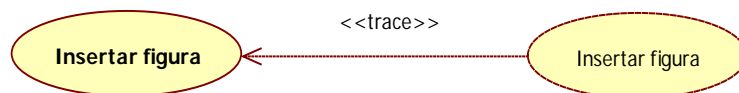


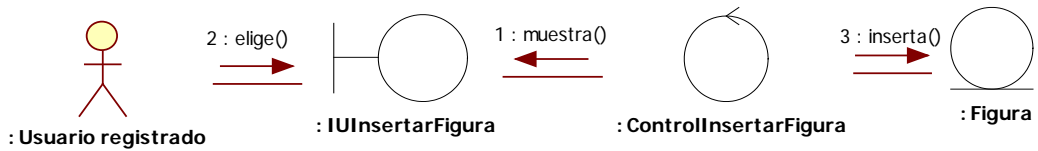
## Eliminar dibujo



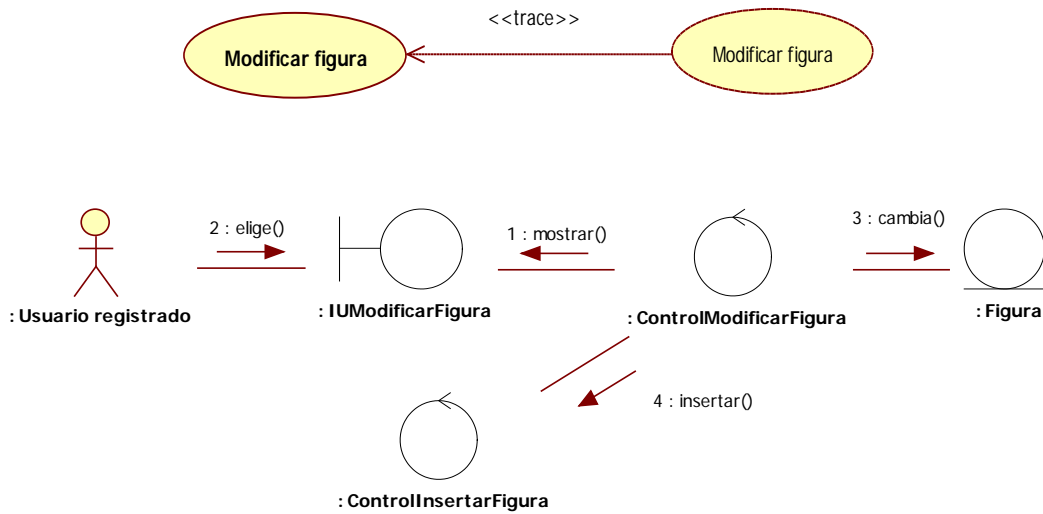
## Dibujo

### Insertar figura





### Modificar figura



## 5.4 Modelo de diseño

Durante el diseño modelamos el sistema y su arquitectura para que soporte los requisitos funcionales y no funcionales. En esta fase se realizará por un lado un diagrama detallado de las clases que existirán en el modelo de implementación. Y también se adjuntará un diagrama de secuencia que mostrará la evolución de las clases a lo largo del tiempo y la interacción entre ellas.

Pero antes de mostrar los diagramas de clase y secuencia, se verá un diagrama de despliegue siendo una visión general de la organización de los elementos de procesamiento que intervendrán en el sistema.

### 5.4.1 Arquitectura del sistema

Los nodos de los que consta el sistema son:

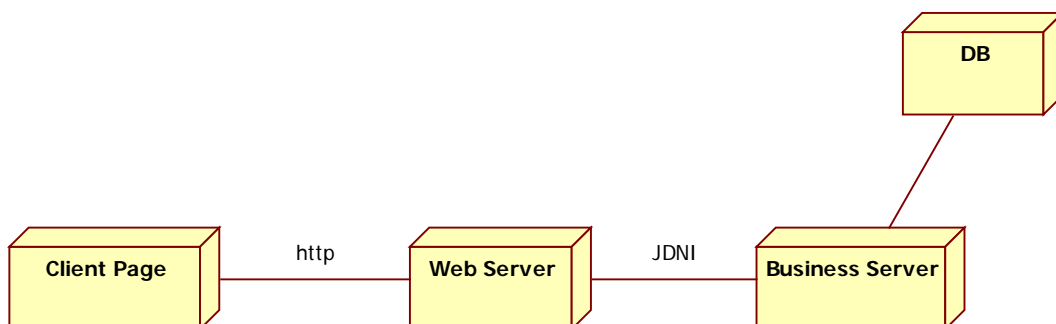
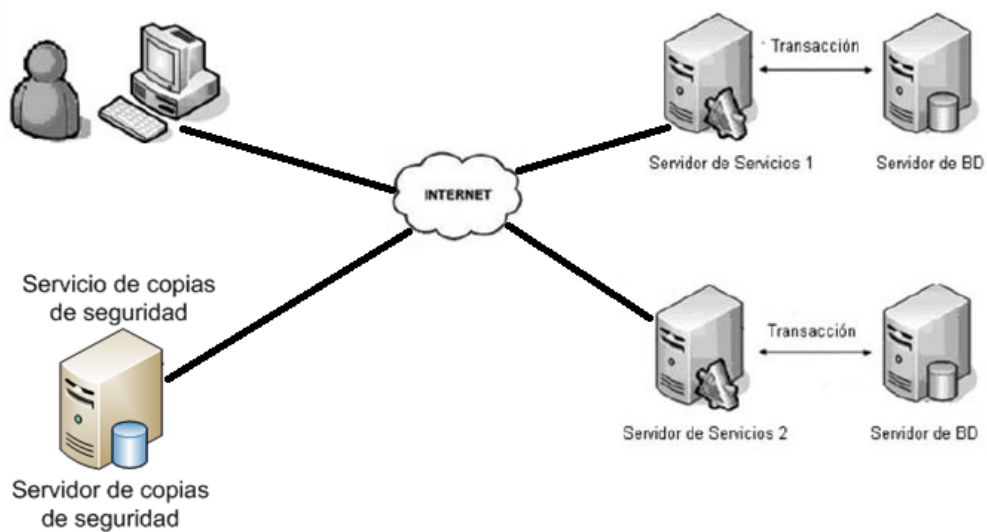
Usuarios del sistema: son lo que harán uso de la aplicación, mediante aportación de datos y realización de diferentes operaciones. HTTP será el protocolo de transacciones con el servidor.

Administradores del sistema: encargados en la gestión de los recursos de la aplicación y en administrar a los usuarios respondiendo dudas, atendiendo reclamaciones, etc.

Servidor Web: atiende a las peticiones del usuario y se comunica con el servidor de la base de datos.

Servidor de DB: encargado de almacenar los datos del sistema, atendiendo a un gran número de peticiones.

Servidor de copias de seguridad: realiza copias de seguridad de la base de datos para prevenir en caso de desastres en el servidor de DB.



## 5.4.2 Organización del modelo de diseño

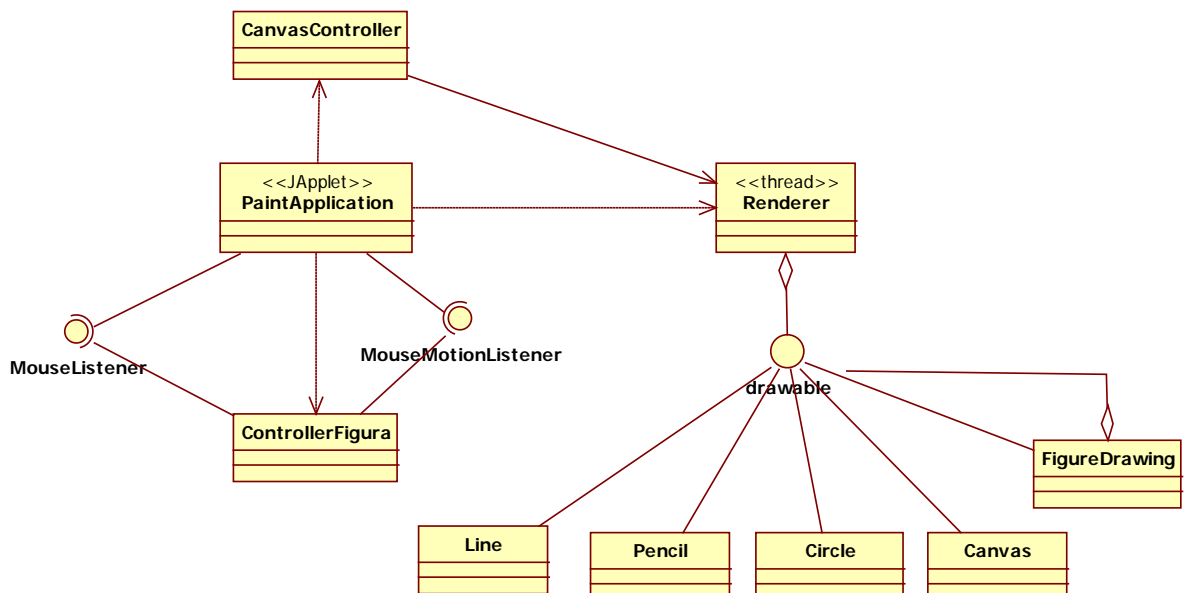
Se realizarán los diagramas de clase para representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación. A su vez están los diagramas de secuencia que muestra la interacción de un conjunto de objetos en una aplicación y se modela para cada caso de uso.

## 5.4.3 Diagramas de clases

Los diagramas de clases están divididos según las partes del proyecto:

### Cliente Applet / Entidades

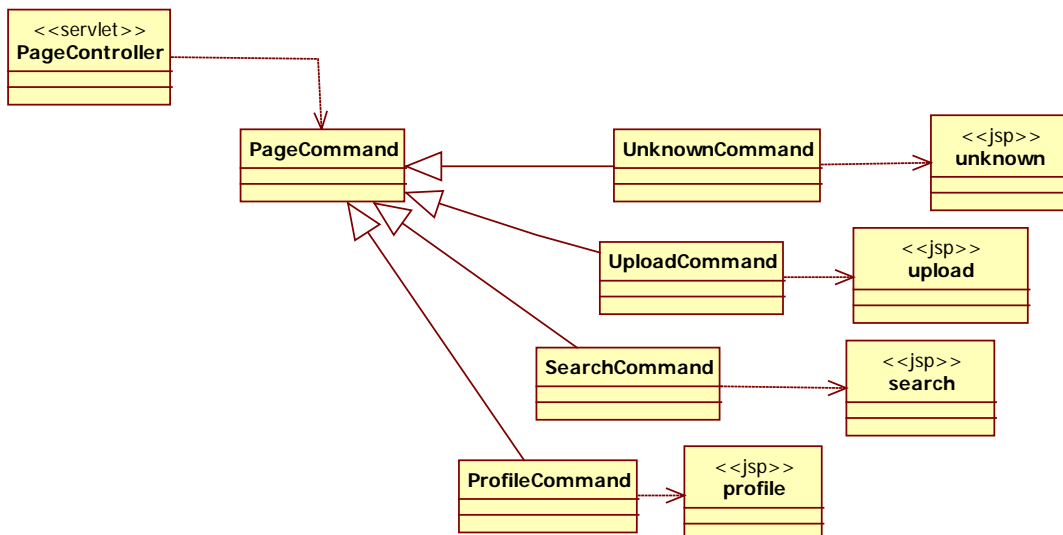
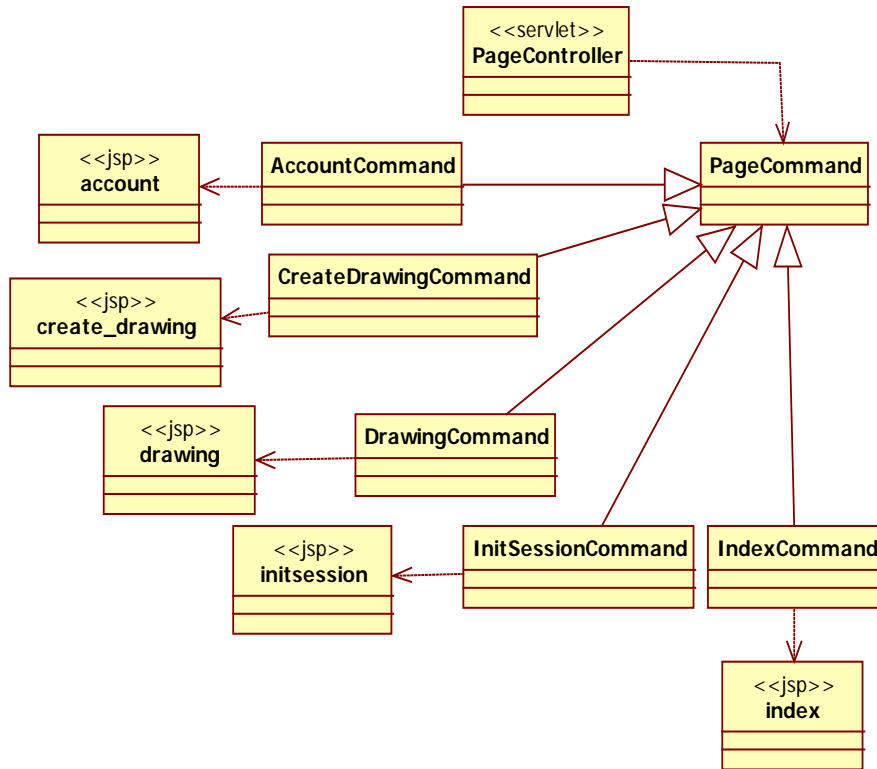
En este diagrama podemos ver las entidades que son los elementos para hacer dibujos, siguiendo el patrón de diseño Composite donde FigureDrawing engloba a dichas entidades. Por otra parte PaintApplication es el JApplet que llamará la página Web 2.0 para ejecutar la herramienta de dibujo y mediante el hilo Renderer irá mostrando en el panel de la herramienta de dibujo los elementos (figuras, trazos, etc.) que se van insertando en el dibujo.

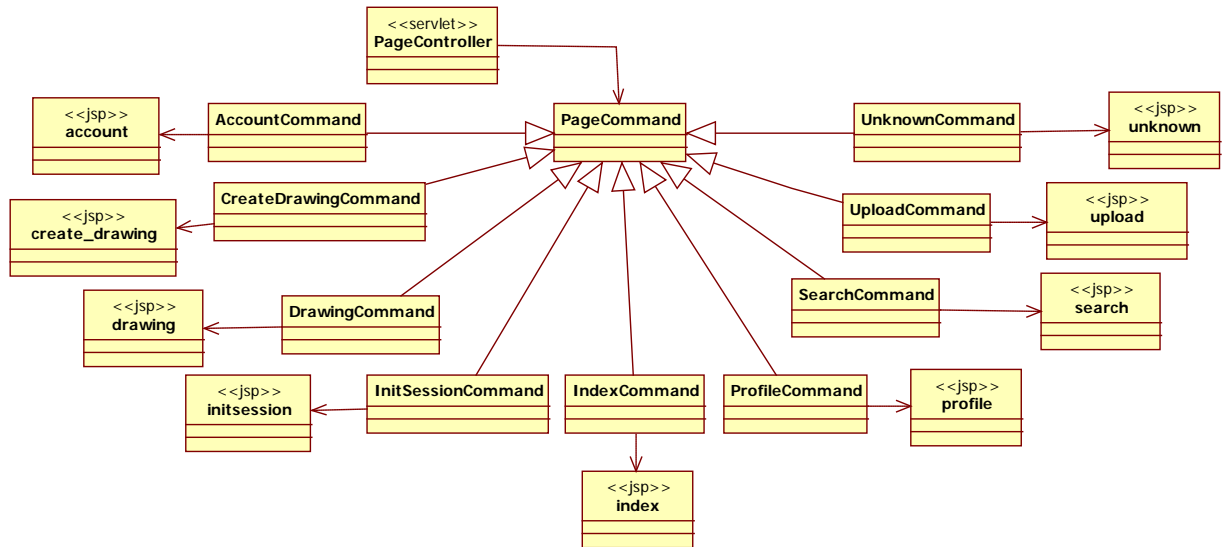




## Servidor

En la parte del servidor se optó por implementar el patrón de diseño FrontCommand para ir llamando a las diferentes páginas teniendo en cuenta el nombre de la clase para acceder a la vista.





Para apreciar mejor lo que hace cada clase, dividimos en diagramas más pequeños los servlet que acceden cada página JSP.

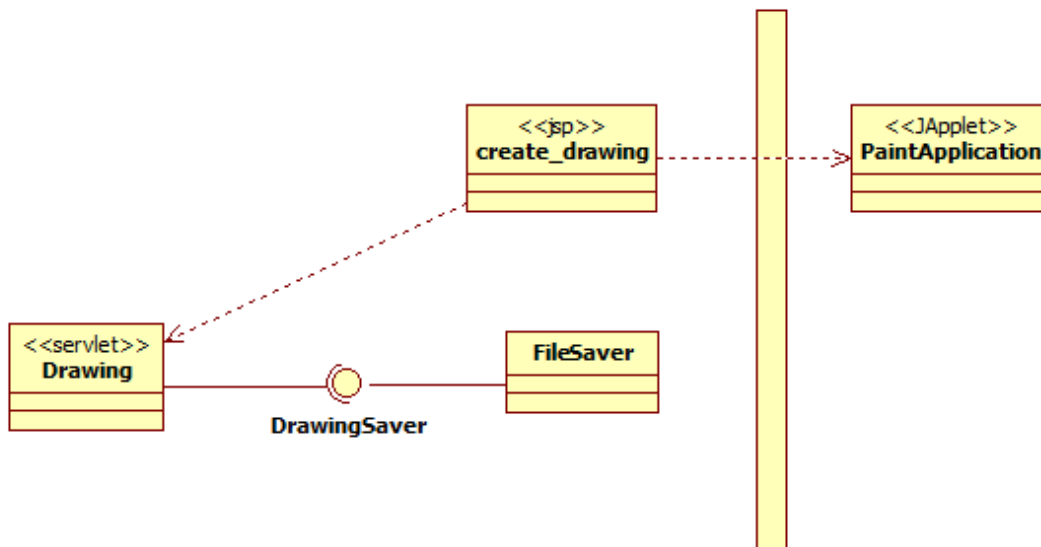
Iniciar sesión



Registrar usuario



Crear dibujo: la página create\_drawing invoca al PaintApplication para ejecutar el JApplet. Posteriormente cuando termine el dibujo accede al servlet para guardar el dibujo en el servidor.

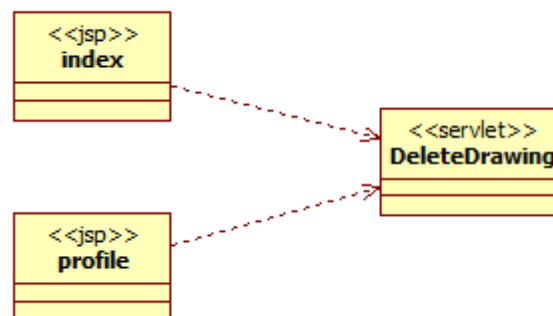


```
<applet code='ClientePaint/PaintApplication.class' archive='ProyectoPaintCliente.jar' height='498' width='720'></applet>
```

Subir dibujo: accede al servlet para posteriormente guardar los datos del dibujo a la base de datos



Borrar dibujo: se puede acceder desde el index (siempre que el usuario haya iniciado sesión) y también desde el perfil de usuario.



Añadir comentario



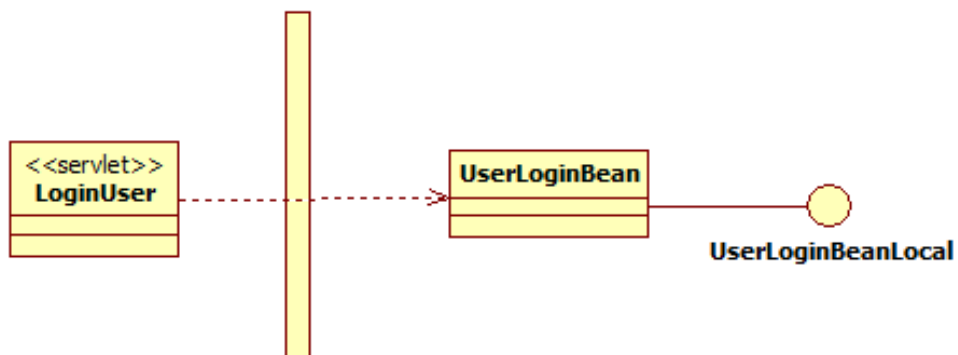
Añadir voto



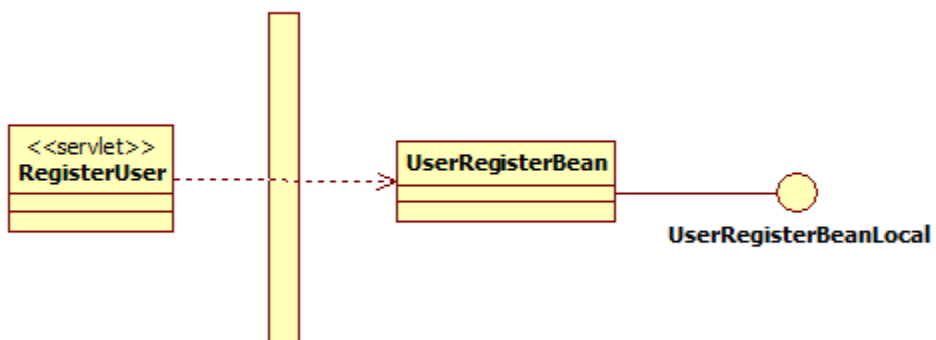
### Negocio

En la Capa de Negocio donde se utilizan los EJB podemos ver en cada diagrama la comunicación de cada servlet con el Bean. En se ha puesto ningún estereotipo a ninguna clase Bean porque todos son stateless.

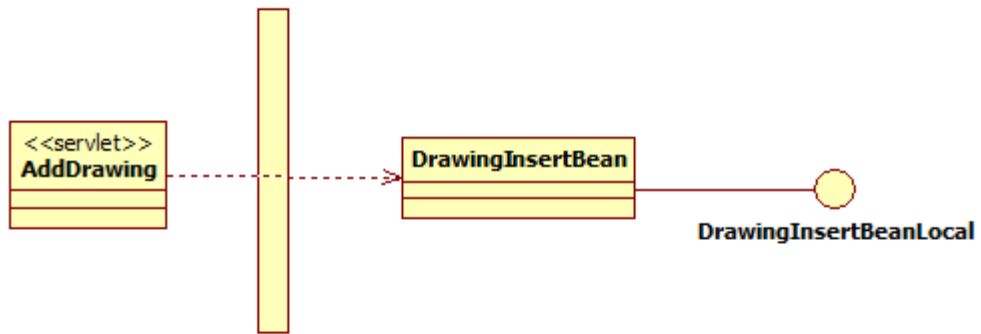
Iniciar sesión



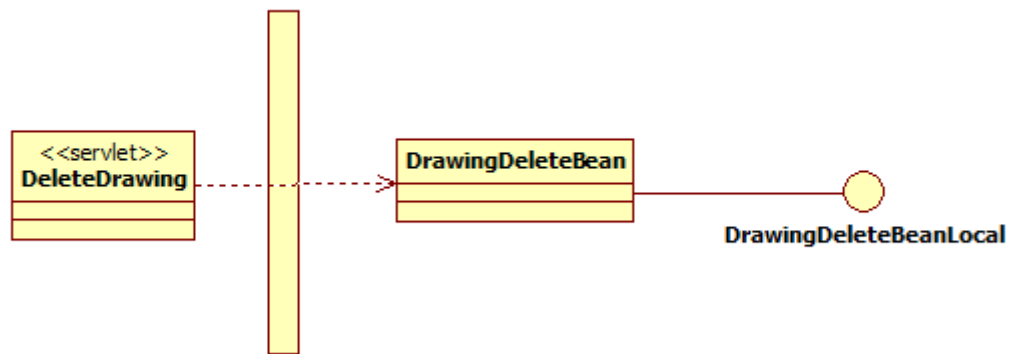
Registrar usuario



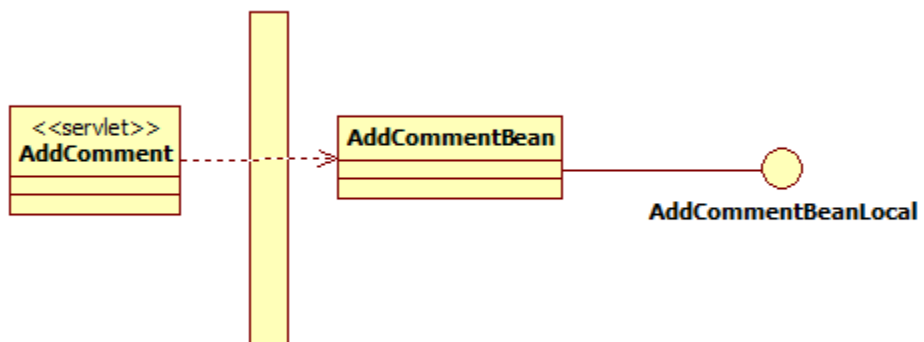
Insertar datos del dibujo



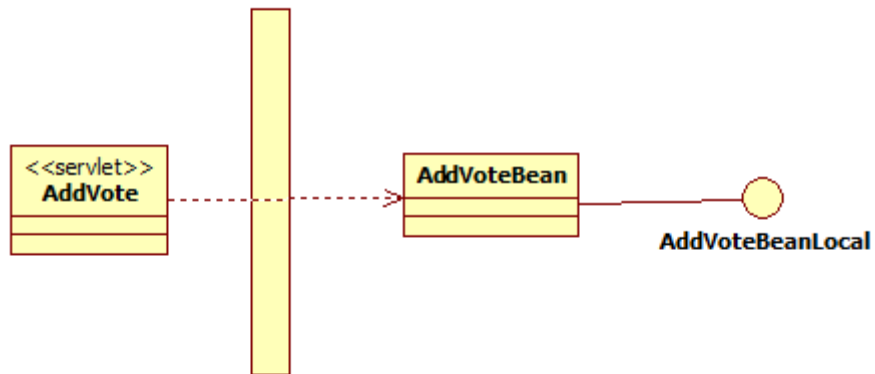
Borrar dibujo



Añadir comentario



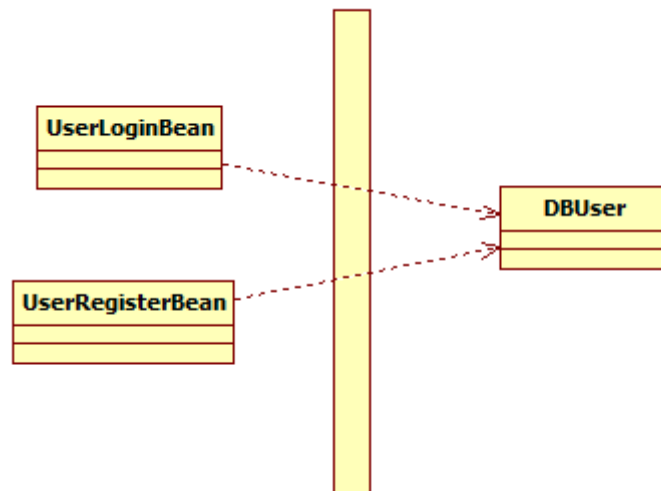
Añadir voto



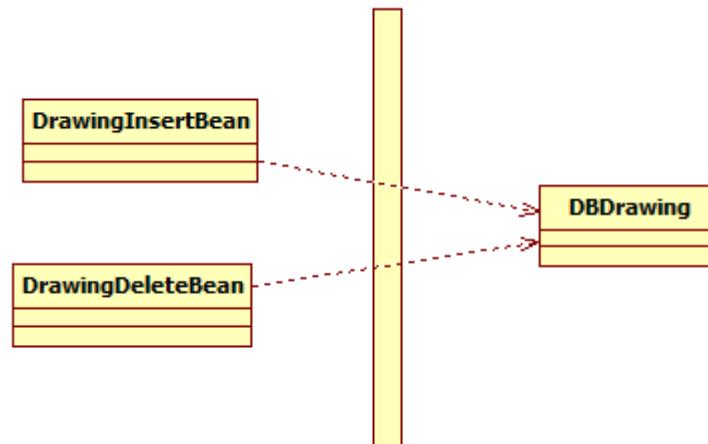
### Datos

Por último en la Capa de Datos se representa la comunicación de los Beans con cada clase que gestiona la base de datos mediante sentencias mysql.

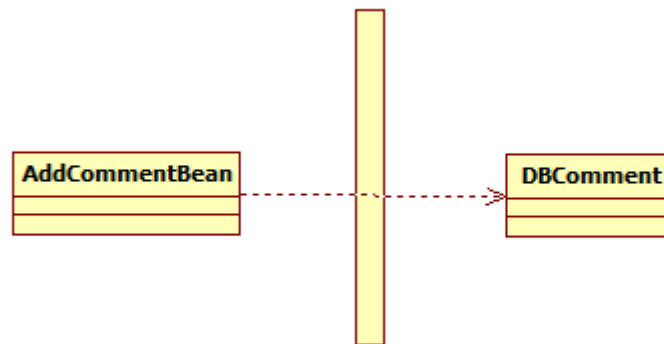
Iniciar y registrar usuario están comunicados con la clase DBUser encargada de gestionar los usuarios.



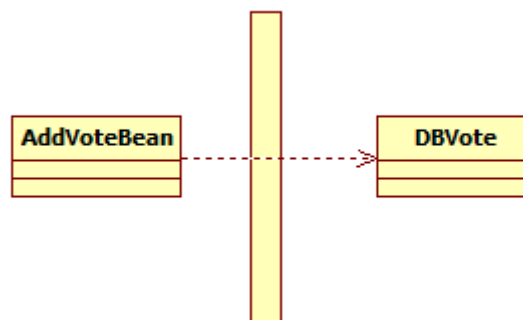
Añadir y borrar datos del dibujo se comunican con DBDrawing para gestionar los dibujos.



Añadir comentarios se comunica con DBComment para gestionar los comentarios del dibujo.



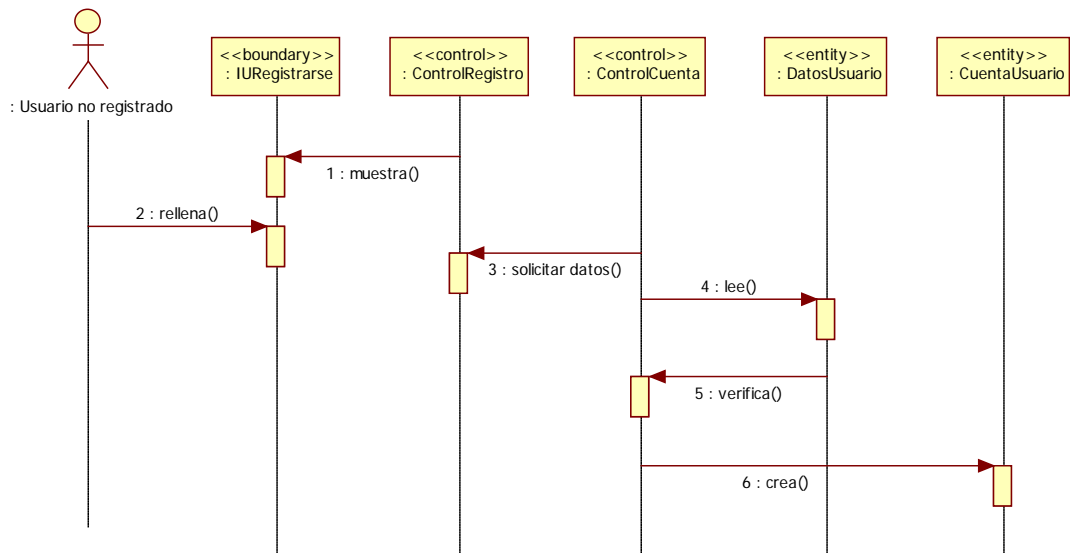
Añadir voto se comunica con DBVote para gestionar los votos del dibujo.



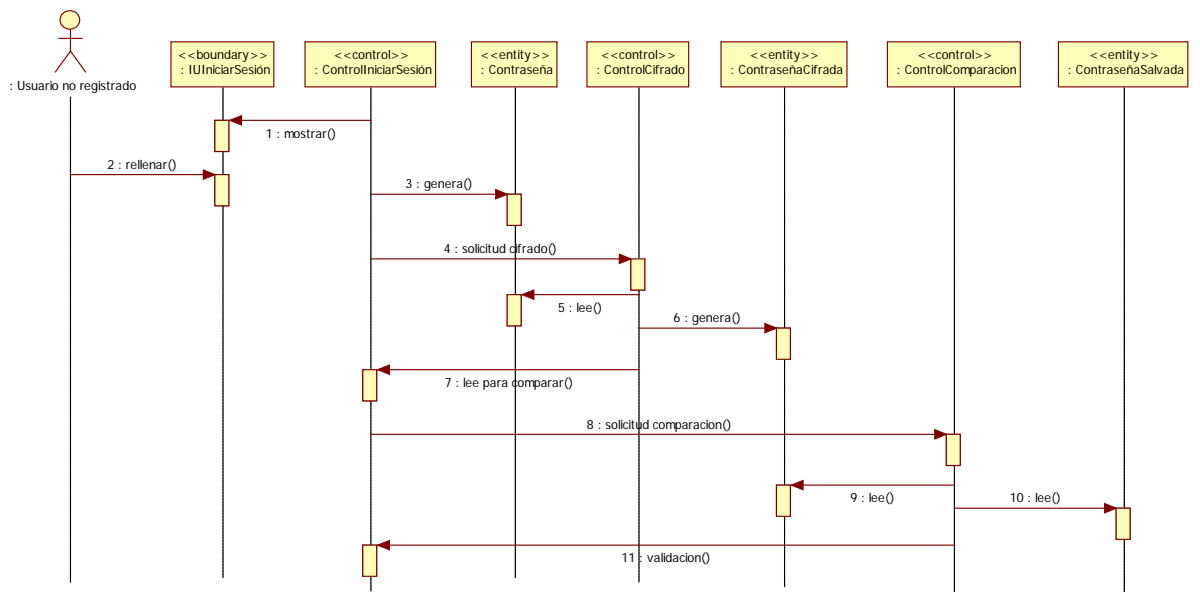
## 5.4.4 Diagramas de secuencia

### Gestión de usuario:

#### Registrarse



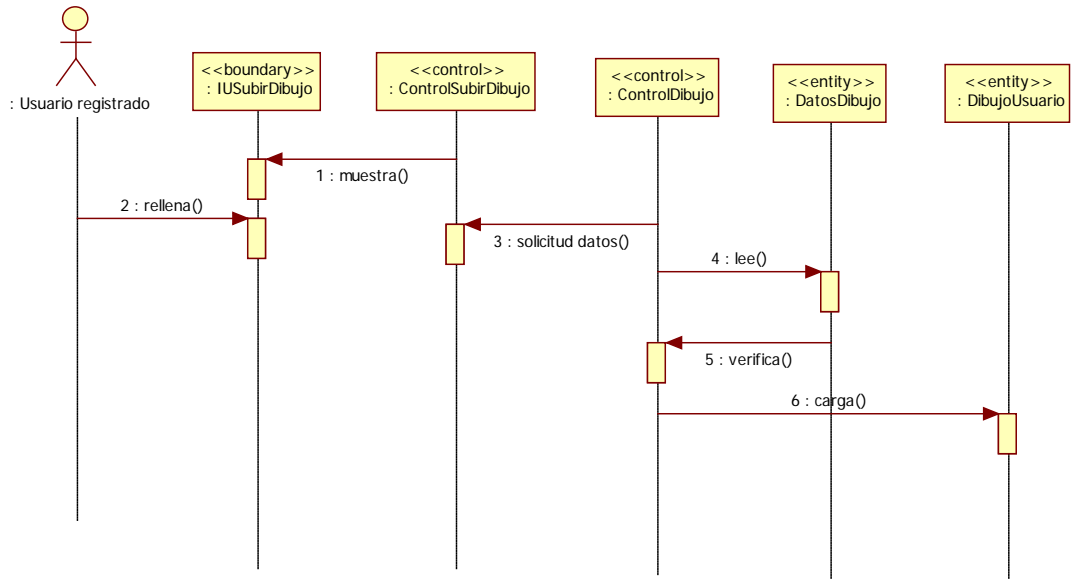
#### Iniciar sesión



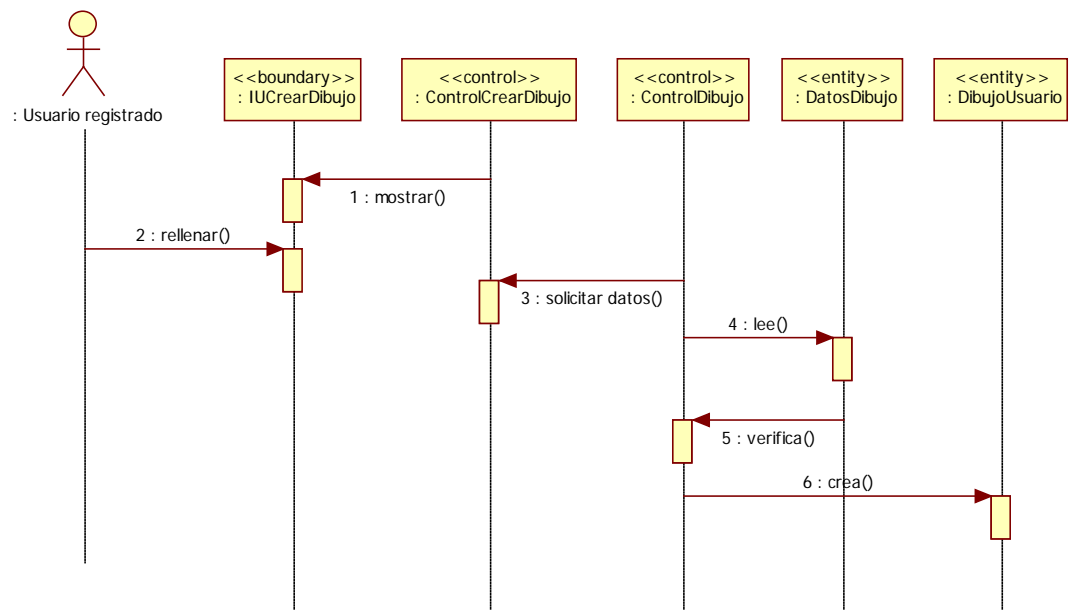


## Gestión de dibujo:

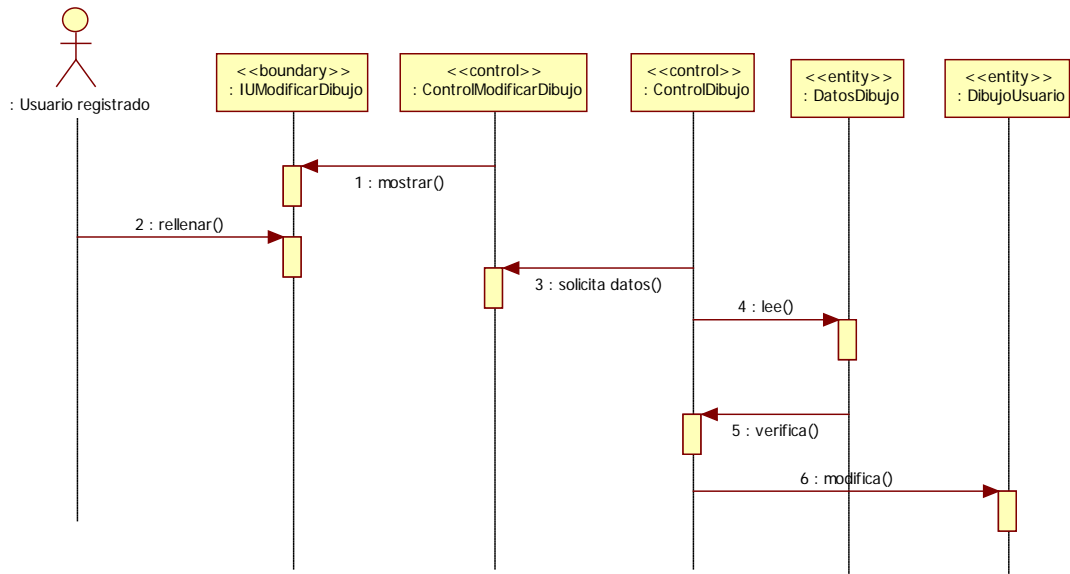
### Subir dibujo



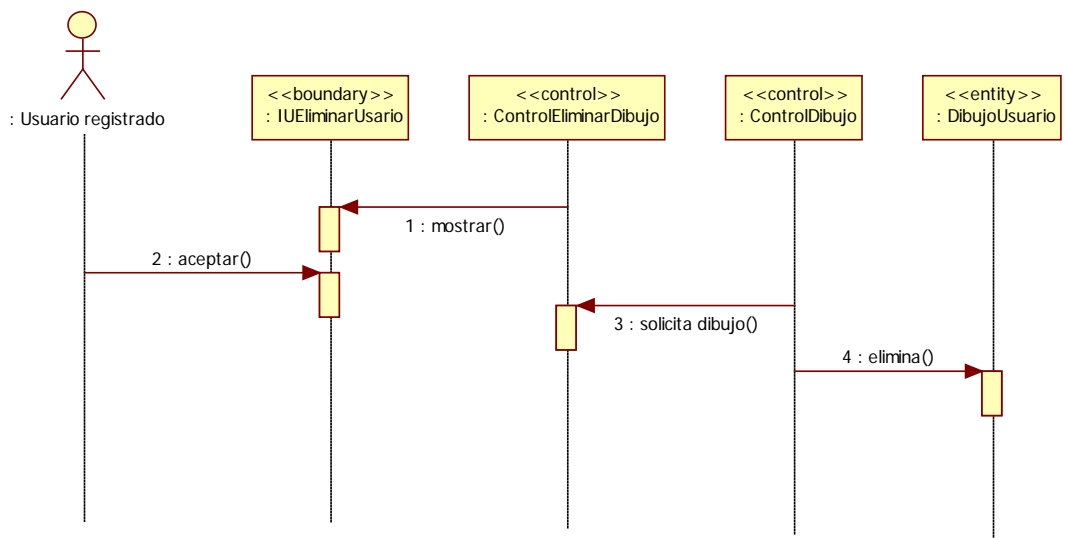
### Crear dibujo



## Modificar dibujo

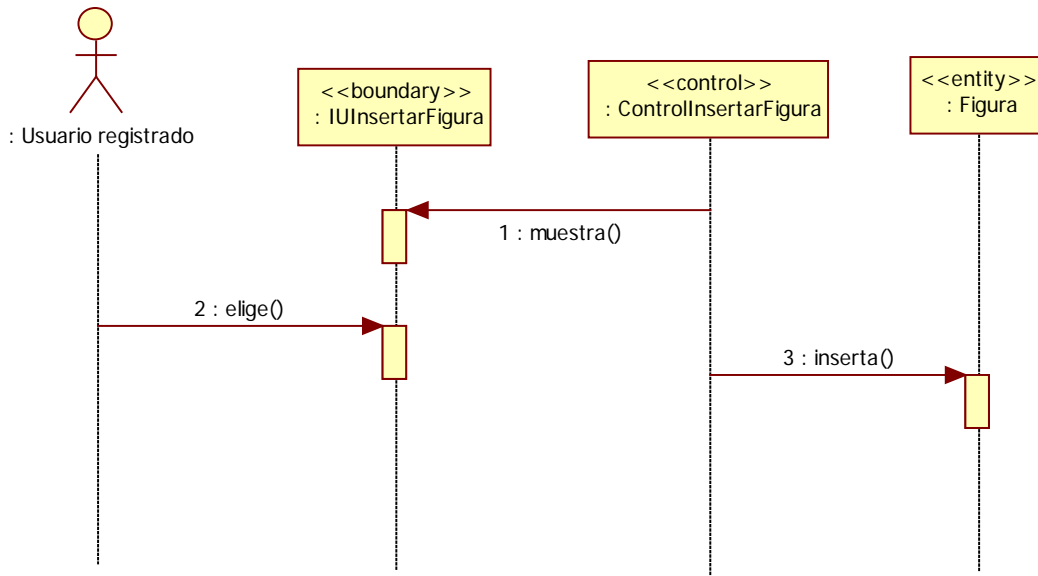


## Eliminar dibujo

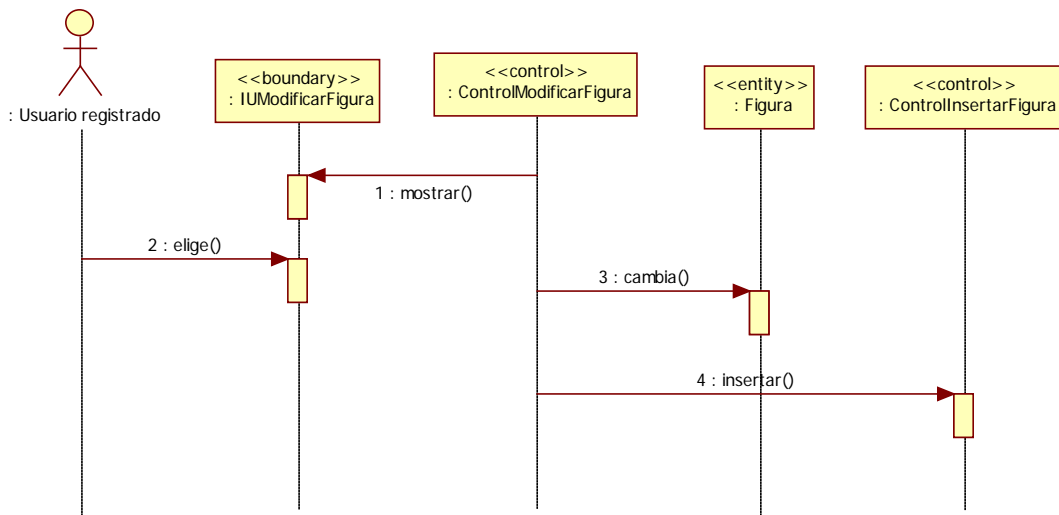


**Dibujo:**

**Insertar figura**



**Modificar figura**



## 5.5 Implementación

Para el desarrollo de la aplicación se han tenido en cuenta casi todos los aspectos fundamentales que debe cumplir una Web 2.0 que soporta una base de datos y gestiona varias operaciones sobre ella. En esta sección se indica las decisiones y problemas que hayan podido afectar a la aplicación.

Uno de los principales problemas que hubo que resolver fue el parpadeo que suele ocurrir en los Applet. Para ello se utilizó el proceso doble-buffering que consiste en dos pasos:

- Primero el usuario ve aparecer de golpe la imagen en la pantalla y mientras el usuario está viendo esa imagen, el programa está generando la siguiente para mostrarla de golpe a continuación, y así una y otra vez.
- Segundo, la técnica de doble-buffering involucra un objeto Image, que se puede pasar directamente a varios métodos. Esta capacidad para manipular objetos Image permite descomponer las rutinas de dibujo en componentes funcionales, en lugar de un enorme método paint().

@Override

```
public void paint(Graphics g) {  
  
    this.jPanel1.repaint();  
  
    this.jPanel1.setBackground(Color.decode("#f1f1f1"));  
  
    this.jPanel2.getGraphics().drawImage(offscreen, 0, 0, this.jPanel2);  
  
    this.jPanel2.setBackground(Color.decode("#f1f1f1"));  
  
}
```

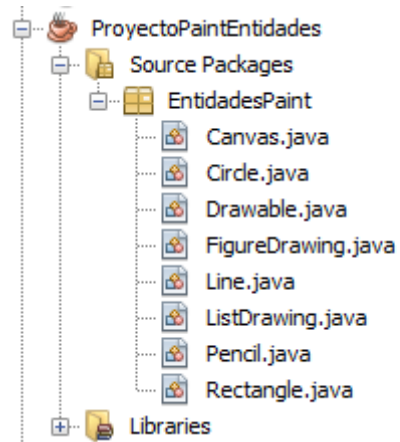
En el caso de la aplicación de este TFG, no ha supuesto sobrecargar la función paint del Applet ya que sólo son necesarios dos paneles (jPanel1 y jPanel2).

El doble-buffering sólo debe usarse para animaciones gráficas, no como método normal.

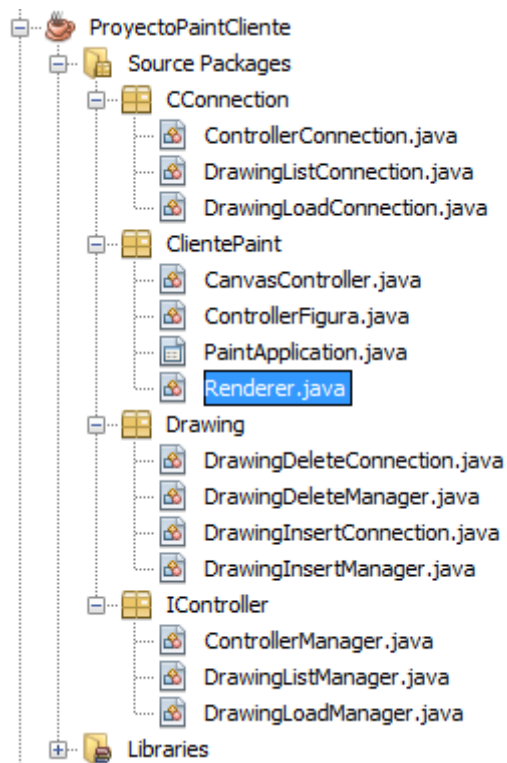
También hay que considerar JavaScript y CSS que han sido necesarios debido a que varias vistas requieren funcionalidad de JavaScript, como por ejemplo mostrar un pop-up de aviso por si se quiere eliminar un dibujo; y luego para las opciones de estilo ha sido necesario varias hojas CSS con el fin de tener separado el estilo de un botón en concreto, estando en hoja CSS el estilo de la plantilla de la página web y con esto se consigue no mezclar tantos estilos que dificulten después alguna modificación.

A nivel de arquitectura el proyecto se dividió en cinco partes para su implementación:

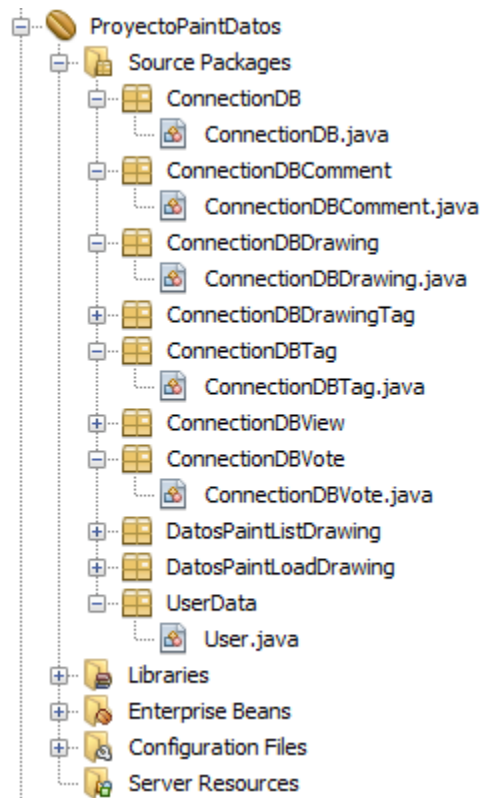
Entidades



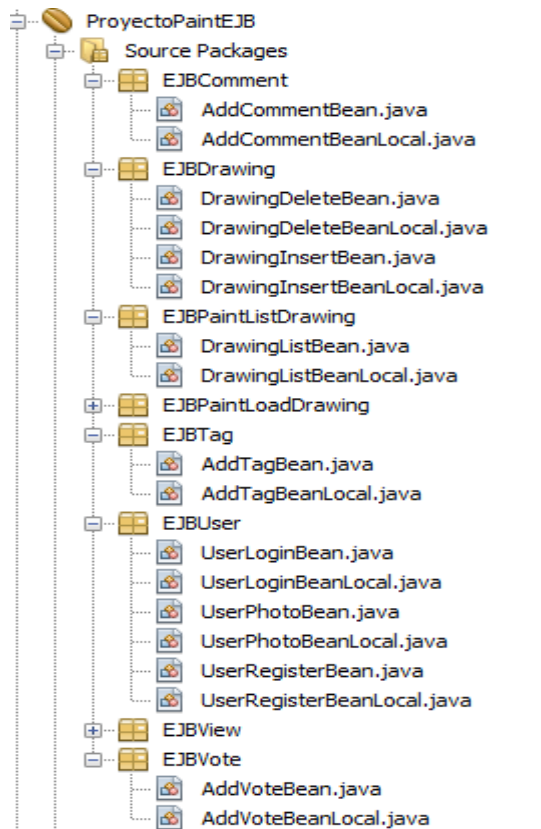
Applet que tiene como librería las Entidades



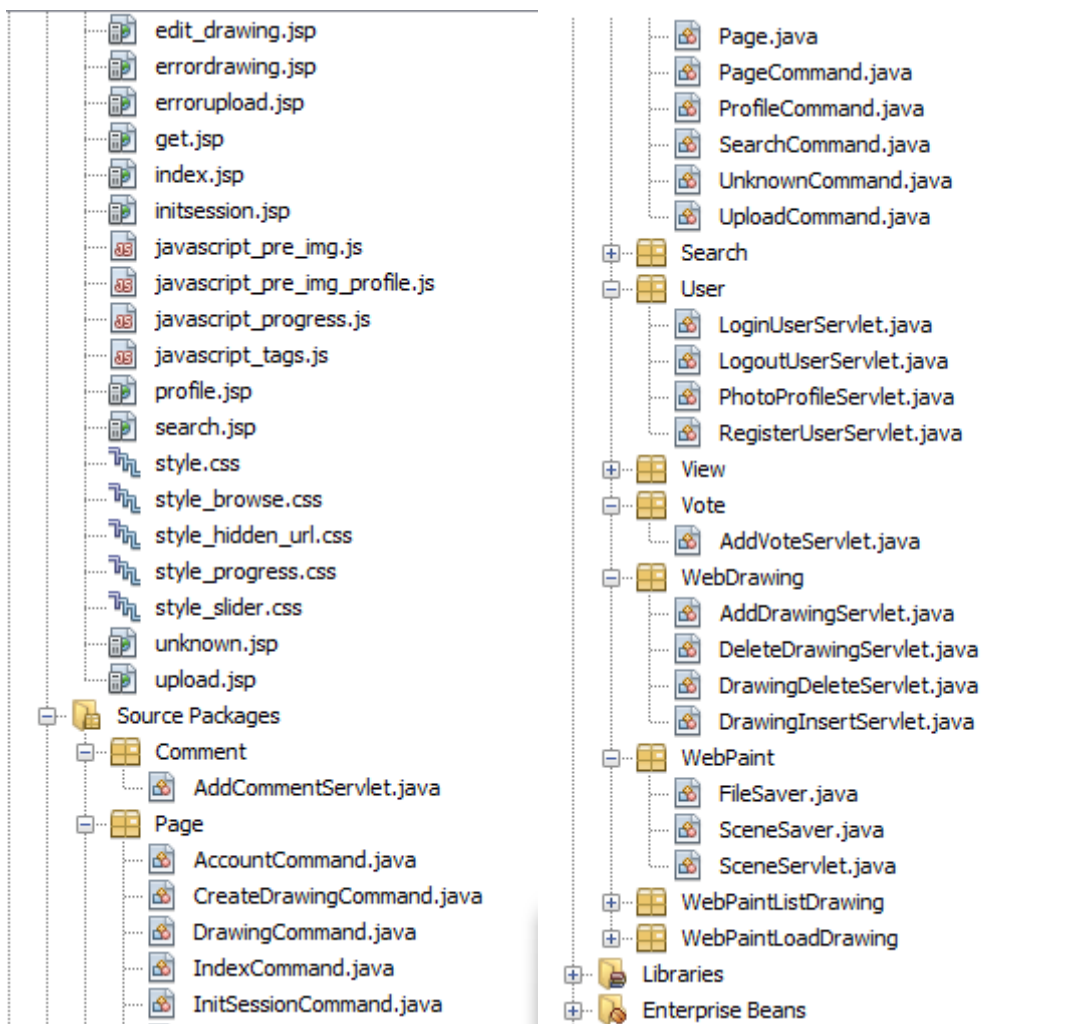
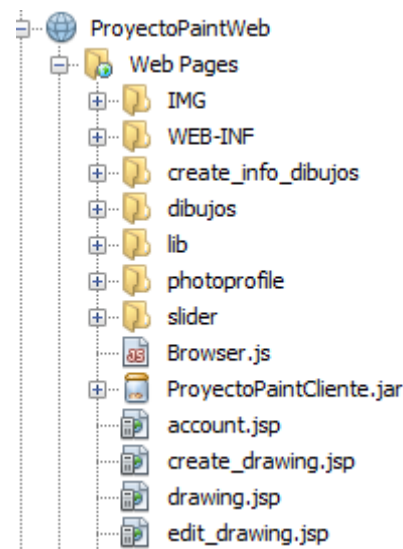
Datos tiene como API el JDBC



Negocio – EJB incluye como librería Datos



Servidor web engloba el Applet y el EJB para el despliegue de la página web.



La herramienta para crear dibujos ha sido desarrollada con un Applet.

Un Applet es una aplicación muy parecida a un Frame que va incrustada en el código de una página web. Sirve para usar funciones que no son accesibles desde la programación misma de la página, o para suministrar datos a una interfaz Javascript. Esta tecnología puede ser usada en cualquier sistema operativo que tenga instalada una JVM (Maquina Virtual de Java); la que por defecto va añadida en los navegadores web.

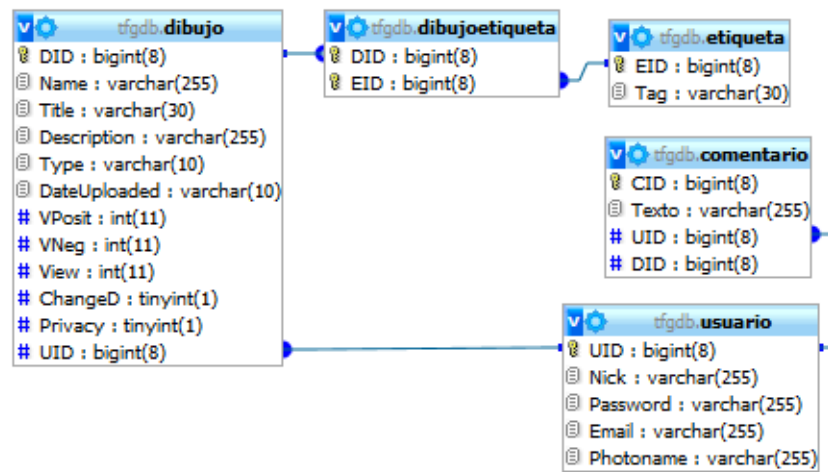
Cuando se crea el applet.java y compilarlo a applet.class, se tendrán que poner en un servidor de páginas (web server) para que sea el web server quien mande la página, el applet y todas sus clases, al usuario o la maquina remota, para que ésta última sea quien lo compile y ejecute. Se debe tener actualizado la JVM del navegador para evitar problemas.

Incluyendo un Applet en una página html:

```
<script src="https://www.java.com/js/deployJava.js" type="text/javascript">
</script><script type="text/javascript">
//
  var attributes = { archive: 'http://docs.oracle.com/javase/tutorial/JWS/samples/uiswing/TumbleItemProject/TumbleItem.jar',
    codebase: 'http://docs.oracle.com/javase/tutorial/JWS/samples/uiswing/TumbleItemProject',
    code: 'components.TumbleItem', width: '600', height: '95' };
  var parameters = { permissions: 'sandbox', nimgs: '17', offset: '-57',
    img: 'images/tumble', maxwidth: '120' };
  deployJava.runApplet(attributes, parameters, '1.7');
//]]&gt;
&lt;/script&gt;&lt;noscript&gt;A browser with Javascript enabled is required for this page to operate properly.&lt;/noscript&gt;</pre></div><div data-bbox="138 482 862 610" data-label="Text"><p>Las ventajas que tienen los Applet es que no tienen accesos a sitios comprometidos, por ejemplo no pueden escribir ficheros a menos que el usuario le otorgue permisos y proporciona una forma fácil de ejecutar aplicaciones desde el navegador, compatibles desde cualquier navegador, funcionan en cualquier SO, pero por el contrario no se pueden iniciar hasta que la JVM esté en funcionamiento, sino está firmado como confiable tiene acceso limitado al usuario y algunas organizaciones sólo permite la instalación de software a los administradores (la universidad es un ejemplo) por lo que los usuarios sin privilegios y necesiten instalar el plugin no podrán ver el Applet.</p></div><div data-bbox="138 625 862 788" data-label="Text"><p>Aun así los Applet pueden mostrar secuencias de imágenes y agregar efectos visuales, mostrar imágenes con sonidos y agregar efectos sonoros, presentación de gráficos interactivos reaccionando a acciones que se toman con el ratón sobre el gráfico, animaciones de textos, crear diagramas y gráficas, juegos sencillos, etc. Y al ser en lenguaje java da mucha facilidad a la arquitectura del software si bien puedes tener el Applet en la que se disponen los controles, luego una clase que describe el canvas que nos permite dibujar figuras sobre su área de trabajo, y por último otra clase que describe las distintas figuras o crear una jerarquía de clases que descendan de una clase abstracta y definir en las clases derivadas la función que la dibuja en un contexto gráfico.</p></div><div data-bbox="484 921 511 939" data-label="Page-Footer"><p>63</p></div>
```



Finalmente la tabla de relaciones de la base de datos:



Donde el usuario sube o crea uno o varios dibujos, publica comentarios o agrega etiquetas a los dibujos.

## 6 Conclusiones y trabajo futuro

Una vez terminada la fase de desarrollo y pruebas, se exponen las siguientes conclusiones, así como las posibles ampliaciones en la aplicación creada.

### Conclusiones

Destacar principalmente que se ha logrado cumplir los objetivos principales del proyecto marcados al inicio. Se han elegido las herramientas de software adecuadas que han permitido el desarrollo del proyecto cumpliendo con los requisitos planteados.

Con JavaScript se complica un poco a la hora de detectar errores, pero se ha facilitado en gran medida la programación de la aplicación en algunos casos puntuales como subir los dibujos u obtener la URL de un dibujo para poder ser compartida.

Por otro lado el Applet interactivo que permite dibujar a los usuarios para ser compartidos. Proporcionan funcionalidad a las páginas, permite la presentación de gráficos interactivos reaccionando a acciones que se toman con el mouse sobre el gráfico, añade animaciones de textos, crear gráficas, etc.

A lo largo del desarrollo de este proyecto se han utilizado diversas técnicas y herramientas propias de diferentes ramas de la titulación de Ingeniería en Informática, principalmente de asignaturas como Metodología de la Programación, Bases de Datos, Ingeniería del Software y Multimedia.

Con la realización de este proyecto, el empleo de una amplia variedad de tecnologías ha sido muy enriquecedor, permitiéndome obtener un valioso nivel de conocimiento de cada una de ellas y de cómo ser capaz de operar con todas ellas de manera conjunta en pos de lograr un producto final con la funcionalidad deseada.

### Trabajo futuro

Tras dar por concluida la fase de desarrollo se puede exponer varias ideas y posibles opciones para incrementar la utilidad de la misma.

1. Creación de dibujos con la participación de más de un usuario de forma simultánea.
2. Desarrollar una versión de la aplicación para dispositivos móviles.
3. Permitir comunicación entre usuarios por medio de mensajes privados o incluso un chat.
4. Añadir publicidad de patrocinadores.
5. Añadir capacidad al crear o editar dibujos.



## Anexo I: Competencias

En este apartado se mostrarán las distintas competencias que se pretenden cubrir en este proyecto:

### **CII01**

***Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación vigente.***

Esta competencia queda cubierta con los capítulos Estado actual y objetivo del Trabajo de Fin de Grado y Requisitos Hardware y Software. En ellos se especifican, por un lado, los motivos de diseño, desarrollo, etc. explicando exhaustivamente las fases de definición de requisitos, análisis y diseño de los casos de uso necesarios para satisfacer las necesidades de la aplicación Web, y por otro lado se justifica la utilización del hardware más adecuado para la implementación de la misma, teniendo en cuenta parámetros de fiabilidad, calidad y seguridad.

En resumen, con esta elección de hardware y software, la implementación de este trabajo de fin de grado da como resultado una aplicación robusta y conforme a los principios éticos y a la legislación vigente.

### **CII02**

***Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.***

De todos los elementos de un sistema de gestión de proyectos, la planificación es posiblemente la más importante, puesto que una mala o inexistente planificación conducirá a una mala realización en el proyecto, lo cual repercutirá enormemente tanto en la calidad como en el plazo de terminación del proyecto, ocasionando grandes perjuicios. Esta es la razón por la que esta competencia queda cubierta, puesto que la buena planificación de este trabajo de fin de grado ha dado lugar al propósito fundamental del mismo, que no es más que la finalización en los plazos establecidos así como conseguir de manera adecuada los objetivos propuestos inicialmente.

### **CII04**

***Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.***

Pliego de especificaciones técnicas que han de regir en la contratación de una plataforma compuesta por todo el desarrollo de la plataforma web y todo lo relacionado con su implementación.

- Objeto del contrato y condiciones: El contrato tiene como objeto, la implementación de la Aplicación Web 2.0 para la Gestión de Dibujos para uso de la ULPGC.
- Ámbito geográfico: El ámbito geográfico en que se prestará el servicio será mundial, ya que cualquier visitante externo puede acceder a la plataforma.
- Requisitos de fiabilidad: El desarrollo de la aplicación debe acometerse de tal forma que su resultado garantice alta fiabilidad y comportamiento estable.
- Plazo de ejecución: El contratista se compromete a elaborar el software de conformidad con lo establecido en este documento y con la oferta que hubiese presentado, así como a entregarlo al contratante. El software se considerará debidamente entregado cuando se haya recibido la plataforma y a su vez esté disponible para su inmediato funcionamiento y se haya facilitado el manual de usuario y la documentación técnica correspondiente.
- Documentación: Las empresas licitadoras deberán poner claramente de manifiesto las posibles actualizaciones de la aplicación.
- Transparencia Tecnológica: Durante la ejecución de los trabajos objeto del contrato, el adjudicatario se compromete a facilitar en todo momento a las personas designadas por la Universidad de las Palmas de Gran Canaria a tales efectos la información que ésta solicite para disponer de un pleno conocimiento de las circunstancias en que se desarrollan los trabajos, así como de los eventuales problemas que pueden plantearse y de las tecnologías, métodos y herramientas utilizados para resolverlos.
- Propiedad del resultado de los trabajos: Todos los documentos y resultados de los trabajos realizados serán propiedad de la Universidad de las Palmas de Gran Canaria que podrá ejercer el derecho de explotación en su centro de trabajo. La empresa adjudicataria podrá hacer uso de los mismos, ya sea como referencia o base para trabajos futuros, siempre que cuente con la autorización expresa del contratante.
- Entrega de la solución: Todo el proceso de suministro de software, documentación, manuales de instalación y configuración, manual de administración y de usuario, desarrollo de pruebas, plan de marcha atrás, se hará conforme al Proceso de Gestión de la Entrega definido por la Dirección de la Universidad de las Palmas de Gran Canaria.
- Entorno Pre-explotación: De ser necesarias tareas correctivas una vez implementada la solución objeto de este contrato, con el fin de facilitar los trabajos, y realización de pruebas sin afectar al sistema en producción, será preciso que cualquier actividad a

realizar sobre la plataforma se realice previamente en el entorno de pre-explotación similar al de producción. Será preciso verificar y testear el correcto funcionamiento en este entorno, como paso previo al despliegue en producción.

## **CII18**

### **Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.**

Un análisis de las legislaciones que se han promulgado en diversos países arroja que las normas jurídicas que se han puesto en vigor están dirigidas a proteger la utilización abusiva de la información reunida y procesada mediante el uso de computadoras, e incluso en algunas de ellas se ha previsto formar órganos especializados que protejan los derechos de los ciudadanos amenazados por los ordenadores.

Desde hace aproximadamente diez años la mayoría de los países europeos han hecho todo lo posible para incluir dentro de la ley, la conducta punible penalmente, como el acceso ilegal a sistemas de cómputo o el mantenimiento ilegal de tales accesos, la difusión de virus o la interceptación de mensajes informáticos.

En la mayoría de las naciones occidentales existen normas similares a los países europeos. Todos estos enfoques están inspirados por la misma de preocupación de contar con comunicaciones electrónicas, transacciones e intercambios tan confiables y seguros como sea posible.

Dar un concepto sobre delitos informáticos no es una labor fácil y esto en razón de que su misma denominación alude a una situación muy especial, ya que para hablar de "delitos" en el sentido de acciones tipificadas o contempladas en textos jurídico penales, se requiere que la expresión "delitos informáticos" este consignada en los códigos penales, lo cual en nuestro país, al igual que en muchos otros, no ha sido objeto de tipificación aún; sin embargo, muchos especialistas en derecho informático emplean esta alusión a los efectos de una mejor conceptualización.

- Normativa y regulación de la informática en el ámbito internacional: En el contexto internacional, son pocos los países que cuentan con una legislación apropiada. Entre ellos, destacan, Estados Unidos, Alemania, Austria, Gran Bretaña, Holanda, Francia, España, Argentina y Chile. Por esta razón a continuación se mencionan algunos aspectos relacionados con la ley en los diferentes países, así como con los delitos informáticos que persigue.
  - Alemania. Este país sancionó en 1986 la Ley contra la Criminalidad Económica, que contempla los siguientes delitos:
    - Espionaje de datos.

- Estafa informática.
  - Alteración de datos.
  - Sabotaje informático.
- Austria. La Ley de reforma del Código Penal, sancionada el 22 de Diciembre de 1987, sanciona a aquellos que con dolo causen un perjuicio patrimonial a un tercero influyendo en el resultado de una elaboración de datos automática a través de la confección del programa, por la introducción, cancelación o alteración de datos o por actuar sobre el curso del procesamiento de datos. Además contempla sanciones para quienes comenten este hecho utilizando su profesión de especialistas en sistemas.
  - Estados Unidos. Este país adoptó en 1994 el Acta Federal de Abuso Computacional que modificó al Acta de Fraude y Abuso Computacional de 1986 con la finalidad de eliminar los argumentos hipertécnicos acerca de qué es y que no es un virus, un gusano, un caballo de Troya y en que difieren de los virus, la nueva acta proscribe la transmisión de un programa, información, códigos o comandos que causan daños a la computadora, a los sistemas informáticos, a las redes, información, datos o programas. La nueva ley es un adelanto porque está directamente en contra de los actos de transmisión de virus. Asimismo, en materia de estafas electrónicas, defraudaciones y otros actos dolorosos relacionados con los dispositivos de acceso a sistemas informáticos, la legislación estadounidense sanciona con pena de prisión y multa, a la persona que defraude a otro mediante la utilización de una computadora o red informática. En el mes de Julio del año 2000, el Senado y la Cámara de Representantes de este país, tras un año largo de deliberaciones, establece el Acta de Firmas Electrónicas en el Comercio Global y Nacional. La ley sobre la firma digital responde a la necesidad de dar validez a documentos informáticos, mensajes electrónicos y contratos establecidos mediante Internet, entre empresas (para el B2B) y entre empresas y consumidores (para el B2C).
  - Chile. Chile fue el primer país latinoamericano en sancionar una Ley contra delitos informáticos, la cual entró en vigencia el 7 de junio de 1993. Esta ley se refiere a los siguientes delitos:
    - La destrucción o inutilización de los de los datos contenidos dentro de una computadora es castigada con penas de prisión. Asimismo, dentro de esas consideraciones se encuentran los virus.
    - Conducta maliciosa tendiente a la destrucción o inutilización de un sistema de tratamiento de información o de sus partes componentes o que dicha conducta.

- Francia. En enero de 1988, este país dictó la Ley relativa al fraude informático, en la que se consideran aspectos como:
    - Intromisión fraudulenta que suprima o modifique datos.
    - Conducta intencional en la violación de derechos a terceros que haya impedido o alterado el funcionamiento de un sistema de procesamiento automatizado de datos.
      - Conducta intencional en la violación de derechos a terceros, en forma directa o indirecta, en la introducción de datos en un sistema de procesamiento automatizado o la supresión o modificación de los datos que éste contiene, o sus modos de procesamiento o de transmisión.
      - Supresión o modificación de datos contenidos en el sistema, o bien en la alteración del funcionamiento del sistema (sabotaje) impida, obstaculice o modifique su funcionamiento.
      - Conducta maliciosa que altere, dañe o destruya los datos contenidos en un sistema de tratamiento de información.
  - Gran Bretaña. Debido a un caso de hacking en 1991, comenzó a regir en este país la Computer Misuse Act (Ley de Abusos Informáticos). Mediante esta ley el intento, exitoso o no, de alterar datos informáticos es penado con hasta cinco años de prisión o multas. Esta ley tiene un apartado que especifica la modificación de datos sin autorización.
  - Holanda. El 10 de Marzo de 1993 entró en vigencia la Ley de Delitos Informáticos, en la cual se penaliza los siguientes delitos:
    - El hacking.
    - La ingeniería social (arte de convencer a la gente de entregar información que en circunstancias normales no entregaría).
    - La distribución de virus.
- Normativa y regulación de la informática en el ámbito europeo: Hasta ahora, el principal esfuerzo europeo por regular el tema de los delitos 21 de noviembre de 2001. Este documento fue firmado por los representantes de cada país miembro del Consejo de Europa, aunque su eficacia depende de su posterior refrendo por los órganos nacionales de cada país firmante. El "Convenio sobre la Ciberdelincuencia" permitió la definición de los delitos informáticos y algunos elementos relacionados con éstos, tales como "sistemas informáticos", "datos informáticos", o "proveedor de servicios".



Estos delitos informáticos fueron clasificados en cuatro grupos:

1. Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.
    - Acceso ilícito a sistemas informáticos.
    - Interceptación ilícita de datos informáticos.
    - Interferencia en el sistema mediante la introducción, transmisión, provocación de daños, borrado, alteración o supresión de éstos.
    - Abuso de dispositivos que faciliten la comisión de delitos
  
  2. Delitos informáticos.
    - Falsificación informática que produzca la alteración, borrado o supresión de datos informático que ocasionen datos no auténticos.
      - Fraudes informáticos.
  
  3. Delitos relacionados con el contenido.
    - Delitos relacionados con la pornografía infantil.
  
  4. Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.
    - Es conviene destacar que en el "Convenio sobre la Ciberdelincuencia" se encomienda a cada Parte que tome las medidas necesarias para tipificar como delito en su derecho interno cada uno de los apartados descritos en cada categoría.
  
    - En la Disposición 14221 del BOE núm. 226 de 2010, encontramos el Instrumento de Ratificación del Convenio sobre la Ciberdelincuencia, hecho en Budapest el 23 de noviembre de 2001.
- Normativa y regulación de la informática en el ámbito nacional:
    - Leyes y Decretos Ley. Ley Orgánica de Protección de datos de carácter personal: régimen sancionador aplicable (BOE no298 de 14/XII/99 que publicó la Ley Org. 15/1999 de 13 de Dic.) Objeto: Proteger y garantizar las libertades públicas y derechos fundamentales de las personas, especialmente su HONOR e INTIMIDAD personal y familiar. Aspectos de interés: Serán responsables: "Los responsables de los ficheros o de los tratamientos" y "los encargados de los tratamientos".  
Ley 7/1998 de 13 de Abril que regula las condiciones generales de contratación.

R.D. 1906/1999 de 17/XII que regula la contratación telefónica.

R.D. Ley 14/1999 de 17/XII sobre Firma Electrónica (BOE No224 de 18/XII).

- Firma electrónica: Dispositivo electrónico que permite la identificación del signatario de las operaciones realizadas por Internet. Identifica: El firmante (autenticación) y Evita el retracto (no repudio).

Código Penal

Ley Orgánica 10/1995 de 23/XI

- Tipifica delitos y faltas por el uso de la informática, concretamente contra la Intimidad, Patrimonio, Socioeconómicos y Propiedad Intelectual.
- Título X: "Delitos contra la intimidad, derecho a la propia imagen y la inviolabilidad del Domicilio".

Finalidad para la que se recogen los datos

- Desconfíe si los datos que le solicitan son excesivos para la finalidad con la que se recogen o innecesarios para el servicio que se le presta.
- Tenga en cuenta que cuando introduce su dirección de correo electrónico en un directorio, lista de distribución o grupo de noticias, dicha dirección puede ser recogida por terceros para ser utilizada con una finalidad diferente, como por ejemplo, remitirle publicidad no deseada.
- Cuando navegue por Internet, sea consciente de que los servidores Web que visita pueden registrar tanto las páginas a las que accede como la frecuencia y los temas o materias por las que busca, aunque no le informen de ello.

Seguridad en el intercambio de datos

- Utilice, siempre que sea posible, las últimas versiones de los programas navegadores, ya que cada vez suelen incorporar mejores medidas de seguridad.
- Considere la posibilidad de activar en dichos programas las opciones que alerten sobre los intercambios de datos no deseados y no rellene aquellos datos que no desee hacer públicos (por ejemplo, dirección de correo electrónico, nombre, apellidos, etc.).
- No realice transacciones comerciales electrónicas a través de proveedores con sistemas inseguros o no fiables. Consulte el manual de su navegador para averiguar cómo informa de que se ha establecido una conexión con un servidor seguro.
- Recuerde que existen sistemas de dinero electrónico que preservan el anonimato de sus compras en Internet.

- Utilice los mecanismos de seguridad que tenga a su alcance para proteger sus datos de accesos no deseados. El medio más fiable para conseguirlo es el cifrado de los mismos.
  - Salvo que se utilicen mecanismos de integridad, autenticación y certificación (firma digital, notarios electrónicos, etc.) no confíe ciegamente en que la persona u organización que le remite un mensaje es quien dice ser y en que el contenido del mismo no se ha modificado, aunque esto sea así en la inmensa mayoría de las ocasiones.
- Para terminar: Siempre que se le soliciten datos personales que no esté obligado legalmente a suministrar, sopesese los beneficios que va a recibir de la organización que los recoge frente a los posibles riesgos de utilización irregular de los mismos. Ante cualquier duda sobre la legalidad de la utilización de sus datos de carácter personal, póngase en contacto con la Agencia de Protección de Datos.
    - La protección jurídica de programas de ordenador. Piratería informática
      - El Real Decreto Legislativo 1/1996, por el que se aprueba el Texto Refundido sobre Propiedad Intelectual, la protección jurídica de los programas de ordenador, antes regulada por la Ley de Protección Jurídica de Programas de Ordenador y por la Ley de Propiedad Intelectual, crea un marco jurídico en contra de la piratería informática.
      - El Texto Refundido desarrolla una serie de medidas para combatir la piratería informática, como la posibilidad de que los fabricantes de programas de ordenador soliciten a la justicia española la realización de un registro sorpresa en empresas en las que existan sospechas fundadas o evidencias de delito.
      - España es uno de los países en los que se puede acudir a esta medida cautelar.
      - De esta manera se erradica la posibilidad de que los presuntos infractores puedan destruir las pruebas existentes, lo cual, indudablemente ocurrirá si se les notifica por adelantado la realización de un registro.
    - ¿En qué casos se infringe la Ley?
      - Al copiar o distribuir un programa de ordenador o la documentación que le acompaña, incluidas aplicaciones, datos, códigos y manuales, sin permiso expreso o licencia del propietario de los derechos de explotación.
      - Al utilizar un programa sin la correspondiente licencia o autorización del fabricante, con independencia de que se utilice en un solo ordenador o en varios de forma simultánea.
      - Al utilizar programas de ordenador en un número de copias superior al autorizado por el fabricante en sus contratos o licencias de uso.

- En empresas y demás organizaciones, al fomentar, consciente o inconscientemente, permitir, obligar o presionar a los empleados a realizar o distribuir copias no autorizadas del programa.
  - Al efectuar copias no autorizadas porque alguien lo requiere u obliga a ello. Al ceder o prestar el programa de forma que pueda ser copiado o al copiarlo mientras está en su posesión en calidad de cedido o prestado.
  - Al crear, importar, poseer o negociar con artículos destinados a burlar o neutralizar cualquier medio técnico aplicado para proteger el programa de ordenador.
- Medidas Judiciales. Si finalmente existe evidencia de delito, las medidas judiciales que pueden adoptarse son:
    - Solicitar al Juez un registro sorpresa de las instalaciones del presunto infractor, tanto por la vía civil, como por la penal.
    - Solicitar al Juez la adopción urgente de medidas cautelares de protección.
    - Exigir indemnizaciones acordes con los daños materiales y morales causados.
    - El cierre del centro de actividad del infractor.
    - El secuestro de todos aquellos medios destinados a suprimir los dispositivos técnicos que protegen un programa desarrollado y comercializado por un fabricante de programas.

#### **TFG01**

***Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.***

Esta competencia se alcanza gracias a la completa realización de este trabajo de fin de grado, abarcando todos los ámbitos dentro de las tecnologías específicas de la Ingeniería Informática.



## Anexo II: Legislación vigente

Se incluye a continuación la legislación vigente que afecta a este trabajo fin de grado:

### Software libre

El software libre (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término en el idioma inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de este, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

Las aplicaciones de software libre utilizadas en el desarrollo del proyecto han sido las siguientes:

NetBeans es un entorno de desarrollo libre bajo los términos de General Public License (GPL) y también de COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL).

XAMPP es gratuito y libre conforme los términos de la licencia GNU.

Mozilla Firefox es navegador libre bajo una triple licencia GNU GPL, GNU LGPL o Mozilla Public License (MPL).

### Licencia de Software

Una licencia de software otorga al usuario derecho legal a utilizar un software. Por cada programa de software de Microsoft que se utiliza, se otorga una licencia al usuario y ésta se

documenta en el Contrato de Licencia de Usuario Final (CLUF). El acuerdo de licencia da al usuario el derecho de utilizar el software. El software está protegido por la ley de derechos de autor, que establece que el producto no se puede copiar sin autorización del dueño de derechos de autor.

Las aplicaciones con licencia de software utilizadas son:

Windows 7 bajo una licencia del tipo Licencia Original.

Paquete Microsoft Office bajo una licencia del tipo Licencia Original

### **Ley de protección de datos**

Tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

Solamente se podrá recoger los datos de carácter personal utilizando medios que no sean fraudulentos desleales; siendo datos personales que sean adecuados, pertinentes y no excesivos en relación con el proyecto.

Se deberá almacenar los datos personales de manera que el titular de los datos pueda ejercer su derecho de acceso cuando lo considere oportuno.

Y se deberán cancelar los datos cuando hayan dejado de ser necesarios para la finalidad del proyecto.

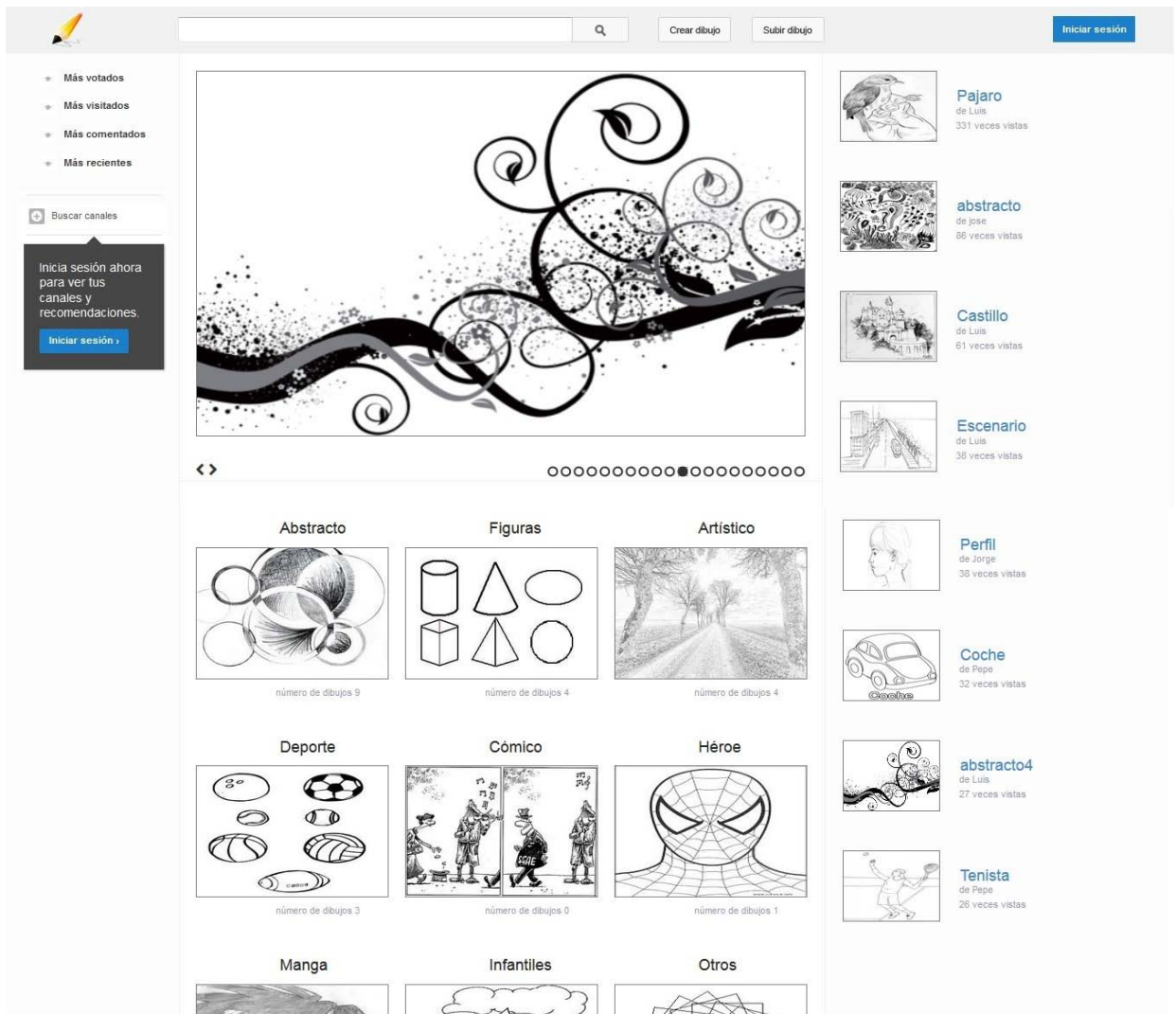
## Anexo III: Manual de usuario

El manual de usuario está dividido en dos partes bien diferenciadas, según el nivel de privilegios que posea el visitante de la aplicación. Obviamente los usuarios con más privilegios tienen acceso a las áreas del nivel inferior anterior.

Se explicará las zonas que un usuario sin identificar puede visitar (Visitante o Usuario anónimo) y luego las zonas correspondientes a un usuario (Usuario) identificado y validado en el sistema.

### Visitante o Usuario anónimo

Como Visitante o Usuario anónimo tiene acceso a la página principal donde podrá ver en un Slider los dibujos más recientes, también los dibujos más votados y los diferentes canales de dibujo como puede ser Abstracto, Figuras, Artístico, etc.





Cuando el Visitante quiera ver un dibujo, se mostrará la información del dibujo que viene siendo el título del dibujo, la foto perfil del usuario y la fecha con la que subió el dibujo, los votos positivos y negativos, una breve descripción y los comentarios publicados.

Más votados  
Más visitados  
Más comentados  
Más recientes

Buscar canales

Inicia sesión ahora para ver tus canales y recomendaciones.  
Iniciar sesión >

Crear dibujo Subir dibujo Iniciar sesión

abstracto

jose 22 1

Inicia sesión para poder votar Información Compartir Añadir a

Publicado el 03/10/2013  
dibujo abstracto

Inicia sesión para poder comentar

Luis nuevo comentario  
Responder

Jorge este dibujo si me gusta mucho  
Responder

Luis

Pajaro de Luis 332 veces vistas

abstracto de Jose 86 veces vistas

Castillo de Luis 62 veces vistas

abstracto4 de Luis 28 veces vistas

ffff de Jorge 20 veces vistas

Pez de Luis 5 veces vistas

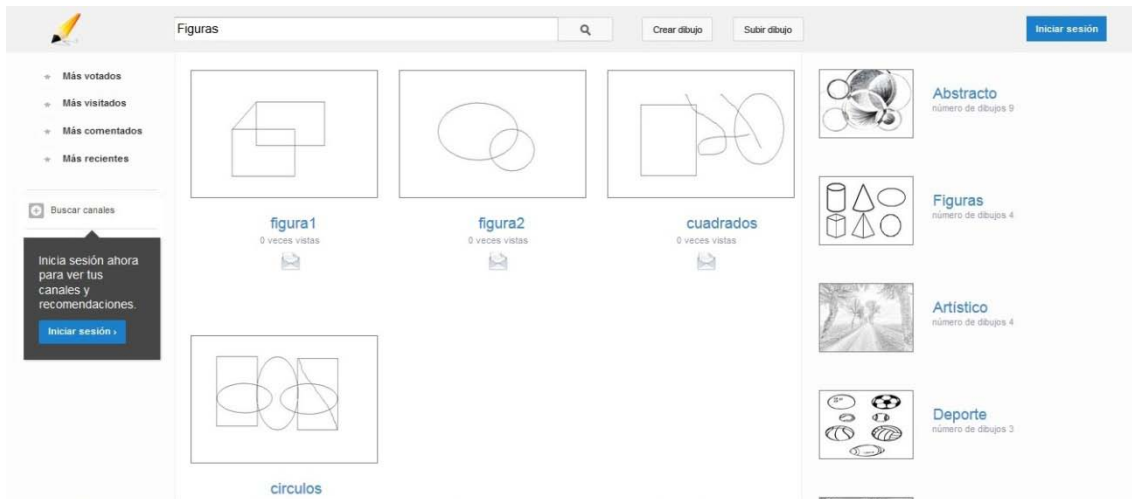
raro de Luis 1 veces vistas

cuadros de Jorge 0 veces vistas

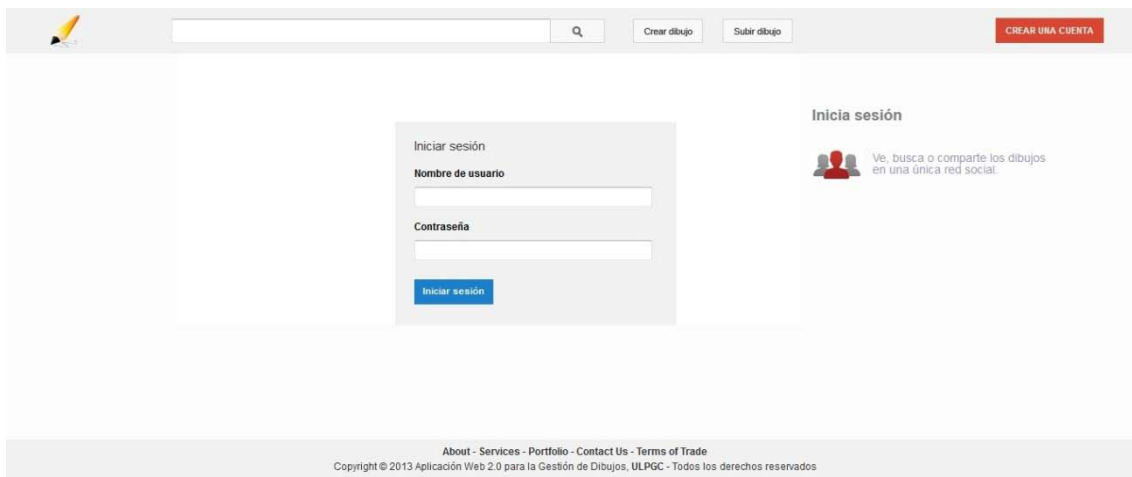
cuadros de Jorge 0 veces vistas

Hay dos formas de buscar dibujos:

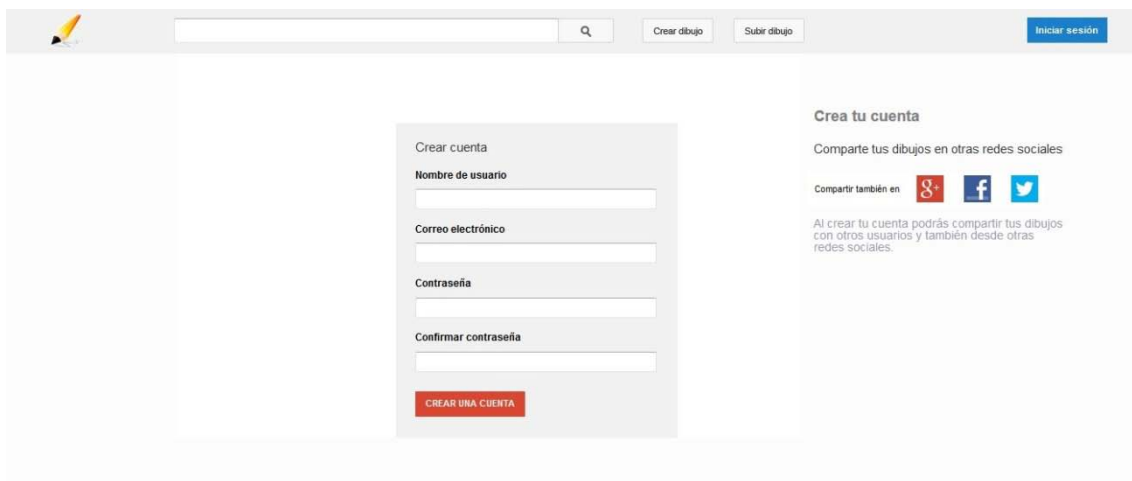
1. Seleccionando el canal de dibujos y mostrarían todos los dibujos relacionados con ese canal.
2. Desde el buscador escribiendo el nombre del canal y mostraría los mismos resultados que en el Paso 1, o bien, escribiendo el título del dibujo y accedería a la información de dicho dibujo.



Por último, el Visitante podrá registrarse para poder ser Usuario de la aplicación. Primero irá a Iniciar sesión y como no tiene una cuenta de Usuario, elegirá CREAR UNA CUENTA.



Para crear una cuenta solamente se pedirá nombre de usuario, email (para confirmar registro) y contraseña.



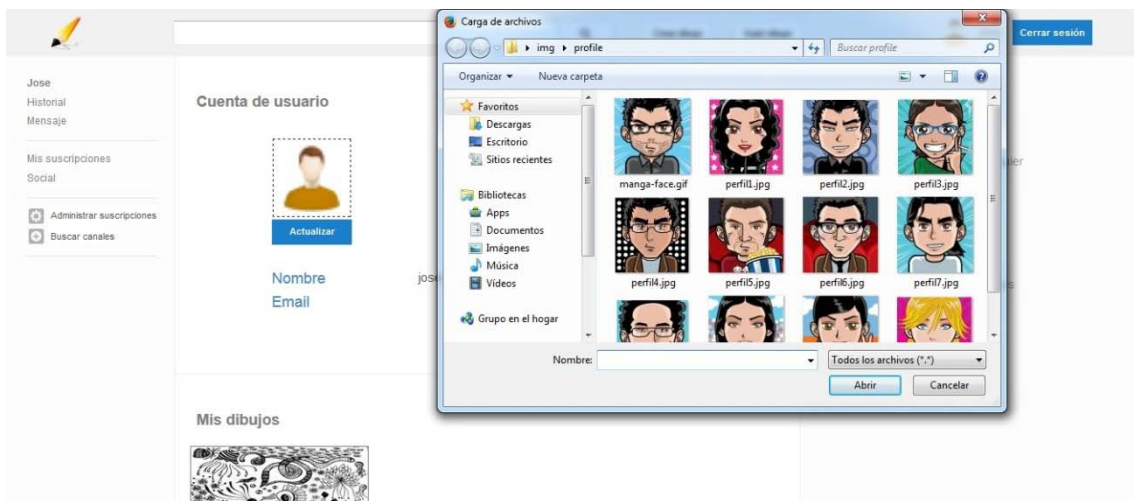
## Usuario

Como se mencionó antes, el Usuario de la aplicación podrá hacer lo que hace el Visitante y otras funciones que el Visitante no tiene acceso.

Una de las opciones es personalizar el perfil de Usuario. Pulsando sobre la foto perfil del Usuario desde la cualquier página accedes al perfil de Usuario y puedes modificar la foto perfil del usuario.



El Usuario al pulsar sobre la foto perfil que aparece en información o con el botón Actualizar, podrá seleccionar una imagen para su perfil siendo en formato .jpg o .gif.



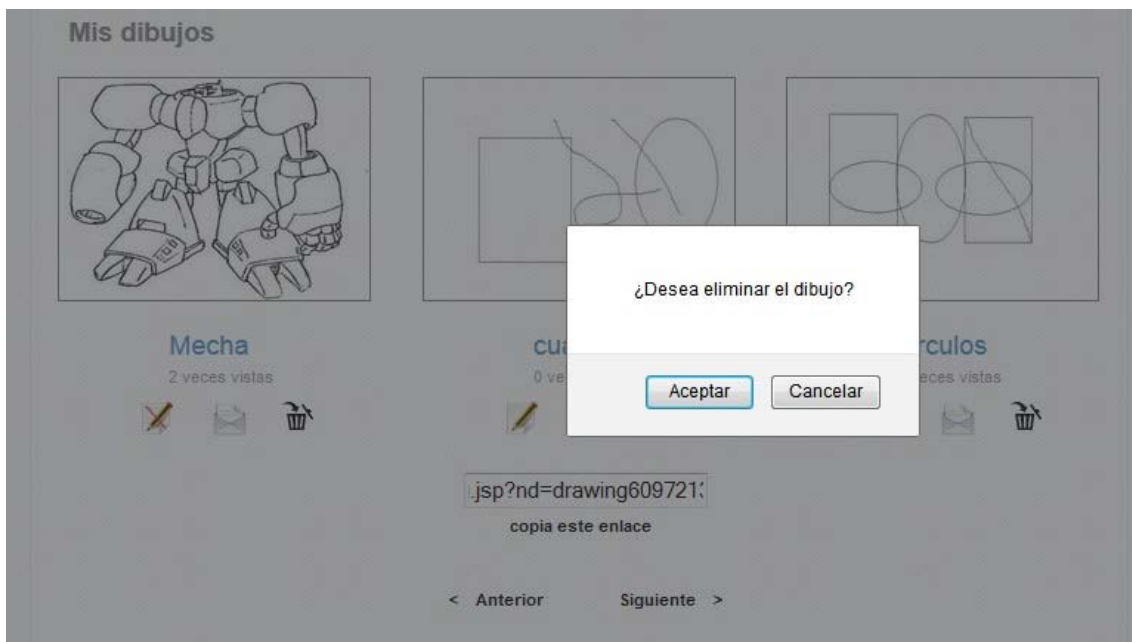
Finalmente se muestra la foto perfil del Usuario actualizada.



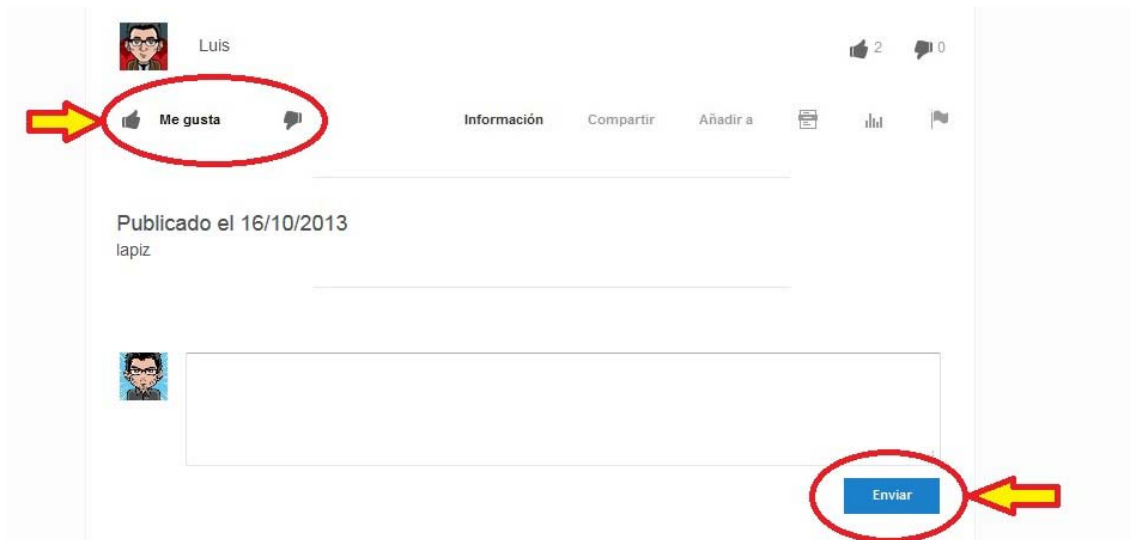
Al final de página se muestran los dibujos del Usuario pudiendo editar (en el caso de ser un dibujo creado por el Usuario), mostrar url de cualquier dibujo para después compartirlo o eliminar el dibujo que desee. Esta misma información la puede ver el Usuario en la página principal.



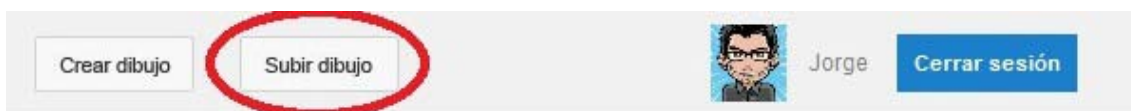
Antes de eliminar el dibujo se mostrará un pop-up para confirmar si realmente quiere eliminarlo.



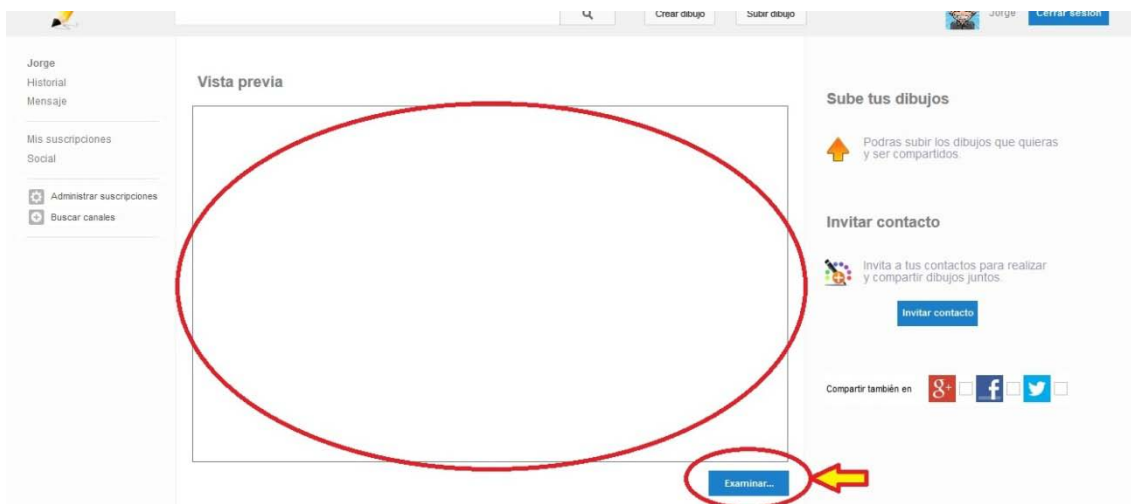
Cuando se accede a la información del dibujo se puede votar y/o dejar un comentario al dibujo.



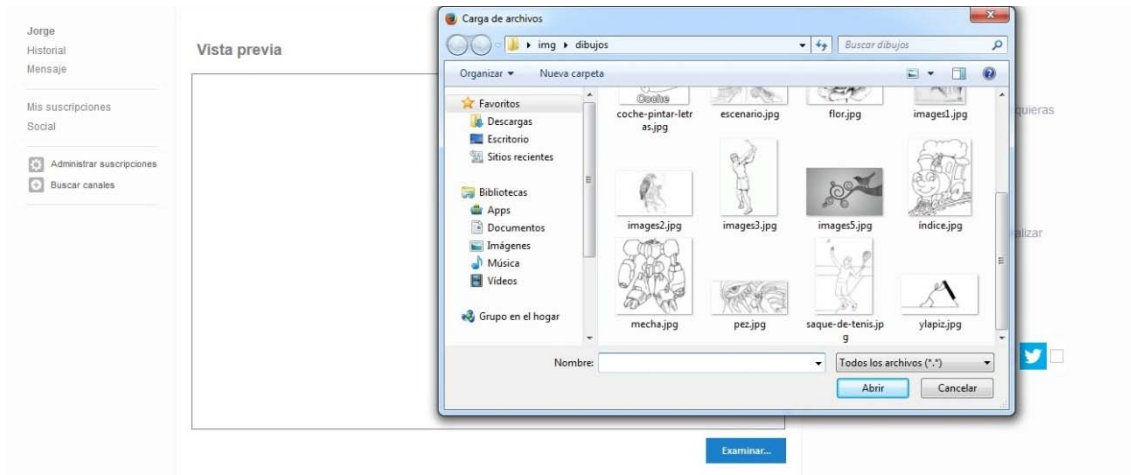
Para subir un dibujo es pulsando en Subir dibujo que aparece en la cabecera de la página.



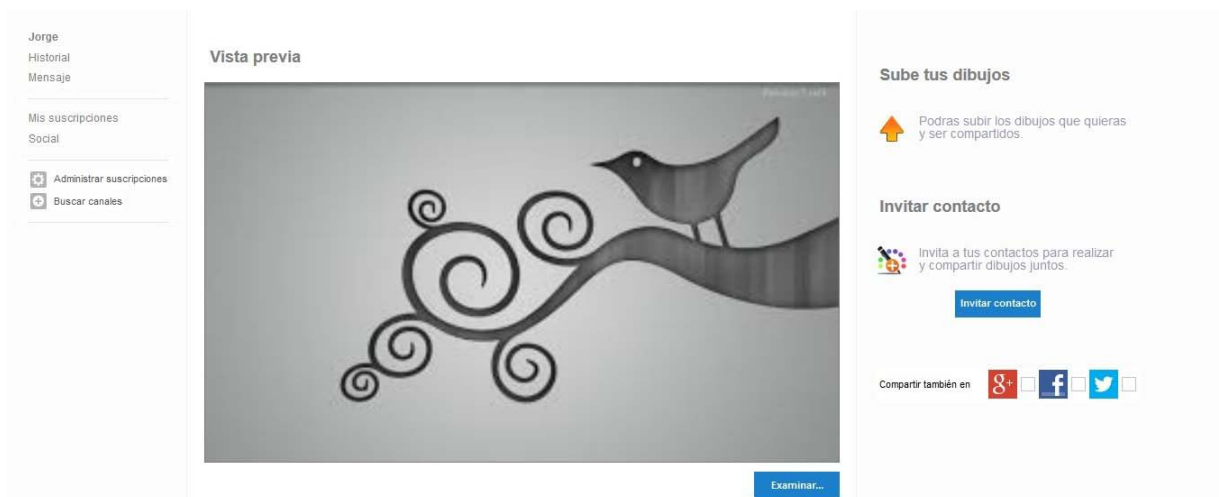
El procedimiento es casi igual que al actualizar la imagen perfil del Usuario, pulsando sobre la imagen en blanco o el botón Examinar.



Se selecciona el dibujo que se quiere subir.



En la vista previa se puede ver cómo queda la imagen dentro de la vista previa.



Cuando se quiera subir la imagen se deberá poner un título, una breve descripción (opcional), etiquetas (opcional), suscriptores (opcional), privacidad para saber si el dibujo será público (cualquier Usuario podrá verlo) o privado (sólo el Usuario que subió el dibujo podrá verlo).

Título

Privacidad

Descripción

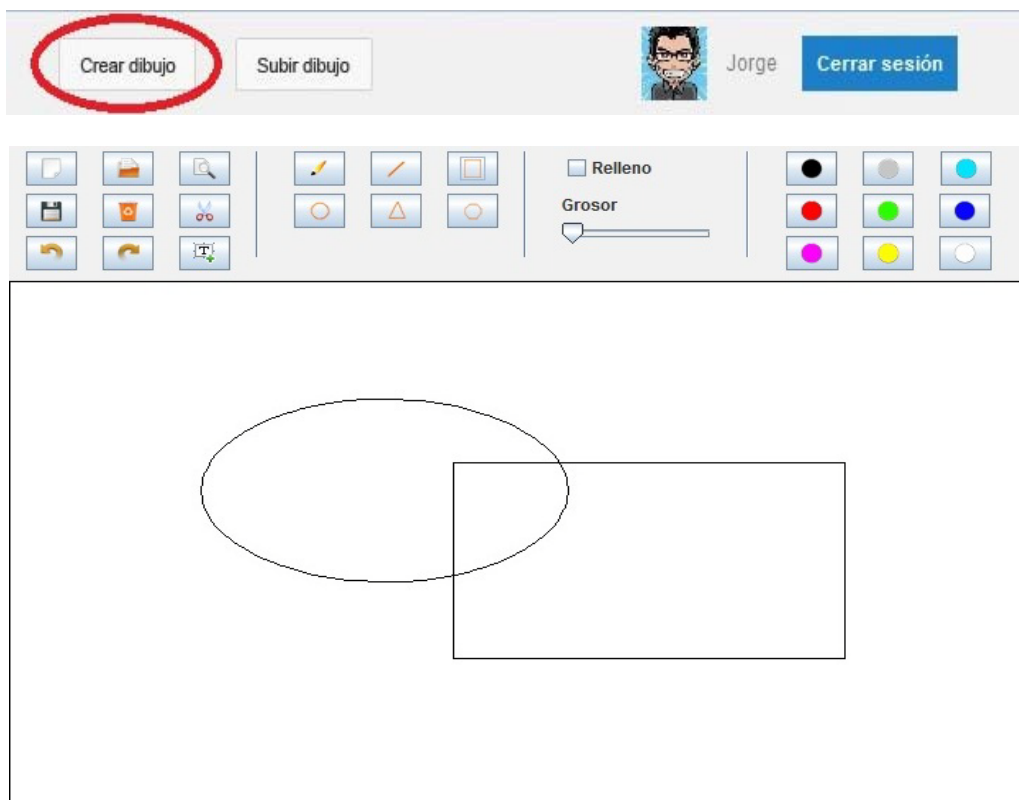
Publicación suscriptores

Tipo

Etiquetas

Una vez subido se mostrará una barra de carga y al finalizar el dibujo estará subido al servidor. Posteriormente se mostrará en la página perfil del Usuario.

Por último el Usuario podrá crear sus dibujos haciendo uso del editor de dibujo con las herramientas que ofrece.

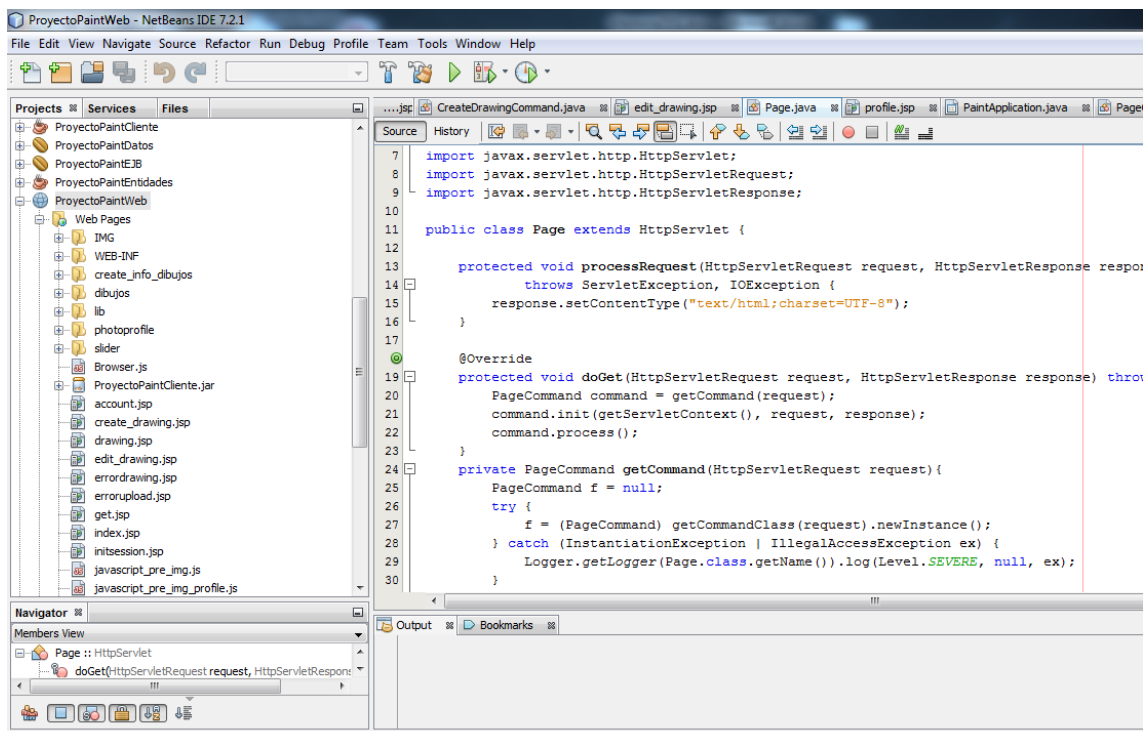


Una vez terminado, el procedimiento para almacenar el dibujo en el servidor es idéntico a Subir dibujo, relleno los mismos campos y mostrándose en la página perfil del Usuario.

## Anexo IV: [ENTORNO DE DESARROLLO]

El entorno de desarrollo de este proyecto es NetBeans, siendo un entorno de desarrollo libre creado principalmente para el lenguaje de programación Java. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamado *módulo*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

La plataforma Java Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE tiene varias especificaciones de API, tales como JDBC, RMI, XML, etc y define cómo coordinarlos. También configura algunas especificaciones únicas para componentes y estas incluyen servlets, JavaServer Pages y varias tecnologías de servicios web.



La aplicación de este proyecto usa dicha arquitectura de N capas. Está dividido en dos Java Applications siendo las entidades que representan los elementos del editor de dibujo y el cliente que contiene el JApplet desde el lado del cliente para conectarse con el servidor por medio de servlets. Después en dos Java EJB module siendo la capa de Negocio que se centra en la



comunicación del servidor con las operaciones para gestionar los datos mediante sentencias mysql usando la API JDBC para conectarse con la base de datos. Y por último un Java Web Application que junto a los dos Java Applications representa la capa de representación y es el encargado del despliegue del proyecto.

```
ProyectoPaintWeb (run)  Java DB Database Process  GlassFish Server 3.1.2
ant -f D:\JAD\Docs\Proyectos\JAVA\NetBeans\ProyectoPaintWeb -DforceRedeploy=false -Ddirectory.deployment.supported=true -Dnb.wait.for.cact
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsp:
Starting GlassFish Server 3.1.2
GlassFish Server 3.1.2 is running.
Incrementally deploying ProyectoPaintWeb
Completed incremental distribution of ProyectoPaintWeb
run-deploy:
Browsing: http://localhost:8080/ProyectoPaintWeb/Page?Command=Index
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 30 seconds)
```

Gracias a esto, la gestión de los componentes desplegados permite a los desarrolladores concentrarse más en la lógica de negocio de los componentes en lugar de tareas de mantenimiento de bajo nivel.

## Bibliografía

- [1] about.com, ¿Qué es un Applet en Java?,  
<http://aprenderinternet.about.com/od/Glosario/g/Applet-En-Java.htm>
- [2] adictosalared, <http://www.adictosalared.com/2010/04/imgur-servicio-gratuito-para-alojary-compartir-todo-tipo-de-imagenes/#.UpNkHeJSY4w>, 2010.
- [3] adobe, Preguntas más frecuentes,  
<http://www.adobe.com/es/products/photoshop/faq.html>
- [4] Apache, XAMPP, <http://www.apachefriends.org/es/xampp.html>
- [5] BOE, Principios de la protección de datos, [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-1999-23750](https://www.boe.es/diario_boe/txt.php?id=BOE-A-1999-23750)
- [6] Cuidatusdatos, ¿Qué hay que hacer para cumplir con la Lopd?,  
<http://cuidatusdatos.com/infocomocumplirlopd.html>
- [7] deviantart, Conoce la nueva herramienta Sta.sh,  
<http://noticias.deviantart.com/journal/Conoce-la-nueva-herramienta-Sta-sh-287578827>, 2012.
- [8] Gimp, Feature Overview, <http://www.gimp.org/features>, 2013.
- [9] Ivar Jacobson, Grady Booch y James Rumbaugh, El Proceso Unificado de Desarrollo Software, Addison Wesley, 1999.
- [10] ite.educacion.es, La máquina virtual de Java,  
[http://www.ite.educacion.es/formacion/materiales/91/cd/para\\_saber\\_mas/la\\_mquina\\_virtual\\_de\\_java.html](http://www.ite.educacion.es/formacion/materiales/91/cd/para_saber_mas/la_mquina_virtual_de_java.html)
- [11] Microsoft, Paint, <http://windows.microsoft.com/es-es/windows7/products/features/paint>
- [12] myjavazone, JApplet, <http://www.myjavazone.com/2010/06/japplet.html>, 2010.
- [13] NetBeans, Docs & Support, <https://netbeans.org/kb/index.html>
- [14] sc.ehu.es, Un programa de dibujo simple,  
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/applets/ejemplos/dibujo/dibujo.htm>
- [15] UNED, Web 2.0, [http://www.uned.es/biblioteca/guia\\_rapida/web2punto0.htm](http://www.uned.es/biblioteca/guia_rapida/web2punto0.htm)
- [16] what'snew, Kleki la herramienta de pintura que destaca en google-chrome,  
<http://wwwwhatsnew.com/2011/10/26/kleki-la-herramienta-de-pintura-que-destaca-en-google-chrome>, 2011.