

Universidad de las Ciencias Informáticas
“Facultad # 6”



**Título: “Análisis de la Herramienta CASE AISOFT
para la Arquitectura de Información”**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es): Yadelis Bárbara González Michel
Yuriem Fernández Tirador

Tutor(es): Ing. Renier Pérez García
Ing. Yunier Saborit Ramírez

Co tutor: Lic. Rodrigo Ronda León

Junio 2009

“Pensar es fácil; actuar, difícil; transformar los pensamientos en actos es lo más difícil.”

Johann W. Goeth.

DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de este trabajo y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año ____

Yadelis Bárbara González Michel

Yuriem Fernández Tirador

Ing. Renier Pérez García

Ing. Yunier Saborit Ramírez

AGRADECIMIENTOS

"[Insertar agradecimientos (opcional)]"

DEDICATORIA

"[Insertar dedicatoria (opcional)]"

RESUMEN

La Universidad de las Ciencias Informáticas hoy cuenta con más de 150 proyectos productivos entre los que se encuentran los proyectos de exportación que se desarrollan principalmente con la República Bolivariana de Venezuela, proyectos nacionales que se desarrollan con más del 80% de los ministerios cubanos y proyectos de informatización internos de nuestra universidad, de estos proyectos hasta el momento de la experiencia obtenida podemos concluir que la aplicación de la arquitectura de información ha afectado el resultado final de los mismo por haber incurrido en un conjunto de errores y una mala aplicación de este proceso, se considera que el mayor error es que no existe una herramienta que asista la arquitectura de información ya que para llevar a cabo este proceso se utilizan varias herramientas por separadas y ninguna abarca completamente los procesos y tareas de la misma. Por lo que se propone una herramienta CASE AISOFT que asista la Arquitectura de Información, a la cual se le realizó el análisis.

El presente trabajo muestra el análisis de la herramienta planteada anteriormente, el cual cumple con las necesidades reales del cliente. Para ello se realizó un estudio de las metodologías y herramientas para el desarrollo de un software que son más utilizadas en la actualidad, las actividades que realiza el analista de sistema, los objetivos de la Ingeniería de Requisitos, las técnicas para la captura de requisitos y los patrones de casos de uso que pueden ser aplicados para lograr una correcta modelación de estos. Finalmente se generaron los artefactos como resultado del proceso de análisis de la Herramienta, para culminar con la validación de los requerimientos mediante un prototipo no funcional y una carta de aceptación del cliente.

PALABRAS CLAVE

Arquitectura de Información, Herramienta CASE, análisis, requerimiento, prototipo.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	7
1.1 Arquitectura de Información.	7
1.1.1 Etapas de la Arquitectura de Información.....	8
1.1.2 Procesos y tareas de la Arquitectura de Información.	9
1.1.3 Herramientas que asisten la Arquitectura de Información.....	10
1.1.3.1 STAR TREE	11
1.1.3.2 CMAP TOOLS.....	11
1.1.3.3 DIGIDOC MAP	12
1.1.3.4 SPACE TREE	13
1.1.3.5 AXURE RP	13
1.2 Definiciones básicas.....	14
1.2.1 ¿Qué es un requerimiento?	14
1.2.2 Clasificación de requerimientos.....	15
1.2.3 Importancia de la etapa de requerimientos	15
1.2.4 La estructura de la Ingeniería de Requerimientos.....	16
1.2.5 Proceso de la Ingeniería de Requerimientos.....	17
1.3 Metodologías de desarrollo de software.	17
1.3.1 Metodología Programación Extrema (XP).	18
1.3.1.1 Proceso de Captura de Requisitos con XP.....	19
1.3.2 Metodología Scrum.....	19
1.3.3 Metodología Proceso Unificado de desarrollo de software (RUP).....	20
1.3.3.1 Roles y artefactos.	20
1.3.3.2 Flujo de trabajo de Requisitos.....	21
1.3.3.5 Estrategias y técnicas de captura de requisitos.....	22

1.4 Lenguaje de modelado.....	25
1.5 Herramienta de modelado.....	26
1.6 Patrones de Casos de Uso	27
Conclusiones.....	28
CAPÍTULO 2 ANÁLISIS DE LA HERRAMIENTA CASE AISOFT	30
2.1 Modelo del dominio.....	30
2.1.1 ¿Qué es un Modelo de Dominio?	30
2.1.2 ¿Por qué hacer uso de un Modelo de Dominio para el caso de estudio tratado?	31
2.1.3 Descripción del Dominio.....	32
2.1.3.1 Informe Diagnóstico	33
2.1.3.2 Definición de Objetivos del Sitio.....	33
2.1.3.3 Definición de Objetivos del Sitio.....	34
2.1.3.4 Definición de Audiencia.....	34
2.1.3.5 Definición de Contenidos del Sitio	35
2.1.3.6 Definición de la Estructura del Sitio	35
2.1.3.7 Definición de los Sistemas de Navegación	36
2.1.4 Diagrama de clases del modelo del dominio	37
2.1.4.1 Definición de las clases del modelo del dominio.....	37
2.2 Levantamiento de Requisitos.....	38
2.2.1 Requisitos Funcionales.....	38
2.2.2 Requisitos no Funcionales.....	40
2.3 Modelo de Casos de Uso del Sistema	43
2.3.1 Diagrama de actores del sistema.....	44
2.3.2 Actores del sistema	44
2.3.3 Diagrama de casos de uso del sistema.....	45
2.3.4 Descripción de los casos de uso del sistema	48
2.4 Validación.....	117
2.4.1 Prototipo no funcional de interfaz gráfica.....	117
2.4.2 Aceptación del Cliente	119
Conclusiones.....	121

CONCLUSIONES GENERALES	122
RECOMENDACIONES.....	123
BIBLIOGRAFÍA	124
ANEXOS.....	126
GLOSARIO DE TÉRMINOS	148

INTRODUCCIÓN

Los avances tecnológicos experimentados durante las últimas décadas del siglo XX impulsaron una serie de procesos que algunos autores han definido “como una nueva revolución productiva o industrial”. Tal fenómeno se caracteriza por la importancia cada vez mayor de la innovación tecnológica y del conocimiento como factor preponderante en la generación de valor.

En este escenario las tecnologías de la información y las comunicaciones (TIC) han adquirido una gran importancia debido a un conjunto de factores, como: la expansión acelerada y los cambios revolucionarios en el sistema de las telecomunicaciones; los procesos vinculados al desarrollo de la red de redes internet; la introducción y crecimiento exponencial de las computadoras personales; así como la demanda de programas de cómputo especializados. Tales elementos se encuentran asociados al desarrollo y uso creciente de una tecnología multifuncional: el **software**, cuyas características dificultan su definición. Es un elemento dual de servicio y producto, intangible y necesario para muchas actividades; desempeña un papel clave en la reconfiguración actual de las nuevas industrias, indispensable para el procesamiento de datos, necesario para que funcionen los hardwares y además se incorpora a productos industriales de uso cotidiano. Su gran dinamismo económico propicia el desarrollo de nuevas áreas y crea nuevas oportunidades de empleo. Ha generado una industria importante, cuyos campos fundamentales son **la Ingeniería de Software** y los servicios informáticos (ISSI) que tienen una estructura compleja y una gran capacidad de innovación.

(1)

“De la Ingeniería de Software podemos decir que es la rama de la Ingeniería que aplica los principios de la Ciencia de la Computación y las matemáticas para lograr soluciones de software efectivas, es decir, permite elaborar consistentemente productos de software correctos y utilizables. La ingeniería de software es una disciplina que define un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad. Dentro de esta ciencia se incluye la ingeniería de requisitos que se enfoca en mejorar la captura, especificación, validación y la gestión de requisitos del software. Para esto, la IR estudia y define métodos, notaciones y herramientas para garantizar una adecuada definición de los requisitos”. (2)

La correcta especificación de requisitos determina en gran medida el éxito de un proyecto de desarrollo de software, esto trae como beneficio que se puedan gestionar las necesidades del mismo en forma estructurada, mejorar la capacidad de predecir los cronogramas, así como sus resultados, disminuir sus costos y retrasos, mejorar su calidad, la comunicación entre los clientes y desarrolladores y evitar los rechazos de los usuarios finales.

Según como afrontan las organizaciones el desarrollo del software aplicando la ingeniería de software, ésta puede comportarse como factor de riesgo o amenaza para el negocio; o por el contrario como una poderosa oportunidad de negocio. Todas las empresas quieren producir más rápido, mejor y con menores costes, y sin duda esto es posible porque la naturaleza del software no es origen de riesgos y problemas, sino una fuente de oportunidades. Cada vez más directivos están comprendiendo que la forma de gestión del desarrollo de software, puede hacer de la materia prima de su negocio un material arisco de resultados impredecibles, o una ventaja competitiva. La evolución hacia entornos de ingeniería del software requiere cambios severos en la organización, así como el convencimiento, implicación y empuje de la dirección. Pero sobre todo el diseño de un modelo de producción propio que sepa aprovechar la personalidad de la organización, y responder a las particularidades de su negocio.

(1)

En este entorno de amenazas y oportunidades se encuentra la Universidad de las Ciencias Informáticas (UCI) centro estratégico para el futuro, la UCI es una institución educacional en constante evolución, que forma profesionales de la informática con una alta capacitación científica, comprometidos con el desarrollo del país. Este centro experimental, surgido en el año 2002, garantiza la preparación académica y el entrenamiento profesional de excelencia a miles de jóvenes informáticos, con segundos perfiles profesionales validados a través del vínculo directo y sistemático con la producción, el empleo de las tecnologías más actuales y diseños curriculares flexibles. Lleva a cabo, importantes proyectos en sectores como la salud, la educación y los servicios al ciudadano y gobierno, a partir del uso de una moderna infraestructura tecnológica y novedosos métodos de organización, que impulsan la informatización de la sociedad cubana y el desarrollo de la Industria Cubana del Software. En su sexto año de funcionamiento, la UCI impulsa la producción de software y servicios profesionales como una de sus apuestas de futuro. Se ha creado la Infraestructura Productiva (IP) que cuenta con una administración innovadora, una perspectiva competitiva, novedosos métodos de vinculación estudio – trabajo y la seguridad de que el futuro de Cuba está en sus producciones intelectuales. La IP está diseñando un modelo de producción de software propio capaz de adaptarse al

dinamismo y peculiaridad de la UCI, dentro de este modelo podemos destacar **la Arquitectura de Información** como una nueva disciplina encargada del estudio, análisis, organización y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos. Con el fin de que la asimilación de contenidos por parte del usuario sea eficiente y efectiva, y también para que la información sea accesible y usable. Usualmente el trabajo de los Arquitectos de Información se concretan en la generación de un conjunto de materiales entregables en los que se plasma de forma efectiva la estructura del espacio de información, el diseño de la interacción del usuario con el sistema y el funcionamiento de la interfaz. Estos entregables tienen la finalidad de servir:

- A los clientes como resultado del proceso de diseño o reingeniería del software llevado a cabo durante la fase de análisis, definición y desarrollo del proyecto.
- A los diseñadores gráficos como material de base para la producción de maquetas gráficas sobre las que se basarán los maquetadores.
- A los ingenieros informáticos para desarrollar el back-end (la programación) del software.

La UCI hoy cuenta con más de 150 proyectos productivos entre los que se encuentran los proyectos de exportación que se desarrollan principalmente con la República Bolivariana de Venezuela, proyectos nacionales que se desarrollan con más del 80% de los ministerios cubanos y proyectos de informatización internos de nuestra universidad, de estos proyectos hasta el momento de la experiencia obtenida podemos concluir que la aplicación de la arquitectura de información ha afectado el resultado final de los mismo por haber incurrido en un conjunto de errores y una mala aplicación de este proceso, algunos de estos errores son:

- No existe la manera de llevar un control de la arquitectura de información en los proyectos productivos, pues actualmente el responsable de esta actividad no conoce el estado actual en que se encuentra la misma.
- Los materiales entregables no cuentan con un formato estándar, es decir algunos arquitectos de información entregan los documentos en PDF utilizando como herramienta Adobe Acrobat Reader y otros en punto doc con la herramienta Microsoft Office Word.
- No existe un lugar físico con las pautas de arquitectura definidas legalmente (creadas por la dirección técnica) donde ellos puedan acceder; por lo tanto los arquitectos no tienen una definición clara de los entregables, al cliente y a los de uso interno del proyecto y utilizan

plantillas encontradas en otros proyectos (pueden ser plantillas no actualizadas). Teniendo como resultado que cada proyecto contenga distinta documentación de arquitectura de información.

- Otro de estos errores es la utilización de varias herramientas informáticas para llevar a cabo el proceso de AI por ejemplo la herramienta Microsoft Office Word como procesador de texto, el MindManager para la realización de mapas conceptuales, el Microsoft Office Visio y el Adobe Photoshop para la realización de prototipos de interfaz grafica, entre otras.

Actualmente existen herramientas para realizar la AI pero solo satisfacen la realización de mapas conceptuales, siendo de todas estas, el Axure RP la herramienta más completa; pero ninguna de las mismas concibe plenamente esta disciplina.

Por lo que, por toda esta situación problemática, se ha planteado el siguiente **problema científico** ¿Cómo integrar las funcionalidades de las herramientas informáticas que se utilizan para modelar los procesos y tareas de la Arquitectura de Información?

Dar solución a este problema, permitiría que los arquitectos de información con una sola herramienta informática realizarían toda la documentación de arquitectura de información, también tendrían centralizadas las pautas de la misma, generándose las plantillas con un formato estándar. Además de que el director técnico de la IP contaría con un chequeo constante del estado de la arquitectura de información de los proyecto de la UCI.

Por lo tanto, como **objeto de estudio**: La Arquitectura de Información como parte del desarrollo de software. Siendo el **campo de acción**: Las herramientas informáticas que asisten los procesos y tareas de la Arquitectura de Información en la UCI.

Con el propósito de dar solución al problema se precisa como **objetivo general**: realizar el análisis a una propuesta de herramienta CASE AISOFT para la Arquitectura de Información en la UCI.

Para alcanzar dicho objetivo se trazaron las siguientes **tareas científicas**:

1. Estudio del estado del arte de las diferentes herramientas que asisten a la AI.
2. Estudio del estado del arte de los procesos de ingeniería de requisitos de la metodología RUP.

3. Elaboración del modelo de dominio.
4. Especificación de los requisitos funcionales y no funcionales.
5. Modelado de los casos de uso del sistema.
6. Validación de los casos de uso.

Para mayor comprensión de los contenidos de este trabajo se ha determinado que el documento se ha estructurado en 2 capítulos:

Capítulo #1. Fundamentación Teórica: en este capítulo se realiza un estudio del estado del arte de los temas relacionados con las herramientas que asisten a la Arquitectura de Información como proceso novedoso dentro de los proyectos productivos, describiendo sus etapas, y técnicas a aplicar. También se analiza la ingeniería de software resaltando dentro de este tema la disciplina de la ingeniería de requisitos como elemento de gran importancia dentro del desarrollo de software, así como el análisis de las metodologías y como alguna de estas contemplan el desarrollo de requisitos. Se detallan las estrategias y técnicas de captura de requisitos.

Capítulo #2. Análisis de la Herramienta CASE AISOFT: Antes de enfrentarse al desarrollo de una herramienta informática, es necesario tener conocimiento del problema, es decir, de la situación actual que enfrenta la organización. Comenzar a implementar una herramienta informática sin conocer sus características principales ni el vocabulario propio del organismo, puede traer consigo que el producto final no sea el esperado por los clientes, ni los usuarios. De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de los problemas reales que se vinculan con la herramienta informática a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. RUP propone en su fase de inicio, para llegar a un mejor entendimiento de la organización, la realización de un modelo de negocio y/o un modelo de dominio para precisamente evitar, en gran medida, los resultados no deseados por las partes involucradas en el proceso de desarrollo. Por lo que se realiza la propuesta de solución para la herramienta CASE, aplicando el flujo de trabajo de levantamiento de requisitos de la metodología RUP generándose los siguientes artefactos. Luego de transcurrida los flujos de Modelado de Negocio y Requerimientos de la Herramienta, esta se puso a consideración de los clientes hasta que finalmente fue aceptado por los mismos mediante una carta de aceptación, mostrando que se logró cumplir con sus expectativas y que los requerimientos que se definieron fueron los que cumplían con las necesidades de los clientes. Para asegurar la validez del Modelo de Casos de Uso del Sistema se aplicó un prototipo no funcional de la herramienta CASE que permitió evaluar el mismo, incluyendo

su interfaz y sus funcionalidades de entradas y salidas. Siendo el paso final luego del proceso de refinamiento de dicho modelo.

Este trabajo incluye además las conclusiones, recomendaciones y referencias bibliográficas utilizadas en su elaboración.



CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza un estudio del estado del arte de los temas relacionados con las herramientas que asisten a la Arquitectura de Información como proceso novedoso dentro de los proyectos productivos, describiendo sus etapas, y técnicas a aplicar. También se analiza la ingeniería de software resaltando dentro de este tema la disciplina de la ingeniería de requisitos como elemento de gran importancia dentro del desarrollo de software, así como el análisis de las metodologías y como alguna de estas contemplan el desarrollo de requisitos. Se detallan las estrategias y técnicas de captura de requisitos.

1.1 Arquitectura de Información.

La arquitectura de información no es más que una nueva disciplina encargada del estudio, análisis, organización y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos. Con el fin de que la asimilación de contenido por parte del usuario sea eficiente y efectiva, y también para que la información sea accesible y usable. Usualmente el trabajo de los Arquitectos de Información se concretan en la generación de un conjunto de materiales entregables en los que se plasma de forma efectiva la estructura del espacio de información, el diseño de la interacción del usuario con el sistema y el funcionamiento de la interfaz. Estos materiales tienen la finalidad de servir como material que se suministrará:

- A los clientes como resultado del proceso de diseño o reingeniería del software llevado a cabo durante la fase de análisis, definición y desarrollo del proyecto.
- A los diseñadores gráficos como material de base para la producción de maquetas gráficas sobre las que se basarán los maquetadores.
- A los ingenieros informáticos para desarrollar el back-end (programación) del software.

A continuación en la figura 1 se muestra unas series de preguntas y respuestas muy relacionadas con este término de Arquitectura de Información:



Figure 1. Preguntas de la Arquitectura de Información

1.1.1 Etapas de la Arquitectura de Información.

Este trabajo generalmente se realiza en tres etapas, la primera etapa encargada de la búsqueda de información del proyecto, los usuarios, los contenidos y los servicios, la segunda al análisis, organización y representación de la información, y la tercera al diseño del software como es el prototipo elemental de interfaz gráfica. Esto concretaría la línea completa de investigación-organización-diseño.



Figure 2. Etapas de la Arquitectura de Información

Para apoyar esto, y basándose principalmente en criterios empíricos, se describen una serie de procesos y tareas de arquitectura de información los cuales no son más que el diseño centrado en el usuario.

1.1.2 Procesos y tareas de la Arquitectura de Información.

El diseño centrado en el usuario se divide de manera independiente, en las tareas de:

- Definición de Objetivos
- Definición de audiencia
- Definición de contenido
- Definición del sistema de navegación
- Definición de estructura

Sin embargo todas estas tareas se realizaron de manera iterativa al mismo tiempo cuenta con dos procesos fundamentales los cuales son Análisis y Diseño Conceptual. Además este diseño tiene la ventaja para el arquitecto de información que el cliente del producto estará evaluando en cualquier momento del mismo. A continuación se ve representado el diseño centrado en el usuario. (3)

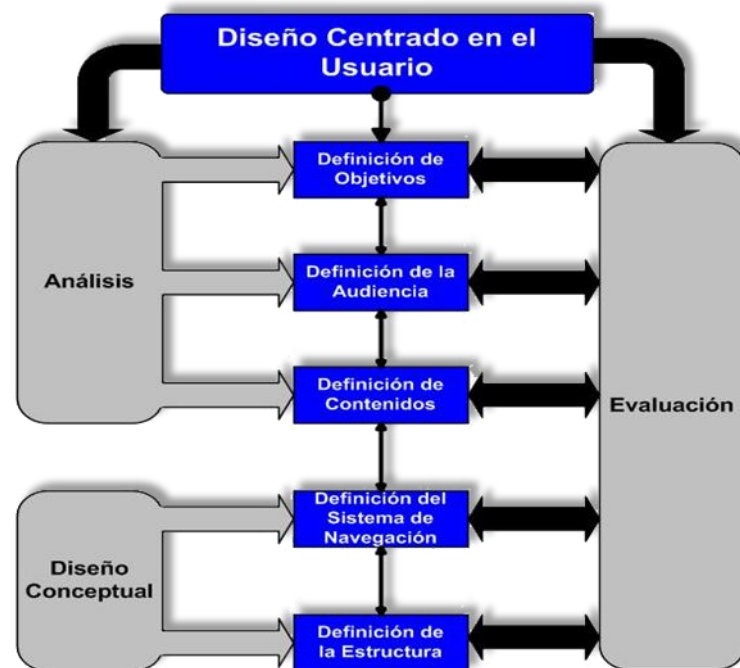


Figure 3. Procesos y tareas de la Arquitectura de la Información

1.1.3 Herramientas que asisten la Arquitectura de Información.

¿Qué son las herramientas CASE?

“Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero”.

Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. (4)

Toda disciplina requiere de una buena herramienta para su sostenibilidad absoluta y desarrollo armónico. Así como el diseñador posee programas como lo son Photoshop, Corel Draw, Flash, 3D; el desarrollador cuenta con HTML, Java Script, PHP, ASP, entre otras.

¿Con que cuenta el Arquitecto de Información para desarrollar la estructura del espacio de información en los diferentes sistemas, sitios web o software en general?

A continuación se hace referencia a un conjunto de herramientas que existen y que han sido desarrolladas para realizar mapas estructurales, pero que no presentan toda la madurez necesaria para asistir esta disciplina completamente. Por lo que se puede concluir que NO existe una herramienta CASE que asista todas las etapas, procesos y tareas de la arquitectura de información.

1.1.3.1 STAR TREE



Este producto permite la creación de nodos (vínculos) a través de una estructura móvil y ágil, anexa imágenes o iconos a cada uno de sus nodos.

Realizar cambios a la posición del mapa sea radial, superior, inferior, derecho o izquierdo.

Ventajas:

- Se aplica y edita bajo el programa Star Tree Studio© y su visualización se hace en el entorno Star Tree Viewer©.
- La fácil creación de nodos o desplazamiento de los mismos en otros nodos.
- Direccionar cada nodo a un URL externo, permitiendo presentaciones con links.
- Manejo de imágenes, iconos, colores, fuentes y tamaños.
- Pliegue y repliegue del mapa, además puede ser visualizado en su aplicación Star Tree Viewer©.
- Salvar presentaciones en HTML

Desventajas

- Límite en la creación de nodos.
- Para visualizarlo en HTML se necesita tener instalado Java Script.
- No es un programa de descarga gratis. (5)

1.1.3.2 CMAP TOOLS



Un programa que permite realizar mapas en posiciones libres, con líneas de unión conceptual. Excelente para asociar vínculos cruzados, enlaces a documentos relacionados del sitio e imágenes de fondo, líneas de unión que permiten formas diversas de conexión.

Ventajas:

- Libre posición de creación en el marco del programa.
- Sin límite de espacio horizontal y vertical en el diseño de la estructura.
- Anexo de imágenes de fondo a cada casilla, a su vez el cambio de forma, color y fuente.
- Compartir esquemas de Cmap Tools creados en la Web.
- Se puede almacenar en formato HTML.
- El programa se puede descargar de la página CmapTools.

Desventajas:

- Implementa una sola carpeta para almacenar imágenes, archivos y otros, la cual no es bien especificada en la ayuda del sistema.
- Si el mapa es muy extenso y amplio no es aconsejable publicarlo en la Web.
- Al guardar el desarrollo en HTML, es salvada una imagen. (6)

1.1.3.3 DIGIDOC MAP



Una aplicación elaborada con el propósito de trabajarse en HTML y Java Script. Destaca las líneas de unión para ser relacionadas con comentarios y vínculos a URL. Un buen modelo para los principiantes de la AI y sus primeros desarrollos, modelos de trabajo y proyectos.

Ventajas:

- La aplicación muestra tablas para ingresar datos básicos a las casillas, además de lista de recursos.
- Para los que se estén iniciando en organización de la información.
- Guarda en formato HTML, el cual puede ser modificado fácilmente.
- Cambio de posición visual, fuentes, colores, desplegar o replegar niveles.
- La aplicación es gratis y se maneja también vía Internet sin instalarla.

Desventajas:

- Se debe tener instalado Java Script.
- Su estructura recta lo hace ver muy rígido.
- Las tablas de información pueden ser complejas y se tropieza mucho en la edición.
- La presentación del mapa en despliegue derecho, no permite la continuidad del background. (7)

1.1.3.4 SPACE TREE



Esta aplicación fue creada por la Universidad de Meriland HCIL; primordialmente enfocada en la estructura taxonómica de proyectos científicos, como lo es el *Taxon Tree*. Una herramienta dinámica que corre sobre Java Script y guarda los datos en formato XML.

Ventajas:

- Movimiento dinámico, cambios de posición dentro del espacio.
- Bookmark para los vínculos importantes del mapa.
- Cambio de posicionamiento o nivel de los nodos.
- Repliegue, desplazamiento y marca de camino (línea de navegación) del link seleccionado.
- Gran capacidad en la creación de nodos.

Desventajas

- Se necesita instalar Java Script para correr el programa.
- Guarda los archivos en formato XML, otra presentación alterna lo hace en PNG.
- Si se desea hacer presentaciones sobre el Space Tree, se debe correr sobre el mismo programa. (8)

1.1.3.5 AXURE RP



Es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas Web. Cuenta con todo lo que se puede necesitar para crear los prototipos de forma más eficiente.

Es una herramienta muy utilizada por los profesionales que se encargan del diseño de la interfaz de usuario y funcionalidad. Es utilizada principalmente por

diseñadores, arquitectos de información, expertos en usabilidad y directores de producto.

Ventajas

- Flexibilidad y sencillez de uso
- Genera una documentación final.
- Crea el mapa conceptual de las páginas Web, así como su taxonomía.

Desventajas

- El documento de especificaciones es un DOCX. (9)

1.2 Definiciones básicas

1.2.1 ¿Qué es un requerimiento?

La definición más general alrededor de esta noción es la que brinda el Instituto de Ingeniería Electrónica y Eléctrica (IEEE):

- (1) “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”.
- (2) “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal”.
- (3) “Una representación documentada de una condición o capacidad documentada como las descritas en (1) y (2)”. (10)

Esta definición expresa la perspectiva clásica de los requerimientos como elementos de un producto, o criterios para acuerdos. Sin embargo, otros autores son más específicos frente a la relación de los requerimientos con relación al sistema que van a representar:

“... Los requerimientos son una especificación de lo que debe ser implementado. Estos son descripciones de cómo el sistema se debe comportar, de las propiedades y atributos del mismo. Deben ser una restricción del proceso de desarrollo del sistema...” (11)

Esta definición está muy ligada a lo que constituye el desarrollo de un sistema. También consta la definición establecida por RUP:

“... Una condición o capacidad que debe ser conformada por el sistema (RUP)...” (12)

1.2.2 Clasificación de requerimientos

Existen dos grandes clasificaciones de requerimientos que según su comportamiento en el software se clasifican en requerimientos funcionales y requerimientos no funcionales los cuales tienen que ser identificados y documentados. Los funcionales definen qué hace el sistema (describen todas las entradas y salidas), es decir, las funciones del sistema. Por su parte, los no-funcionales definen los atributos que le indican al sistema cómo realizar su trabajo (eficiencia, hardware, software, interfaces, usabilidad), los mismos a su vez se dividen en varias categorías.

Requerimientos funcionales:

- Requerimientos Lógicos: Son las actividades que debe realizar el software

Requerimientos no funcionales:

- Requerimientos Físicos: Son los requerimientos del comportamiento del sistema divididos en requerimientos de: comportamiento, fiabilidad, seguridad, hardware e interfaces de comunicación.
- Requerimiento de comportamiento: Son los que describen todos los aspectos de interfaces entre el software y su medio ambiente (hardware, otro software y humanos).
- Requerimiento de no comportamiento: Son los atributos de calidad de desarrollo. Incluyen cualquier restricción sobre atributos de construcción estática del sistema.
- Requerimientos de implantación: Indican como deberá ser implantado el sistema.
- Requerimientos de Rendimiento: Especifican cuál es el rendimiento mínimo aceptable para el sistema.
- Requerimientos de Uso: Especifican el máximo tiempo aceptable para demostrar el uso del sistema.
- Requerimientos de dominio: Se derivan del dominio del sistema más que de las necesidades específicas de los usuarios. Estándares de organización en el aspecto de las Interfaces de Usuario, con la base de datos. (2)

1.2.3 Importancia de la etapa de requerimientos

La etapa de Definición de Requerimientos es de gran importancia para el proceso de desarrollo de un software ya que establece y mantiene el acuerdo entre los clientes y el equipo de desarrollo de lo que

debe hacer el sistema, también para que los desarrolladores tengan una mejor comprensión de los requisitos que debe de cumplir el mismo, para definir los límites del software, para proporcionar una base, para planear el contenido técnico de iteraciones. Para definir una interfaz de usuario, centrándose en las necesidades y las metas de los usuarios. Para alcanzar estas metas, es importante, primero que todos, entender la definición y el alcance del problema que estamos intentando solucionar con este sistema.

1.2.4 La estructura de la Ingeniería de Requerimientos

Describe de manera detallada y precisa cada uno de los aspectos del ciclo de vida de un conjunto de requerimientos. Este proceso presenta dos grandes ramas: El Desarrollo de requerimientos, y la Administración de requerimientos. Véase Figura 4 y figura 5.

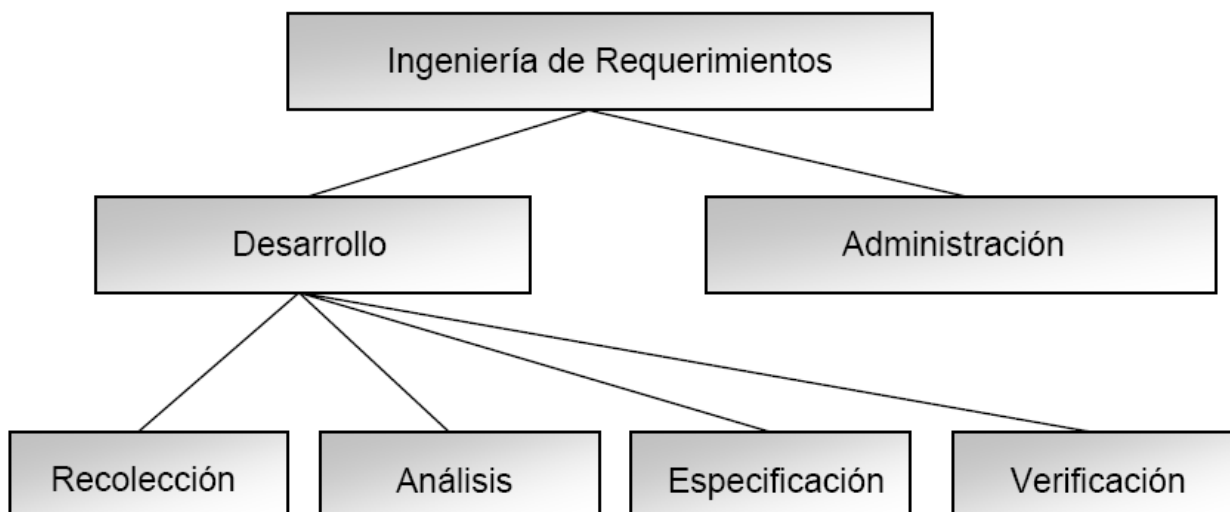


Figure 4. Estructura de la Ingeniería de Requerimientos.

- **Recolección:** Es el Proceso a través del cual los clientes y el desarrollador de un sistema de software; descubren, revisan, articulan, y entienden las necesidades de los usuarios del sistema y las restricciones que se dan sobre el software y el desarrollo del mismo.
- **Análisis:** Es el proceso de analizar las necesidades de los clientes y los usuarios para llegar a una definición de los requerimientos de software.

- Especificación: Consiste en el desarrollo de un documento que de manera clara y precisa contenga y especifique cada uno de los requerimientos del sistema de software.
- Verificación: Es el proceso de asegurar que la especificación de requerimientos de software sea acorde con los requerimientos del sistema, conforme a los estándares de documentación de la fase de requerimientos, y que a su vez este documento sea una base sólida para la arquitectura y el diseño. (13)

1.2.5 Proceso de la Ingeniería de Requerimientos

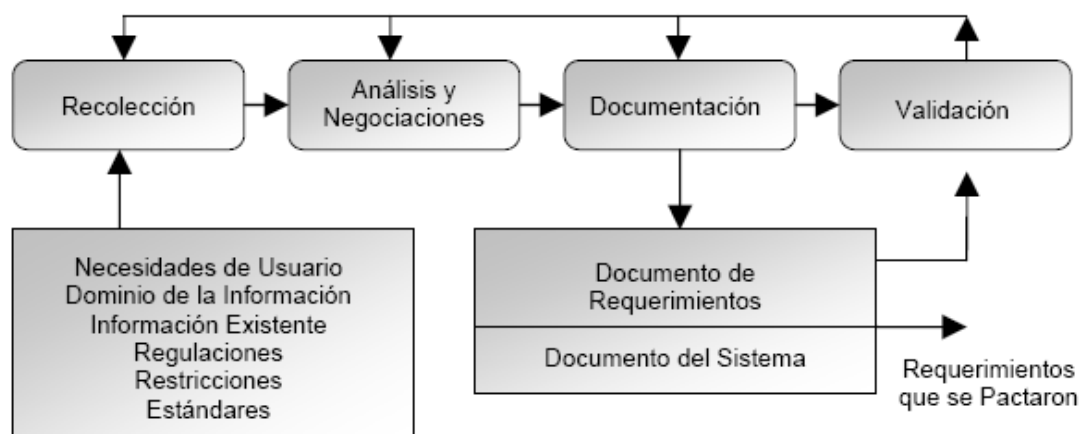


Figure 5. Proceso de la Ingeniería de Requerimientos.

1.3 Metodologías de desarrollo de software.

La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.

A continuación se detallan algunas de las metodologías de desarrollo de software más utilizadas internacionalmente.

1.3.1 Metodología Programación Extrema (XP).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera (el código es revisado y discutido mientras se escribe) es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajos distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrán añadir funcionalidades si es necesario.

1.3.1.1 Proceso de Captura de Requisitos con XP.

Las Historias de Usuario: Es la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [12]. Beck en su libro [2] presenta un ejemplo de ficha (customer story and task card) en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado de cosas por terminar y comentarios. A efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación (para no superar el tamaño de una iteración). Las historias de usuario son descompuestas en tareas de programación (task card) y asignadas a los programadores para ser implementadas durante una iteración. (14)

1.3.2 Metodología Scrum.

Scrum es una metodología para el desarrollo ágil de productos, expuesta por *Hiroataka Takeuchi* e *Ikujiro Nonaka*, en 1986 en el que ponen de manifiesto que:

El mercado competitivo de los productos tecnológicos, además de los conceptos básicos de calidad, coste y diferenciación, exige también rapidez y flexibilidad.

Los nuevos productos representan cada vez un porcentaje más importante en el volumen de negocio de las empresas.

El mercado exige ciclos de desarrollo más cortos.

Define además un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Esto resume Scrum, no es una metodología de análisis, ni de diseño, si no que es una forma de gestionar proyectos de software. (15)

1.3.3 Metodología Proceso Unificado de desarrollo de software (RUP).

El Proceso Racional Unificado (RUP) es un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software: figura 6.

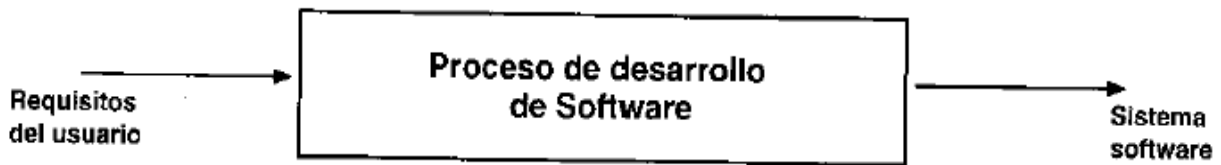


Figure 6. Proceso de desarrollo de software.

Constituye una metodología estándar para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Pone a nuestra disposición la forma disciplinada de asignar tareas y responsabilidades, además de implementar las mejores prácticas en la Ingeniería de Software como son: desarrollo iterativo, control de cambios, verificación de la calidad, administración de requisitos, entre otros.

RUP está centrado en la arquitectura y guiado por los casos de uso, incluye artefactos que son los productos tangibles del proceso como por ejemplo: el modelo de casos de uso, el código fuente, etc. Y roles como el papel que desempeña una persona en un determinado momento, teniendo en cuenta que una persona puede desempeñar distintos roles a lo largo del proceso.

1.3.3.1 Roles y artefactos.

Rol	Artefacto
Analista de Sistemas	Modelo de clases del dominio Listado de los Requisitos Actor Modelo de casos de uso
Especificador de casos de uso	Descripción de los casos de uso.

Diseñador de Interfaz de usuario	Prototipo de interfaz de usuario
----------------------------------	----------------------------------

Rol: Define un comportamiento o responsabilidades de un individuo o grupo de individuos trabajando en equipo, en el contexto de una organización de ingeniería de software.

Artefacto: Las actividades tienen artefactos de entrada y de salida. Un artefacto es un producto de trabajo en un proceso: los trabajadores utilizan artefactos para realizar actividades y producen artefactos como resultado de sus actividades. Los artefactos son responsabilidad de un único trabajador y promueven la idea de que toda pieza de información en el proceso debe ser responsabilidad de un rol específico. Un trabajador es el “propietario” de un artefacto, pero otros trabajadores pueden usarlo y tal vez modificarlo si tienen permiso para ello.

1.3.3.2 Flujo de trabajo de Requisitos.

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se construya. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

Los objetivos del flujo:

- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan las funcionalidades del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requisitos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

En este flujo de trabajo, y como parte de los requisitos de facilidad de uso, se diseña la interfaz gráfica de usuario. Para ello habitualmente se construyen prototipos de interfaz gráfica de usuario que se contrastan con el usuario final. (16)

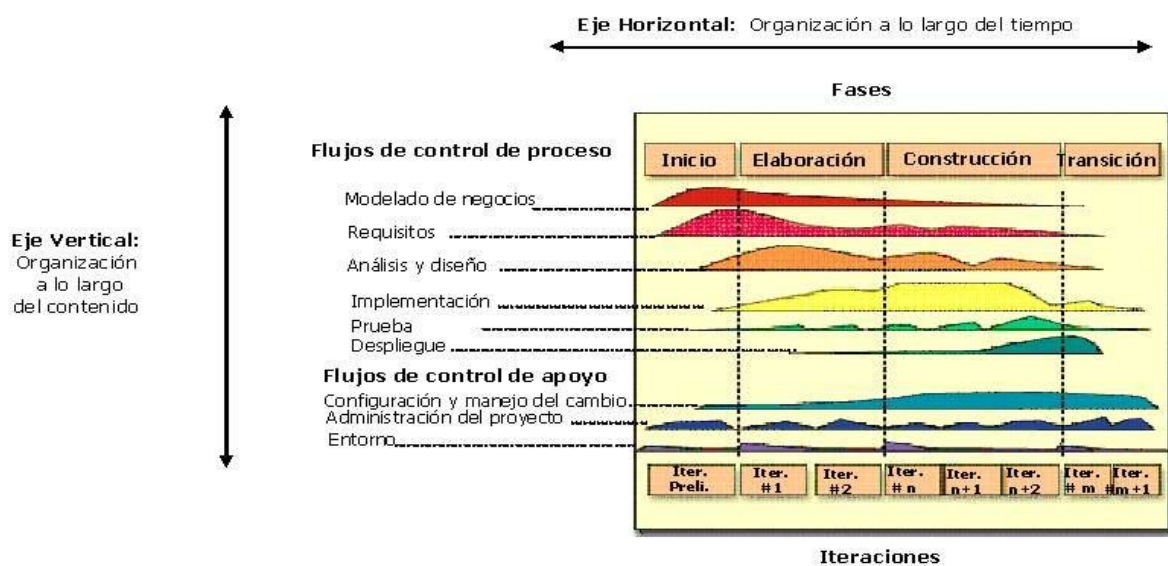


Figure 7. Proceso Unificado de desarrollo de software (RUP)

1.3.3.5 Estrategias y técnicas de captura de requisitos.

Desde el inicio del desarrollo de sistemas, los ingenieros nos hemos encontrado con un gran problema, la identificación de los requisitos del sistema. Esto es debido a que no es un proceso que pueda ser determinado matemáticamente. Es un proceso en el cual los datos son extraídos de las personas y estos datos pueden variar, dependiendo de la persona a la cual estemos consultando. Es por eso que la Ingeniería de Requisitos ha trabajado arduamente para tratar de desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y segura. A continuación se presentan algunas de las técnicas más usadas en esta disciplina.

Entrevistas.

En esta técnica, el encargado de definir los requerimientos realiza una serie de preguntas a personas o grupos, que pueden ser futuros o potenciales usuarios del sistema que se desea definir. El objetivo es reunir la mayor cantidad de información posible que permita al analista descubrir opiniones, sentimientos y experiencias generales, que no necesariamente tengan relación con el conocimiento de la potencial solución. Las preguntas pueden ser enfocadas a distintos elementos del sistema como por ejemplo usuario, proceso, producto, entre otros. Estas preguntas deben ser de alto nivel y realizadas al comienzo del proceso para así lograr la recaudación de aspectos globales del problema. El éxito del uso de esta técnica dependerá de la experiencia y sensibilidad del entrevistador, ya que además de preparar estratégicamente el curso de la entrevista, debe ser capaz de interpretar el significado de las respuestas.

Las preguntas suelen distinguirse en dos categorías: abiertas y cerradas. Las preguntas abiertas permiten que los encuestados respondan con su propia terminología. Generalmente estas son más reveladoras, ya que los interrogados no están limitados en sus respuestas. Son especialmente útiles en la etapa exploratoria de la investigación, cuando se busca penetrar en el pensamiento del encuestado.

A continuación se presentan algunos ejemplos de preguntas abiertas:

¿Cuál es la razón por la que quiere resolver este problema?

¿Cómo se resuelve el problema actualmente?

¿Cuál es el valor de una solución exitosa?

¿Quién usará el sistema que se va a construir?

¿Cómo desearía llevar a cabo esta actividad?

Por su parte, las preguntas cerradas predeterminan todas las posibles respuestas y el interrogado elige entre las opciones presentadas. Estas preguntas las podemos utilizar cuando estamos estableciendo el criterio de priorización de los casos de uso con el cliente. Para cada uno preguntamos si es a corto plazo, a futuro, o Indispensable. Como vemos, la respuesta está acotada a tres opciones. También podemos volver a utilizarla cuando tenemos que negociar algún requerimiento con el cliente. (13)

Brainstorm o Lluvia de Ideas.

Esta técnica busca el incentivo de la creatividad de los participantes del proyecto, creando una atmósfera de trabajo donde todos puedan aportar con ideas de la solución final sin que estas sean enjuiciadas ni criticadas hasta que ya nadie tenga una nueva idea que aportar. La lluvia de ideas se suele dividir en tres fases que se caracterizan por su relativa secuencia y los resultados que se obtienen. En la fase de Descubrimiento de Hechos, que consta con una etapa de ambientación especialmente importante para aquellos que no tienen experiencia en el tema, se determina y delimita el problema a tratar, para luego investigar documentación de experiencias anteriores, en caso de que exista. La segunda fase, Producción de Ideas, es la más importante, pues es aquí donde se busca producir una gran cantidad de ideas de todos los participantes, para la solución del problema. La tercera etapa, corresponde al Descubrimiento de Soluciones, y es aquella donde se seleccionan las ideas que parecen ser factibles, ponderándolas en caso de que sea necesario para generar la lista que más tarde debe ser analizada en detalle. Los participantes deben ser guiados por un director que se caracterice por una gran capacidad creativa. Este debe ser un moderador del encuentro, además de ser el encargado de definir claramente el problema a tratar. Por último, se desea que no existan diferencias jerárquicas entre los participantes, para que todos se sientan a gusto y sus ideas no sean inhibidas.

Prototipos.

Esta técnica comienza con la definición básica de los requerimientos generales por parte de los usuarios y clientes con el fin de identificar las características globales del sistema. Luego, se diseña y construye una maqueta del sistema que permita la representación de aquellos aspectos del software que serán visibles para el usuario. Esta maqueta es la que se conoce como prototipo y es el que debe ser evaluado por el cliente y el usuario con el fin de refinar los requerimientos del sistema. Se han identificado dos tipos de prototipos, los Prototipos Rápidos, que son aquellos que facilitan la evaluación de los requerimientos en la etapa previa al diseño general del sistema; y los Prototipos Evolutivos, a medida que se avanza en el ciclo de vida del proyecto, los prototipos sufren modificaciones y mejoras con el fin de llegar hasta el producto final.

Cuestionarios.

Esta técnica consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. Este cuestionario

será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista. (11)

A continuación en la tabla 1 se muestra las herramientas que se utilizan para las diferentes fases de la ingeniería de requerimiento, algunas de estas expuestas anteriormente.

Herramientas	Extracción	Análisis	Especificación	Validación
Entrevistas y cuestionarios	X			
Sistemas existentes	X	X		
Grabaciones de video y de audio	X	X		
Brainstorming (Lluvia de ideas)	X	X		
Arqueología de Documentos	X	X		
Observación	X			
Prototipo Throw Away (no funcional)	X	X	X	
Prototipo Evolutionary (funcional)	X		X	X
Análisis DOFA		X		
Cadena de Valor		X		
Modelo Conceptual		X	X	
Diagrama de Pescado	X	X	X	
Glosario	X	X	X	X
Diagrama de Actividad		X	X	
Casos de uso	X	X	X	X
Casa de Calidad o QFD				X
Checklist	X		X	

Tabla 1. Herramientas que se utilizan para las diferentes fases de la ingeniería de requerimientos.

1.4 Lenguaje de modelado.

“El Lenguaje Unificado de Modelado (Unified Modeling Language, UML), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software”. (16) De hecho UML es una parte esencial del proceso Unificado. Sus desarrollos fueron

paralelos. Como todo lenguaje, UML proporciona un vocabulario y unas reglas para permitir una comunicación. Éste particularmente, está compuesto por elementos que no son más que abstracciones que constituyen los bloques básicos de construcción, los cuales pueden unirse mediante relaciones para conformar los diagramas.

Ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. UML 2.0 detalla los artefactos en el sistema y puede modelar además de los sistemas de software, sistemas de hardware y organizaciones del mundo real.

El hecho de usar la notación UML para el intercambio de información de diseño e ideas, requiere de un software que ofrezca todas las herramientas necesarias, para hacer eficientemente este tipo de trabajo. Esta clase de software es a lo que denominamos herramienta de modelado y son usados para capturar, guardar, rechazar e integrar automáticamente información y diseño de documentación, labores difíciles de lograr con un simple procesador de texto.

1.5 Herramienta de modelado.

La herramienta escogida para el modelado de la aplicación fue el Visual Paradigm, debido fundamentalmente a que es multiplataforma.

Visual Paradigm es una herramientas CASE que da soporte al modelado visual con UML 2.0, ofreciendo distintas perspectivas del sistema. Es independiente de la plataforma y está dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, así como garantizar la calidad del producto final.

Posee entre sus principales características las siguientes:

Es profesional: brinda la posibilidad de crear un conjunto bastante amplio de artefactos utilizados con mucha frecuencia durante la confección de un Software. Todos estos, cumpliendo con el Standard UML 2.0.

Es amigable: puede ser utilizado en varios idiomas, sus componentes se encuentran relacionados, por lo que se hace muy fácil la creación de cualquier tipo de diagrama.

Brinda un número considerable de estereotipos: lo que permite un mayor entendimiento de los diagramas.

Facilidades para redactar especificaciones de casos de uso: es posible crear plantillas para las especificaciones de casos de uso y describirlos.

Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para plataformas como PHP.

Interoperabilidad con otras aplicaciones: brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose. Además permite importar y exportar XML y XMI.

Generación de código ORM: (Object–Relational Map) permite generar a partir de un Diagrama de Entidad Relación una Base de Datos Relacional y el código necesario para acceder a esta base de datos utilizando PHP entre otros lenguajes.

Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java. (Paradigm).

1.6 Patrones de Casos de Uso

Es importante aclarar que un Patrón de Caso de Uso (PCU) no describe un uso particular del sistema, sino que captura una técnica para hacer sostenible, reusable y entendible el modelo.

Uno de los PCU más utilizados es:

Patrón CRUD (Creating, Reading, Updating and Deleting): este patrón posee un caso de uso, llamado “Información CRUD” o “Gestionar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de información de un tipo determinado; ya sea crearla, leerla, actualizarla y eliminarla. Es importante destacar que este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.

Actores múltiples: Rol común (Multiple Actors: Common Role): dos actores interpretan el mismo papel en determinado caso de uso. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol. Este patrón debe aplicarse cuando, desde el punto de

vista de un caso de uso hay solo una entidad externa interactuando con cada instancia del caso de uso.

ADICIÓN: Extensión Concreta. En cuanto a la extensión, el caso de uso extendido es concreto por lo que puede ser instanciado directamente o realizarse a partir del caso de uso base. El caso de uso base puede ser abstracto o concreto.

A medida que avanzan las iteraciones de construcción del software nuevas funcionalidades se adicionan a los casos de uso ya definidos; asimismo cuando se realizan nuevas versiones de una aplicación, se enriquecen las funcionalidades descritas.

Este fenómeno puede traer grandes imprevistos tanto para el equipo de desarrollo como para el cliente cuando se intenta modificar lo que ya está aprobado o implementado. La simplicidad de las especificaciones se ve afectada al incrementar secuencias de eventos entre las originales, modificando por mucho la estructura de los artefactos, dificultando la identificación de las nuevas funcionalidades para los trabajadores del proceso, el modelo y el mantenimiento de los componentes.

Los clientes por su parte pueden llegar a perderse entre descripciones de lo viejo y lo nuevo, tomando más tiempo la asimilación del cambio conceptual de las funcionalidades preconcebidas.

Conclusiones

Como resultado del estudio realizado se estudio las herramientas existentes que aportan conocimiento e interés para la AI, pero sin embargo no existe actualmente una herramienta CASE que asista a la realización de la Arquitectura de Información completamente. De las mencionadas el Axure RP tiene una gran aproximación a la realización de la misma pero aun no se adapta al proceso como se realiza actualmente en los proyectos productivos de la Universidad.

También se estudió los conceptos de requerimientos y sus clasificaciones, así como la definición fundamental de la Ingeniería de Requerimientos, donde se concluye que se divide en dos grande bloques, Desarrollo y Administración de Requisitos.

De los cuales, el presente trabajo solo se centrará en el Desarrollo de Requisitos, el cual está compuesto por 4 tareas fundamentales:

- Recolección
- Análisis

- Especificación
- Verificación

Se realizó el estudio de tres metodologías de desarrollo que más se utilizan a nivel mundial, XP, Scrum y RUP y como estas contemplan la Ingeniería de Requisitos.

De estas se decidió aplicar el Flujo de Trabajo de Requisitos de la Metodología RUP, aparejado al uso de técnicas de captura de requisitos para apoyar dicho proceso, de las técnicas mencionadas se utilizaron las entrevista, cuestionario, lluvia de ideas y prototipo. Utilizando como herramienta de modelado Visual Paradigm.

Por lo cual en el siguiente Capítulo se realizará una propuesta de Herramienta CASE ajustada a las necesidades reales de los Arquitectos de Información de los Proyectos Productivos en la Universidad.



CAPÍTULO 2

ANÁLISIS DE LA HERRAMIENTA CASE AISOFT

Antes de enfrentarse al desarrollo de una herramienta informática, es necesario tener conocimiento del problema, es decir, de la situación actual que enfrenta la organización. Comenzar a implementar una herramienta informática sin conocer sus características principales, puede traer consigo que el producto final no sea el esperado por clientes ni usuarios. De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. RUP propone en su fase de inicio la realización de un conjunto de estos modelos. Por lo que en este capítulo se realiza la propuesta de solución para la herramienta CASE, aplicando el flujo de trabajo de levantamiento de requisitos de la metodología RUP generándose sus artefactos. Además de dos métodos de validación para garantizar que la propuesta cumpla con las necesidades reales de los clientes.

2.1 Modelo del dominio.

2.1.1 ¿Qué es un Modelo de Dominio?

Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio, para esto se realiza un modelo de dominio (o Modelo Conceptual) que es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software.

Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML con un Diagrama de Clases en los que se muestra:

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos.

Cualquiera que sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo de dominio contempla cualquiera de las soluciones analizadas. El modelo de dominio es global, es decir se realiza para todos los casos de uso y no para uno en particular.

Características:

- No hay cronología.
- No se diferencia entre dentro y fuera del sistema.
- Es global, no por caso de uso.
- No es completo: esquemático, las asociaciones están resumidas.
- No debe ser demasiado extenso debido a que esto podría producir confusión, los grados de refinamiento se deben ir logrando posteriormente.

Dado todo lo anterior planteado irrumpe la interrogante tratada en el próximo tópico.

2.1.2 ¿Por qué hacer uso de un Modelo de Dominio para el caso de estudio tratado?

El negocio que se estudió, tiene muy bajo nivel de estructuración, con soluciones muy diversas y dispersas, aunque todas llevan el mismo propósito, por lo que no se tiene la posibilidad de identificar los procesos de negocio, las personas que lo inician, quienes son los beneficiados en cada uno de esos procesos, pero sin embargo si podemos identificar al menos eventos, transiciones, “cosas” involucradas a este entorno que no se logra determinar con facilidad y a eso se le llama concepto (objeto) y por tanto lo que se trata de hacer es un modelo de dominio, simplemente reunir todos esos conceptos que están en un entorno de trabajo en un diagrama (diagrama de modelo de dominio).

Considerando tales escenarios y según RUP, cuando esto sucede, se recomienda desarrollar un modelo del dominio, el cual tiene como alternativa al modelo de casos de usos del negocio, graficadas en la Figura 8.

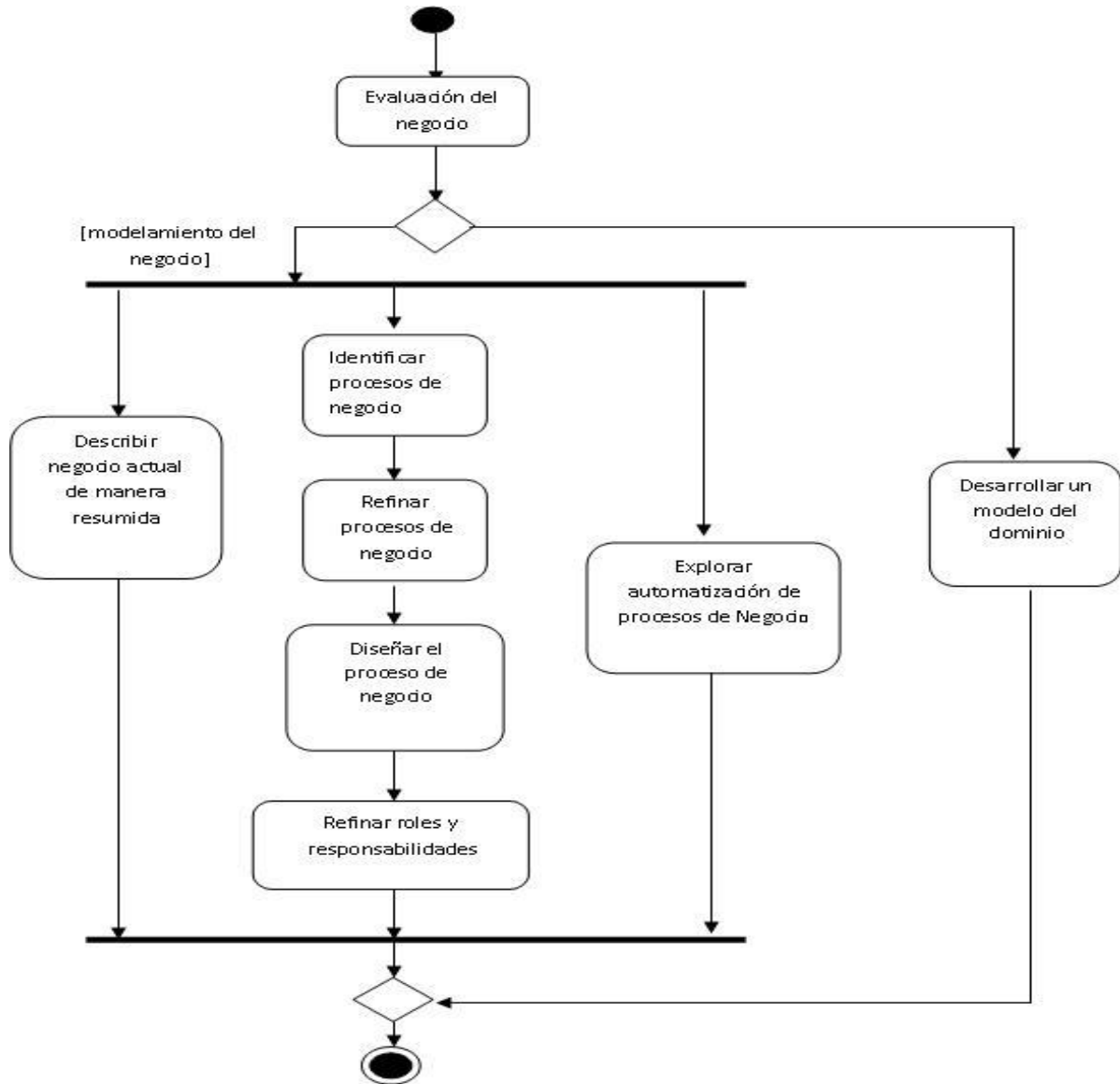


Figure 8. Flujo de trabajo para el modelado del negocio según RUP.

2.1.3 Descripción del Dominio.

El proceso de desarrollo de software de la uci es un proceso novedoso y en constante dinamismo el cual está haciendo adaptado a partir de modelos de procesos existentes y buenas prácticas internacionales, dentro de este proceso se encuentra la disciplina de arquitectura de información, la

cual surgió por la necesidad de organizar estructuralmente la información para una mayor satisfacción de los usuarios finales de los sistemas desarrollados.

La Arquitectura de Información es la ciencia de organizar y rotular sitios web, intranets, comunidades en línea y software para promover la usabilidad y la accesibilidad. Esta disciplina estará cumpliendo sus objetivos cuando un usuario entre por primera vez a un sistema y pueda entender y encontrar de forma rápida y sin esfuerzo la información ofrecida. También está compuesta por los siguientes elementos con los cuales es posible conseguir las metas de organización y visibilidad de los contenidos. Cada una de ellas debe ser investigada, desarrollada y documentada adecuadamente:

- Análisis de Homologo.
- Informe Diagnóstico.
- Definición de Objetivos del Sitio.
- Definición de Audiencia.
- Definición de Contenidos del Sitio.
- Definición de la Estructura del Sitio.
- Definición de los Sistemas de Navegación.

2.1.3.1 Informe Diagnóstico

Se hace una evaluación del producto actual donde se tienen en cuenta los aspectos positivos y negativos del diseño gráfico, estructura y organización de la información, presentación del contenido, accesibilidad y usabilidad, así como el impacto que tienen estos elementos sobre el nivel de audiencia del portal.

2.1.3.2 Definición de Objetivos del Sitio

A través de esta etapa se busca definir cuáles serán los objetivos centrales que deberá tener el sistema y establecer la forma de cumplirlos.

Se recomienda comenzar por generar un objetivo central y luego definir varios objetivos secundarios; es importante que estos objetivos sean explicados adecuadamente al comienzo del desarrollo, con el fin de que todo el equipo tenga claro el horizonte que debe tener el proyecto.

Para generar objetivos que sean válidos y comprensibles, se aconseja escribirlos a través de frases que se inicien con verbos que expresen las acciones a realizar. De cada uno de estos objetivos se pueden desprender tareas concretas que permitan avanzar en el cumplimiento de ellos.

2.1.3.3 Definición de Objetivos del Sitio

A través de esta etapa se busca definir cuáles serán los objetivos centrales que deberá tener el sistema y establecer la forma de cumplirlos.

Se recomienda comenzar por generar un objetivo central y luego definir varios objetivos secundarios; es importante que estos objetivos sean explicitados adecuadamente al comienzo del desarrollo, con el fin de que todo el equipo tenga claro el horizonte que debe tener el proyecto.

Para generar objetivos que sean válidos y comprensibles, se aconseja escribirlos a través de frases que se inicien con verbos que expresen las acciones a realizar. De cada uno de estos objetivos se pueden desprender tareas concretas que permitan avanzar en el cumplimiento de ellos.

2.1.3.4 Definición de Audiencia

Cuando se ha terminado adecuadamente la etapa anterior con la generación de los objetivos del sistema, se debe dar un siguiente paso, que es el de determinar las principales audiencias (públicos) hacia las cuales se orientará el sistema. Hay varias consideraciones que tener en cuenta, al definir qué tipos de audiencia se atenderán desde las diferentes pantallas del mismo, para esto existen definiciones de audiencia:

- Por capacidad física
- Por capacidad técnica
- Por conocimiento de la institución
- Por necesidades de información
- Por ubicación geográfica

Las que, sin importar las características del sistema, siempre deberán estar presentes, y por lo mismo, deben traducirse en acciones para poder atenderlas adecuadamente.

2.1.3.5 Definición de Contenidos del Sitio

Una vez que se han identificado los objetivos del sitio y la audiencia, se debe proceder a hacer las definiciones más concretas que permitan decidir qué contenidos son los que van a tener el sistema que se desarrolle. Para identificar contenidos, se deben utilizar como insumo los materiales que se hayan obtenido en la etapa de identificación de Objetivos y de Audiencias, ya que en ambos la búsqueda giró en torno a las necesidades que tenían los usuarios del sistema.

Por lo mismo, es muy relevante que tanto los objetivos como la audiencia del sitio se hayan definido muy bien en forma previa, porque de lo contrario no habrá posibilidad de atender al usuario de manera adecuada. Para lograr esta etapa de manera eficiente se siguen los siguientes pasos:

- Agrupar y etiquetar el contenido: Que consiste en poner en práctica las técnicas que permiten ordenar los contenidos, agrupándolos en conjuntos coherentes y dándoles nombres que los identifiquen.
- Análisis de sistemas Similares: Que no es más que la búsqueda de otros sistemas que sean similares al nuestro, con el fin de revisar de qué manera han resuelto los mismos problemas que deberemos atender. Es importante que se haga una pauta previa de comparación, con el fin de saber de antemano cuáles son los parámetros que se medirán y gracias a eso optimizar la revisión que se haga.

2.1.3.6 Definición de la Estructura del Sitio

Hecho el trabajo de identificación de contenidos reseñado en las etapas anteriores de este documento, se debe avanzar hacia las definiciones relacionadas con la forma que tendrá el sistema que se está desarrollando. Ello implicará trabajar en áreas concretas, a través de las cuales se definirá la estructura del mismo, el árbol de contenidos y los sistemas de navegación que se ofrecerá a los usuarios para que avancen a través de sus contenidos. Para lograr estos objetivo realizaremos la siguiente tarea constituida por tres etapas:

- La creación de la Estructura: Se refiere al proceso de identificar la forma que tendrá el sistema. En este sentido es muy importante hacer una diferencia entre estructura y diseño.
- Mapas Permanentes del Sitio: Es el proceso de crear un árbol de contenido en el que se muestre de manera práctica cuántas secciones tendrá el sistema y cuántos niveles habrá dentro de cada uno. Cuando se usa la idea de crear un árbol, se refiere exactamente a generar un diagrama que cuente con un tronco, ramas y hojas, para mostrar las zonas principales, secundarias y contenidos finales que se irán incorporando.

- Definición de los Sistemas de Navegación: Consiste en generar los sistemas de acceso a dichos contenidos. A través de estos, los usuarios podrán avanzar por sus diferentes áreas, sin perderse. En la generación de dichos sistemas se debe atender a dos elementos que serán muy importantes, el elemento Textual y el Contextual.

2.1.3.7 Definición de los Sistemas de Navegación

Cuando se ha terminado el trabajo en la estructura, ya se cuenta con los insumos suficientes como para avanzar hacia la generación del diseño visual de las pantallas del producto. Para ello la recomendación es trabajar en tres etapas sucesivas e incrementales, como son:

- Diseño de las Estructuras de Páginas: Este considera la generación de dibujos sólo lineales que describen los componentes de cada una de las pantallas del producto, con el objetivo de verificar la ubicación de cada uno de ellos. El ideal es que se dibujen diagramas con todas las pantallas que tendrá el mismo, ya que de esta manera será posible que diseñadores y desarrolladores tengan un documento concreto de trabajo, a través del cual resuelvan todas las dudas de los elementos que componen esta página.
- Bocetos de Diseño: Consiste en la generación de dibujos digitales acabados de la forma que tendrán las páginas principales del producto que se desarrolla. Para desarrollar los elementos gráficos se utilizan como insumos los dibujos de estructura que se han generado anteriormente.
- Maqueta Web: Consiste en generar todo el producto en tecnología HTML utilizando imágenes y contenidos reales (prototipo no funcional). las pantallas que se generen permitirán que el diseñador genere las plantillas de trabajo y el desarrollador de software las utilice como elementos para introducir la programación que sea necesaria para la creación de la herramienta informática.

2.1.4 Diagrama de clases del modelo del dominio

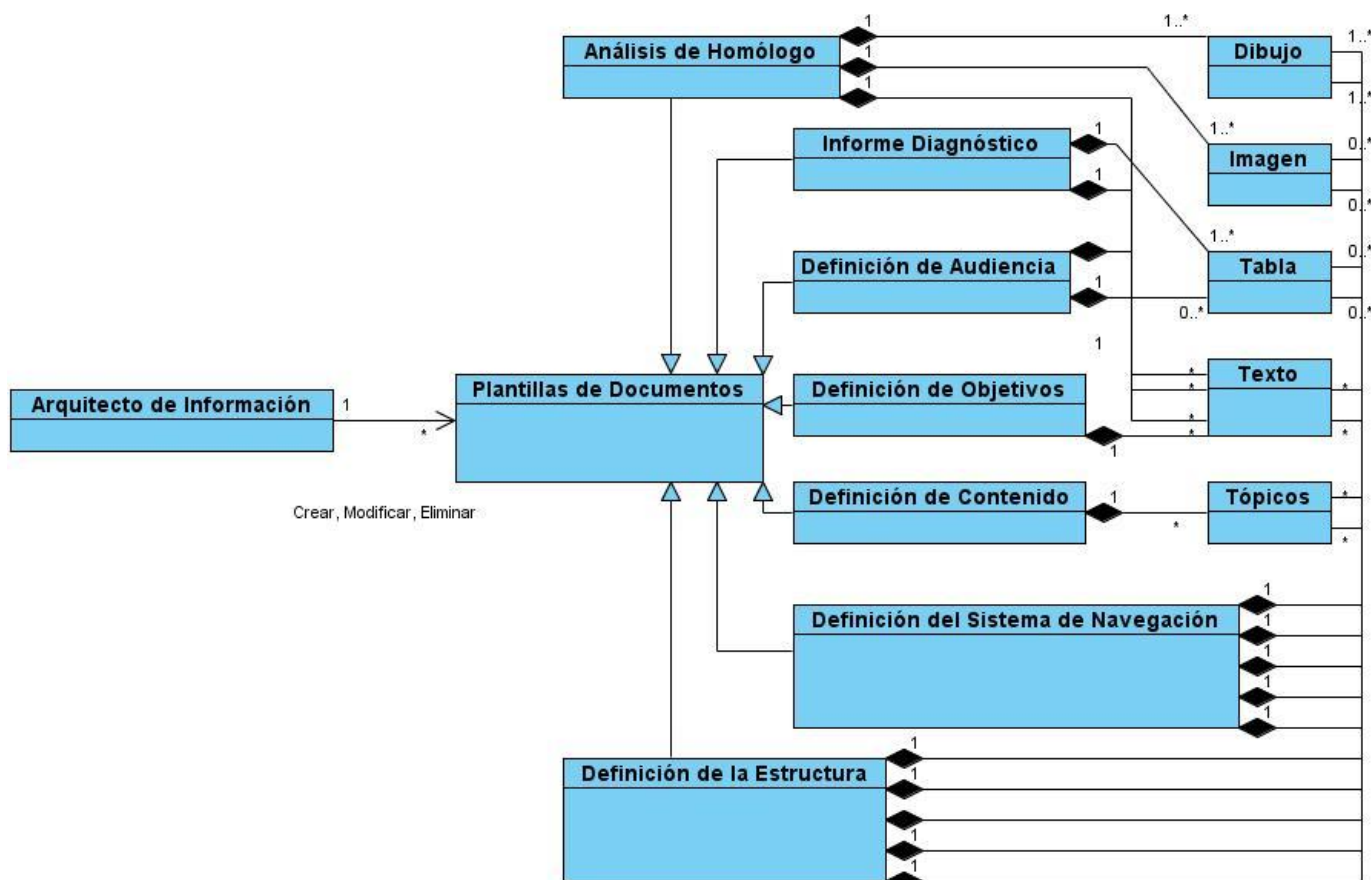


Figure 9. Modelo del Dominio

2.1.4.1 Definición de las clases del modelo del dominio

Incluimos este glosario para definir términos y describir los vocablos más importantes que se manejan en el dominio del problema tratado, para una mayor comprensión de las entidades del diagrama de modelo de dominio mostrado en la figura 6. Ya que hay que identificar todos los conceptos que se utilizarán en el mismo. A continuación una pequeña descripción de los mismos:

- Arquitecto de Información: Persona calificada para trabajar la Arquitectura de Información, que puede modificar, crear o eliminar las plantillas que componen la Arquitectura de Información.
- Plantillas Documentos: Se denominara plantilla a cada uno de los documentos entregables que conforman la Arquitectura de Información.

- Componentes: Son todos los elementos (imágenes, tablas tópico, etc.) con que cuenta el Arquitecto de Información para realizar las plantillas, donde cada una de ellas tendrá sus componentes en específico.

2.2 Levantamiento de Requisitos.

2.2.1 Requisitos Funcionales

Después de conocer y predefinir los objetivos y conceptos que encierran a nuestra herramienta informática debemos proceder con la interrogante ¿Qué debe hacer dicha herramienta para que se cumplan estos objetivos? Para darle respuesta a la pregunta se enumeran a través de requisitos funcionales las acciones que la herramienta informática deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario y las acciones ocultas que debe realizar la herramienta. Teniendo en cuenta los objetivos trazados las funcionalidades de la herramienta serán:

Ref. #	Función
RF-01	Autenticar usuario
RF-02	Crear nuevo proyecto
RF-03	Mostrar datos generales del proyecto
RF-04	Modificar datos generales del proyecto
RF-05	Eliminar proyecto
RF-06	Salvar proyecto en la maquina local
RF-07	Abrir proyecto local
RF-08	Abrir plantilla local
RF-09	Conectar al servidor
RF-10	Salvar proyecto en el servidor
RF-11	Actualizar proyecto del servidor
RF-12	Eliminar proyecto en el servidor
RF-13	Descargar proyecto del servidor
RF-14	Adicionar plantilla al proyecto
RF-15	Mostrar listado de plantillas

RF-16	Mostrar contenido de la plantilla
RF-17	Salvar plantilla en la maquina local
RF-18	Eliminar plantilla local
RF-19	Modificar plantilla local
RF-20	Crear nuevo usuario
RF-21	Modificar datos del usuario
RF-22	Eliminar usuario
RF-23	Mostrar datos del usuario
RF-24	Insertar imagen a la plantilla
RF-25	Eliminar imagen de la plantilla
RF-26	Insertar tabla a la plantilla
RF-27	Eliminar tabla de la plantilla
RF-28	Modificar Tabla de la plantilla
RF-29	Insertar tópico
RF-30	Eliminar tópico
RF-31	Modificar tópico
RF-32	Insertar relación
RF-33	Eliminar relación
RF-34	Mostrar barra de herramienta
RF-35	Ocultar barra de herramienta
RF-36	Mostrar árbol de proyectos
RF-37	Ocultar árbol de proyectos
RF-38	Mostrar área de botones
RF-39	Ocultar área de botones
RF-40	Cambiar Formato de texto a negrita
RF-41	Cambiar Formato de texto a Cursiva
RF-42	Cambiar Formato de texto a Subrayado
RF-43	Aplicar Formato de texto Viñetas
RF-44	Aplicar Formato de texto Numeración
RF-45	Realizar acción de copiar texto /elemento
RF-46	Realizar acción de cortar texto /elemento
RF-47	Realizar acción de pegar texto /elemento

RF-48	Realizar acción de rehacer
RF-49	Realizar acción de deshacer
RF-50	Mostrar proyectos registrados
RF-51	Mostrar usuarios registrados
RF-52	Visualizar proyecto más activo de la semana
RF-53	Visualizar últimos proyectos registrados
RF-54	Consultar registro de traza
RF-55	Enviar por correo
RF-56	Exportar a PDF
RF-57	Buscar contenido local
RF-58	Eliminar secciones a la plantilla
RF-59	Adicionar secciones a la plantilla
RF-60	Definir componentes de la plantilla
RF-61	Cambiar Idioma
RF-62	Buscar contenido en el servidor
RF-63	Enviar correo al administrador
RF-64	Verificar actualizaciones de la herramienta
RF-65	Verificar actualizaciones de las plantillas

2.2.2 Requisitos no Funcionales

Los requisitos no funcionales se desarrollarán sobre áreas clave de calidad, cada una de las cuales presenta características que permitirán ejecutar con eficiencia y eficacia las funcionalidades planteadas para el sistema que de una u otra forma puedan limitarlo, son propiedades o cualidades que el producto debe tener, así como criterios que pueden usarse para juzgar la operación del mismo.

Requisitos de Apariencia o Interfaz Extrema

- Un diseño simple, permitiendo que no sea necesario muchas instrucciones para utilizar el sistema.

Requisitos de Software

Requisitos óptimos del software del servidor

- Sistema Operativo recomendado: Linux (Debian, Ubuntu)
- Servidor de Base Datos: PostgreSQL 8.2.9 o superior
- Máquina virtual de Java versión 1.6

Requisitos óptimos del software del cliente

- Sistema Operativo recomendado: Linux (Debian, Ubuntu) o Windows XP Professional
- Máquina virtual de Java versión 1.6

Requisitos de Usabilidad

- Todos los textos y mensajes en pantalla aparecen en idioma español.
- Su funcionamiento es intuitivo y requiere de información mínima.

Requisitos de Seguridad

- Debe quedar registrada una traza de todas las acciones realizada por los usuarios.
- Los permisos de los usuarios en el sistema se establecerán desde el módulo de administración.

Requisitos de Hardware

Requisitos óptimos del hardware del servidor

- Procesador de tipo Pentium IV a 1.8 gigahercios (GHz) o superior.
- 1 gigabytes (GB) de memoria RAM o más.
- Se requiere de un disco duro de 80 GB o superior.
- Periféricos: tarjeta de red 100 megabytes (MB).

Requisitos óptimos del hardware del cliente

- Procesador de tipo Pentium IV a 1.8 gigahercios (GHz) o superior.
- 256 megabytes (MB) de memoria RAM o más.
- Se requiere de un disco duro de 60 GB o superior.
- Periféricos: tarjeta de red 100 megabytes (MB).

Requisitos Restricciones de diseño

- La arquitectura de software utilizada para el diseño e implantación de la herramienta es una arquitectura en capas.
- Bajo el modelo de desarrollo de la plataforma Java 2 Enterprise Edition (J2EE) que define un estándar para el desarrollo de aplicaciones multicapas basándolas en componentes modulares y estandarizados.

De forma general las tecnologías utilizadas por capas son las siguientes:

Capa de presentación:

Para las operaciones de ingreso y visualización de datos: el framework Swing utilizado para la presentación GUI (Graphical User Interface) en aplicaciones Desktop, para los reportes el framework JasperReports, utilizado para la generación de reportes en formatos PDF, XLS, HTML, CSV y XML, para la generación de Gráficos, el framework JFreeChart que se integra al JasperReports.

Capa de Negocio:

Para la implementación de la lógica del negocio el framework Spring, para la Seguridad Autenticación y Autorización el framework JAAS (Java Authorization and Authentication), para el envío de correos JavaMail de Sun y para trazabilidad de los usuarios el framework Log4j.

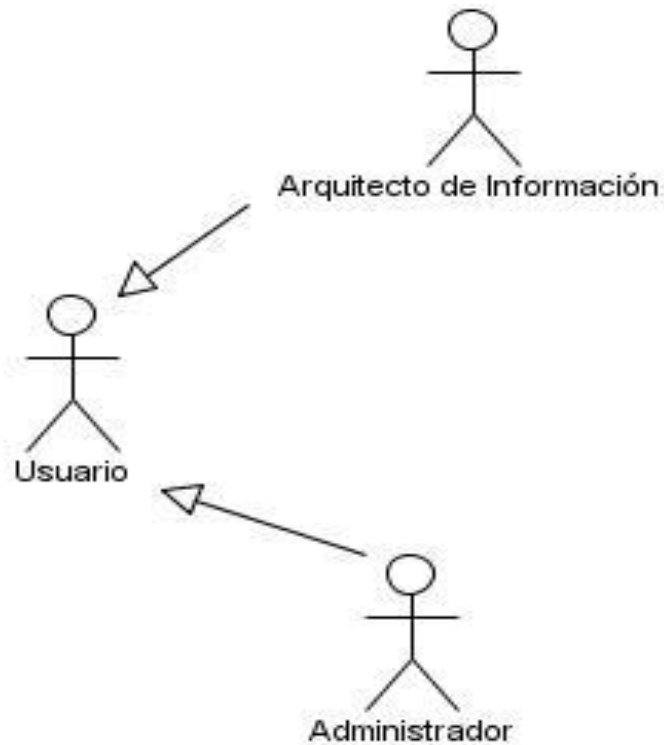
Capa de Acceso a Datos: Persistencia en la BD relacional y Transacciones de Aplicación el framework Hibernate. IDE: Netbeans

- Gestor de Base de Datos: PostgreSQL

2.3 Modelo de Casos de Uso del Sistema

La modelación del sistema describe la funcionalidad propuesta del mismo y constituye una representación de los requisitos funcionales anteriormente encontrados, donde se define la interrelación usuario-sistema como base a sus expectativas y los posibles resultados de valor que les pueda brindar a sus actores.

2.3.1 Diagrama de actores del sistema.



2.3.2 Actores del sistema

Actor	Justificación
Arquitecto de información	Es el que se beneficiará con las funcionalidades que brinda la herramienta, ya que esta le facilitará su trabajo como arquitecto de información ayudándole a generar sus plantillas.
Administrador	Es la persona que configura las plantillas de la herramienta con privilegio de modificar cualquier estructura de las mismas, es el encargado de realizar todo el proceso administrativo de la herramienta, así como la gestión de los usuarios.

Usuario Es quien representa al arquitecto de información y al administrador cuando utilizan la herramienta con la misma funcionalidad.

2.3.3 Diagrama de casos de uso del sistema.

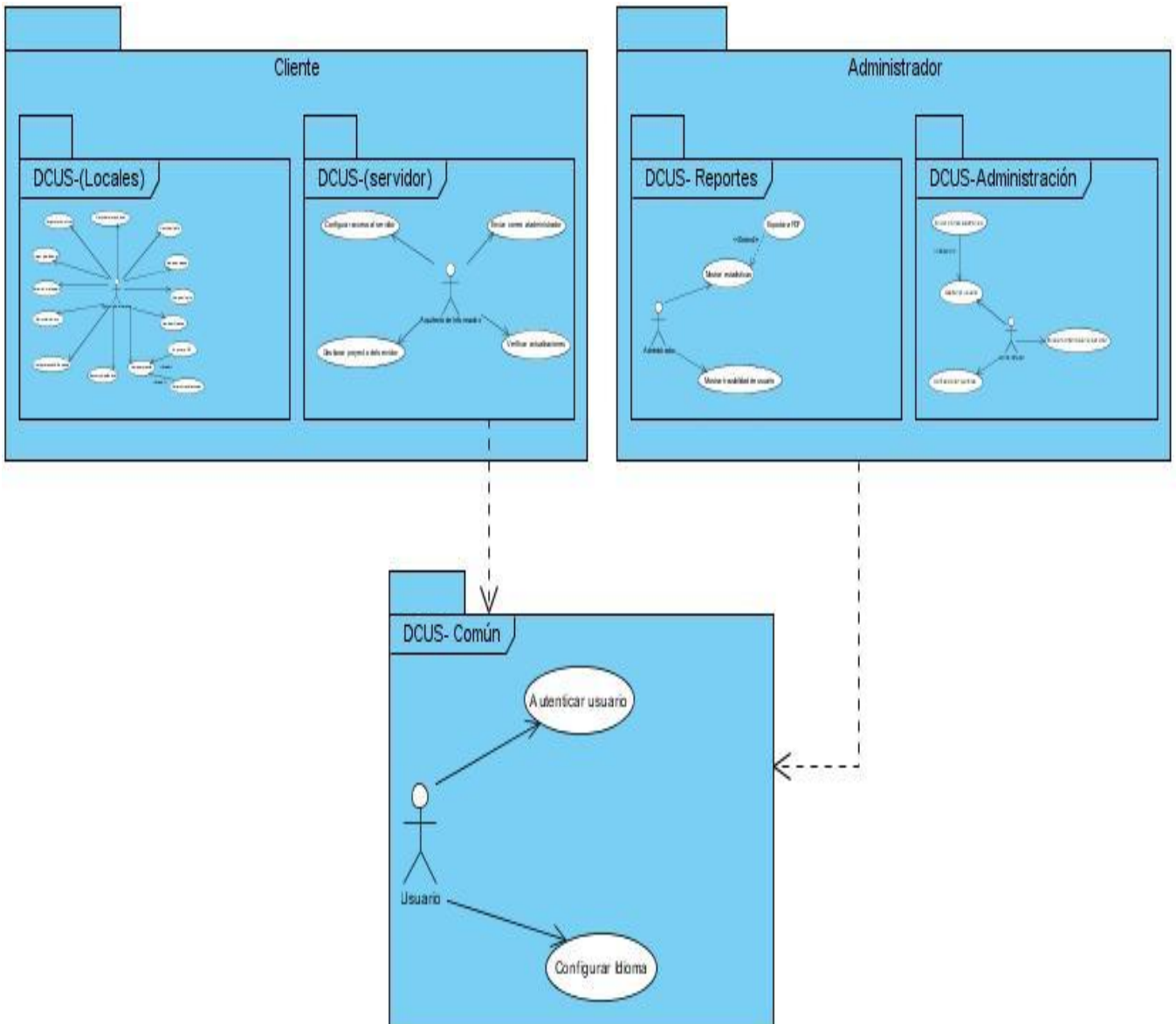
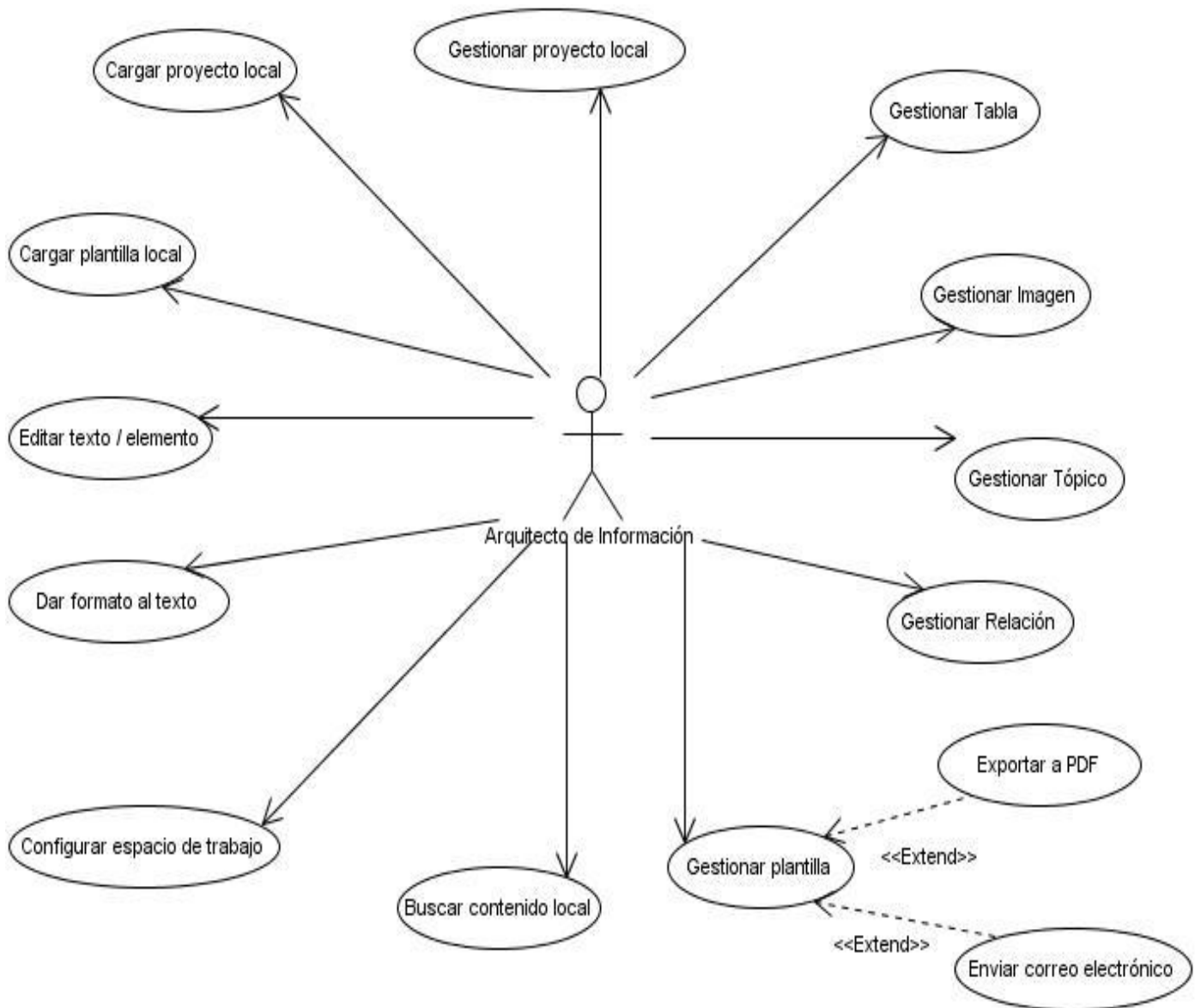
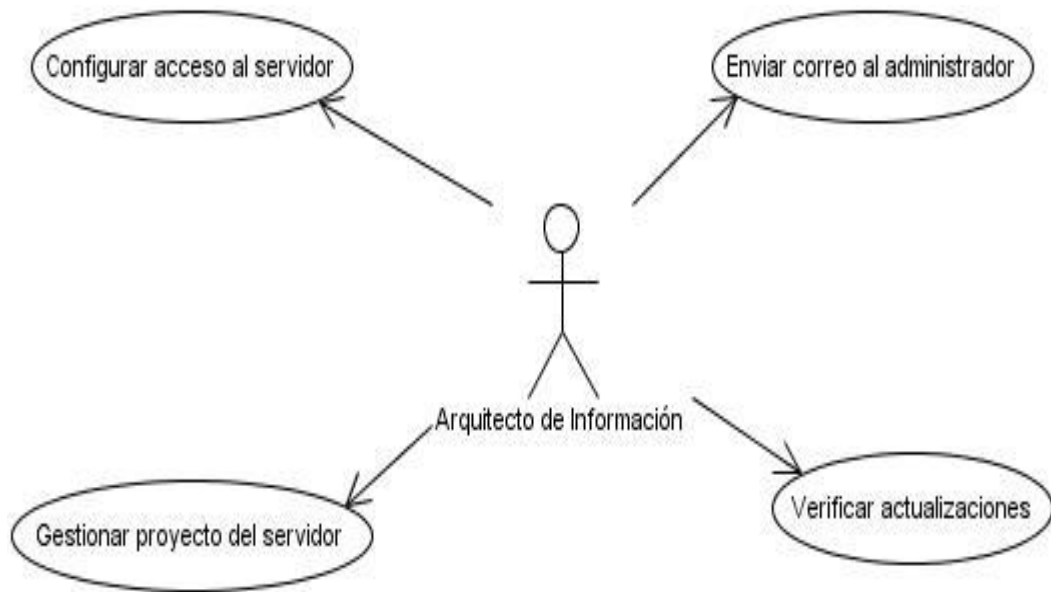


Figure 10. Diagrama de Casos de Uso del Sistema por paquete.

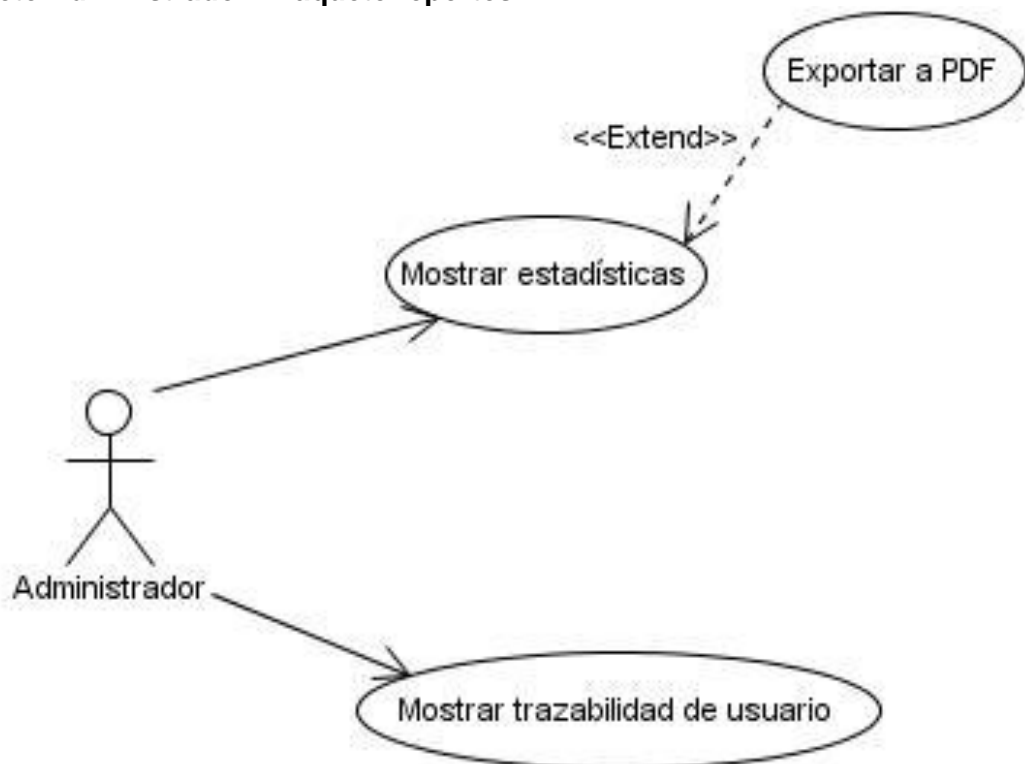
DCUS Paquete Cliente - Paquete locales



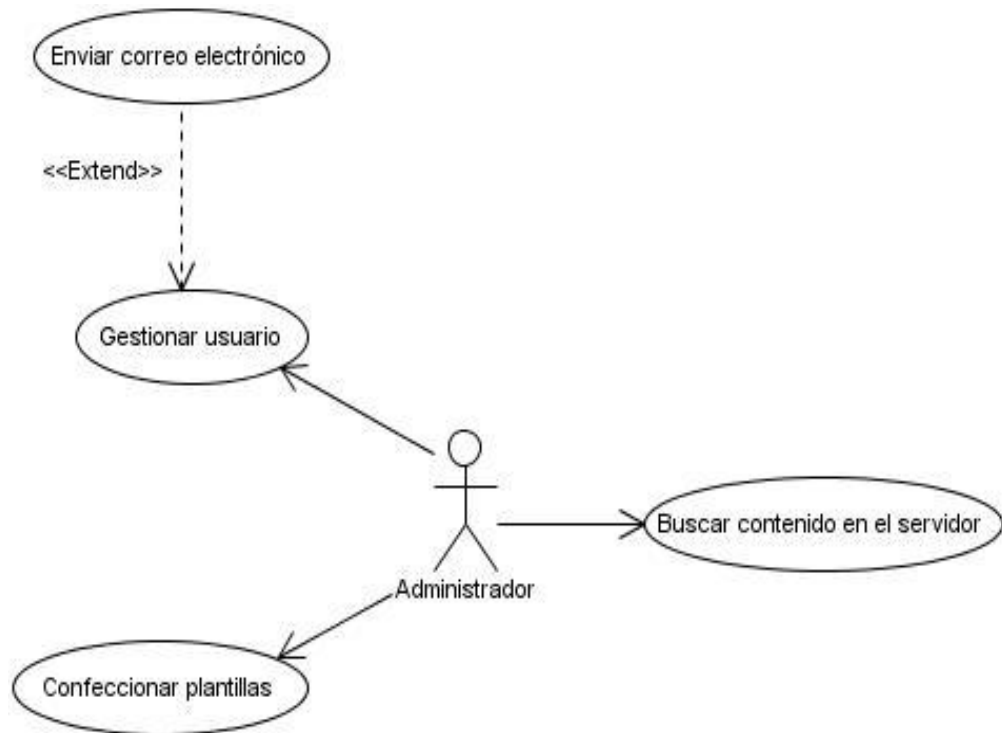
DCUS Paquete Cliente - Paquete servidor



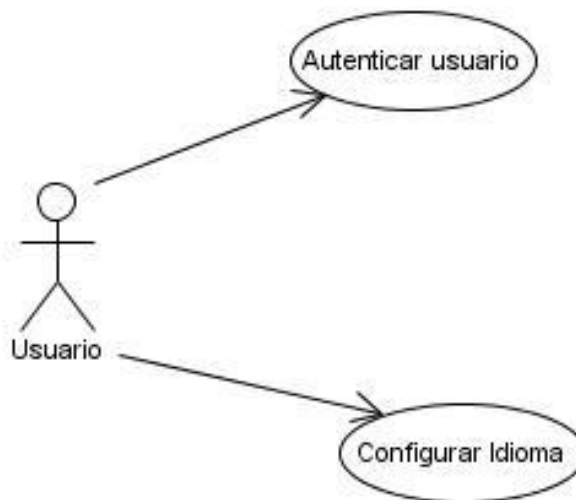
DCUS Paquete Administrador - Paquete reportes



DCUS Paquete Administrador - Paquete Administración



DCUS Paquete Común



2.3.4 Descripción de los casos de uso del sistema

A partir de los requisitos funcionales descritos se llevará a cabo el proceso de descripción y expansión de los casos de uso del sistema, dividiéndolos en dos grupos, críticos y no críticos.

Casos de Usos Críticos.

CUS #	Nombre del caso de uso del sistema	Referencias
-------	------------------------------------	-------------

CUS-01	Autenticar usuario	RF-01
CUS-02	Gestionar proyecto local	RF-02, RF-03, RF-04, RF-05, RF-06
CUS-03	Cargar proyecto local	RF-07
CUS-04	Cargar plantilla local	RF-08
CUS-05	Configurar acceso al servidor	RF-09
CUS-06	Gestionar proyecto del servidor	RF-10, RF-11, RF-12, RF-13
CUS-07	Gestionar plantilla	RF-14, RF-15, RF-16, RF-17, RF-18, RF-19
CUS-08	Gestionar usuario	RF-20, RF-21, RF-22, RF-23
CUS-09	Gestionar Imagen	RF-24, RF-25
CUS-10	Gestionar Tabla	RF- 26, RF-27, RF-28
CUS-11	Gestionar Tópico	RF-29, RF-30, RF-31
CUS-12	Gestionar Relación	RF-32, RF-33
CUS-13	Configurar espacio de trabajo	RF-34, RF-35, RF-36, RF-37, RF-38, RF-39
CUS-14	Dar formato al texto	RF-40, RF-41, RF-42, RF-43. RF-44
CUS-15	Editar texto / elemento	RF-45, RF-46, RF-47, RF-48. RF-49
CUS-16	Mostrar estadísticas	RF-50, RF-51, RF-52, RF-53
CUS-17	Mostrar trazabilidad de usuario	RF-54
CUS-18	Enviar correo electrónico	RF-55
CUS-19	Exportar a PDF	RF-56
CUS-20	Buscar contenido local	RF-57
CUS-21	Confeccionar plantillas	RF-58, RF-59, RF-60
CUS-22	Configurar Idioma	RF-61
CUS-23	Buscar contenido en el servidor	RF-62

Casos de Usos No Críticos.

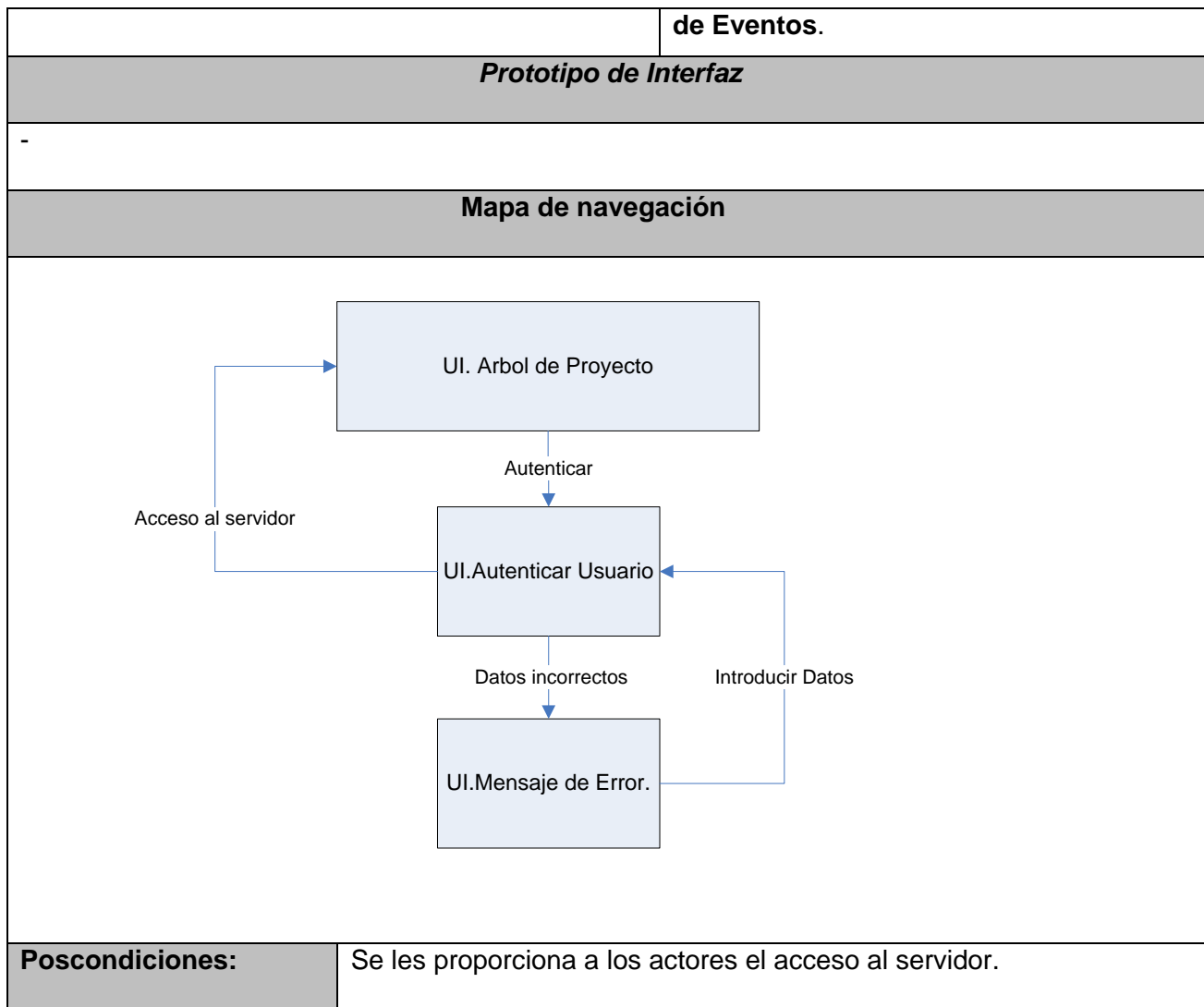
CUS #	Nombre del caso de uso del sistema	Referencias
CUS-24	Enviar correo al administrador	RF-63
CUS-25	Verificar actualizaciones	RF-64, RF-65

Expansión de los casos de uso

Con el propósito de lograr una mejor comprensión de los procesos a automatizar, se especifican los casos de uso del sistema mediante la descripción textual de los mismos.

Descripción Textual del Caso de Uso del sistema: Autenticar Usuario.

Caso de Uso:	Autenticar Usuario.	
Actores:	Arquitecto de Información, Administrador (inicia).	
Resumen:	El Caso de Uso se inicia cuando el Arquitecto de información o el administrador desean autenticarse en el servidor. El sistema permite introducir usuario y contraseña. Después de introducidos estos datos, el sistema los verifica para comprobar si existe algún error. Terminando así el caso de uso.	
Precondiciones:	Debe estar configurado el acceso al servidor.	
Referencias:	RF-01.	
Prioridad:	Crítico	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. El caso de uso se inicia cuando el actor desea autenticarse para entrar al servidor.	2. El sistema brinda la posibilidad de conectarse al servidor. Y permite introducir: <ul style="list-style-type: none"> • Usuario. • Contraseña.
	3. El actor introduce el usuario y la contraseña.	4. <i>El sistema verifica los datos.</i>
		5. El sistema da acceso al servidor de Base de Datos.
		6. El caso de uso termina.
Flujos Alternos		
	Acción del Actor	Respuesta del Sistema
		4. a Si los datos introducidos están incorrectos se muestra un mensaje de error: "Existen datos escritos incorrectamente."
		4. a. 2 Regresa al paso 2 del Flujo Normal



Descripción Textual del Caso de Uso del sistema: Gestionar proyecto local.

Caso de Uso:	Gestionar proyecto local.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el actor desea realizar algunas de las acciones definidas para un proyecto. El sistema brinda al usuario la posibilidad de crear un nuevo proyecto, modificar datos de un proyecto, eliminar o salvar un proyecto en específico. En caso que seleccione la opción de crear un nuevo proyecto, el sistema dará la

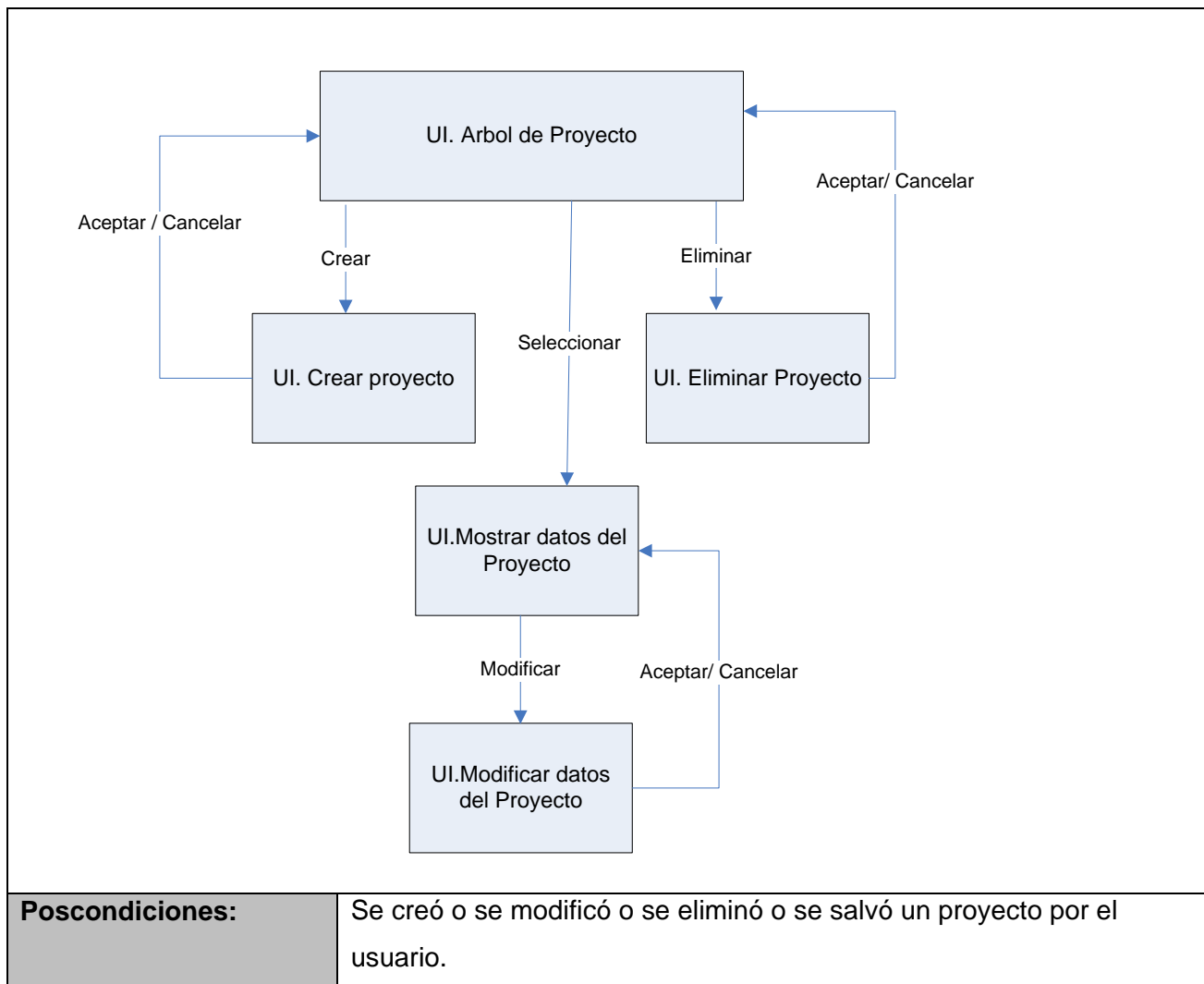
	<p>posibilidad de insertar los datos específicos del mismo. Si el usuario elige la opción de modificar el proyecto, el sistema mostrará los datos que pueden ser editables y una vez realizados los cambios, guardará las modificaciones. Si el usuario selecciona la opción eliminar un proyecto, el sistema le permitirá al usuario verificar que en verdad desea eliminar dicho proyecto y una vez confirmado lo eliminará. Si el usuario selecciona la opción salvar un proyecto, el sistema le permitirá realizar una salva del mismo en la PC. Terminando así el caso de uso.</p>
Precondiciones:	-
Referencias:	RF-02, RF-03, RF-04, RF-05, RF-06.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea realizar las acciones definidas para un proyecto.	<p>2. El sistema muestra un listado con todos los proyectos existentes y brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Crear nuevo proyecto. Ver Sección1: “Nuevo proyecto”. • Eliminar proyecto. Ver Sección2: “Eliminar proyecto”. • Modificar datos del proyecto. Ver Sección3: “Modificar datos del proyecto”. • Salvar el proyecto en la PC. Ver Sección4: “Salvar proyecto”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección1: “Nuevo proyecto”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción de crear nuevo proyecto.	<p>2. El sistema brinda la posibilidad de introducir los datos del proyecto.</p> <ul style="list-style-type: none"> • Nombre del proyecto.

	<ul style="list-style-type: none"> • Arquitecto de Información. • Breve descripción del proyecto. <p>Seleccionar:</p> <ul style="list-style-type: none"> • Fecha de creación. • Tipo de proyecto. • Plantillas correspondientes al proyecto. <p>Y permite:</p> <ul style="list-style-type: none"> • Crear el proyecto. • Cancelar la acción.
3. Introduce los datos y selecciona la opción de crear el proyecto.	4. <i>Valida los datos</i>
	5. <i>El sistema guarda los datos y crea la carpeta del proyecto.</i>
	6. Muestra la carpeta del proyecto creada en el árbol de proyecto y visualiza en el área de trabajo los datos. Y permite: <ul style="list-style-type: none"> • Modificar los datos del proyecto.
7. Consulta los datos.	8. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
	4. a.1 Muestra un mensaje de error "Existen campos escritos incorrectamente".
	4. a.2 Regresa al paso 2 de la Sección1: "Nuevo proyecto" .
7. a Selecciona la opción de modificar los datos cuando están incorrectos.	7. a.1 Brinda la posibilidad de modificar los datos del proyecto. Ver paso 2 de la

	Sección3: “Modificar datos del proyecto”.
Sección2: “Eliminar proyecto”.	
Acción del Actor	Respuesta del Sistema
1. El usuario escoge del árbol de proyecto el proyecto que desea eliminar y selecciona la opción de eliminar proyecto.	2. Muestra el mensaje de confirmación “¿Desea eliminar el proyecto seleccionado?” Y permite: <ul style="list-style-type: none"> • Eliminar el proyecto. • Cancelar la acción.
3. Selecciona la opción de eliminar proyecto.	4. <i>Elimina el proyecto seleccionado.</i>
	5. Muestra el mensaje de información “El proyecto ha sido eliminado”.
	6. Actualiza el árbol de proyecto quitando el proyecto eliminado.
	7. Regresa al paso 2 del Flujo Normal de Eventos.
	8. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Sección3: “Modificar datos del proyecto”.	
Acción del Actor	Respuesta del Sistema
1. El usuario escoge del árbol de proyecto el proyecto que desea modificar y selecciona la opción de modificar proyecto.	2. El sistema permite modificar los siguientes datos del proyecto: <ul style="list-style-type: none"> • Nombre del proyecto. • Arquitecto de información. • Descripción del proyecto. • Fecha de creación.

	<ul style="list-style-type: none"> • Tipo de proyecto. • Plantillas del proyecto. <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar los datos del proyecto. • Cancelar la acción.
3. Introduce los nuevos datos y selecciona la opción de modificar los datos.	4. <i>Valida los datos introducidos.</i>
	5. <i>Actualiza los datos del proyecto y los guarda.</i>
	6. Muestra un mensaje de información “Se han actualizado los datos del proyecto”.
	7. El sistema visualiza los datos del proyecto después de las modificaciones.
	8. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
	4. a.1 Si existen datos incorrectos se muestra el mensaje de error: “Existen campos escritos incorrectamente, por favor, rectifique estos datos”.
	4. a.2 Regresa al paso 2 de la Sección3: “Modificar datos del proyecto” .
Sección4: “Salvar proyecto”	
Acción del Actor	Respuesta del Sistema
1. El usuario escoge del árbol de proyecto el proyecto que desea y selecciona la opción de salvar el proyecto en la máquina local.	2. El sistema brinda la posibilidad de direccionar el lugar donde desea realizar la salva del proyecto Y permite:

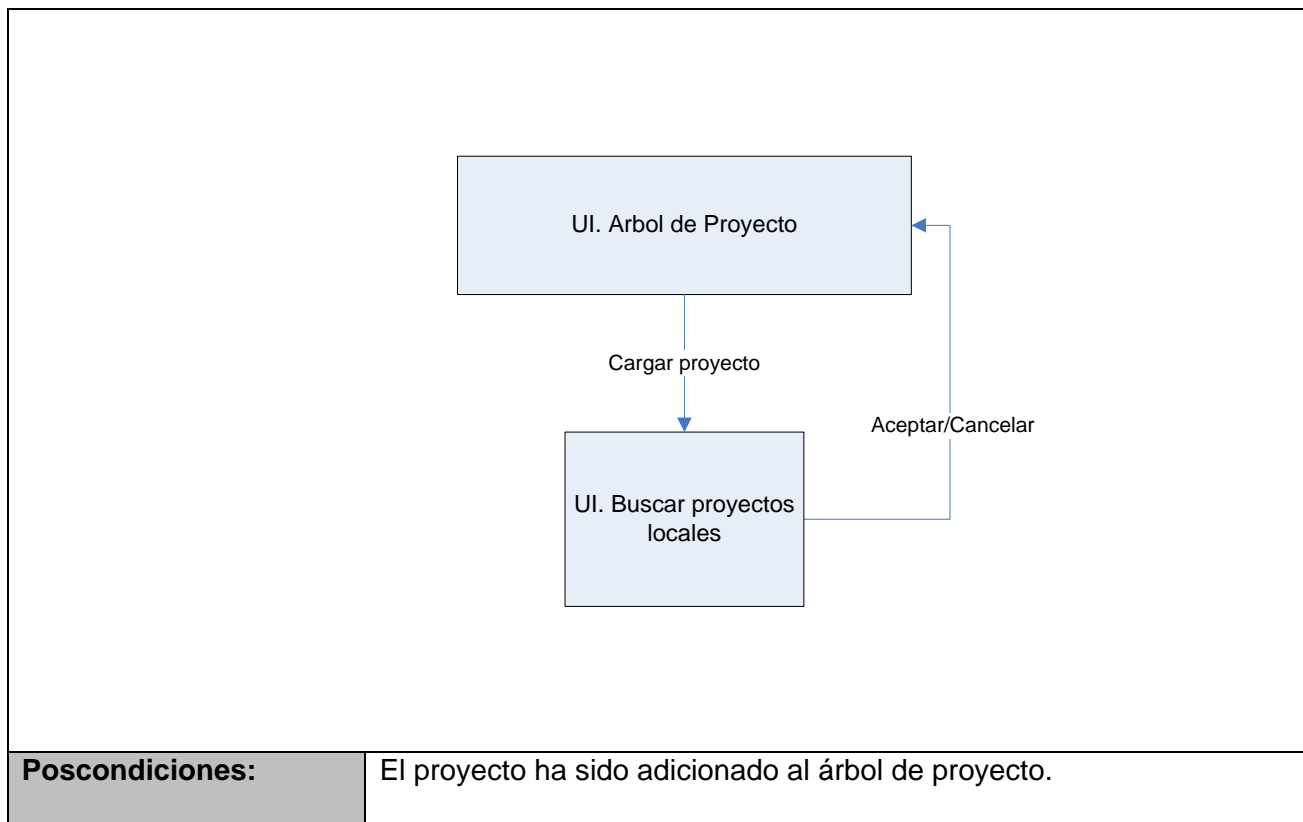
	<ul style="list-style-type: none"> • Salvar el proyecto en la máquina local. • Cancelar la acción.
3. El usuario realiza la búsqueda del lugar de la máquina local donde desea guardar el proyecto y selecciona la acción de salvar.	4. <i>El sistema guarda la carpeta de proyecto seleccionada.</i>
	5. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo: <ul style="list-style-type: none"> • UI. Mostrar Datos Proyecto • UI. Crear Proyecto • UI. Modificar Datos del proyecto • UI. Eliminar Proyecto 	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Cargar proyectos locales.

Caso de Uso:	Cargar proyectos locales.
Actores:	Arquitecto de Información (inicia).
Resumen:	El caso de uso se inicia cuando el actor desea abrir un proyecto que ha sido guardado anteriormente en la máquina local. El sistema permite realizar una búsqueda de los proyectos que desea abrir, para posteriormente visualizarlo. Terminando así el caso de uso.
Precondiciones:	Debe haber sido guardado el proyecto en la máquina local.
Referencias:	RF-07

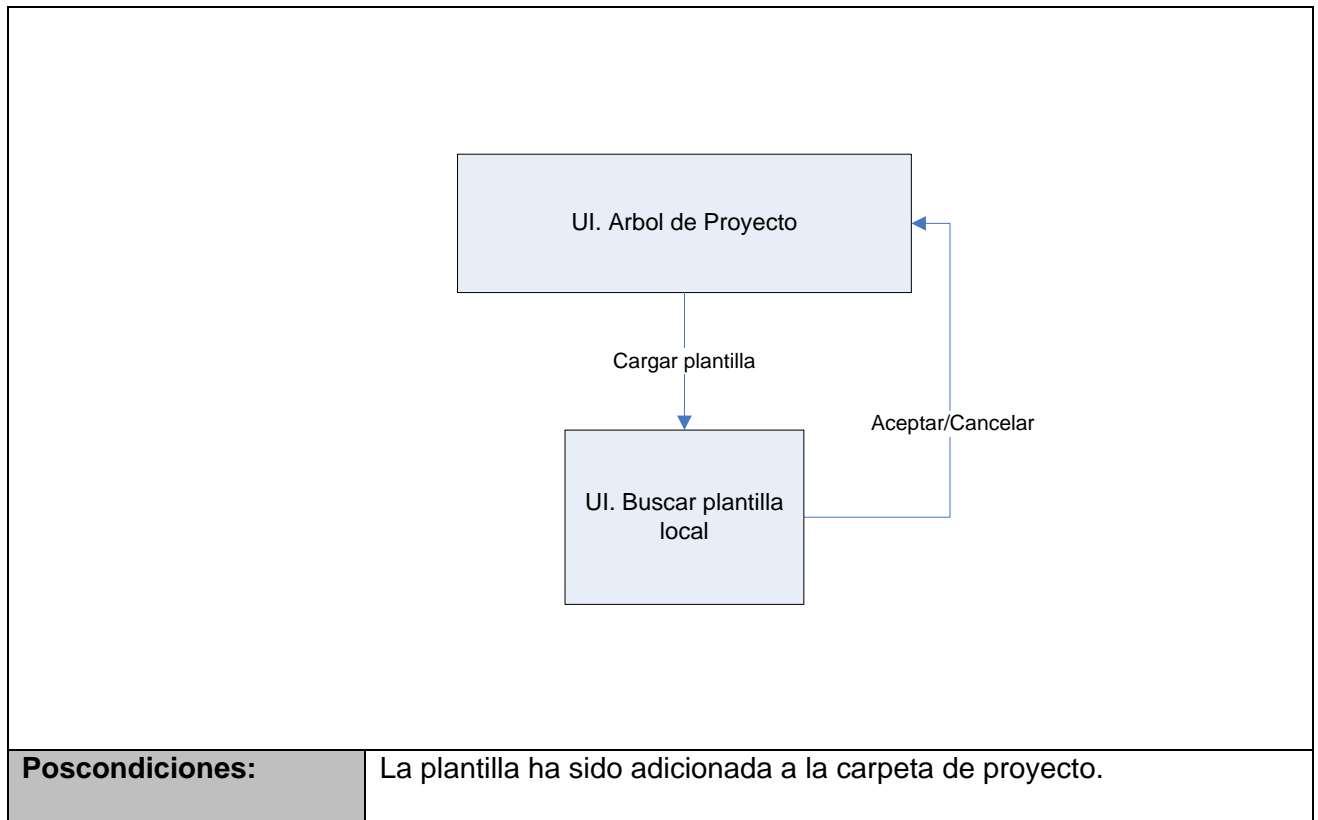
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario desea abrir un proyecto guardado en la máquina local.	2. El sistema brinda la posibilidad de realizar una búsqueda de la ubicación del proyecto que se desea abrir Y permite: <ul style="list-style-type: none"> • Abrir proyectos locales. • Cancelar la acción.
3. El usuario realiza la búsqueda de la ubicación donde se encuentran guardados los proyectos en la máquina local, la selecciona, y escoge la acción abrir proyectos locales.	5. Actualiza el árbol de proyecto.
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Cargar plantillas locales.

Caso de Uso:	Cargar plantillas locales.
Actores:	Arquitecto de Información (inicia).
Resumen:	El caso de uso se inicia cuando el actor desea abrir una plantilla que ha sido guardada anteriormente en la máquina local. El sistema permite realizar una búsqueda de las plantillas que desea abrir, para posteriormente visualizarla. Terminando así el caso de uso.
Precondiciones:	Debe haber sido guardada la plantilla en la máquina local. Debe haber sido creado el proyecto en la herramienta al que desea adicionarle la plantilla.
Referencias:	RF-08
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

<p>1. El caso de uso se inicia cuando el usuario desea abrir una plantilla guardada en la máquina local.</p>	<p>2. El sistema brinda la posibilidad de realizar una búsqueda de la ubicación de la plantilla que se desea abrir seleccionando el proyecto de la herramienta al que desea adicionarle la plantilla.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Abrir la plantilla seleccionada de la máquina local. • Cancelar la acción.
<p>3. El usuario realiza la búsqueda de la ubicación donde se encuentran guardadas las plantillas en la máquina local, y escoge el proyecto de la herramienta al que desea adicionarle la plantilla y selecciona la acción abrir plantillas locales.</p>	<p>5. Actualiza el árbol de proyecto con la nueva plantilla adicionada.</p>
	<p>6. El caso de uso termina.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<p>*a. El actor selecciona la opción de cancelar.</p>	<p>*a. 1. Regresa a la vista anterior.</p>
	<p>*a. 2. El caso de uso termina.</p>
Prototipo de Interfaz	
-	
Mapa de navegación	



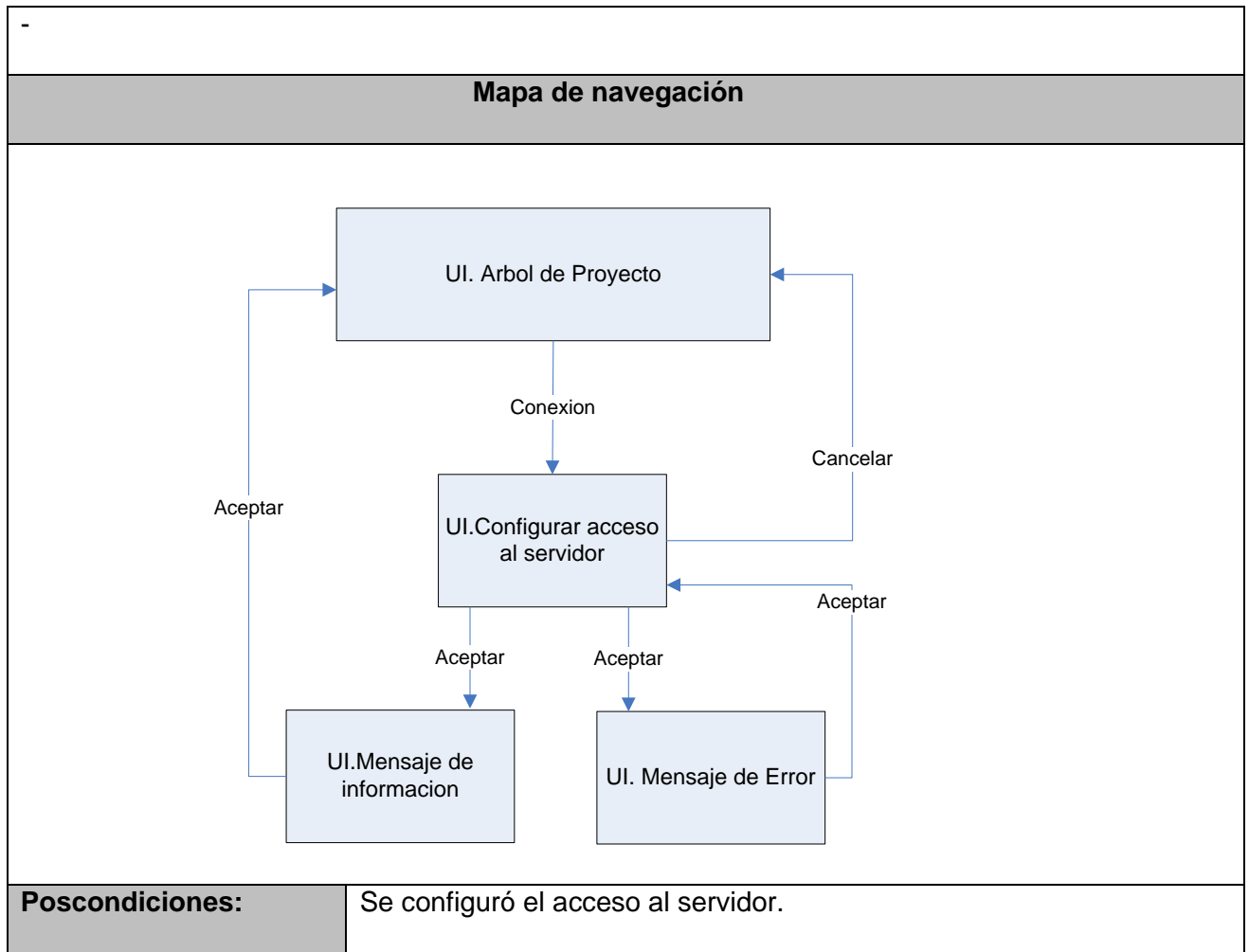
Descripción Textual del Caso de Uso del sistema: Configurar acceso al servidor.

Caso de Uso:	Configurar acceso al servidor.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el actor desea configurar el acceso al servidor. El sistema brinda la posibilidad de configurar el acceso al servidor, dejándole introducir una serie de datos, para posteriormente permitir la conexión al mismo. Terminando así el caso de uso.
Precondiciones:	-
Referencias:	RF-09
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea configurar el acceso al servidor.	2. El sistema brinda la posibilidad de configurar el acceso al servidor mostrando

	<p>una interfaz donde le permite al usuario introducir los datos para la conexión, como son:</p> <ul style="list-style-type: none"> • Servidor • Proxy • Puerto • Usuario • Contraseña <p>Y permite seleccionar:</p> <ul style="list-style-type: none"> • No Usar Proxy. Ver Sección1: “No usar proxy”. • Usar Proxy. Ver Sección2: “Usar proxy”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección1: “No usar proxy”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de no usar proxy.	<p>2. El sistema muestra la interfaz con los campos de puerto y proxy deshabilitados.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar la conexión. • Cancelar la acción.
3. El usuario introduce los datos de los campos que no han sido deshabilitados y selecciona la opción de aceptar la conexión.	4. <i>Valida los datos.</i>
	5. Muestra un mensaje de información “Se ha conectado al servidor”.
	6. Visualiza una iconografía que indica al usuario que la herramienta se ha conectado al servidor.
	7. El caso de uso termina.
Flujo Alterno	

Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.
4. a. Los datos del servidor están incorrectos.	4. a.1 Muestra un mensaje de error “Los datos del servidor están incorrecto”.
	4. a.2 Regresa al paso 3 de la Sección1: “No usar proxy” .
	4. a.3 Visualiza una iconografía que indica al usuario que la herramienta no se ha conectado al servidor.
	4. a.4 El caso de uso termina.
4. b. El usuario o la contraseña están incorrectos.	4. b.1 Muestra un mensaje de error “El usuario o contraseña no son los corrector”.
	4. b.2 Regresa al paso 3 de la Sección1: “No usar proxy” .
	4. b.3 Visualiza una iconografía que indica al usuario que la herramienta no se ha conectado al servidor.
	4. b.4 El caso de uso termina.
Sección2: “Usar proxy”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de usar proxy.	2. El sistema muestra la interfaz con todos los campos habilitados. Y permite: <ul style="list-style-type: none"> • Aceptar la conexión. • Cancelar la acción.
3. El usuario introduce los datos de los campos y selecciona la opción de aceptar la conexión.	4. <i>Valida los datos</i>
	5. Muestra un mensaje de información “Se ha conectado al servidor”.
	6. Visualiza una iconografía que indica al usuario que la herramienta se ha

	conectado al servidor.
	7. El caso de uso termina.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.
4. a. Los datos de conexión (Proxy / Puerto) están incorrectos.	4. a.1 Muestra un mensaje de error: “Los datos de conexión (Proxy / Puerto) están incorrecto”.
	4. a.2 Y regresa al paso 3 de la Sección2: “Usar proxy” .
	4. a.3 Visualiza una iconografía que indica al usuario que la herramienta no se ha conectado al servidor.
	4. a.4 El caso de uso termina.
4. b. El usuario o la contraseña están incorrectos.	4. b.1 Muestra un mensaje de error: “El usuario o contraseña no son los corrector”.
	4. b.2 Y regresa al paso 3 de la Sección2: “Usar proxy” .
	4. b.3 Visualiza una iconografía que indica al usuario que la herramienta no se ha conectado al servidor.
	4. b.4 El caso de uso termina.
4. c. Los datos del servidor están incorrectos.	4. c.1 Muestra un mensaje de error “Los datos del servidor están incorrecto.”
	4. c.2 Y regresa al paso 3 de la Sección2: “Usar proxy” .
	4. c.3 Visualiza una iconografía que indica al usuario que la herramienta no se ha conectado al servidor.
Prototipo de Interfaz	



Descripción Textual del Caso de Uso del sistema: Gestionar proyecto del servidor.

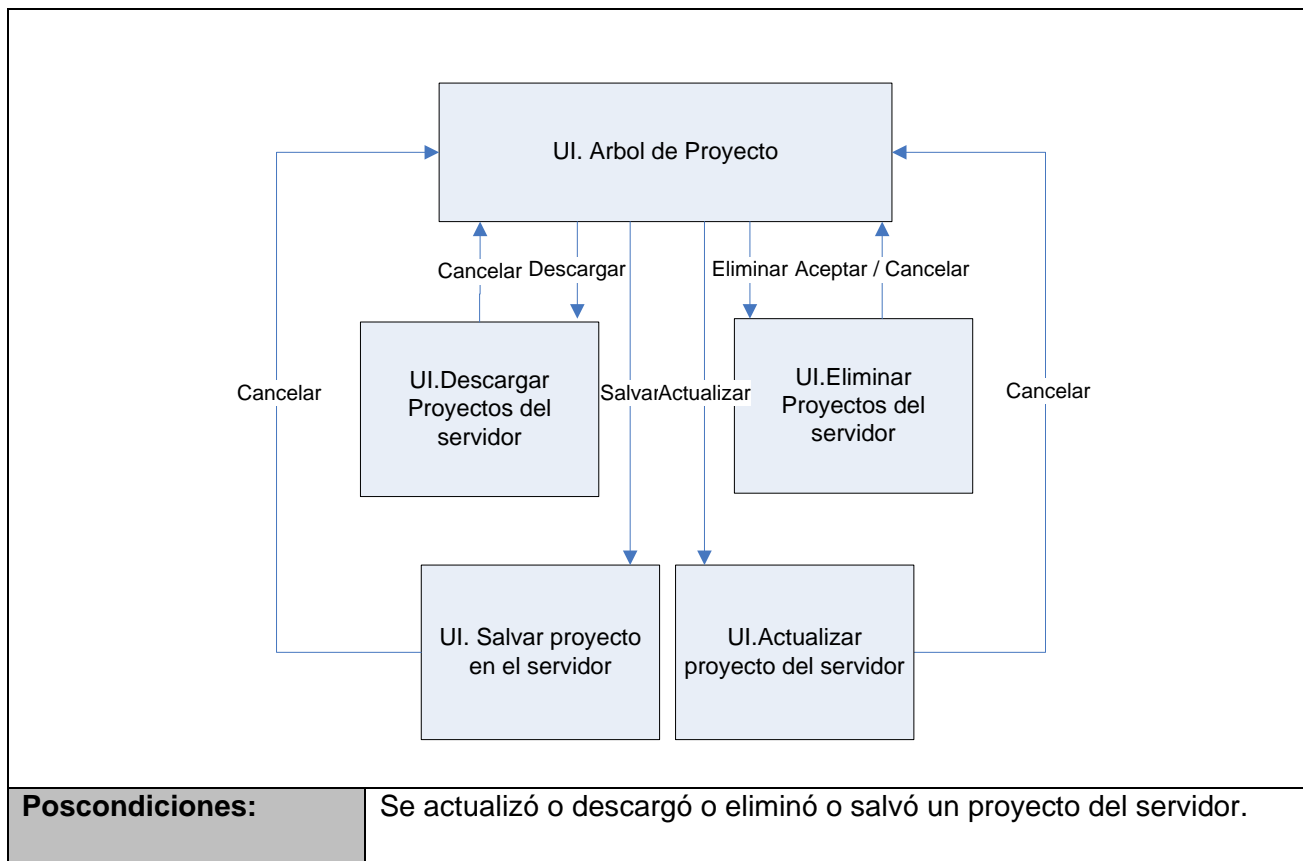
Caso de Uso:	Gestionar proyecto del servidor.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea realizar algunas de las acciones definidas para un proyecto en el servidor. El sistema brinda la posibilidad de descargar, actualizar, eliminar o salvar un proyecto en el servidor. En caso que seleccione la opción descargar un proyecto, el sistema copiara la carpeta de proyecto con sus plantillas correspondientes en la herramienta. Si el usuario selecciona la opción de actualizar proyecto, el sistema permitirá al usuario

	reemplazar la carpeta de proyecto con sus plantillas correspondiente y una vez realizados los cambios, guardará las nuevas modificaciones. Si el usuario selecciona la opción eliminar un proyecto, el sistema le permitirá al usuario verificar que en verdad desea eliminar dicho proyecto y una vez confirmado lo eliminará del servidor. Si el usuario selecciona la opción salvar proyecto, el sistema le permitirá realizar una salva del mismo en el servidor. Terminando así el caso de uso.
Precondiciones:	Debe estar configurado el servidor. El usuario debe estar autenticado.
Referencias:	RF-10, RF-11, RF-12, RF-13.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea realizar las acciones definidas para un proyecto en el servidor.	2. El sistema muestra los proyectos salvados en el servidor y brinda la posibilidad de: <ul style="list-style-type: none"> • Descargar proyecto del servidor. Ver Sección1: “Descargar proyecto del servidor”. • Eliminar proyecto del servidor. Ver Sección2: “Eliminar proyecto del servidor”. • Actualizar proyecto del servidor. Ver Sección3: “Actualizar proyecto del servidor”. • Salvar proyecto en el servidor. Ver Sección4: “Salvar proyecto en el servidor”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección1: “Descargar proyecto del servidor”.	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de descargar proyecto del servidor.	2. El sistema brinda la posibilidad de seleccionar los proyectos con sus plantillas

	<p>correspondientes que desea descargar.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Descargar proyecto(s) del servidor. • Cancelar la acción.
3. El usuario selecciona la(s) carpeta(s) de proyecto deseada(s) y la opción de descargar.	4. <i>Descarga la carpeta de proyecto para la herramienta.</i>
	5. Muestra la carpeta de proyecto descargada en el árbol de proyecto con sus plantillas correspondientes.
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1 Regresa a la vista anterior.
	*a.2 El caso de uso termina.
Sección2: “Eliminar proyecto del servidor”.	
Acción del Actor	Respuesta del Sistema
1. El usuario escoge el proyecto en el servidor que desea eliminar y selecciona la opción de eliminar proyecto.	<p>2. El sistema muestra los proyectos salvados en el servidor y brinda la posibilidad de seleccionar el que desea eliminar.</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Eliminar proyecto(s) en el servidor • Cancelar la acción.
3. El usuario selecciona los proyectos deseados y la acción de eliminarlo.	<p>4. Muestra el mensaje de confirmación “¿desea eliminar el elemento seleccionado?”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Confirma que desea eliminar el elemento. • Cancelar la acción.
5. El usuario realiza la confirmación de eliminar el	6. <i>Se elimina el proyecto del servidor.</i>

elemento.	
	7. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Sección3: “Actualizar proyecto del servidor”.	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción de actualizar proyecto del servidor.	2. <i>El sistema verifica los proyectos existentes en el servidor.</i>
	3. El sistema muestra los proyectos salvados en el servidor y brinda la posibilidad de seleccionar el que desea actualizar. Y permite: <ul style="list-style-type: none"> • Actualizar proyecto(s) en el servidor. • Cancelar la acción.
4. El usuario selecciona los proyectos deseados y la acción de actualizar.	5. Muestra el mensaje de confirmación “Se ha actualizado el proyecto en el servidor”
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Sección4: “Salvar proyecto del servidor”.	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el proyecto que desea salvar en el servidor y selecciona la opción de salvar.	2. <i>El sistema verifica si el proyecto señalado existente en el servidor.</i>
	3. Si el proyecto existe en el servidor, muestra un mensaje de información “El

	<p>proyecto existe en el servidor, ¿Desea actualizarlo?”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Actualizar proyecto en el servidor. • Cancelar la acción.
	<p>4. Si el proyecto no existe en el servidor. El sistema muestra un mensaje de confirmación “¿Desea salvar el proyecto en el servidor?”.</p> <ul style="list-style-type: none"> • Salvar proyecto en el servidor. • Cancelar la acción.
4. El usuario selecciona la acción de salvar proyecto en el servidor.	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1 Regresa a la vista anterior.
	*a.2 El caso de uso termina.
3. a El actor selecciona la opción de actualizar proyecto en el servidor.	3. a.1 Brinda la posibilidad de actualizar. Ver paso 2 de la Sección 3: “Actualizar proyecto del servidor” .
Prototipo de Interfaz	
-	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Gestionar plantilla.

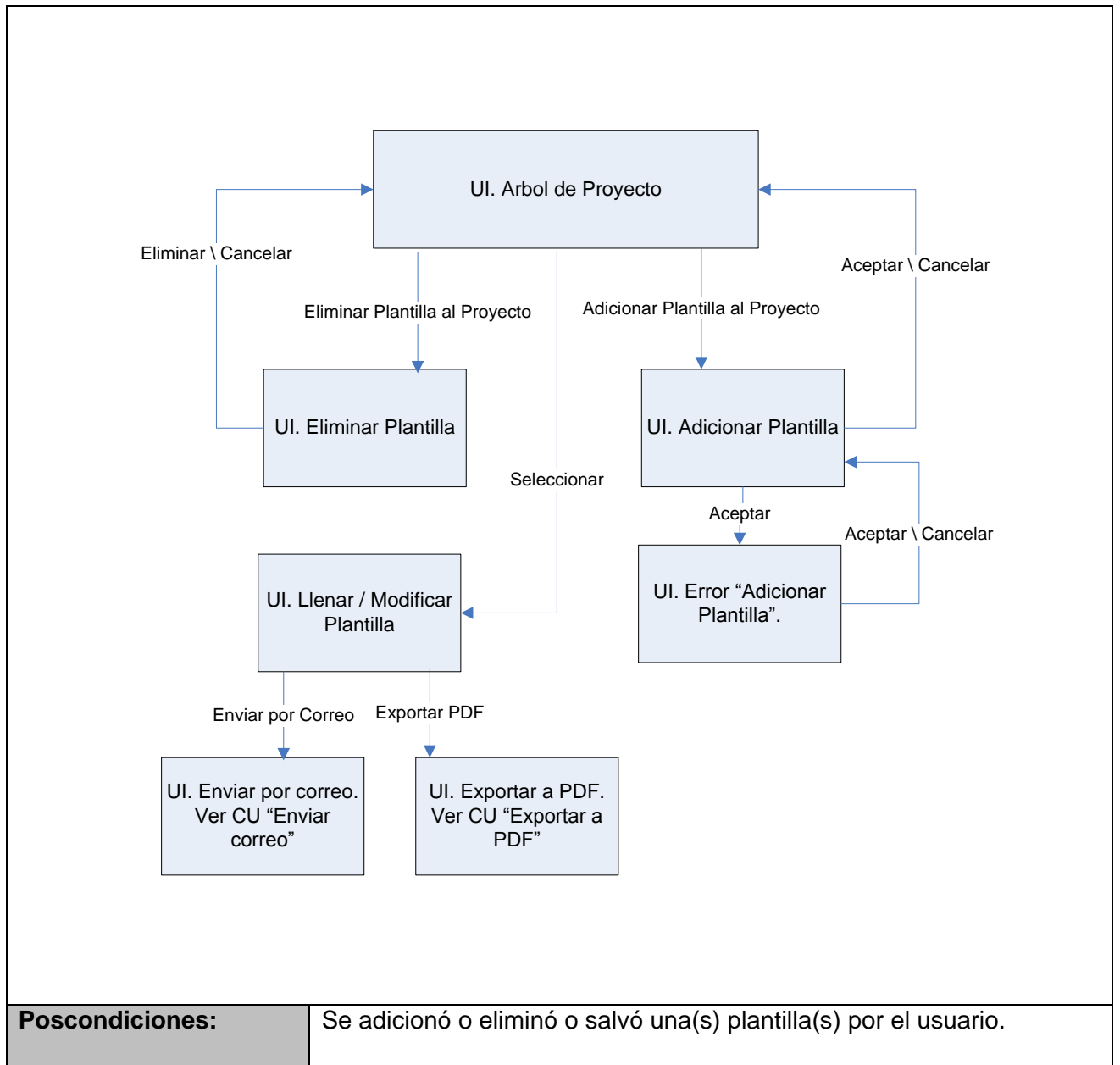
Caso de Uso:	Gestionar proyecto del servidor.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea realizar algunas de las acciones definidas para un plantilla. El sistema brinda al usuario las posibilidades de adicionar, llenar, salvar o eliminar una plantilla de un proyecto creado. En caso que seleccione la opción de adicionar una(s) plantilla(s) al proyecto, el sistema dará la posibilidad de seleccionar la(s) que desea insertar. Si selecciona la opción de llenar plantilla, el sistema brinda la posibilidad de llenar los datos correspondientes a la plantilla, así como exportarla a PDF o enviarla por correo. Si el usuario selecciona la opción eliminar la plantilla de un proyecto, el sistema le permitirá al usuario verificar que en verdad

	desea eliminar dicha plantilla y una vez confirmada la eliminará. Terminando así el caso de uso.
Precondiciones:	Debe haber sido creado el proyecto. Deben estar creadas las plantillas por el administrador.
Referencias:	RF-14, RF-15, RF-16, RF-17, RF-18, RF-19.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea realizar las acciones definidas para una plantilla.	2. El sistema muestra en el árbol de proyecto las carpetas de proyectos con sus plantillas correspondientes y brinda la posibilidad de: <ul style="list-style-type: none"> • Adicionar plantilla(s) al proyecto creado. Ver Sección1: “Adicionar plantilla”. • Llenar / Modificar plantilla. Ver Sección2: “Llenar / Modificar plantilla”. • Eliminar la plantilla del proyecto. Ver Sección3: “Eliminar plantilla”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección1: “Adicionar plantilla”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de Adicionar plantilla(s) al proyecto.	2. <i>El sistema verifica las plantilla(s) existentes del proyecto seleccionado.</i> Además muestra un listado con la(s) plantilla(s) que no posee actualmente, permitiendo seleccionar la(s) que desea incorporar. Y permite: <ul style="list-style-type: none"> • Adicionar plantilla(s) al proyecto.

	<ul style="list-style-type: none"> • Cancelar la acción.
3. El usuario selecciona la(s) plantilla(s) que desea incorporar al proyecto y selecciona la opción de adicionar plantilla(s).	4. <i>Valida los datos.</i>
	5. El sistema actualiza el árbol de proyecto con las plantilla(s) que fueron adicionada(s) en el mismo.
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.
4. a. El actor selecciona la opción de aceptar y no ha seleccionado ninguna plantilla.	4. a.1. Muestra el mensaje de error “Debe seleccionar al menos una plantilla”.
	4. a.2. Regresa al paso 3 de la Sección1: “Adicionar plantilla” .
Sección2: “Llenar / Modificar plantilla”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción de llenar / modificar la plantilla seleccionada.	2. El sistema muestra los campos de la plantilla además de sus componentes y brinda la posibilidad de llenar los campos de la misma (en caso de que estén llenos permite modificarlos.) Y permite: <ul style="list-style-type: none"> • Exportar a PDF la plantilla. • Enviar por correo la plantilla. • Salvar la plantilla en la PC.
3. El usuario llena o modifica los campos que desee.	4. <i>El sistema guarda los cambios realizados en la plantilla.</i>
	5. El caso de uso termina.
Flujos Alternos	

Acción del Actor	Respuesta del Sistema
2. a. El actor selecciona la acción de exportar a PDF la plantilla.	2. a.1 Brinda la posibilidad de exportar a PDF la plantilla. Ver CU “Exportar a PDF” .
	2. a.2 Regresa al paso 2 de la Sección2: “Llenar / Modificar plantilla” .
2. b. El actor selecciona enviar por correo la plantilla en la PC.	2. b.1 Brinda la posibilidad de enviar por correo electrónico la plantilla. Ver CU “Enviar correo electrónico” .
	2. b.2 Regresa al paso 2 de la Sección2: “Llenar / Modificar plantilla” .
2. c. El actor selecciona salvar la plantilla.	2. c.1 El sistema brinda la posibilidad de realizar una búsqueda de la ubicación donde desea guardar la plantilla. Y permite además: <ul style="list-style-type: none"> • Guardar la plantilla.
2. c.2 El usuario direcciona la ubicación donde desea realizar la salva de la plantilla y selecciona la opción de guardar la plantilla.	2. c.3 <i>Guarda la plantilla en la dirección especificada en la máquina local.</i>
	2. c.4 El caso de uso termina.
Sección3: “Eliminar plantilla”.	
Acción del Actor	Respuesta del Sistema
1. El usuario escoge en el árbol de proyecto, la plantilla que desea eliminar y selecciona la opción de eliminar plantilla.	2. Muestra el mensaje de confirmación “¿desea eliminar la plantilla seleccionada?” Y permite: <ul style="list-style-type: none"> • Eliminar la plantilla del proyecto. • Cancelar la acción.
3. Selecciona la opción de eliminar la plantilla del proyecto.	4. <i>Elimina la plantilla seleccionada.</i>
	5. Muestra el mensaje de información “La plantilla ha sido eliminada”.
	6. Actualiza el árbol de proyecto quitando la

	plantilla eliminada.
	7. Regresa a la vista anterior.
	8. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a. 1. Regresa a la vista anterior.
	*a. 2. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo:	
<ul style="list-style-type: none"> • UI. Texto. • UI. Taxonomía y Mapa. • UI. Ejemplo de Mapa. • UI. Esbozo de estructura. • UI. Taxonomía • UI. Prototipo 	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Gestionar usuario.

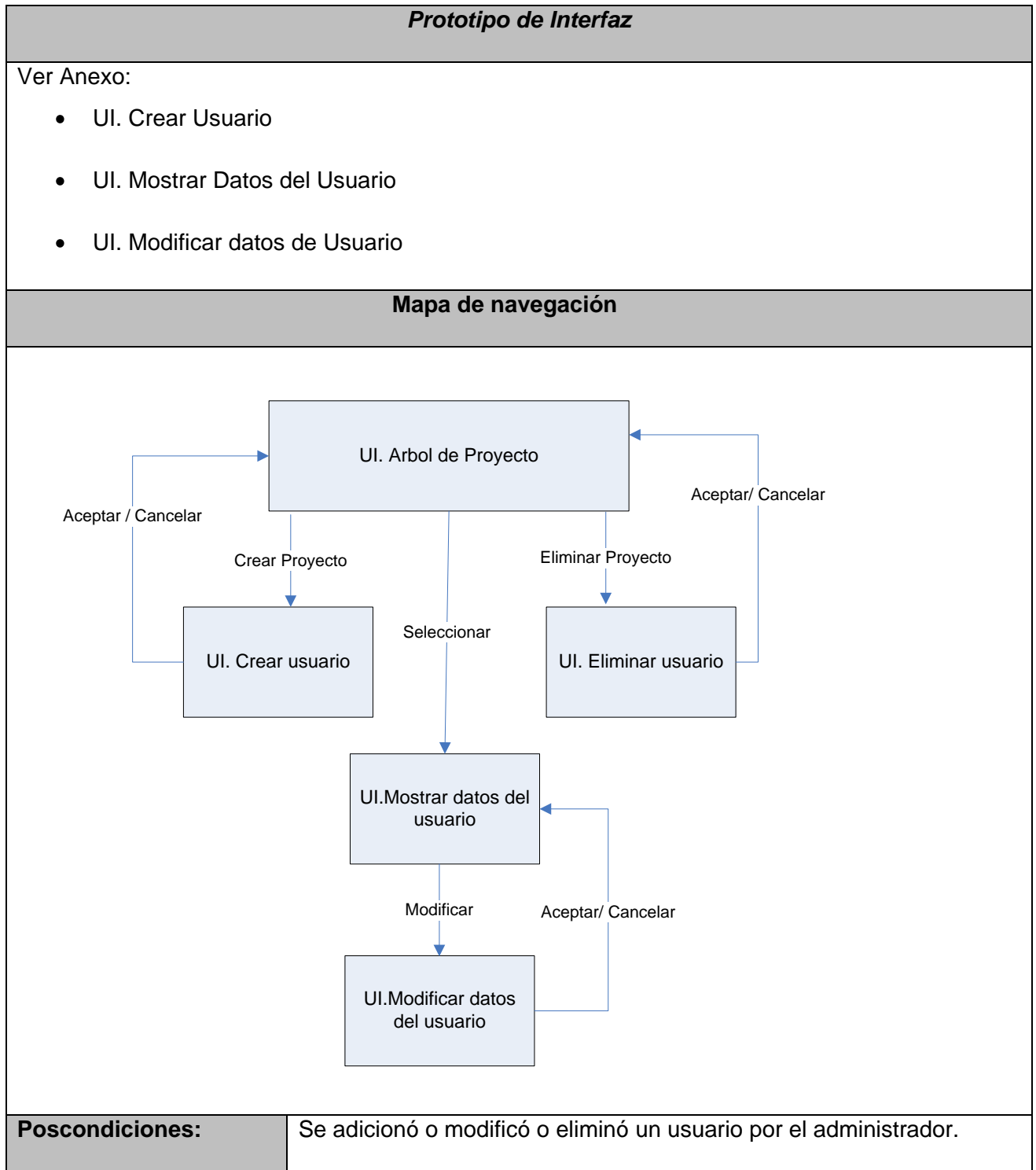
Caso de Uso:	Gestionar usuario.
Actores:	Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea realizar algunas de las acciones definidas para los usuarios que tendrán acceso al servidor. El sistema brinda la posibilidad de crear un nuevo usuario,

	<p>modificar datos de un usuario, eliminar un usuario. En caso que seleccione la opción de crear un nuevo usuario, el sistema dará la posibilidad de insertar los datos específicos del mismo. Si el administrador elige la opción de modificar los datos del usuario, el sistema mostrará los datos que pueden ser editables y una vez realizados los cambios, guardará las modificaciones. Si el administrador selecciona la opción eliminar un usuario, el sistema le permitirá verificar que en verdad desea eliminar dicho usuario y una vez confirmado lo eliminará. Terminando así el caso de uso.</p>
Precondiciones:	<p>Tiene que estar conectado al servidor. El usuario tiene que estar autenticado.</p>
Referencias:	RF-20, RF-21, RF-22, RF-23.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el administrador desea realizar las acciones definidas a un usuario.	<p>2. El sistema brinda la posibilidad de:</p> <ul style="list-style-type: none"> • Crear nuevo usuario. Ver Sección1: “Crear usuario”. • Eliminar usuario. Ver Sección2: “Eliminar usuario”. • Modificar datos del usuario. Ver Sección3: “Modificar datos del usuario”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección1: “Crear usuario”	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la acción de crear usuario.	<p>2. El sistema brinda la posibilidad de introducir los datos del usuario.</p> <ul style="list-style-type: none"> • Nombre del usuario (Arquitecto de Información). • Nombre de los proyectos que atiende.

	<ul style="list-style-type: none"> • Facultad / dirección a la que pertenece. • Número de solapín. <p>Seleccionar:</p> <ul style="list-style-type: none"> • Permisos de usuario (Roles). <p>Y permite:</p> <ul style="list-style-type: none"> • Crear usuario. • Cancelar la acción.
3. Introduce los datos y selecciona la opción de crear usuario.	4. <i>Valida los datos.</i>
	5. <i>Guarda los datos y crea un nuevo usuario.</i>
	6. Brinda la posibilidad de enviar un correo electrónico al usuario añadido.
	7. Muestra el nombre del nuevo usuario y visualiza los datos del mismo. Y permite: <ul style="list-style-type: none"> • Modificar los datos del usuario.
8. Consulta los datos.	9. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.
4. a. Existen datos incorrectos.	4. a.1 Muestra un mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos.”
	4. a.2 Regresa al paso 2 de la Sección 1.
4. b. Existen datos incompletos.	4. b.1 Muestra un mensaje de error “Existen campos incompletos, por favor, rectifique estos datos.”
	4. b.2 Regresa al paso 2 de la Sección 1.

6. a. Enviar un correo electrónico al usuario añadido.	6. a.1. Brinda la posibilidad de enviar un correo electrónico al usuario añadido. Ver CU “Enviar correo electrónico” .
	6. a.2. Regresa al paso 7 de la Sección1: “Crear usuario” .
8. a. El actor selecciona la acción de Modificar datos del proyecto.	8. a.1. Brinda la posibilidad de modificar los datos del usuario. Ver Sección 3: “Modificar datos del usuario” .
Sección2: “Eliminar usuario”	
Acción del Actor	Respuesta del Sistema
1. El administrador escoge el usuario que desea eliminar y selecciona la opción de eliminar usuario.	2. Muestra el mensaje de confirmación “¿desea eliminar el elemento seleccionado?” Y permite: <ul style="list-style-type: none"> • Eliminar usuario. • Cancelar la acción.
3. Selecciona la opción de eliminar usuario.	4. <i>Elimina el usuario seleccionado.</i>
	5. Muestra el mensaje de información “El elemento ha sido eliminado”.
	6. Actualiza el listado de los usuarios del servidor.
	7. Regresa a la vista anterior.
	8. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.
Sección3: “Modificar datos del usuario”	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la acción de modificar los datos del usuario.	2. El sistema permite modificar los siguientes los siguientes datos del usuario: <ul style="list-style-type: none"> • Nombre del usuario (Arquitecto de

	<p>Información).</p> <ul style="list-style-type: none"> • Nombre de los proyectos que atiende. • Facultad / dirección a la que pertenece. • Número de solapín. <p>Seleccionar:</p> <ul style="list-style-type: none"> • Permisos de usuario (Roles). <p>Y permite:</p> <ul style="list-style-type: none"> • Modificar los datos del usuario. • Cancelar la acción.
3. Realiza los cambios y selecciona la opción de modificar los datos del usuario.	4. <i>Valida los datos introducidos.</i>
	5. <i>Actualiza los datos del usuario y los guarda.</i>
	6. Muestra un mensaje de información “Se han actualizado los datos del usuario”.
	7. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
4. a. Existen datos incorrectos.	4. a.1. Muestra el mensaje de error “Existen campos escritos incorrectamente, por favor, rectifique estos datos”.
	4. a.2. Regresa al paso 2 de la Sección 3.
4. b. Existen datos incompletos.	4. a.1 Muestra un mensaje de error “Existen campos incompletos, por favor, rectifique estos datos.”
	4. a.2 Regresa al paso 2 de la Sección 3.
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.
	*a.2. El caso de uso termina.

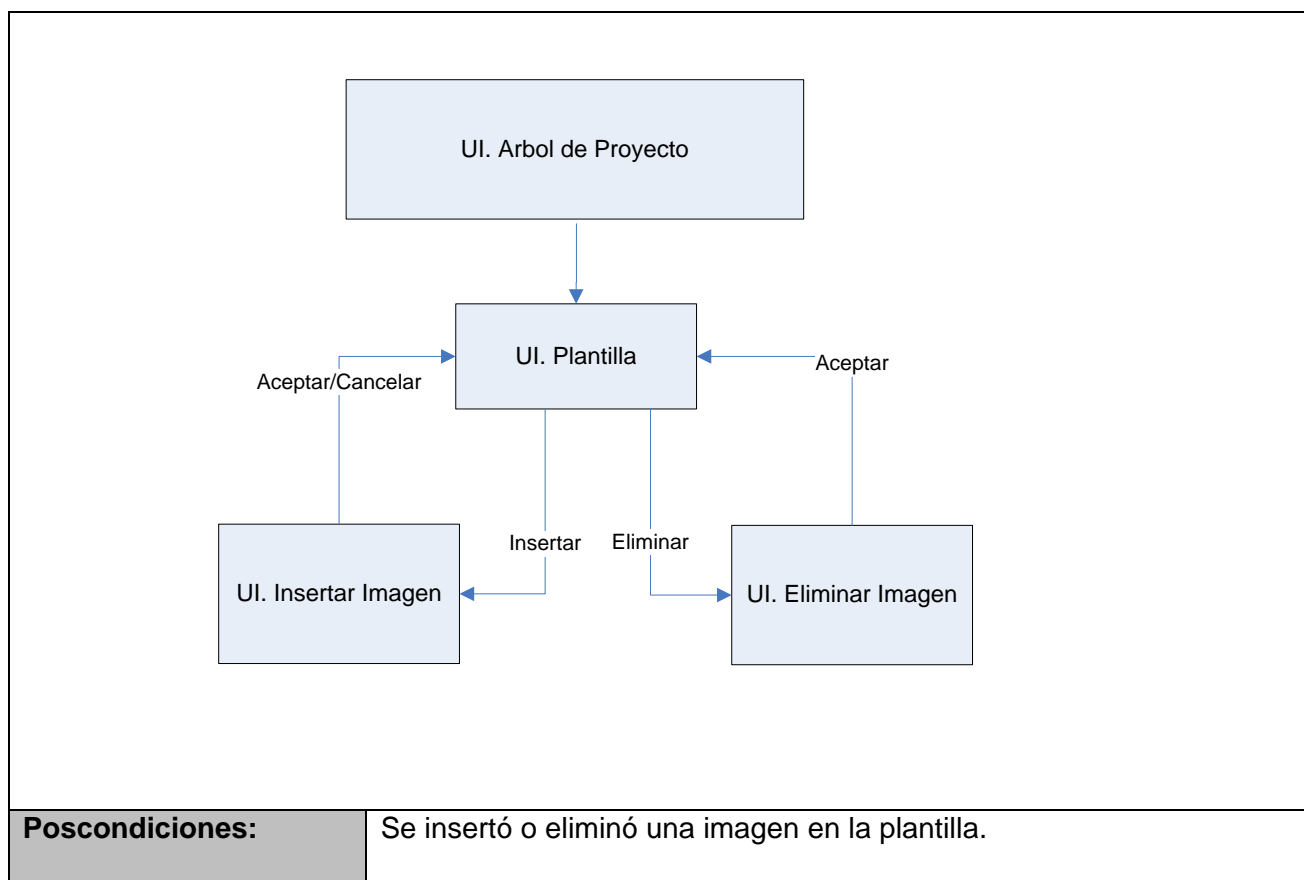


Descripción Textual del Caso de Uso del sistema: Gestionar Imagen.

Caso de Uso:	Gestionar Imagen.
---------------------	-------------------

Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el arquitecto de información desea realizar alguna acción sobre un elemento de contenido de tipo imagen. El sistema brinda la posibilidad de realizar acciones de insertar, eliminar una imagen. En caso que seleccione la opción de insertar una imagen, el sistema dará la posibilidad de insertar una imagen en específico. Si el arquitecto de información selecciona la opción eliminar una imagen, el sistema la eliminará. Terminando así el caso de uso.
Precondiciones:	El elemento debe estar seleccionado. Debe estar adicionada la plantilla.
Referencias:	RF-24, RF-25.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. 1. El caso de uso inicia cuando el Arquitecto de Información desea realizar alguna acción sobre los elementos del contenido de tipo imagen.	2. El sistema muestra la plantilla selecciona en la que se irá a trabajar y brinda la posibilidad de: <ul style="list-style-type: none"> • Insertar imagen. Sección1: "Insertar Imagen". • Eliminar imagen. Ver Sección2: "Eliminar Imagen".
3. El Arquitecto de Información selecciona la acción que desea realizar.	4. El caso de uso termina.
Sección1: "Insertar Imagen"	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de insertar una imagen.	2. El sistema brinda la posibilidad de realizar una búsqueda de la ubicación de la imagen que se desea insertar. Y permite: <ul style="list-style-type: none"> • Insertar la imagen.

	<ul style="list-style-type: none"> • Cancelar la acción.
3. El arquitecto de información realiza la búsqueda de la ubicación donde se encuentra guardada la imagen en la PC, la selecciona, y escoge la acción insertar la imagen en la plantilla.	4. El sistema muestra la imagen insertada.
	5. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*.a. El actor selecciona la opción de Cancelar.	*.a.1. Regresa a la vista anterior.
	*.a.2. El caso de uso termina.
Sección2: “Eliminar Imagen”	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información escoge la imagen que desea eliminar y selecciona la acción de eliminar.	2. El sistema elimina el elemento seleccionado
	3. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	

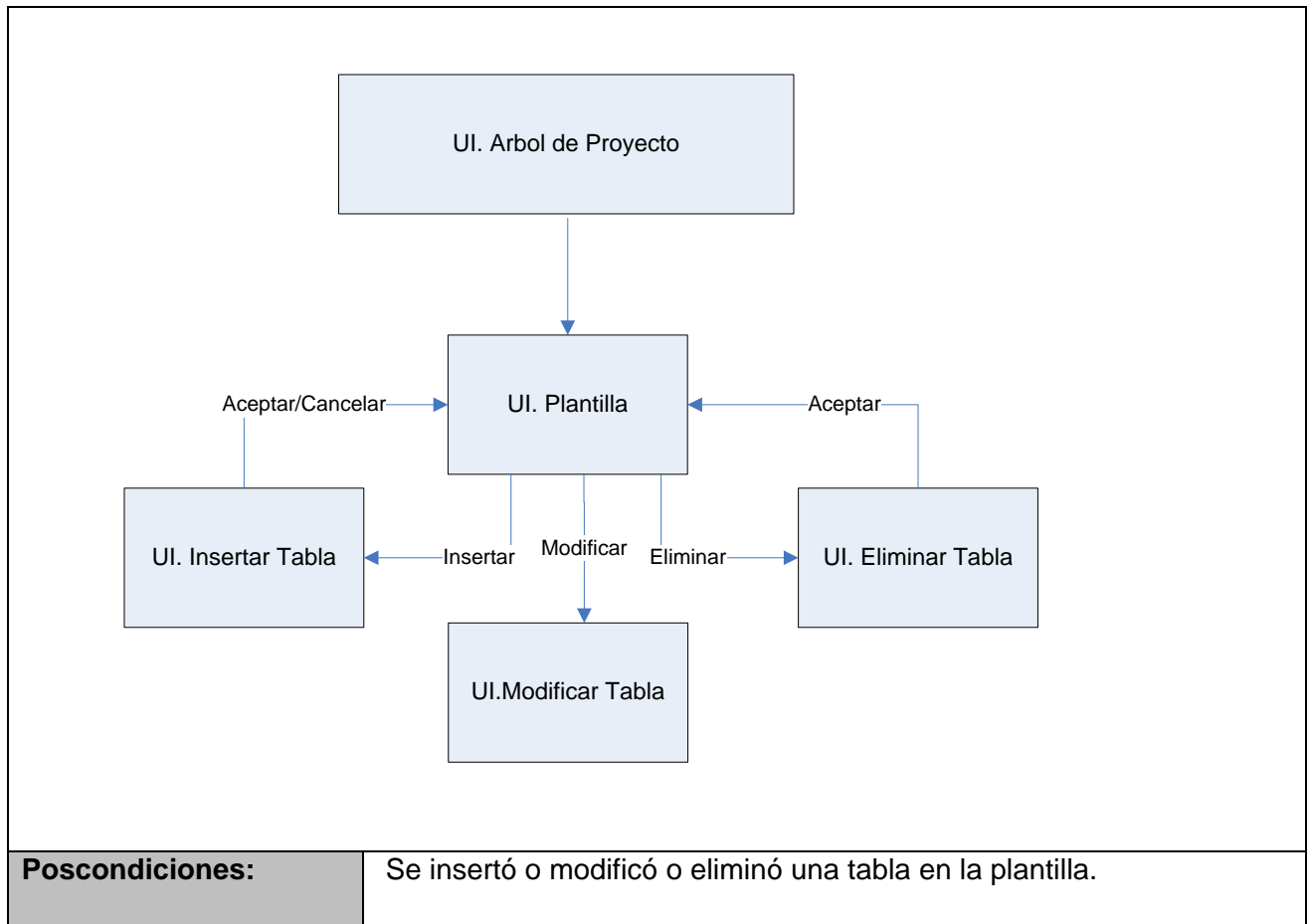


Descripción Textual del Caso de Uso del sistema: Gestionar Tabla.

Caso de Uso:	Gestionar Tabla.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el arquitecto de información desea realizar alguna acción sobre un elemento de contenido de tipo tabla. El sistema brinda la posibilidad de realizar acciones de insertar, modificar o eliminar una tabla. En caso que seleccione la opción de insertar una tabla, el sistema dará la posibilidad de insertar una tabla. Si el arquitecto de información selecciona la opción eliminar una tabla, el sistema lo eliminará. Si el arquitecto de información elige la opción de modificar la tabla, el sistema permite aumentar / eliminar una nueva columna o fila a la tabla, así como modificar el texto de la misma. Terminando así el caso de uso.
Precondiciones:	En el caso de eliminar o modificar, el elemento debe estar

	seleccionado. Debe estar adicionada la plantilla en que se va a trabajar.
Referencias:	RF- 26, RF-27, RF-28.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Arquitecto de Información desea realizar alguna acción sobre los elementos del contenido de tipo tabla.	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Insertar Tabla. Sección1: “Insertar Tabla”. • Eliminar Tabla. Ver Sección2: “Eliminar Tabla”. • Modificar Tabla. Ver Sección3: “Modificar Tabla”.
3. El Arquitecto de Información selecciona la acción que desea realizar.	4. El caso de uso termina.
Sección1: “Insertar Tabla”	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de insertar una tabla.	2. El sistema brinda la posibilidad de: Seleccionar <ul style="list-style-type: none"> • Número de Filas. • Número de columnas. Y Permite: <ul style="list-style-type: none"> • Crear la tabla. • Cancelar la acción.
3. El arquitecto de Información escoge la cantidad de fila y de columnas deseadas y la acción de crear la tabla.	4. El sistema muestra la tabla insertada.
	5. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*a. El actor selecciona la opción de cancelar.	*a.1. Regresa a la vista anterior.

	*a.2. El caso de uso termina.
Sección2: "Eliminar Tabla"	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información escoge la tabla que desea eliminar y selecciona la acción de eliminar.	2. El sistema elimina el elemento seleccionado
	3. El caso de uso termina.
Sección3: "Modificar Tabla"	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información escoge la tabla que desea modificar.	2. El sistema permite aumentar / eliminar una nueva columna o fila a la tabla. Así como modificar el texto de la misma.
	3. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Gestionar Tópico.

Caso de Uso:	Gestionar Tópico.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el arquitecto de información desea realizar alguna acción sobre un elemento de contenido de tipo tópico. El sistema brinda la posibilidad de realizar acciones de insertar, eliminar un tópico, así como modificar su nombre. En caso que seleccione la opción de insertar un tópico, el sistema dará la posibilidad de insertar un tópico específico en el área de trabajo. Si el arquitecto de información selecciona la opción eliminar un tópico, el sistema lo eliminará. Si el arquitecto de información elige la opción de modificar el nombre del tópico, el sistema mostrará el tópico seleccionado con el nombre de manera editable y una vez realizados

	los cambios, guardará las modificaciones. Terminando así el caso de uso.
Precondiciones:	En el caso eliminar o modificar el elemento debe estar seleccionado. Debe estar adicionada la plantilla a la que se le desea adicionar un tópico.
Referencias:	RF-29, RF-30, RF-31.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Arquitecto de Información desea realizar alguna acción sobre los elementos del contenido de tipo tópico.	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Insertar tópico. Sección1: “Insertar Tópico”. • Eliminar tópico. Ver Sección2: “Eliminar Tópico”. • Modificar el nombre del tópico. Ver Sección3: “Modificar Tópico”.
3. El Arquitecto de Información selecciona la acción que desea realizar.	4. El caso de uso termina.
Sección1: “Insertar Tópico”	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de insertar un Tópico.	2. El sistema visualiza una interfaz con el tópico insertado. Y permite: <ul style="list-style-type: none"> • Nombrar el tópico.
3. El arquitecto de Información nombra el nuevo tópico insertado.	4. El caso de uso termina.
Sección2: “Eliminar Tópico”	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información escoge el tópico que desea eliminar y selecciona la acción de eliminar.	2. El sistema elimina el elemento seleccionado.
	3. El caso de uso termina.

Sección3: “Modificar Tópico”	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información selecciona el tópico que desea renombrar.	2. El sistema visualiza una interfaz con el tópico que desea renombrar. Y permite: <ul style="list-style-type: none"> • Renombrar el tópico.
3. El arquitecto de Información renombra el tópico.	4. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] --> B[UI. Plantilla] C[UI. Insertar Topico] -- Insertar --> B D[UI. Eliminar Topico] -- Eliminar --> B E[UI. Modificar Topico] -- Modificar --> B </pre>	
Poscondiciones:	Se insertó o modificó o eliminó un tópico en la plantilla.

Descripción Textual del Caso de Uso del sistema: Gestionar Relación.

Caso de Uso:	Gestionar Relación.
--------------	---------------------

Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el arquitecto de información desea realizar alguna acción sobre un elemento de contenido de tipo relación. El sistema brinda la posibilidad de realizar acciones de insertar, eliminar una relación entre los tópicos. En caso que seleccione la opción de insertar una relación, el sistema dará la posibilidad de insertar una relación al tópico específico en el área de trabajo. Si el arquitecto de información selecciona la opción eliminar una relación, el sistema lo eliminará. Terminando así el caso de uso.
Precondiciones:	En el caso eliminar el elemento debe estar seleccionado. Debe estar adicionada la plantilla a la que se le desea adicionar un tópico. Debe estar insertado un tópico.
Referencias:	RF-32, RF-33.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Arquitecto de Información desea realizar alguna acción sobre los elementos del contenido de tipo tópico.	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Insertar relación a un tópico. Sección1: “Insertar relación”. • Eliminar relación. Ver Sección2: “Eliminar relación”.
3. El Arquitecto de Información selecciona la acción que desea realizar.	4. El caso de uso termina.
Sección1: “Insertar relación”	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de insertar una relación al tópico insertado.	2. El sistema visualiza en el área de componentes los elementos tipo relación.
3. El usuario arrastra del área de componente el elemento relación.	4. El sistema muestra una vista de los tópicos y las relaciones insertadas.
	4. El caso de uso termina.

Sección2: “Eliminar relación”	
Acción del Actor	Respuesta del Sistema
1. El arquitecto de información escoge la relación que desea eliminar y selecciona la acción de eliminar.	2. El sistema elimina el elemento seleccionado.
	3. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] --> B[UI. Plantilla] B -- Insertar --> C[UI. Insertar relación] B -- Eliminar --> D[UI. Eliminar relación] </pre>	
Poscondiciones:	Se insertó o eliminó una relación a un tópico en la plantilla.

Descripción Textual del Caso de Uso del sistema: Configurar espacio de trabajo.

Caso de Uso:	Configurar espacio de trabajo.
Actores:	Arquitecto de Información (inicia).

Resumen:	El Caso de Uso se inicia cuando el usuario desea personalizar el área de trabajo. El sistema brinda la posibilidad de mostrar y ocultar el árbol de proyecto, el área de botones y la barra de herramienta. El usuario selecciona la acción que desea realizar para trabajar con más comodidad. El sistema actualiza la vista de la herramienta con los cambios realizados. Terminando así el caso de uso.	
Precondiciones:	-	
Referencias:	RF-34, RF-35, RF-36, RF-37, RF-38, RF-39.	
Prioridad:	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el usuario desea personalizar su entorno de trabajo.	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Ocultar/Mostrar el área de botones. Ver Sección 1: “Ocultar/Mostrar Área de Botones”. • Ocultar/Mostrar el árbol de proyecto. Ver Sección 2: “Ocultar/Mostrar Árbol de Proyecto”. • Ocultar/Mostrar barra de herramienta. Ver Sección 3: “Ocultar/Mostrar barra de herramienta”. 	
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.	
Sección 1: “Ocultar/Mostrar Área de Botones”		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la acción de ocultar/mostrar área de botones.	2. El sistema brinda la posibilidad de que este visible o no el área de botones, ubicado debajo de la zona editable de la herramienta.	
	3. El caso de uso termina.	

Sección 2: “Ocultar/Mostrar Árbol de Proyecto”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de ocultar/mostrar área de botones.	2. El sistema brinda la posibilidad de que este visible o no el área de botones, ubicado debajo de la zona editable de la herramienta.
	3. El caso de uso termina.
Sección 3: “Ocultar/Mostrar barra de herramienta”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de ocultar/mostrar barra de herramienta	2. El sistema brinda la posibilidad de que este visible o no la barra de herramienta, ubicado debajo de la barra de menú.
	3. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] -- Seleccionar --> B[UI. Ocultar/Mostrar Area de Botones] A -- Seleccionar --> C[UI. Ocultar/Mostrar Arbol de Proyecto] A -- Seleccionar --> D[UI. Ocultar/Mostrar Barra de Herramienta] </pre>	

Poscondiciones:	Se mostró / ocultó el árbol de proyecto o el área de botones o la barra de herramienta.
------------------------	---

Descripción Textual del Caso de Uso del sistema: Dar formato al texto.

Caso de Uso:	Dar formato al texto.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea darle formato a un texto. El sistema brinda al usuario las posibilidades de cambiar el texto en Negrita, Cursiva, Subrayado, Viñetas, Numeración. El usuario selecciona la acción a realizar. El sistema una vez realizados los cambios, guardará las modificaciones. Terminando así el caso de uso.
Precondiciones:	Tiene que estar el texto seleccionado.
Referencias:	RF-40, RF-41, RF-42, RF-43. RF-44.
Prioridad:	Crítico.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea darle formato a un texto.	2. El sistema brinda la posibilidad de aplicarle los siguientes formato al texto: <ul style="list-style-type: none"> • Negrita. Ver Sección1: “Texto en Negrita”. • Cursiva. Ver Sección2: “Texto en Cursiva”. • Subrayado. Ver Sección3: “Texto Subrayado”. • Viñetas. Ver Sección4: “Viñetas”. • Numeración. Ver Sección5: “Numeración”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.

Sección1: “Texto en Negrita”

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

1. El usuario señala el texto al que desea cambiarle el formato y selecciona la acción cambiar la letra a negrita.	2. El sistema muestra el texto seleccionado resaltado en negro.
	3. El caso de uso termina.
Sección2: "Texto en Cursiva"	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto al que desea cambiarle el formato y selecciona la acción cambiar la letra a cursiva.	2. El sistema muestra el texto seleccionado inclinado hacia la derecha.
	3. El caso de uso termina.
Sección3: "Texto Subrayado"	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto al que desea cambiarle el formato y selecciona la acción cambiar la letra a subrayado.	2. El sistema muestra el texto seleccionado encima de una raya.
	3. El caso de uso termina.
Sección4: "Viñetas"	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto al que desea cambiarle el formato y selecciona la acción cambiar la letra a viñetas.	2. El sistema muestra una lista con los diferentes estilos de viñetas (símbolos, imágenes). Y brinda la posibilidad de seleccionar alguna.
3. El usuario selecciona el estilo deseado.	4. Muestra el texto seleccionado a modo de lista con el nuevo estilo de viñetas.
	5. El caso de uso termina.
Sección5: "Numeración"	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto al que desea cambiarle el formato y selecciona la acción cambiar la letra a numeración.	2. El sistema muestra una lista con los diferentes estilos de números.
3. El usuario selecciona el estilo deseado.	4. Muestra el texto seleccionado a modo

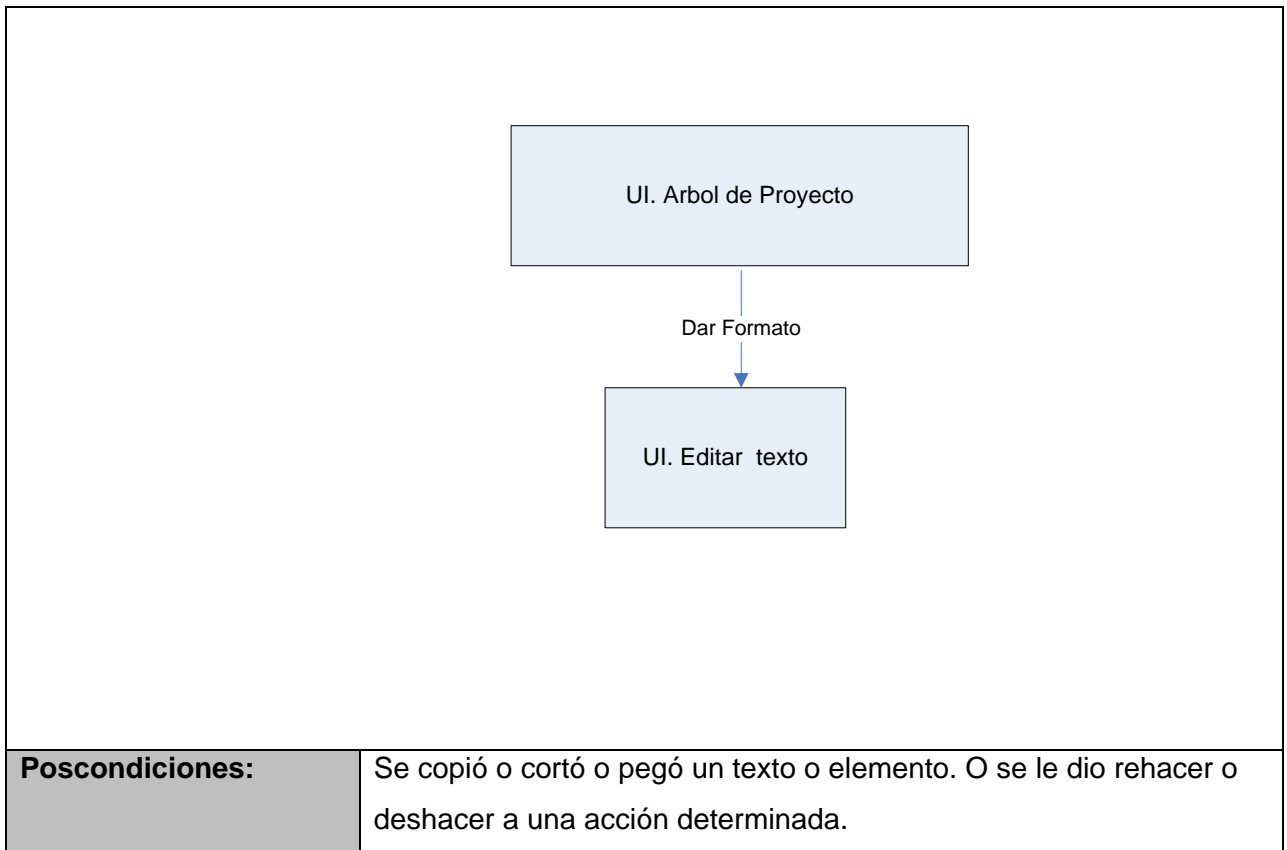
	de lista con el nuevo estilo de numeración deseado.
	5. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] -- Dar Formato --> B[UI. Formato al texto] </pre>	
Poscondiciones:	Se cambió el formato de un texto a Negrita o a Cursiva o a Subrayado o a Viñetas o a Numeración.

Descripción Textual del Caso de Uso del sistema: Editar texto / elemento.

Caso de Uso:	Editar texto / elemento.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea realizar alguna acción sobre un texto o un elemento en específico. El sistema brinda la posibilidad de realizar acciones de copiar, cortar, pegar,

	rehacer, deshacer texto o elemento seleccionado. El usuario selecciona la acción que desea realizar. El sistema una vez realizados los cambios, guardará las modificaciones. Terminando así el caso de uso.
Precondiciones:	En el caso de copiar, cortar, pegar tiene que estar el texto / elemento seleccionado.
Referencias:	RF-45, RF-46, RF-47, RF-48. RF-49.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea realizar alguna acción sobre un texto o un elemento en específico.	2. El sistema brinda la posibilidad de realizar las siguientes acciones al texto / elemento seleccionado: <ul style="list-style-type: none"> • Copiar un texto / elemento. Ver Sección1: “Copiar un texto /elemento”. • Cortar un texto /elemento. Ver Sección2: “Cortar un texto / elemento”. • Pegar un texto / elemento. Ver Sección3: “Pegar un texto / elemento”. • Rehacer acción. Ver Sección4: “Rehacer acción”. • Deshacer acción. Ver Sección5: “Deshacer acción”.
3. El usuario selecciona el texto o elemento y la acción que desea realizar.	4. El caso de uso termina.
Sección1: “Copiar un texto /elemento”	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto / elemento que desea y selecciona la acción copiar lo seleccionado.	2. El sistema realiza una copia del texto o elemento seleccionado, para

	posteriormente reproducirlo en otro fragmento del documento.
	3. El caso de uso termina.
Sección2: “Cortar un texto / elemento”	
Acción del Actor	Respuesta del Sistema
1. El usuario señala el texto / elemento que desea y selecciona la acción cortar lo seleccionado.	2. El sistema elimina (transitoriamente) el texto o elemento seleccionado, para posteriormente colocarlo en otro fragmento del documento.
	3. El caso de uso termina.
Sección3: “Pegar un texto / elemento”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción pegar un texto / elemento, seleccionado anteriormente en cualquiera de las acciones de copiar o cortar.	2. El sistema coloca en la posición indicada el texto / elemento que fue cortado o copiado anteriormente por el usuario.
	3. El caso de uso termina.
Sección4: “Rehacer acción”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción rehacer una acción realizada previamente por el usuario.	2. El sistema repite la última acción realizada por el usuario.
	5. El caso de uso termina.
Sección5: “Deshacer acción”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción deshacer una acción realizada previamente por el usuario.	2. El sistema descarta los cambios hechos recientemente por el usuario.
	5. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	

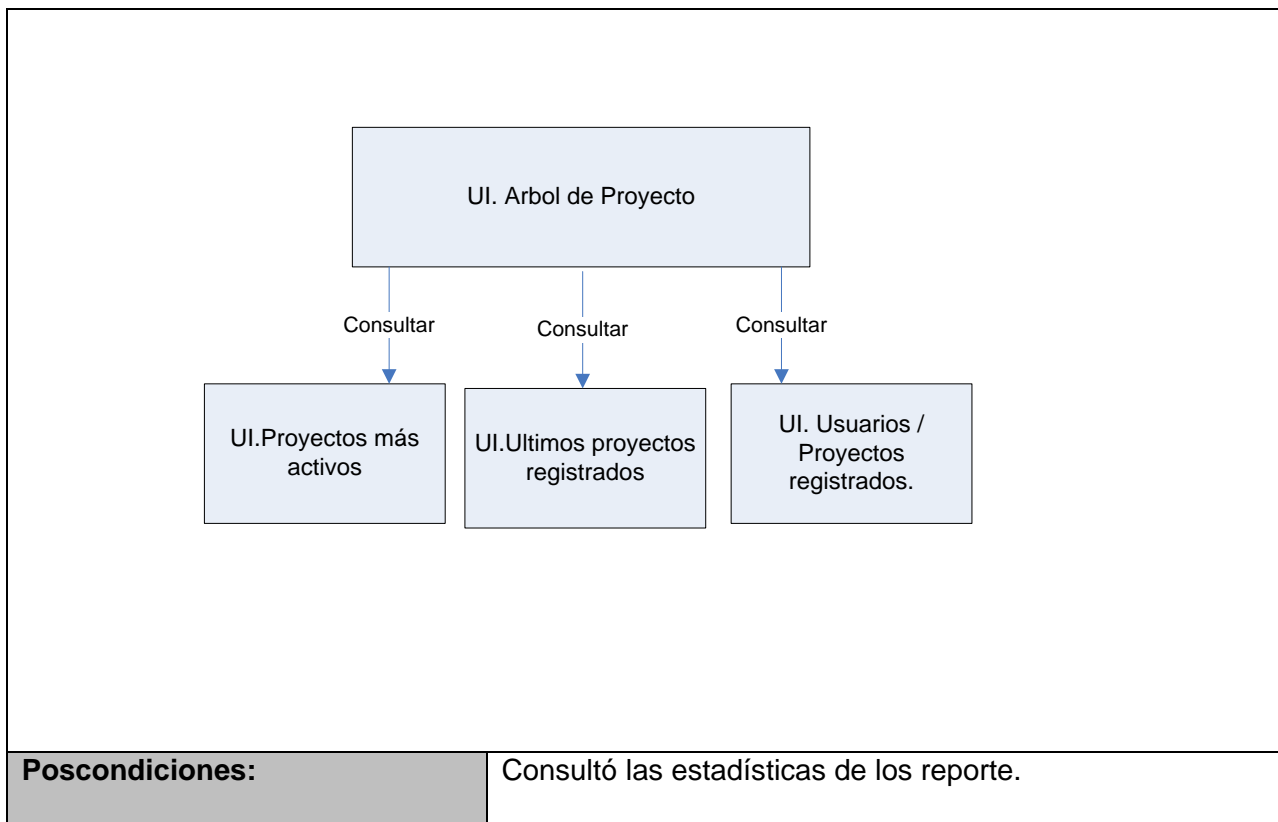


Descripción Textual del Caso de Uso del sistema: Mostrar estadísticas.

Caso de Uso:	Mostrar estadísticas.
Actores:	Administrador (inicia).
Resumen:	<p>El Caso de Uso se inicia cuando el Administrador desea ver alguna de las estadísticas de los reporte del servidor como son, usuarios / proyecto registrados, proyectos más activos de la semana y los últimos proyectos registrados. Si el usuario desea ver los usuarios / proyecto registrados. El sistema muestra un listado con todos los usuarios y proyectos existentes. Si el usuario desea ver los proyectos más activos de la semana. El sistema muestra un listado con los proyectos donde más se han trabajado. Si el usuario desea ver los últimos proyectos registrados. El sistema muestra un listado con los últimos proyectos que han sido registrados en el servidor en la última semana.</p>

	Terminando así el caso de uso.
Precondiciones:	Debe estar conectado al servidor.
Referencias:	RF-50, RF-51, RF-52, RF-53.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario desea ver alguna de las estadísticas de los reporte del servidor.	2. El sistema brinda la posibilidad de realizar las siguientes acciones: <ul style="list-style-type: none"> • Usuarios / proyecto registrados. Ver Sección1: “Usuarios / proyectos registrados”. • Proyectos más activos de la semana. Ver Sección2: “Proyectos más activos de la semana”. • Últimos proyectos registrados. Ver Sección3: “Últimos proyectos registrados”.
3. El usuario selecciona la acción que desea realizar.	4. El caso de uso termina.
Sección1: “Usuarios / proyectos registrados”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción de ver los usuarios / proyectos registrados.	El sistema muestra un listado con los usuarios y proyectos registrados. Y permite: <ul style="list-style-type: none"> • Filtrar por usuario • Filtrar por proyecto
3. Introduce los datos que desea buscar y selecciona la acción de filtrar por criterio de búsqueda.	4. <i>El sistema realiza la búsqueda.</i>
	5. El sistema muestra el resultado de la búsqueda.

	6. El Caso de uso termina.
Sección2: “Proyectos más activos de la semana”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción de ver los proyectos más activos.	2. El sistema muestra un listado con los proyectos más activos. Y permite: <ul style="list-style-type: none"> • Filtrar por proyecto.
3. Introduce el nombre del proyecto que desea buscar y selecciona la acción de filtrar.	4. <i>El sistema realiza la búsqueda.</i>
	5. El sistema muestra el resultado de la búsqueda.
	6. El Caso de uso termina.
Sección3: “Últimos proyectos registrados”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la acción de ver los últimos proyectos registrados.	2. El sistema muestra un listado con los últimos proyectos registrados de la semana. Y permite: <ul style="list-style-type: none"> • Filtrar por proyecto.
3. Introduce el nombre del proyecto que desea buscar y selecciona la acción de filtrar.	4. <i>El sistema realiza la búsqueda.</i>
	5. El sistema muestra el resultado de la búsqueda
	6. El Caso de uso termina.
Prototipo de Interfaz	
Ver Anexo: <ul style="list-style-type: none"> • UI. Reportes 	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Mostrar trazabilidad de usuario.

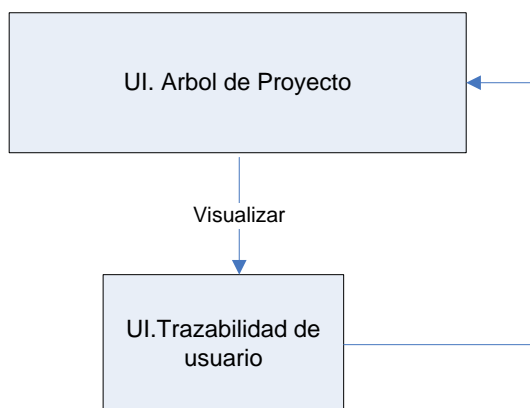
Caso de Uso:	Mostrar trazabilidad de usuario.
Actores:	Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el Administrador desea ver las acciones realizadas por los usuarios. El sistema muestra cada una de ellas y permite filtrar por usuario. El sistema muestra un listado con todos los usuarios. Terminando así el caso de uso.
Precondiciones:	Debe estar conectado al servidor.
Referencias:	RF-54.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. El caso de uso inicia cuando el Administrador desea ver las acciones realizadas por los usuarios.	2. El sistema muestra un listado con los usuarios registrados y las acciones correspondientes. Y permite filtrar por usuario.
3. Introduce el nombre del usuario que desea buscar y selecciona la acción de filtrar.	4. <i>El sistema realiza la búsqueda.</i>
	5. El sistema muestra el resultado de la búsqueda
	6. El Caso de uso termina.

Prototipo de Interfaz

-

Mapa de navegación

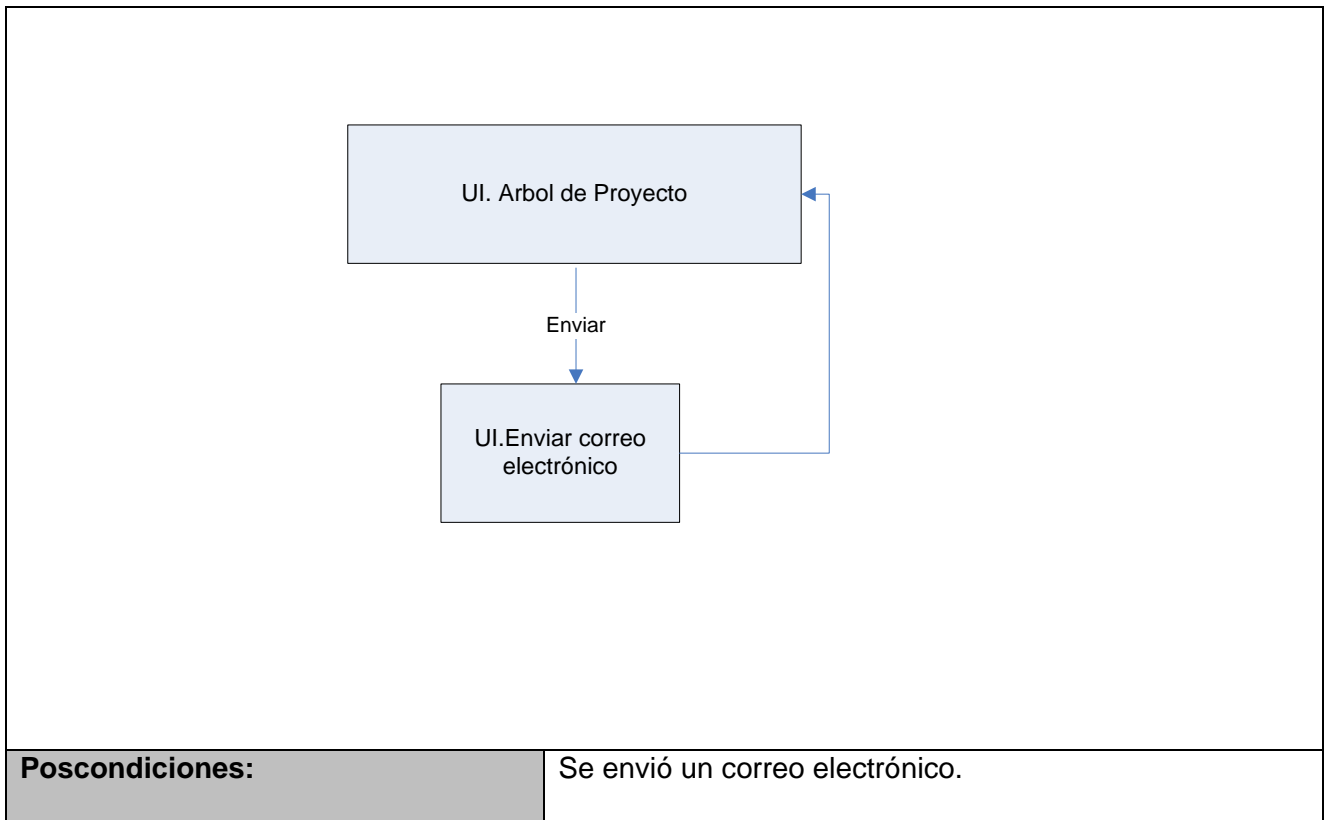


Poscondiciones:	Consultó los registros de traza.
------------------------	----------------------------------

Descripción Textual del Caso de Uso del sistema: Enviar correo electrónico.

Caso de Uso:	Enviar correo electrónico.
Actores:	Arquitecto de Información, Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea enviar

	por correo electrónico alguna plantilla o información. El sistema permite enviar un correo electrónico a cualquier usuario. Terminando así el caso de uso.
Precondiciones:	Deben estar creadas la plantilla o la información a enviar. En caso de ser una plantilla debe haber sido seleccionada.
Referencias:	RF-55.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el usuario decide enviar por correo una plantilla o alguna información.	2. El sistema genera una ventana tradicional de enviar un correo electrónico, dejándole introducir el destinatario del mismo, si desea mandarle copia a otra persona y el asunto, así como el texto deseado. Y permite <ul style="list-style-type: none"> • Enviar el correo a la persona.
3. El usuario escribe todo los datos correspondientes y selecciona la opción de enviar el correo.	4. Regresa a la vista anterior.
	5. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo: <ul style="list-style-type: none"> • UI. Enviar por Correo. 	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Exportar a PDF.

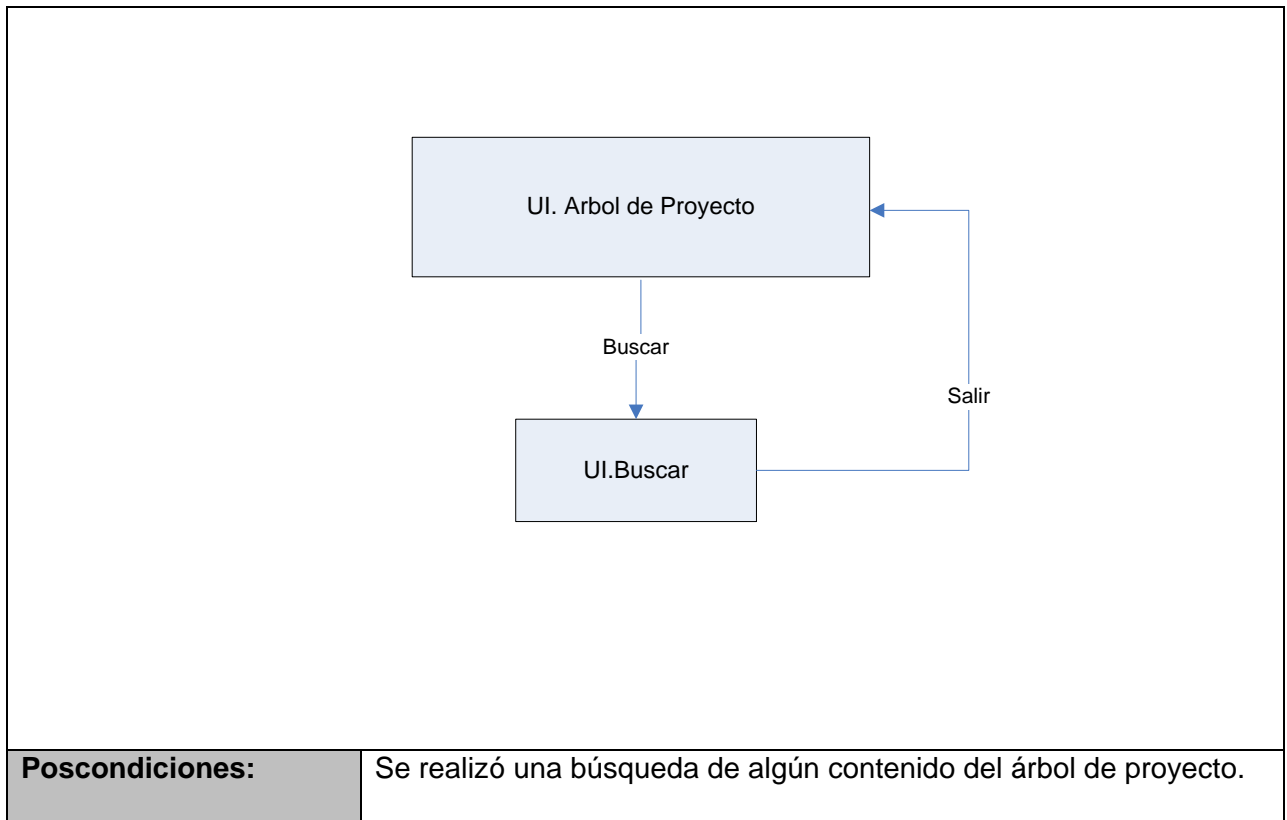
Caso de Uso:	Exportar a PDF.
Actores:	Arquitecto de Información, Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea exportar un documento a PDF. El sistema toma los datos que se desea exportar y genera un archivo PDF. Terminando así el caso de uso. Precondiciones: Debe estar creada la información a exportar.
Precondiciones:	Se generó un documento en formato PDF.
Referencias:	RF-56.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el actor accede	2. El sistema exporta lo seleccionado por el

a la opción de exportar a PDF los datos que se muestran.	actor, en el formato PDF.
	3. Genera un archivo PDF con la información correspondiente.
	4. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo:	
<ul style="list-style-type: none"> • UI. Exportar PDF. 	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] --> B[UI. Exportar a PDF] </pre>	
Poscondiciones:	Se exportó a PDF.

Descripción Textual del Caso de Uso del sistema: Buscar contenido local.

Caso de Uso:	Buscar contenido local.
Actores:	Arquitecto de Información.
Resumen:	El Caso de Uso se inicia cuando el Arquitecto de información o el administrador desean buscar algún contenido en la herramienta en

	el árbol de proyecto, ya sea algún proyecto, plantilla, entre otros. El sistema permite realizar la búsqueda a través de un buscador central. Terminando así el caso de uso.	
Precondiciones:	-	
Referencias:	RF-57.	
Prioridad:	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor desea buscar algún contenido en la Herramienta en el árbol de proyecto.	2. El sistema brinda la posibilidad de buscar algún contenido en el buscador de la herramienta.	
3. El actor introduce la palabra que desea buscar.	4. <i>El sistema realiza la búsqueda en el árbol de proyecto.</i>	
	5. Muestra un listado con todo el resultado de búsqueda.	
	6. El caso de uso termina.	
Prototipo de Interfaz		
-		
Mapa de navegación		



Descripción Textual del Caso de Uso del sistema: Confeccionar plantillas.

Caso de Uso:	Confeccionar plantillas.
Actores:	Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el usuario desea confeccionar las plantillas. El sistema brinda la posibilidad de adicionar y eliminar secciones a las plantillas y definir los componentes para cada plantilla. El administrador selecciona la acción que desea realizar. Terminando así el caso de uso.
Precondiciones:	Debe estar conectado al servidor.
Referencias:	RF-58, RF-59, RF-60.
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

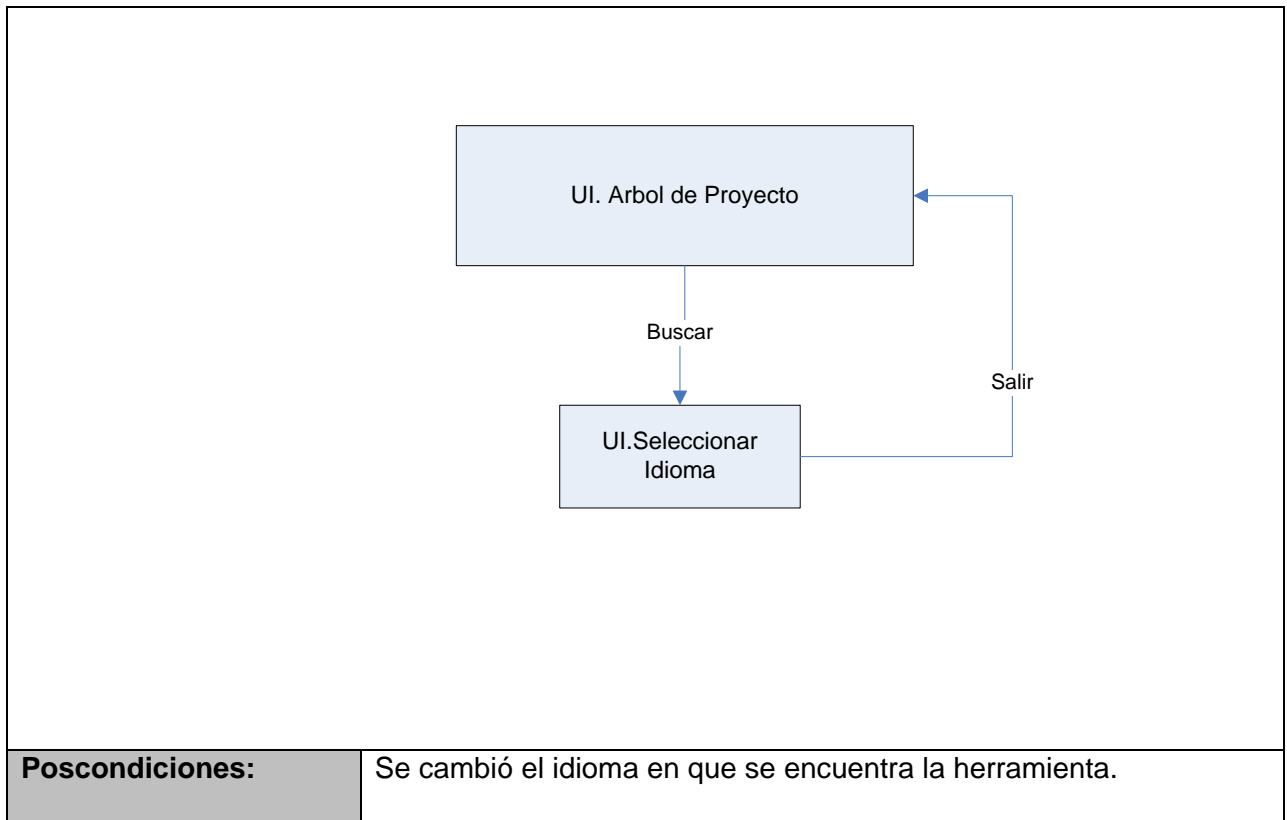
1. El caso de uso inicia cuando el usuario desea configurar las plantillas.	2. El sistema brinda la posibilidad de: <ul style="list-style-type: none"> • Adicionar secciones a la plantilla. Ver Sección 1: “Adicionar sección”. • Eliminar secciones a la plantilla. Ver Sección 2: “Eliminar sección”. • Definir componentes a la plantilla. Ver Sección 3: “Componentes de la plantilla”.
3. Selecciona una de las posibles acciones.	4. El caso de uso termina.
Sección 1: “Adicionar sección”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de adicionar sección.	2. El sistema brinda la posibilidad de adicionarle secciones a la plantilla.
3. El usuario arrastra del área de componente las secciones que desea adicionarle a la plantilla.	4. El sistema visualiza la plantilla confeccionada por el administrador.
	5. El caso de uso termina.
Sección 2: “Eliminar sección”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de eliminar sección.	2. El sistema brinda la posibilidad de eliminar secciones a la plantilla.
3. El usuario elimina las secciones que desea a la plantilla.	4. El sistema visualiza la plantilla sin las secciones que le fueron eliminadas previamente por el administrador.
	5. El caso de uso termina.
Sección 3: “Componentes de la plantilla”	
Acción del Actor	Respuesta del Sistema
1. Selecciona la acción de definir componentes a la plantilla.	2. El sistema brinda la posibilidad de realizar una búsqueda de la ubicación de los componentes que desea insertar. Y permite: <ul style="list-style-type: none"> • Insertar los componentes.

	<ul style="list-style-type: none"> • Cancelar la acción.
3. El administrador realiza la búsqueda de la ubicación donde se encuentran guardados los componentes en la PC, los selecciona, y escoge la acción insertar los componentes.	4. El sistema muestra los componentes insertados.
	5. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo: <ul style="list-style-type: none"> • UI. Nueva Plantilla 	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] --> B[UI. Adicionar seccion] A --> C[UI. Eliminar seccion] A --> D[UI. Definir componentes] </pre>	
Poscondiciones:	Se confeccionó la plantilla.

Descripción Textual del Caso de Uso del sistema: Configurar Idioma.

Caso de Uso:	Configurar Idioma.
Actores:	Arquitecto de Información, Administrador (inicia).
Resumen:	El Caso de Uso se inicia cuando el Arquitecto de información o el administrador desean cambiar el idioma predefinido por la herramienta. El sistema permite cambiar los idiomas de English a español o viceversa. Terminando así el caso de uso.

Precondiciones:	-
Referencias:	RF-61.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el actor desea cambiar el idioma predefinido por la herramienta.	2. El sistema brinda la posibilidad de hacer un cambio de idioma de: <ul style="list-style-type: none"> • English a Español • Español a English.
3. El actor selecciona el idioma deseado.	4. <i>El sistema realiza el cambio de idioma.</i>
	5. Muestra la herramienta actualizada con el idioma seleccionado.
	6. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	



Descripción Textual del Caso de Uso del sistema: Buscar contenido en el servidor.

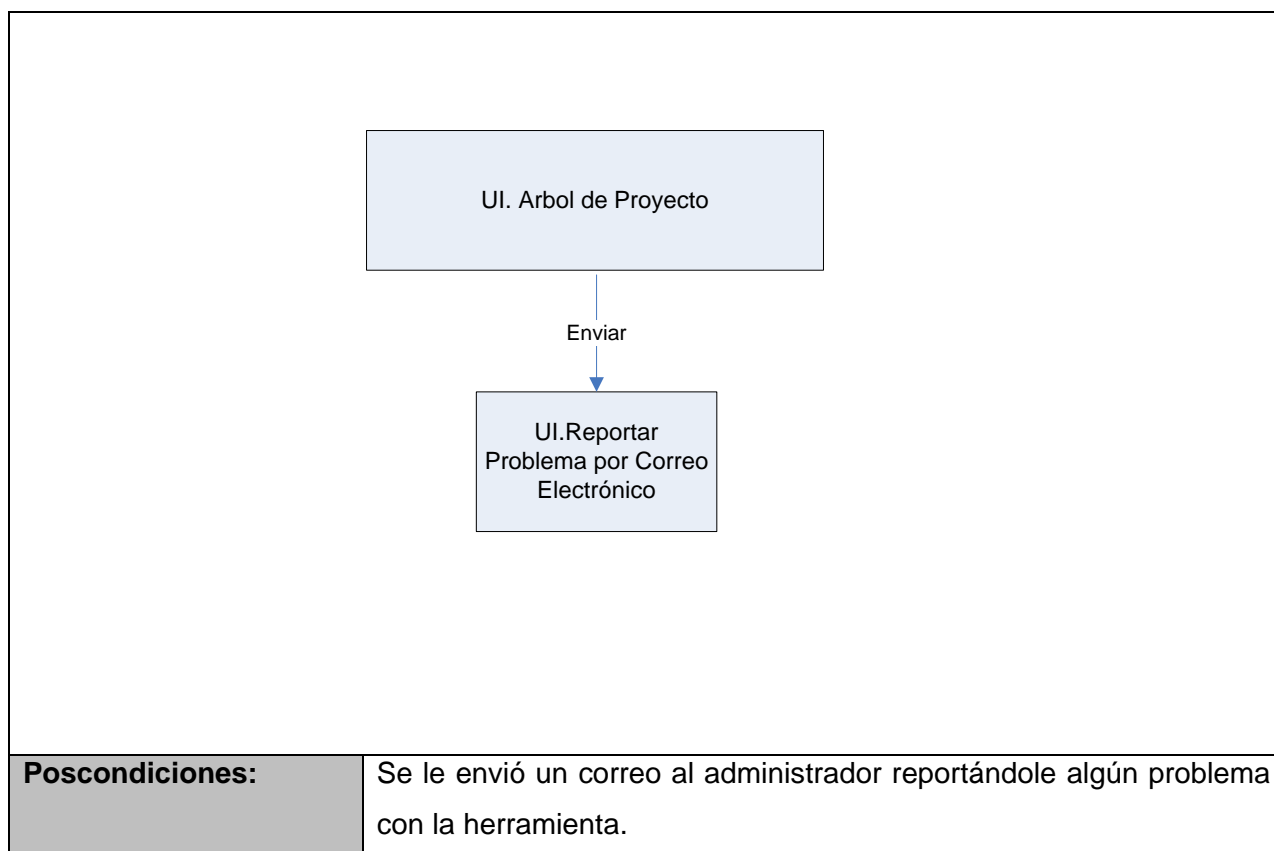
Caso de Uso:	Buscar contenido en el servidor.	
Actores:	Administrador.	
Resumen:	El Caso de Uso se inicia cuando el Administrador desea buscar algún contenido en la herramienta en el árbol de plantilla o en el árbol de usuario, ya sea algún proyecto, plantilla, usuario entre otros. El sistema permite realizar la búsqueda a través de un buscador central. Terminando así el caso de uso.	
Precondiciones:	Debe de estar conectado al servidor.	
Referencias:	RF-62.	
Prioridad:	Crítico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso se inicia cuando el actor desea buscar algún contenido en la Herramienta en el	2. El sistema brinda la posibilidad de buscar algún contenido en el buscador de	

árbol de plantilla o en el árbol de usuario	la herramienta.
3. El actor introduce la palabra que desea buscar.	4. <i>El sistema realiza la búsqueda en el árbol de plantilla o en el árbol de usuario.</i>
	5. Muestra un listado con todo el resultado de búsqueda.
	6. El caso de uso termina.
Prototipo de Interfaz	
-	
Mapa de navegación	
<pre> graph TD A[UI. Arbol de Proyecto] -- Buscar --> B[UI. Buscar] B -- Salir --> A </pre>	
Poscondiciones:	Se realizó una búsqueda de algún contenido del árbol de plantilla o del árbol de usuario.

Descripción Textual del Caso de Uso del sistema: Enviar correo electrónico al administrador.

Caso de Uso:	Enviar correo electrónico al administrador.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el Arquitecto de Información desea

	reportar algún problema al administrador. El sistema permite enviar un correo electrónico al administrador de la herramienta. Terminando así el caso de uso.
Precondiciones:	Debe estar conectado al servidor.
Referencias:	RF-63
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Arquitecto de Información decide reportar un problema al administrador.	2. El sistema brinda la posibilidad de enviar un correo electrónico al administrador de la herramienta, dejándole introducir el asunto, el contexto y algún adjunto.
3. El arquitecto de información introduce los problemas deseados en el área de contexto y selecciona la opción de enviar el correo.	4. <i>Envía el correo al administrador.</i>
	4. Regresa a la vista anterior.
	5. El caso de uso termina.
Prototipo de Interfaz	
Ver Anexo: <ul style="list-style-type: none"> • UI. Enviar por Correo 	
Mapa de navegación	

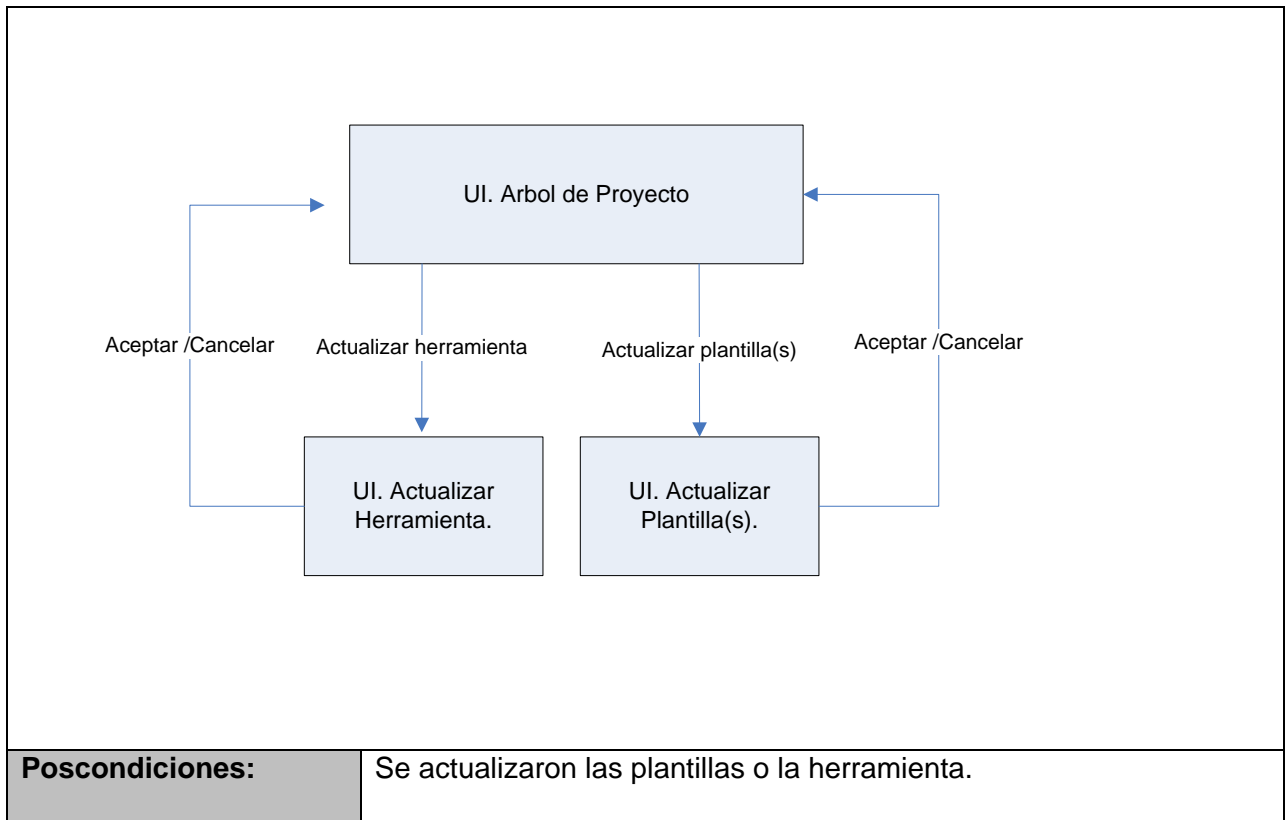


Descripción Textual del Caso de Uso del sistema: Verificar actualizaciones.

Caso de Uso:	Verificar actualizaciones.
Actores:	Arquitecto de Información (inicia).
Resumen:	El Caso de Uso se inicia cuando el Arquitecto de Información desea verificar si existen actualizaciones nuevas de la herramienta o de las plantillas en el servidor. El sistema en caso de tener nuevas actualizaciones brinda la posibilidad de actualizar la herramienta o las plantillas. Terminando así el caso de uso.
Precondiciones:	-
Referencias:	RF-64, RF-65.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el Arquitecto de	2. El sistema brinda la posibilidad de:

Información desea verificar si existen actualizaciones nuevas de la herramienta o de las plantillas en el servidor.	<ul style="list-style-type: none"> • Actualizar la Herramienta. Ver Sección 1 “Actualizar Herramienta”. • Actualizar la(s) plantilla(s). Ver Sección 2 “Actualizar Plantillas”.
3. El arquitecto de información selecciona una de las acciones.	4. El caso de uso termina.
Sección 1 “Actualizar Herramienta”	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de actualizar la herramienta.	2. <i>El sistema compara la herramienta que se está utilizando con la versión del servidor.</i> Y permite: <ul style="list-style-type: none"> • Actualizar la herramienta. • Cancelar la acción.
3. El arquitecto de información selecciona la acción de actualizar la herramienta.	4. <i>Actualiza la herramienta.</i>
	5. El sistema muestra un mensaje de información “La herramienta ha sido actualizada con la versión X”.
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*.a. El actor selecciona la opción de Cancelar.	*.a.1. Regresa a la vista anterior.
	*.a.2. El caso de uso termina.
2. a. El sistema compara la herramienta con la del servidor y no encuentra actualizaciones.	2. a.1 Muestra el mensaje de información “No se encontraron actualizaciones de la herramienta”.

	2. a.2. Regresa al paso 1 de la sección 1“Actualizar Herramienta”.
Sección 2 “Actualizar Plantillas”	
Acción del Actor	Respuesta del Sistema
1. El Arquitecto de Información selecciona la acción de actualizar la(s) plantilla(s).	2. <i>El sistema compara la(s) plantilla(s) que se está utilizando con la versión del servidor.</i> Y permite: <ul style="list-style-type: none"> • Actualizar la(s) plantilla(s). • Cancelar la acción.
3. El arquitecto de información selecciona la acción de actualizar la(s) plantilla(s).	4. <i>Actualiza la plantilla.</i>
	5. El sistema muestra un mensaje de información “La(s) plantilla(s) ha(n) sido actualizada(s)”.
	6. El caso de uso termina.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
*.a. El actor selecciona la opción de Cancelar.	*.a.1. Regresa a la vista anterior.
	*.a.2. El caso de uso termina.
2. a. El sistema compara la(s) plantilla(s) con la del servidor y no encuentra actualizaciones.	2. a.1 Muestra el mensaje de información “No se encontraron actualizaciones de la(s) plantilla(s)”.
	2. a.2. Regresa al paso 1 de la sección 2 “Actualizar Plantillas”.
Prototipo de Interfaz	
-	
Mapa de navegación	



2.4 Validación

La obtención de los artefactos desarrollados durante este trabajo, sirvió como entrada al proceso de diseño del software. Luego de transcurrida los flujos de Modelado de Negocio y Requerimientos de la Herramienta, esta se puso a consideración de los clientes mediante un prototipo no funcional de interfaz gráfica, culminando con la firma de una carta de aceptación donde afirmaban que se logró cumplir con sus expectativas y que los requerimientos que se definieron fueron los que cumplían con sus necesidades.

2.4.1 Prototipo no funcional de interfaz gráfica.

Al realizar la captura de requisitos, el cliente define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. Por esta razón es de importancia la construcción de prototipos al comienzo de la recolección de los requisitos. Se procede a

la creación de un diseño rápido, que se centra en una representación de los aspectos del software que serán visibles para el cliente.

El prototipo lo evalúa el cliente y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, siguiendo el ciclo que se muestra en la Figura 7, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que necesita hacer. (17)

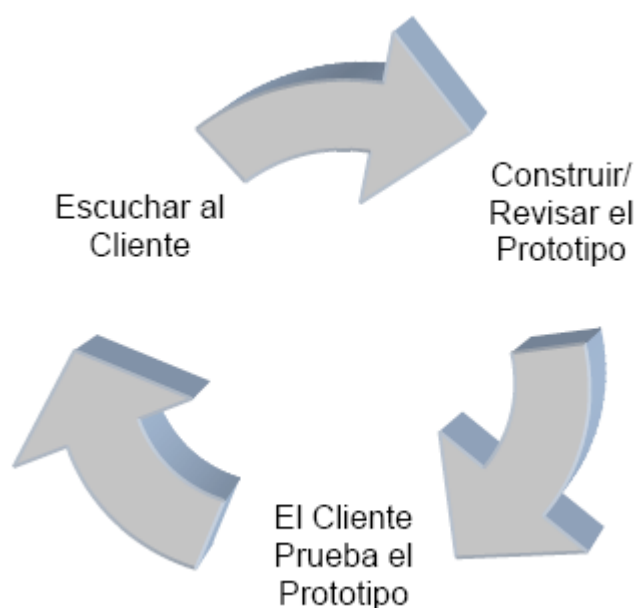


Figura7. Paradigma de Construcción de Prototipos.

En esta etapa, luego de definir los objetivos del software e identificar los requisitos junto al cliente, se procedió a la realización del prototipo, que sería la guía en el desarrollo posterior del diseño de la herramienta. Para la realización de este, se hizo una división estableciéndose prototipos de acuerdo a un área funcional determinada (Cliente - Administrador), que es representada por los casos de uso definidos en el Diagrama de Casos de Uso del Sistema, mostrado anteriormente.

La realización de los prototipos de interfaz de usuario aportó ventajas, pues sirvió como retroalimentación con los clientes, quienes ayudaron en el refinamiento de los mismos. Por cada caso de uso del sistema se crearon nuevos prototipos de interfaces no funcionales de la aplicación. Estas interfaces mostraron las funcionalidades que podría activar cada actor al interactuar con un caso de uso específico a través de formularios. La generación de los prototipos de interfaz fue una forma de

validar los requisitos funcionales y de usabilidad que fueron identificados durante el levantamiento de los requisitos.

2.4.2 Aceptación del Cliente

Luego de concluir el proceso de análisis de la Herramienta CASE AISOFT esta fue aceptada por los clientes, mostrando la validez del trabajo realizado y mediante la cual se expresa la conformidad con los requisitos definidos. Además donde queda plasmado que la herramienta se adapta perfectamente a las necesidades planteadas.

Acta de Aceptación por parte del cliente se presentan a continuación:



Carta de conformidad del cliente con los requisitos.

La Habana, 25 de Mayo del 2009.
"Año del 50 Aniversario del Triunfo de la Revolución"

A: Universidad de las Ciencias Informáticas.

Mediante el presente documento se expresa la conformidad con los requisitos de la Herramienta CASE AISOFT considerando que el trabajo realizado por los estudiantes se adapta perfectamente a las necesidades planteadas.

- El presente trabajo de diploma sirve de base a los futuros desarrolladores de la Herramienta, ya que les garantiza un mejor entendimiento de las funcionalidades del mismo, para posteriormente realizar las mejoras de la Herramienta con mayor facilidad.
- Los prototipos de interfaz no funcionales se adecuan completamente a las necesidades planteadas para la realización de la Herramienta.

De manera general los requisitos identificados constituyen una propuesta que integran muy bien las funcionalidades de una Herramienta CASE que asiste la Arquitectura de Información (AISOFT). Consideramos que el trabajo realizado por los estudiantes es muy bueno, y se ha realizado con profesionalidad.

Y para que así conste, se firma la presente a los 25 días del mes del mes de Mayo del año 2009.

Lic. Rodrigo Ronda León
Bibliotecología y Ciencias de la Información
Especialista en Información. DATYS



Ing. Renier Pérez-García
Director de la Dirección Técnica de la IP
Universidad de las Ciencias Informáticas

Conclusiones

En este capítulo se mostró el Modelo del Dominio utilizado para comprender mejor el funcionamiento del proceso de Arquitectura de Información, haciendo énfasis en su descripción y en la especificación de sus clases conceptuales. Se especificaron los requisitos funcionales y no funcionales de la Herramienta. Se identificaron los actores del sistema, se presentó el Modelo de Casos de Uso del Sistema, donde se realizó la expansión de los casos de uso identificados, mostrándose los prototipos no funcionales de cada uno de ellos en los anexos. Por último se utilizó como método para justificar la validez del trabajo realizado una carta de aceptación del cliente, que avala por sí misma los resultados que se debían conseguir con el análisis de la Herramienta y un prototipo no funcional de Interfaz grafica.

Conclusiones Generales

A partir de la investigación realizada se verificó que no existía una herramienta que abarcara completamente todos los procesos y tareas de la Arquitectura de Información, por lo que se decidió realizar un análisis a una propuesta de Herramienta CASE que asistiera a la misma.

Producto del estudio realizado se procede a las siguientes conclusiones:

- Como respuesta al problema originado, se llegó a la propuesta de realizar un análisis a una herramienta CASE que asistiera la Arquitectura de Información.
- Como primera fase se analizaron los procesos de la ingeniería de requerimientos.
- Se modelaron los artefactos correspondientes,
 - Se realizó la modelación del dominio definiendo sus clases conceptuales lo cual facilitó tener una mayor claridad sobre el entorno de los procesos de la Arquitectura de Información.
 - La especificación de los requisitos permitió definir las características principales y las funcionalidades con que debía contar la Herramienta.
 - Se definieron los actores y los casos de uso del sistema, realizando una descripción detallada de los flujos de eventos que debía seguir la Herramienta.
- La validación del modelo de casos de uso del sistema se realizó mediante un prototipo no funcional de interfaz gráfica, garantizando que los requisitos identificados cumplirían con las necesidades de los usuarios finales. Además se procedió a la firma de una carta de aceptación de los clientes identificados en el proceso.

Por lo que se puede ultimar que los artefactos generados durante el proceso de análisis, constituirán la entrada para el trabajo del diseñador de la herramienta y serán imprescindibles para que el equipo de desarrollo logre implementarla.

Recomendaciones.

Para la obtención de mejores resultados en la siguiente fase del proyecto, donde se pretenderá una versión mejorada de la Herramienta AISOF, con mayores beneficios para los arquitectos de información y muchas más ventajas para llevar a cabo el proceso, tareas y técnicas de la Arquitectura de Información se recomienda incluir las siguientes funcionalidades al sistema:

Técnicas

- Aplicar técnicas de recopilación de información.
- Incorporar técnicas de recopilación de información aplicadas a las diferentes plantillas a las que haga referencia.
- Realizar técnicas (on-line) a los diferentes usuarios a distancia.
- Incorporar tutoriales de ayuda en la aplicación de las técnicas de recopilación de información utilizadas por la Herramienta AISOF.

Además se recomienda continuar con el estudio del proceso de Arquitectura de Información para mayor conocimiento del mismo y en nuevas iteraciones de levantamiento de requisitos incorporar los nuevos avances de la misma.

Bibliografía

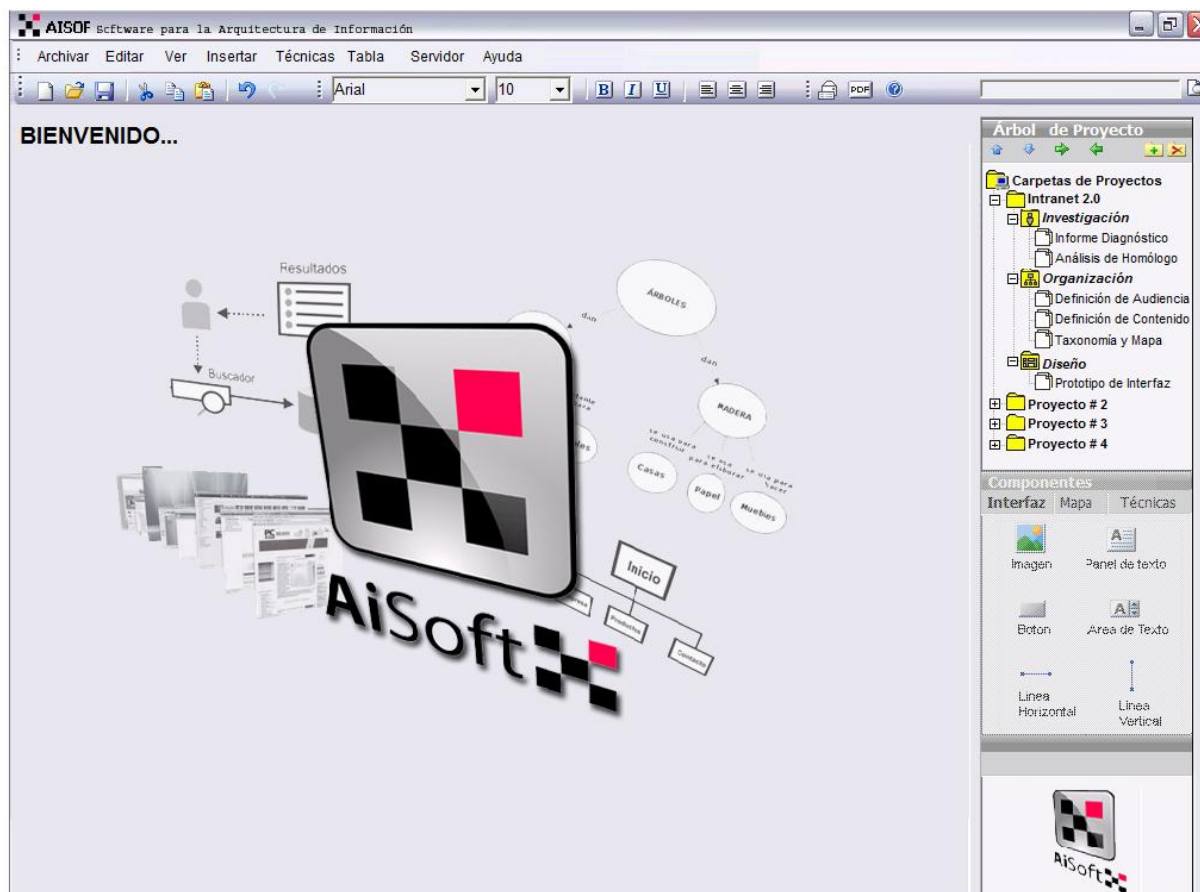
1. **Palacio, J.** *Gestión y procesos en empresas de software*. 2005.
2. **IAN, S. and S. PETER.** *Requirements engineering: A good practice guide*. Chinchester, Inglaterra : s.n., 2000.
3. **Ronda León, Rodrigo.** 2007, Vol. La diagramación en la arquitectura de información.
4. **DELFINO, E.** *Integración de herramientas CASE usando Internet, CORBA, y repositorios de Meta Información*. . s.l. : Universidad de la República Uruguay, (Facultad de Ingeniería), 2005.
5. **www.sap.com.** [Online] [Citado: diciembre 10, 2008.]
<http://www.sap.com/solutions/sapbusinessobjects/large/intelligenceplatform/bi/dashboard-visualization/advanced-visualization/index.epx>.
6. [Online] <http://cmap.ihmc.us/conceptmap.html>.
7. [Online] <http://www.uoc.edu/symposia/spdece05/pdf/ID23.pdf>.
8. [Online] <http://www.cs.umd.edu/hcil/qp/>.
9. **www.axure.com.** [Online] <http://www.axure.com/>.
10. **IEEE.** *Glosario estándar de la terminología de la ingeniería de software estándar*. 610.12-1990, 1997.
11. **DÁVILA, N. D.** *INGENIERÍA DE REQUERIMIENTOS UNA GUÍA PARA EXTRAER, ANALIZAR, ESPECIFICAR Y VALIDAR LOS REQUERIMIENTOS DE UN PROYECTO*. s.l. : Facultad de Ingeniería. Montevideo, UNIVERSIDAD ORT URUGUAY, 2002.
12. **DSIC.** *Rational Unified Process (RUP)*. s.l. : Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2000.
13. **ZAMBRANO, A. N. C.** *Herramienta para el analisis de requerimientos dentro de la pequeña empresa desarrolladazo de software en Bogotá*. . s.l. : Facultad de Ingeniería, Pontificia Universidad Javeriana, 2006.
14. **XP.** *Extreme Programming: A gentle introduction*. XP Extreme Programming. 2008.
15. **SCHWABER, K.** *Agile Project Management with Scrum* Microsoft Press. . 2004.
16. **JACABOSON and BOOCH, et al.** *El Proceso Unificado de Desarrollo de Software*. . s.l. : Addison Wesley, 2000.
17. **Pressman, Roger S.** *Ingeniería del Software. Un Enfoque Práctico.*, . [ed.] Quinta Edición. 2002.

-
18. **LTDA, M.** *Programas para la Arquitectura de la Información. Mapas Estructurales.* 2006.
 19. **JACOBSON and BOOCH, et al.** *El Proceso Unificado de Desarrollo de Software.* . 1999. Addison Wesley.
 20. **LARMAN, Craig.** *“UML y Patrones. Introducción al análisis y diseño orientado a objetos”.* s.l. : Primera Edición por Prentice Hall, Hispanoamericana S.A., 1999. Tomo I, Capítulo 18. .
 21. [www.microsoft.com.](http://office.microsoft.com/es-es/word/FX100487983082.aspx?ofcresset=1) [Online] [Citado: Diciembre 21, 2008.] <http://office.microsoft.com/es-es/word/FX100487983082.aspx?ofcresset=1>.
 22. [www.mindjet.com.](http://www.mindjet.com) [Online] [Citado: Diciembre 21, 2008.] <http://www.mindjet.com/products/mindmanager/default.aspx>.
 23. [www.microsoft.com.](http://office.microsoft.com/es-es/visio/FX100487863082.aspx) [Online] [Citado: Diciembre 21, 2008.] <http://office.microsoft.com/es-es/visio/FX100487863082.aspx>.
 24. [www.adobe.com.](http://www.adobe.com) [Online] <http://www.adobe.com/es/products/photoshop/photoshop/>.

Anexos.

Prototipo elemental de interfaz gráfica de usuario. Modulo Cliente

UI 1. Base



UI. Texto

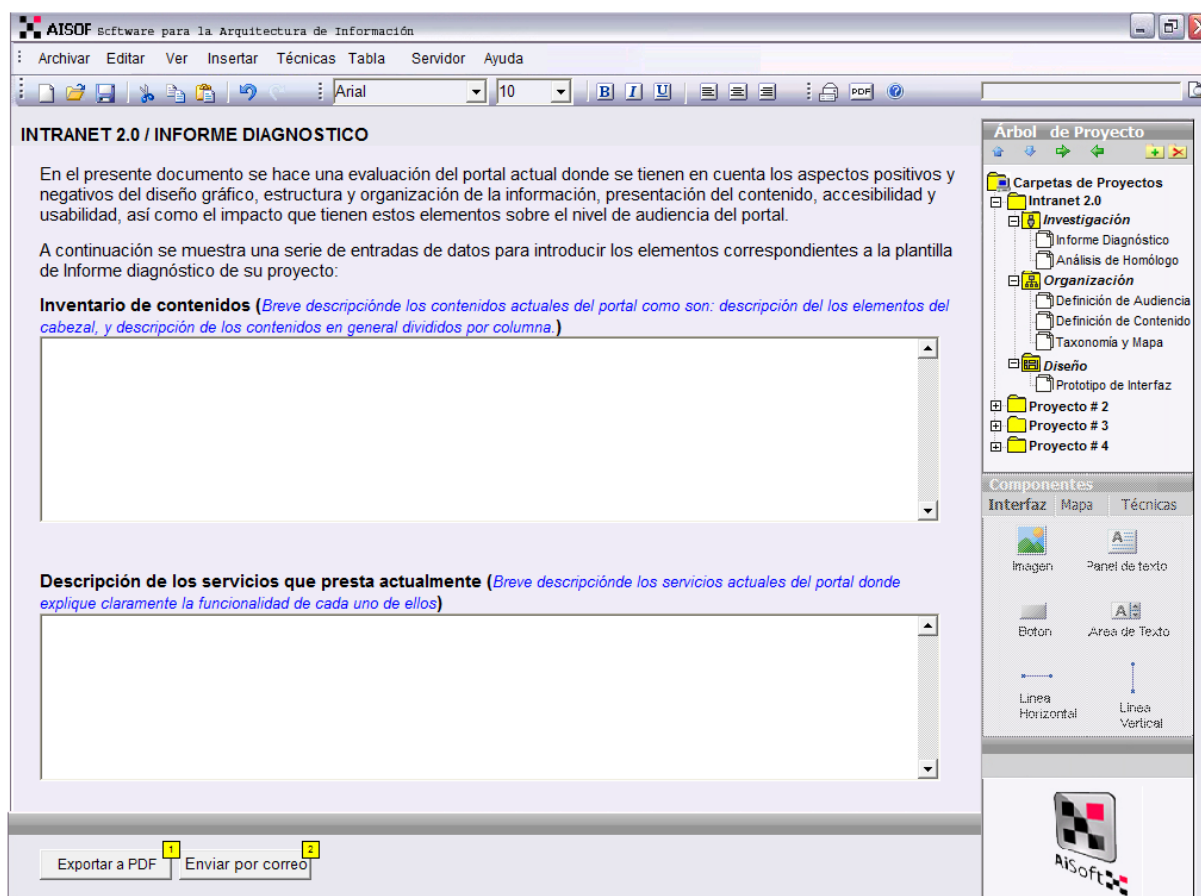


Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Exportar PDF en la Ventana actual.
2	Click	Muestra la UI Enviar por Correo en la Ventana actual.

UI. Exportar PDF

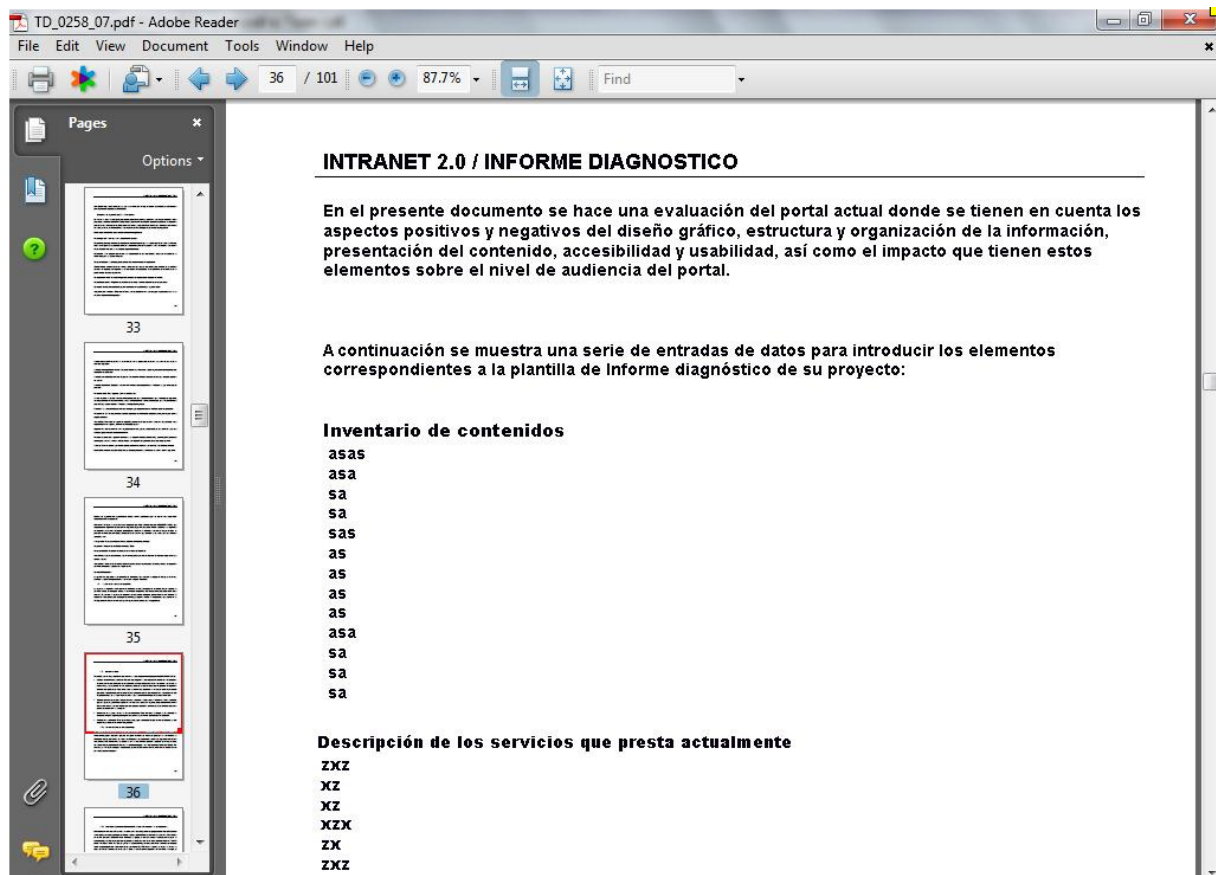
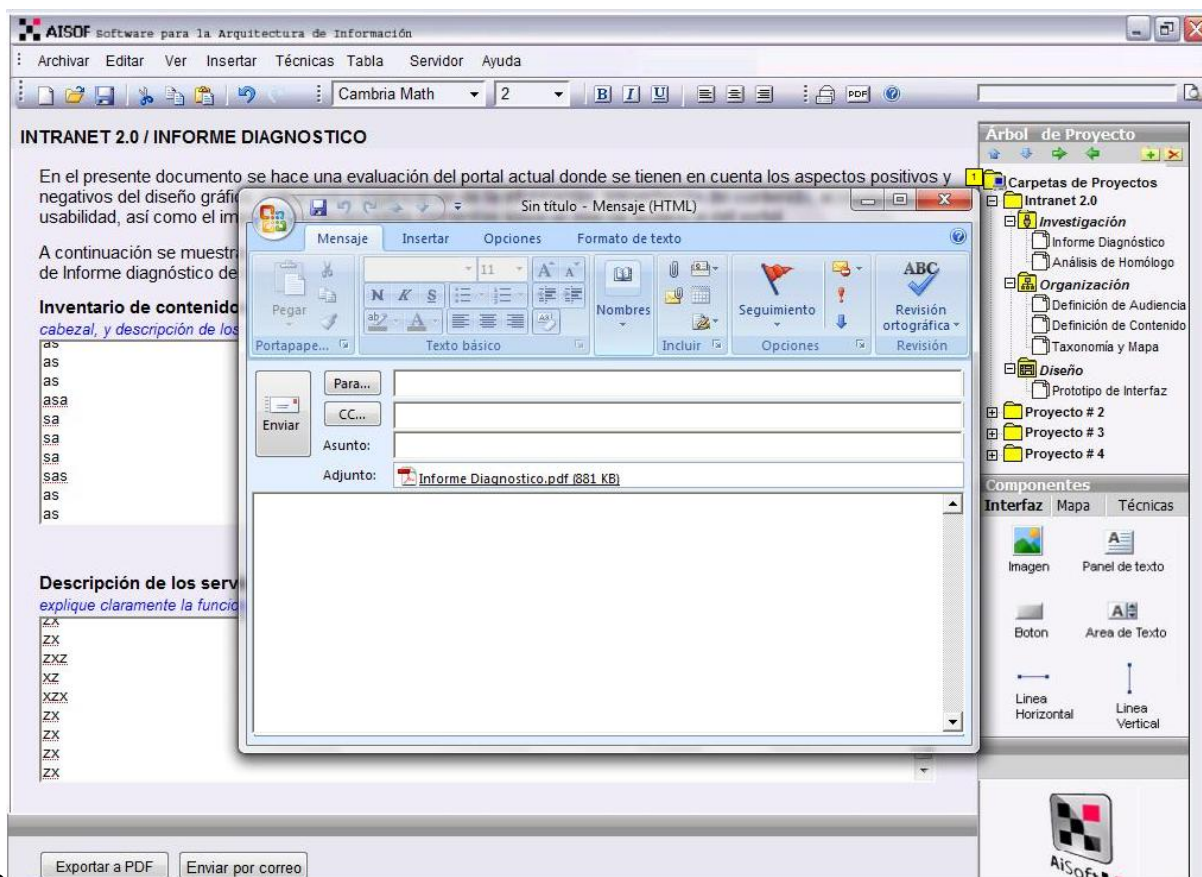


Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Texto en la Ventana actual.

UI. Enviar por

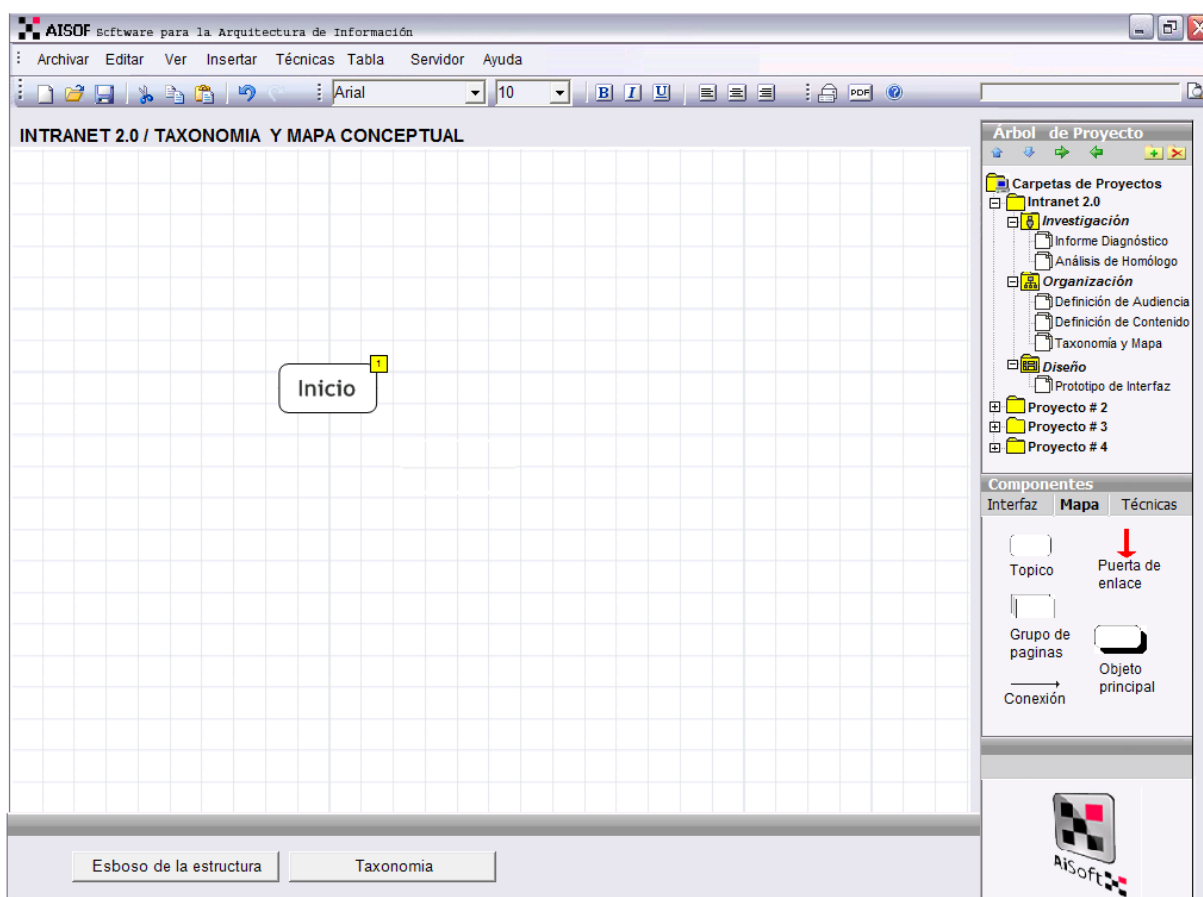


Correo

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Texto en la Ventana actual.

UI. Taxonomía y

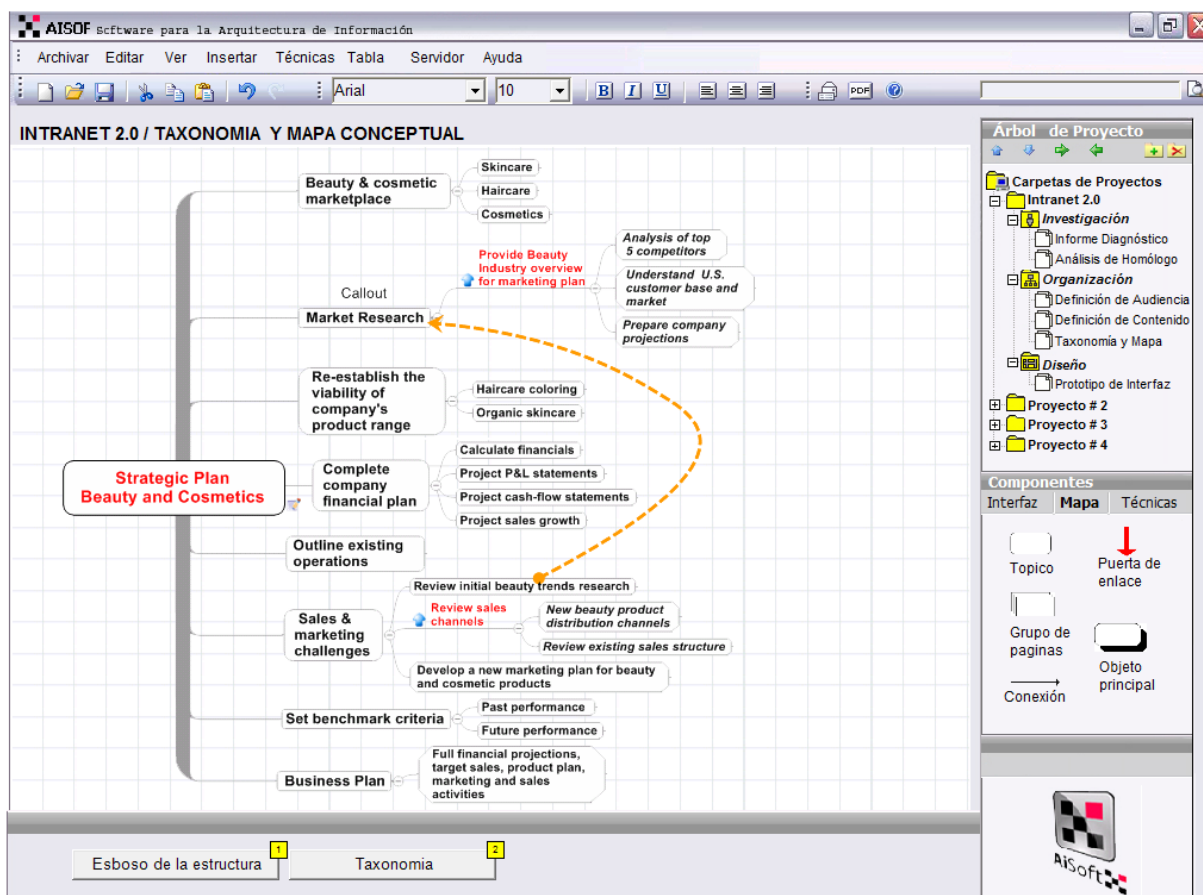


Mapa

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Ejemplo de Mapa en la Ventana actual.

UI. Ejemplo de

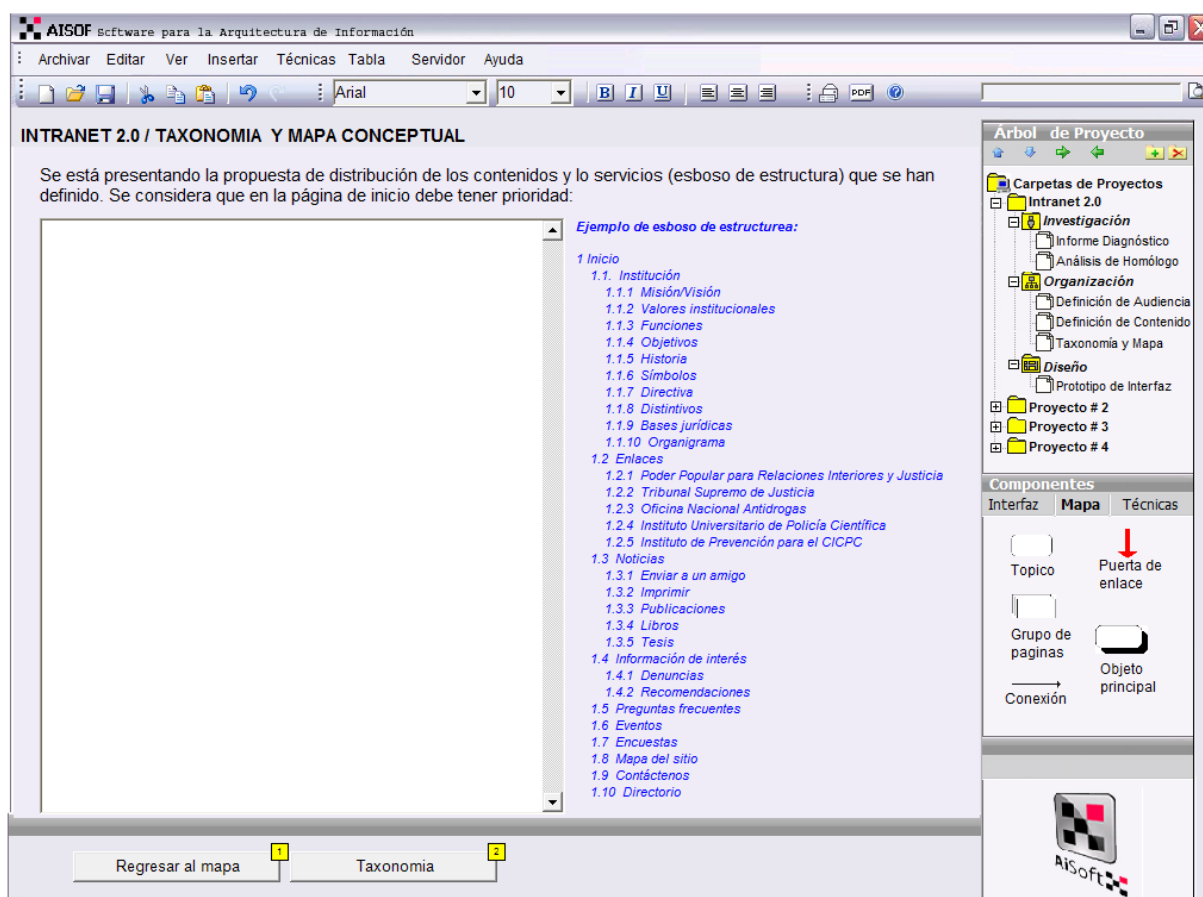


Mapa

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Esbozo de estructura en la Ventana actual.
2	Click	Muestra la UI Taxonomía en la Ventana actual.

UI. Esbozo de



estructura

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Ejemplo de Mapa en la Ventana actual.
2	Click	Muestra la UI Taxonomía en la Ventana actual.

UI. Taxonomía

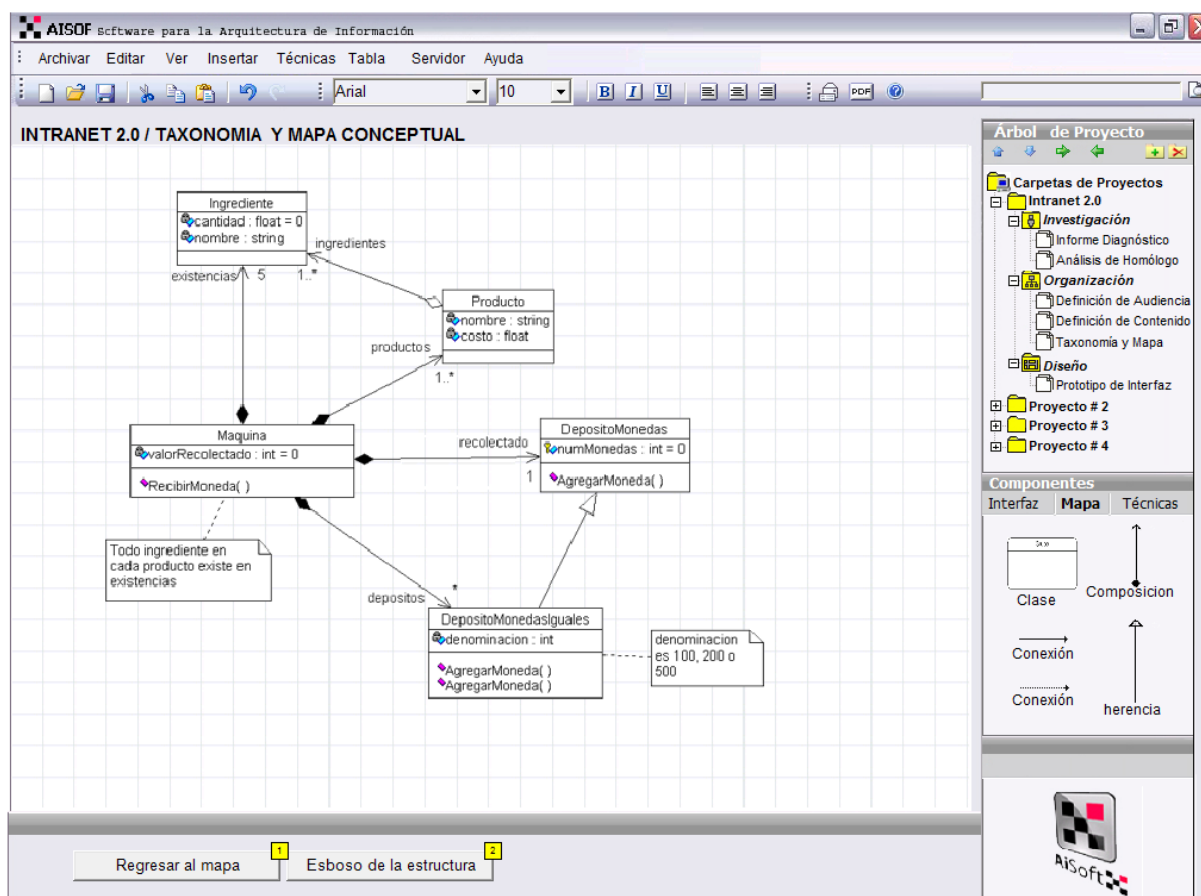
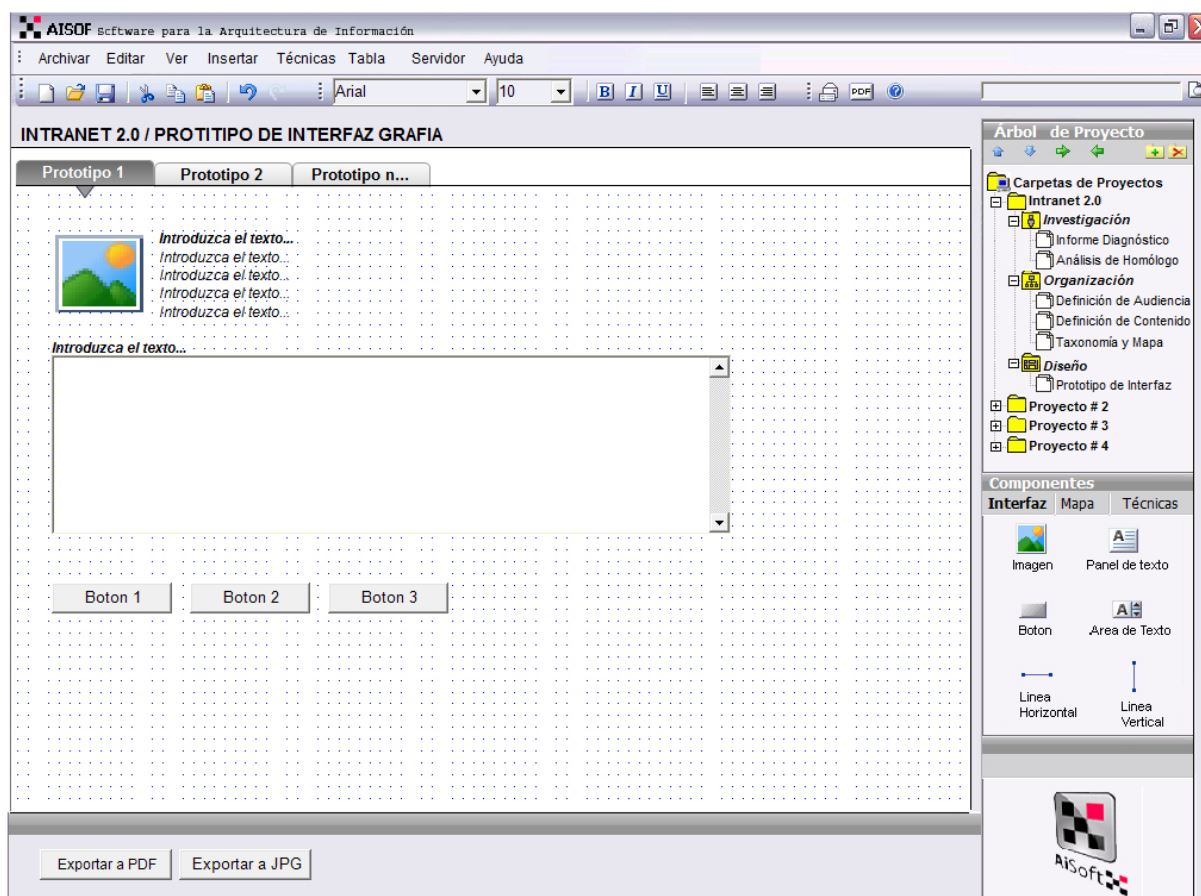


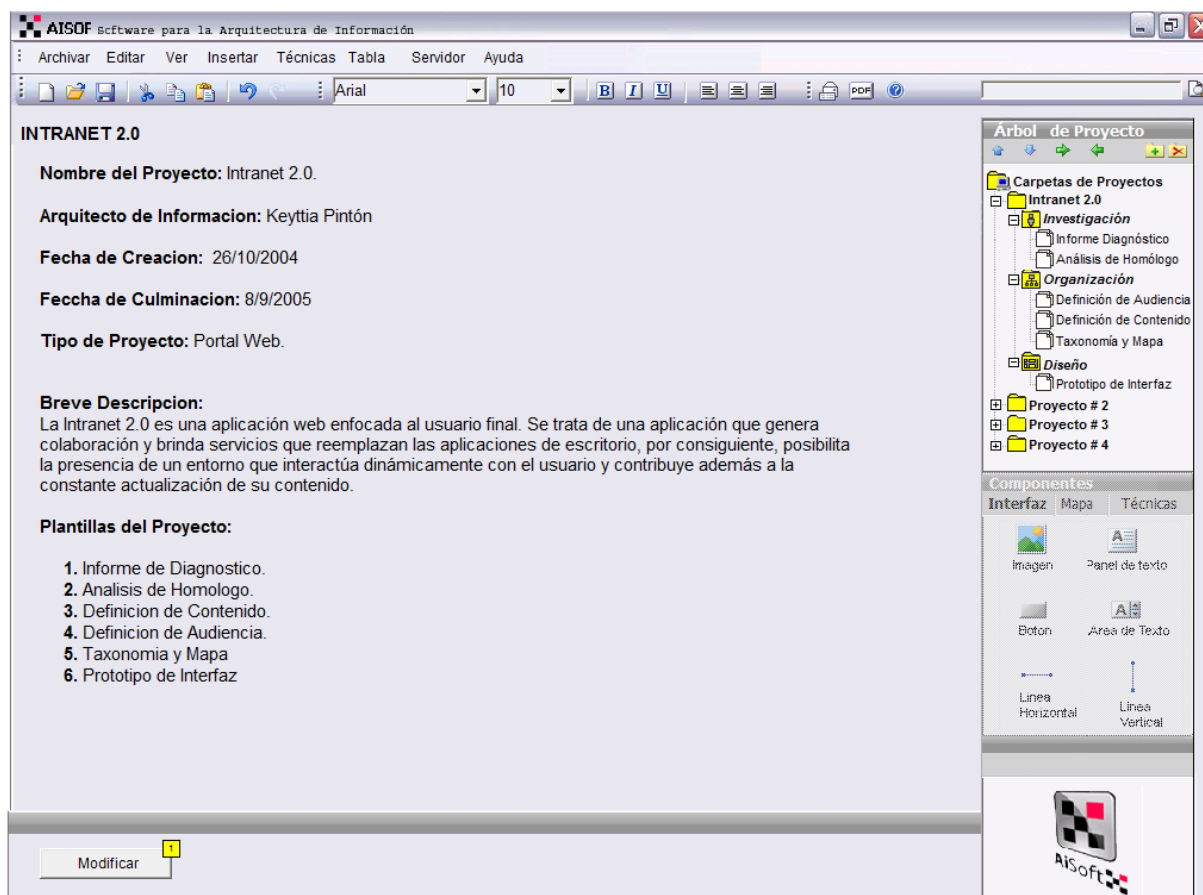
Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Ejemplo de Mapa en la Ventana actual.
2	Click	Muestra la UI Esbozo de estructura en la Ventana actual.

UI. Prototipo



UI. Mostrar Datos



Proyecto

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Modificar Datos del proyecto en la Ventana actual.

UI. Crear

Proyecto

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos Proyecto en la Ventana actual.
2	Click	Muestra la UI Mostrar Datos Proyecto en la Ventana actual.
3	Click	Desplegar el calendario.
4	Click	Desplegar el calendario.

UI. Modificar Datos del

INTRANET 2.0

Nombre del proyecto: Intranet 2.0

Arquitecto de información: Keyttia Pintón

Fecha de creación: 07/14/2007

Fecha de culminación: 07/14/2007

Tipo de proyecto: Sitio Web

Breve Descripción:

La Intranet 2.0 es una aplicación web enfocada al usuario final. Se trata de una aplicación que genera colaboración y brinda servicios que reemplazan las aplicaciones de escritorio, por consiguiente, posibilita la presencia de un entorno que interactúa dinámicamente con el usuario y contribuye además a la constante actualización de su contenido.

La intranet pretende incluir a toda la comunidad universitaria en un mismo contomo web.

Tiene el objetivo de vincular cada uno de los diferentes servicios que brin

Plantillas del proyecto:

- Informe Diagnostico
- Analisis de Homólogo
- Definición de Audiencia
- Definición de Contenido
- Taxonomía y Mapa
- Prototipo de Interfaz

Seleccionar todas las plantillas

Aceptar Cancelar

Arbol de Proyecto

- Carpetas de Proyectos
 - Intranet 2.0
 - Investigación
 - Informe Diagnostico
 - Análisis de Homólogo
 - Organización
 - Definición de Audiencia
 - Definición de Contenido
 - Taxonomía y Mapa
 - Diseño
 - Prototipo de Interfaz
 - Proyecto # 2
 - Proyecto # 3
 - Proyecto # 4

Componentes

Interfaz Mapa Técnicas

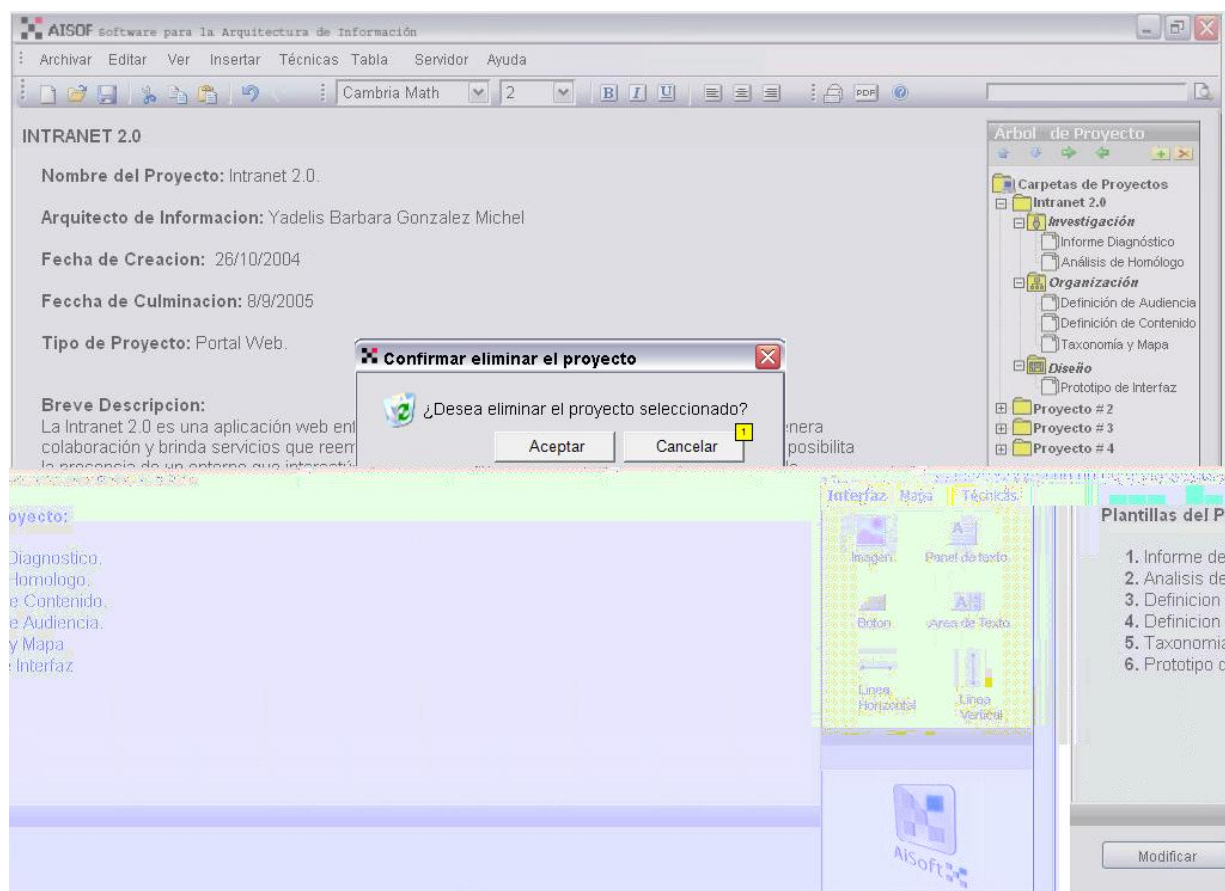
- Imagen Panel de texto
- Boton Area de Texto
- Linea Horizontal Linea Vertical

proyecto

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos Proyecto en la Ventana actual.
2	Click	Muestra la UI Mostrar Datos Proyecto en la Ventana actual.
3	Click	Desplegar el calendario.
4	Click	Desplegar el calendario.

UI. Eliminar



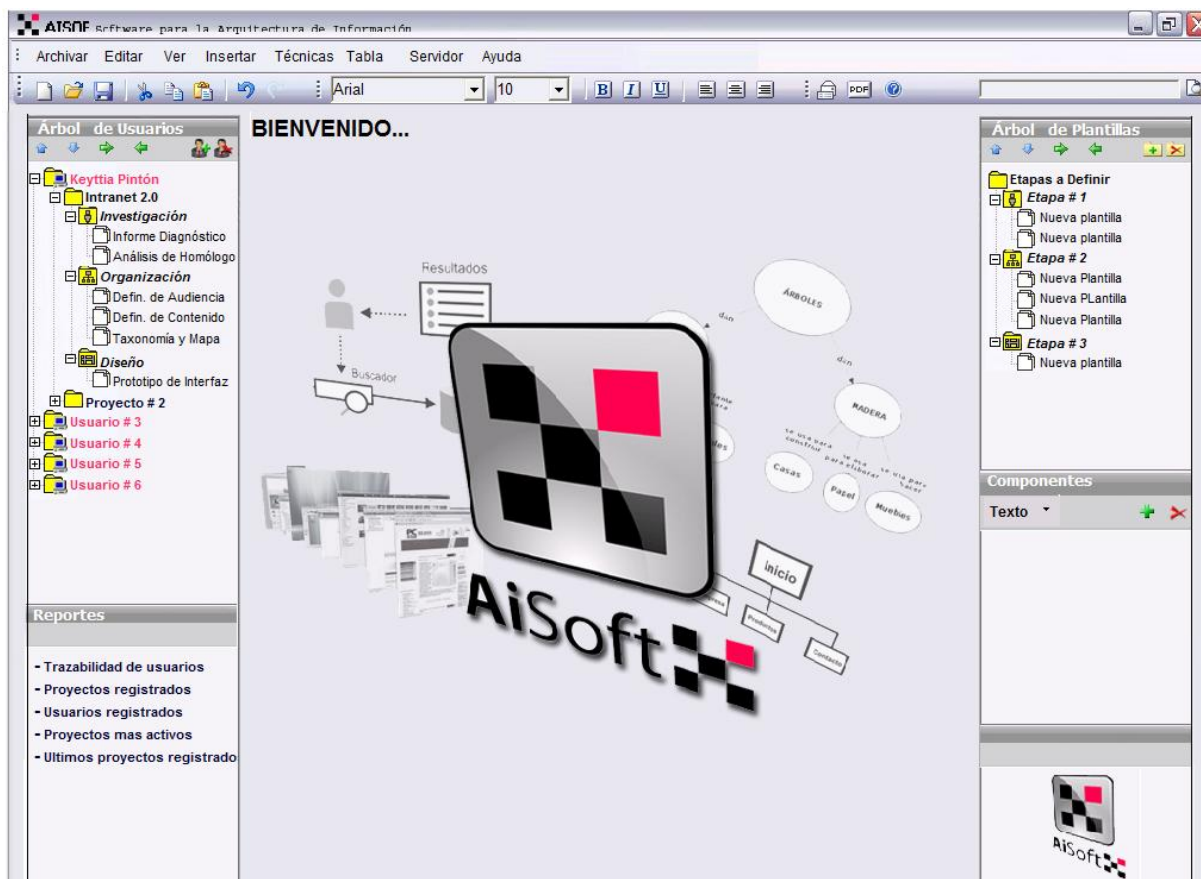
Proyecto

Tabla de eventos

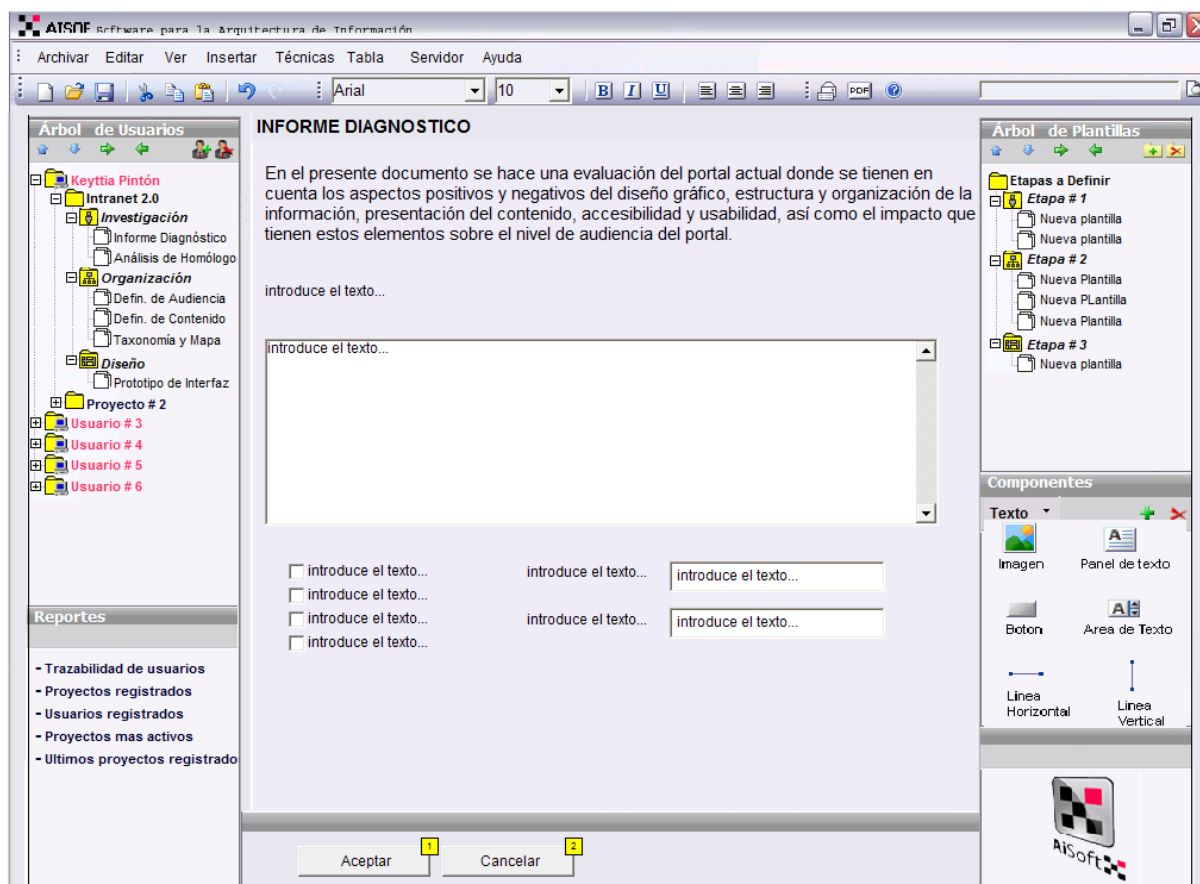
Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos Proyecto en la Ventana actual.

Prototipo elemental de interfaz gráfica de usuario. Modulo Administrador

UI 1. Base



UI. Nueva

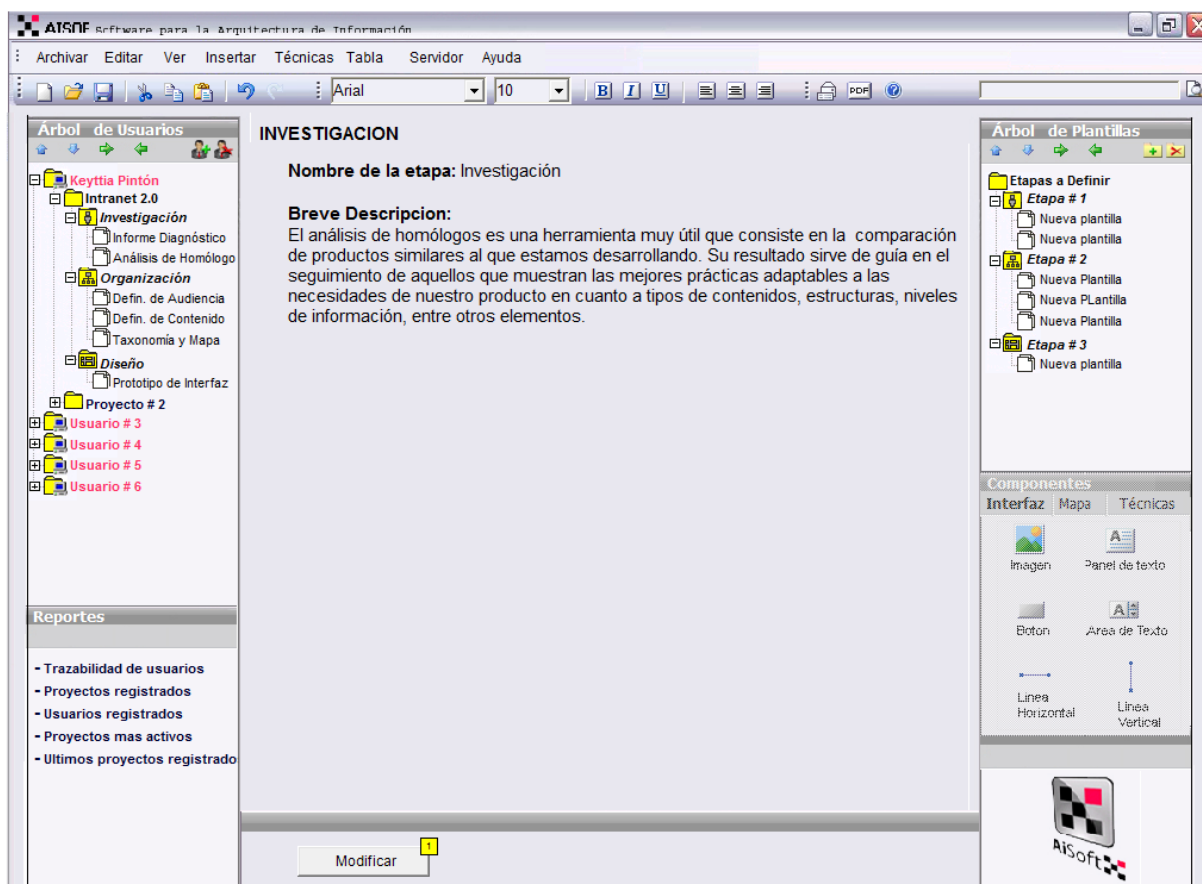


Plantilla

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Base en la Ventana actual.
2	Click	Muestra la UI Base en la Ventana actual.

UI. Mostrar Datos



Etapa

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Modificar Datos de Etapa en la Ventana actual.

UI. Crear Etapa

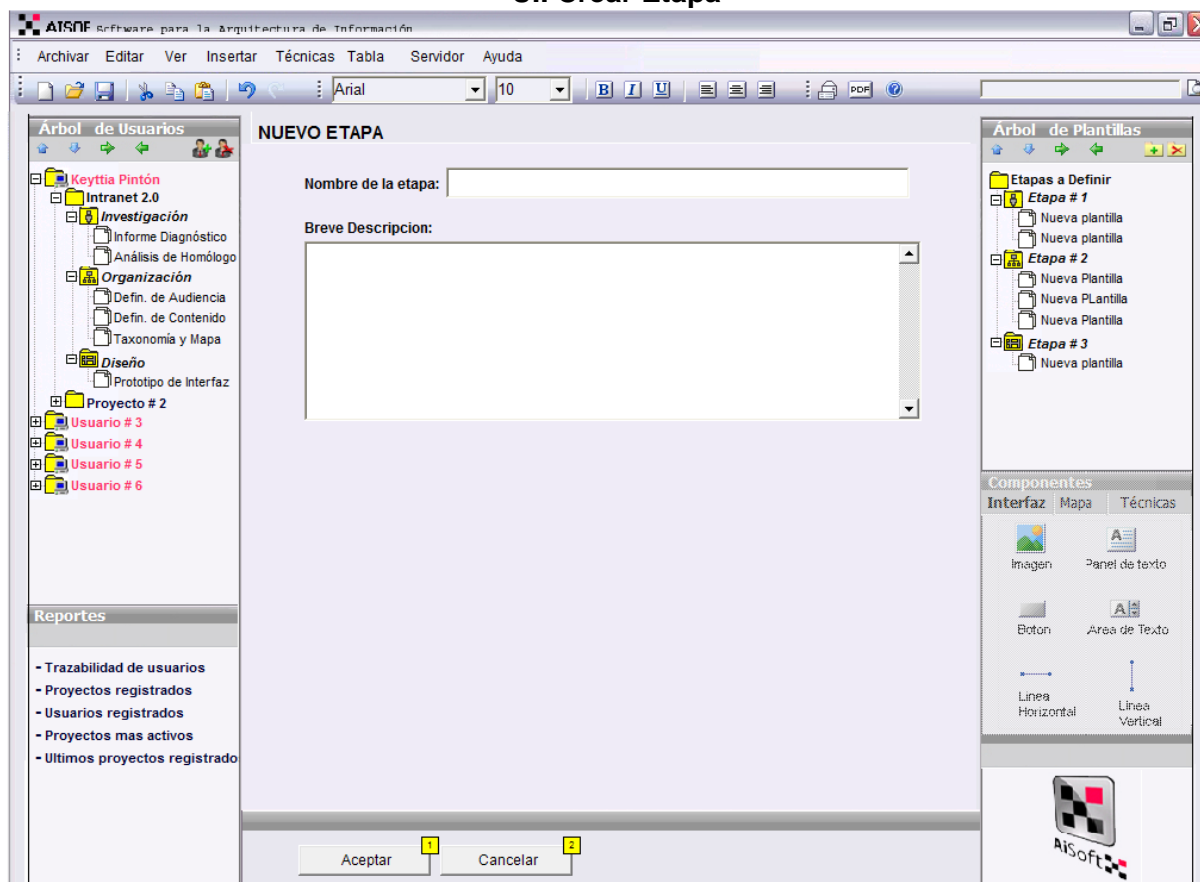
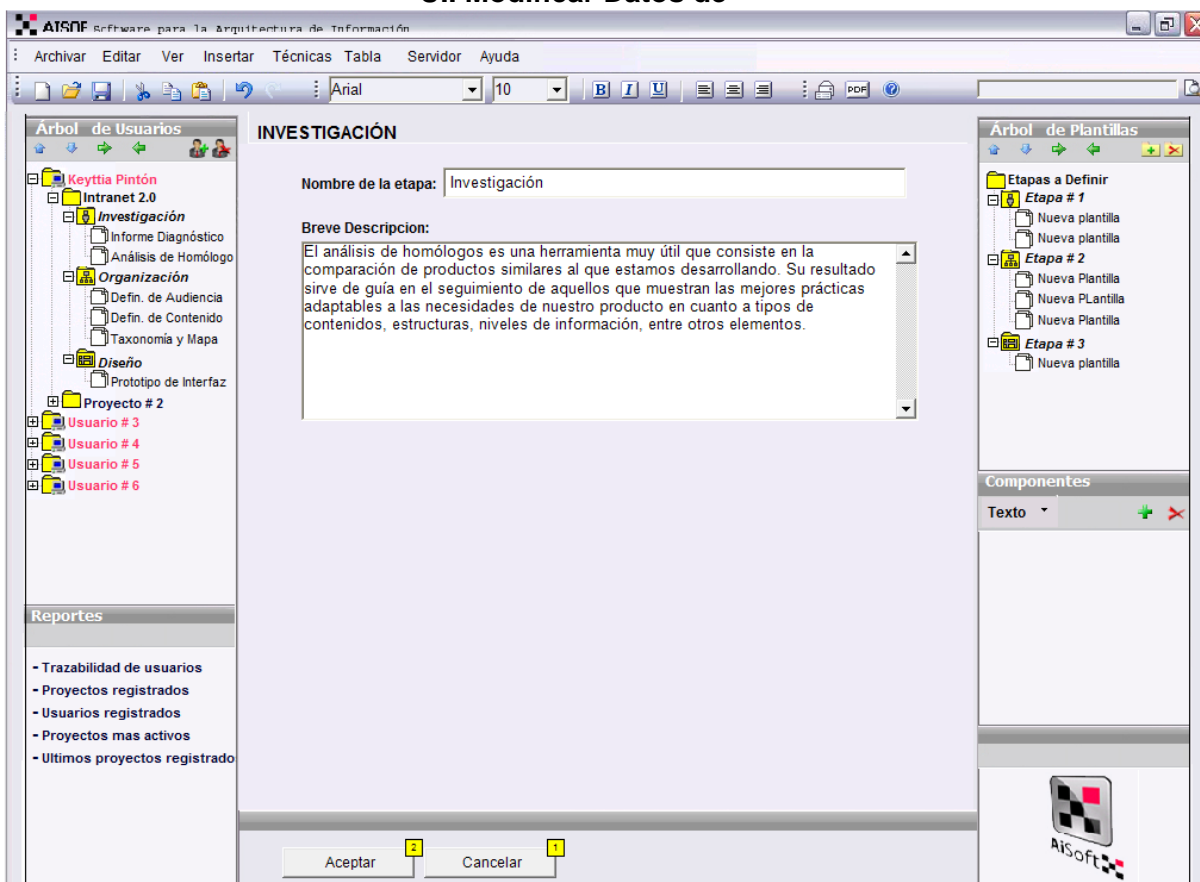


Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos de Etapa en la Ventana actual.
2	Click	Muestra la UI Modificar Datos de Etapa en la Ventana actual.

UI. Modificar Datos de



Etapa

Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos de Etapa en la Ventana actual.
2	Click	Muestra la UI Mostrar Datos de Etapa en la Ventana actual.

UI. Reportes

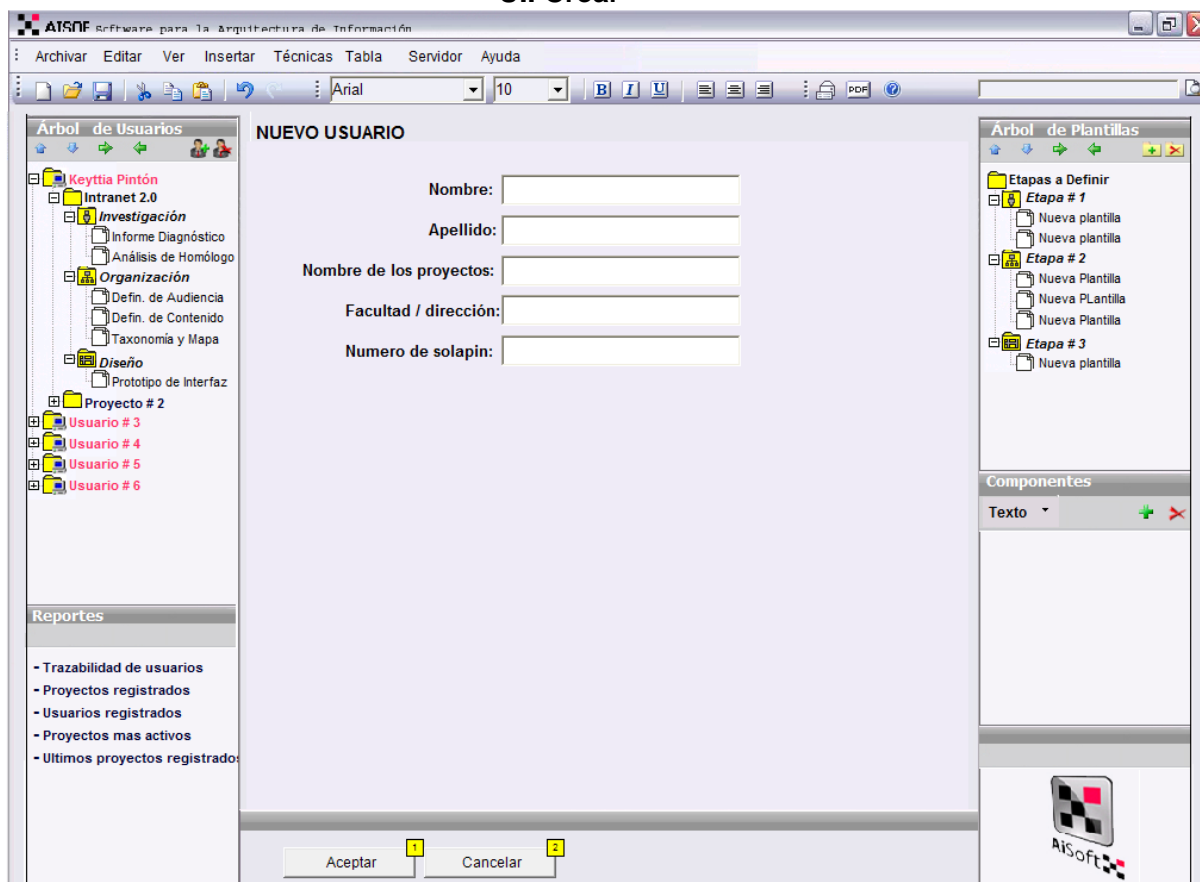
The screenshot displays the AISOF software interface for generating reports. The main window is titled 'AISOF Software para la Arquitectura de Información'. The menu bar includes 'Archivar', 'Editar', 'Ver', 'Insertar', 'Técnicas', 'Tabla', 'Servidor', and 'Ayuda'. The toolbar contains various icons for file operations and printing.

The interface is divided into three main sections:

- Arbol de Usuarios (Left Panel):** A tree view showing the project structure. It includes folders for 'Keyttia Pintón', 'Intranet 2.0', 'Investigación', 'Organización', 'Diseño', 'Proyecto # 2', and 'Usuario # 3' through '# 6'.
- USUARIOS REGISTRADOS (Main Panel):** A search and results area. It features a search input field with the placeholder 'Introduzca la cadena de búsqueda...' and a 'Buscar' button. Below is a table titled 'Resultado de Búsqueda' with columns for 'Nombre y apellidos', 'Telefono', 'Area', and 'Proyecto'. The table shows two entries: 'Lianet Liben Martinez' (584756, Direc. Informatizacion, SICI) and 'Liliana Sanchez Perez' (7589658, Direc. Prod 3, CICPC). At the bottom of the table are 'Anterior', 'Pagina 1 de 20', 'Siguiete', 'Imprimir', and 'Exportar' buttons.
- Arbol de Plantillas (Right Panel):** A tree view for templates, showing 'Etapas a Definir' and three sub-steps: 'Etapa # 1', 'Etapa # 2', and 'Etapa # 3', each with a 'Nueva plantilla' option.

At the bottom right of the interface is the AISOF logo, which consists of a square with a grid pattern and the text 'AISoft' below it.

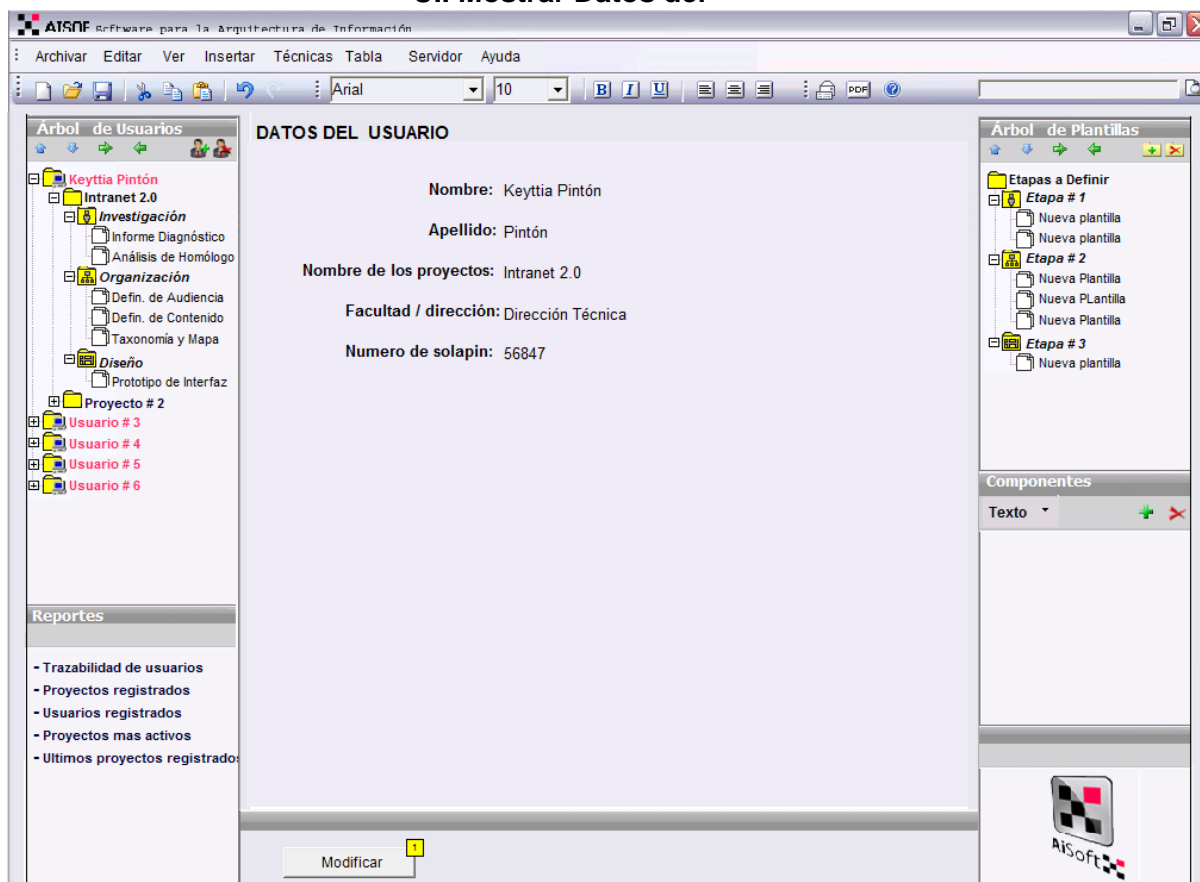
UI. Crear



Usuario
Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos del Usuario en la Ventana actual.
2	Click	Muestra la UI Base en la Ventana actual.

UI. Mostrar Datos del



Usuario Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Modificar datos de Usuario en la Ventana actual.

UI. Modificar datos de Usuario

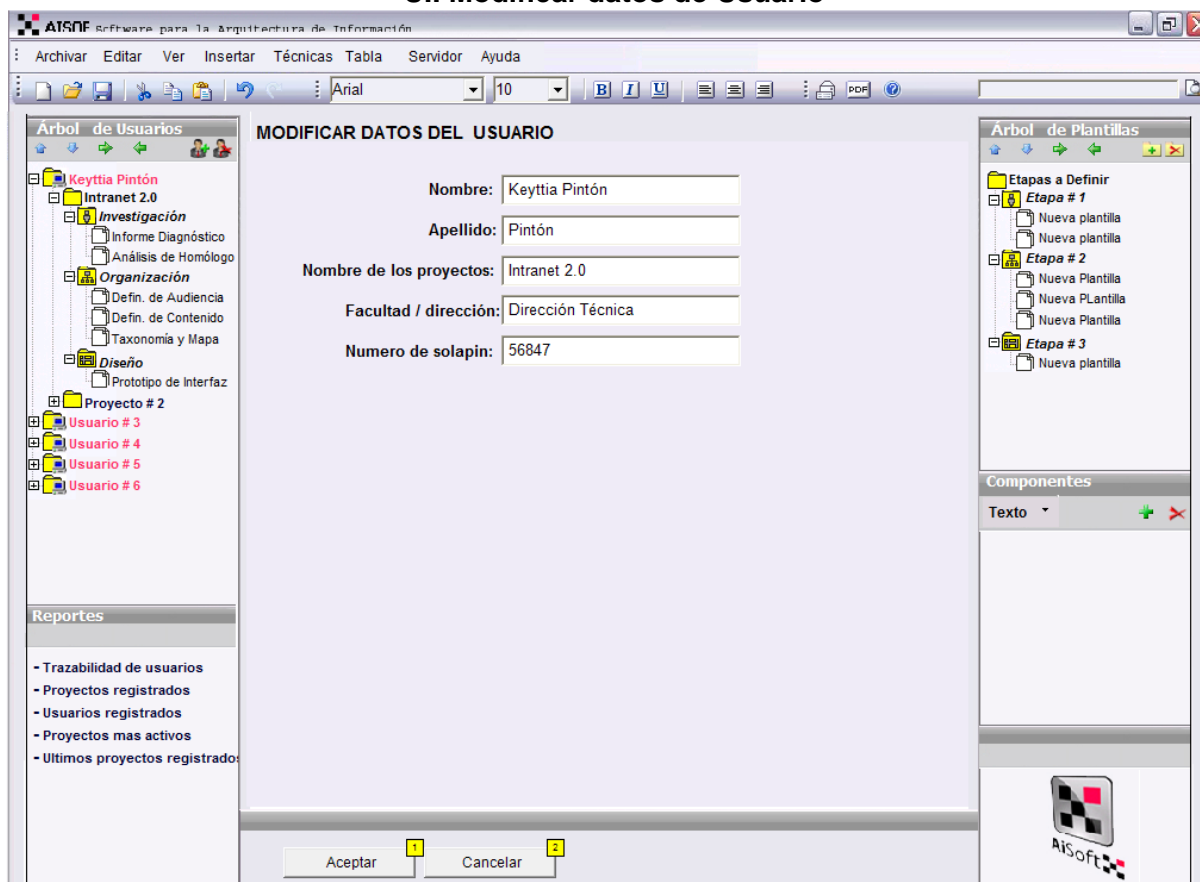


Tabla de eventos

Control	Evento	Acción
1	Click	Muestra la UI Mostrar Datos del Usuario en la Ventana actual.
2	Click	Muestra la UI Mostrar Datos del Usuario en la Ventana actual.

Glosario de términos

ASP: Active Server Pages es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

CorelDRAW: Es un programa avanzado de edición gráfica.

CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador):

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

DOCX: es la extensión (.docx) empleada en los documentos creados por la aplicación ofimática Word en su versión perteneciente al paquete Office 2007.

Flash: Programa de la empresa Adobe que sirve para diseñar webs y banners con animación.

HTML: siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

Javascript: lenguaje de programación desarrollado por Netscape, empleado para la creación de webs interactivas.

Mantenimiento: Propiedad de un sistema que representa la cantidad de esfuerzo requerida para conservar su funcionamiento normal o para restituirlo una vez se ha presentado un evento de falla.

PHP: es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Photoshop: Programa hecho por Adobe para diseño y manejo de gráficas.

PNG: Portable Networks Graphic, Formato de imagen en mapa de bits.

Reusabilidad: Respecto a la calidad de un programa, la reusabilidad hace referencia a poder volver a usar parte de dicho software en otro proyecto.

Stakeholder: Desde el punto de vista del desarrollo de sistemas, un Stakeholders es aquella persona o entidad que está interesada en la realización de un proyecto o tarea, auspiciando el mismo ya sea mediante su poder de decisión o de financiamiento.

URL: (Uniform Resource Locator) Utilizado para especificar un objeto en Internet.

Usabilidad: Facultad de ser usado. En el contexto de las aplicaciones informáticas, se refiere al conjunto de características que inciden en su utilización, atendiendo a cualquier contexto o tipo de usuario.

Taxonomía: Es la ciencia de la clasificación.

XML: Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

3D: termino para referirse a la tridimensional.