

UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA



Sistemas para la Colaboración

- PLANCODE -
PLANIFICADOR COLABORATIVO PARA EL DISEÑO DE ESTRATEGIAS Y ACCIONES DE
EMERGENCIA

ANÁLISIS DE REQUISITOS

Juan Andrada Romero
Jose Domingo López López

11 de abril de 2010

Índice

1. Introducción	1
2. ¿Qué es PlanCoDE?	1
2.1. Diagrama de casos de uso	2
3. Prototipo de interfaz gráfica de usuario	3

1. Introducción

En este documento se realizará un análisis de requisitos que deberán ser satisfechos por medio de una aplicación *grupware*. Se trata de una herramienta colaborativa denominada PlanCoDE.

2. ¿Qué es PlanCoDE?

PlanCoDE es el acrónimo de *Planificador Colaborativo para el Diseño de Estrategias y Acciones de Emergencia*. Es, por tanto, una herramienta que permite la colaboración de diferentes especialistas y cuerpos de seguridad y emergencia (policías, bomberos, etc.) para diseñar y ejecutar un plan de actuación y poder atender una situación de emergencia.

PlanCoDE ofrece las siguientes funcionalidades a sus usuarios:

- **Chat:** los diferentes miembros de grupos especialistas en situaciones de emergencia podrán utilizar el chat que la aplicación proporciona para comunicarse en tiempo real e intercambiar opiniones acerca del plan a realizar.
- **Visualización de mapas:** la aplicación mostrará una imagen del mapa de la zona donde se ha producido la emergencia. Dichos mapas se recuperarán a partir de *Google Maps* o serán imágenes ya almacenadas en disco.
- **Edición de mapas:** los usuarios del sistema podrán realizar trazos libres y colocar diferentes símbolos (barreras, puntos conflictivos, etc.) sobre la imagen del mapa que se ha cargado. De este modo, se pueden realizar rutas sobre el propio mapa y marcar con símbolos aquellos puntos de interés. Del mismo modo, también se podrá eliminar los trazos y objetos introducidos por cada usuario de la aplicación.

Además, PlanCoDE debe tener como requisitos adicionales la **movilidad** y la **tolerancia a fallos** debido a las situaciones críticas en las que dicha herramienta será utilizada. Esto implica disponer de una arquitectura lo más descentralizada posible. Por ello, ésta no será la típica arquitectura cliente-servidor, ya que si en algún momento el servidor no está disponible por la razón que sea, los usuarios no podrían iniciar el sistema para diseñar las estrategias pertinentes. Por tanto, el primer usuario que inicie una sesión se desempeñará la función de servidor y cliente al mismo tiempo permitiendo que resto de usuarios puedan conectarse a él. De modo que cuando se ejecuta la aplicación, ésta debe permitir al usuario crear una nueva sesión o unirse a una sesión existente. En el segundo caso, será necesario introducir la dirección IP y el puerto al que debe conectarse la aplicación. Para esto, el usuario podrá introducir manualmente dicha dirección y puerto, pero también podrá recuperar una lista de servidores (entendiendo servidor como uno de los clientes que iniciaron la sesión por primera vez) a través de un servicio Web e iniciar sesión en el servidor deseado.

Con esta arquitectura, se satisfacen los requisitos críticos de movilidad y tolerancia a fallos, pues cualquier cliente puede conectarse en cualquier momento y lugar y siempre va a existir algún servidor disponible, ya que éste será uno de los propios clientes de la aplicación. Además, si el servicio Web falla al recuperar la lista de servidores, existiría la posibilidad de introducir la información para conectarse con otros clientes de manera manual.

2.1. Diagrama de casos de uso

El diagrama de casos de uso de la aplicación se muestra en la Figura 2.1. Dicho diagrama representa los requisitos de la aplicación que se han comentado en el punto anterior, junto con otras funcionalidades que son necesarias para el correcto funcionamiento de la aplicación colaborativa.

Así, un usuario debe iniciar una sesión para poder utilizar la aplicación con el resto de usuarios y cierra la sesión cuando deja de utilizarla. Del mismo modo, la propia aplicación debe actualizar su estado y enviar a todos los usuarios conectados esta información, es decir, los mensajes que se envían al chat, la imagen del mapa que se desea mostrar y los dibujos que se realizan sobre éste.

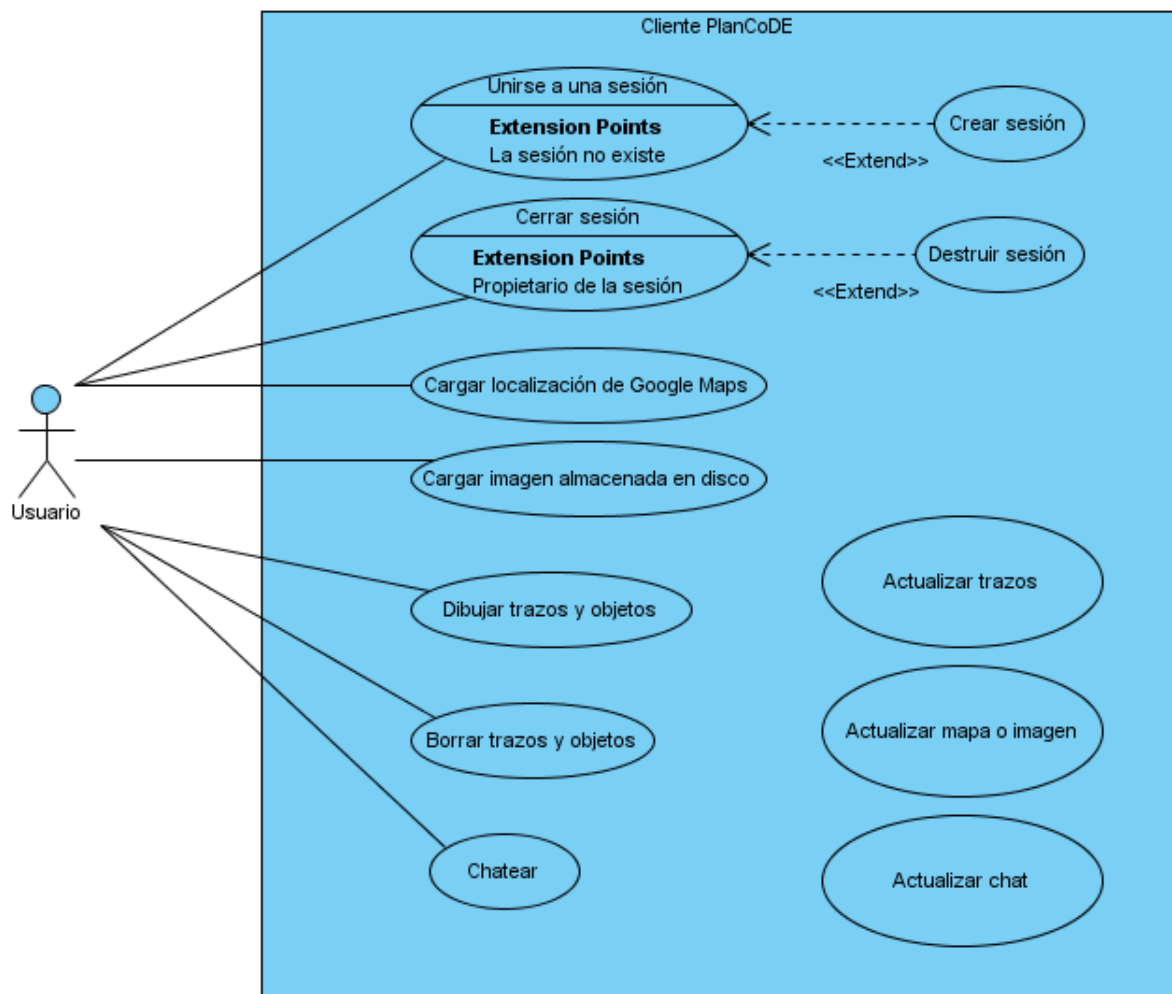


Figura 2.1: Diagrama de casos de uso de la aplicación

3. Prototipo de interfaz gráfica de usuario

Atendiendo a los requisitos que debe tener la herramienta *PlanCoDE* comentados en la sección 2, la interfaz gráfica de usuario de la aplicación será como la que muestra el prototipo de la Figura 3.1.

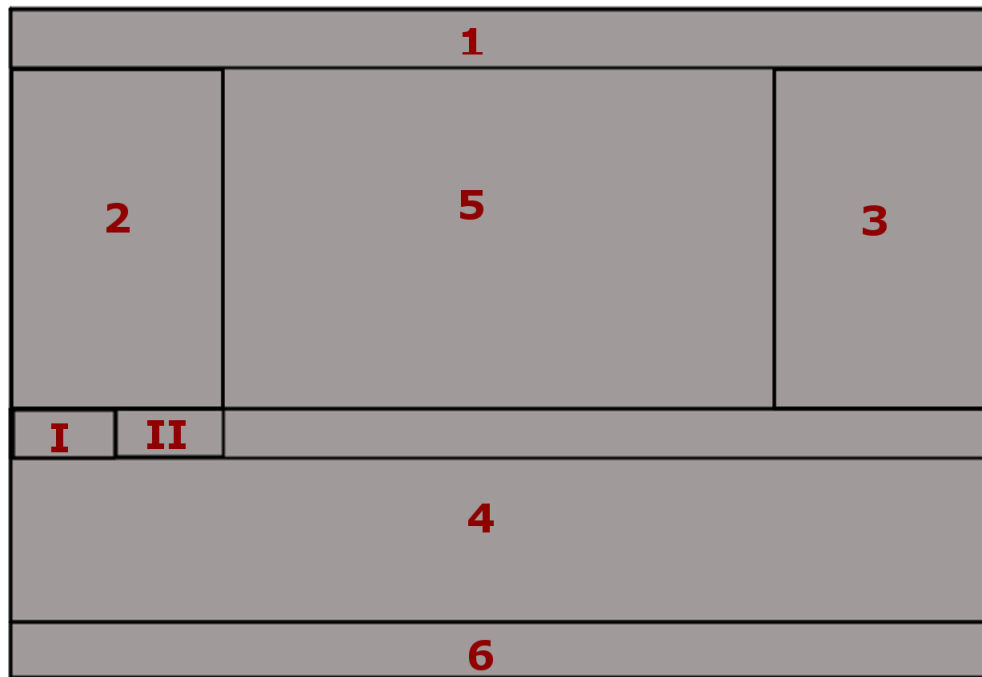


Figura 3.1: Proptotipo de la interfaz gráfica de PlanCoDE

Los elementos que componen la interfaz son:

1. **Barra de herramientas:** en este elemento se mostrarán los diferentes menús de la aplicación.
2. **Caja de herramientas:** aquí se mostrarán las herramientas necesarias para poder dibujar trazos y objetos sobre las imágenes de los mapas.
3. **Lista de usuarios:** en este elemento se muestran los usuarios que están utilizando la aplicación.
4. **Terminal:** en la primera pestaña de este elemento se muestran los mensajes que se envían al chat, y en la segunda, se muestra todo el *log* de acciones que realizan los usuarios.
5. **Área de trabajo:** aquí se muestra la imagen del mapa seleccionado.
6. **Barra de estado:** este elemento proporciona información al usuario acerca de la acción que está realizando.