



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Diseño y prototipado de una plataforma web sobre base cartográfica para el análisis de riesgos en entornos industriales

Ingeniería Informática
Proyecto de Fin de Carrera

Autor: Nayarit Santana Pacheco

Tutores: D. Francisco Mario Hernández Tejera
D. Blas J. Galván González

Fecha: Septiembre de 2013

Proyecto fin de carrera de la Escuela de Ingeniería Informática de la Universidad de Las Palmas de Gran Canaria presentado por la alumna:

Título del Proyecto: DISEÑO Y PROTOTIPADO DE UNA PLATAFORMA WEB SOBRE BASE CARTOGRÁFICA PARA EL ANÁLISIS DE RIESGOS EN ENTORNOS INDUSTRIALES

Autor: Nayarit Santana Pacheco

Tutores: D. Francisco Mario Hernández Tejera
D. Blas J. Galván González

DEDICATORIA

A mis padres y mi hermana, por todo su apoyo y amor.

AGRADECIMIENTOS

Quiero agradecer todo el apoyo y la ayuda proporcionada por mis tutores: D. Mario Hernández Tejera y D. Blas J. Galván González, que me han apoyado y han dado la libertad suficiente en todo momento y han confiado en mi trabajo.

Me gustaría darles las gracias principalmente a mis padres y a mi hermana, que siempre me han dado fuerzas para seguir adelante y no tirar la toalla, que han hecho que crea en mí y en mis posibilidades y siempre me han dado todo su amor y confianza. Sin ellos esto no habría sido posible.

Quisiera agradecerle a mi compañero Samuel por toda la ayuda que me ha prestado y por darme ánimos durante el proyecto, sin duda, su compañía y apoyo ha hecho más llevadero este trabajo. También querría agradecer a algunos miembros del CEANI, a Andrés Cacereño, que me ha proporcionado las bases necesarias sobre los riesgos en entornos industriales y se ha prestado a revisar los resultados obtenidos. Y muy especialmente a Juan Pedro Ramos Ponce, que me ha ayudado guiándome y supervisándome durante todo el proceso de desarrollo.

Por último no puedo olvidarme de mis amigos, a todos ellos gracias por la paciencia y el apoyo, por entender mis ausencias y mostrarme siempre su comprensión y amistad. Especialmente quiero nombrar a Miguel Ángel, que día a día ha aguantado mis nervios y siempre ha encontrado la forma de animarme y relajarme.

Índice

Capítulo 1. Introducción	13
1.1. Estructura de la memoria.....	13
Capítulo 2. Estado actual del tema	15
2.1. PHAST	15
2.2. ZEUS	17
2.3. Propiedades Comunes	21
Capítulo 3. Objetivos	23
Capítulo 4. Recursos necesarios.....	25
Capítulo 5. Metodología	27
5.1. Análisis.....	27
5.2. Diseño.....	27
5.3. Implementación	28
5.4. Validación	28
Capítulo 6. Análisis	29
6.1. Descripción del problema / Contextualización de la plataforma	29
6.1.1. Accidentes graves	29
6.1.2. Legislación sobre “accidentes graves”	30
6.1.3. Real Decreto 1254/1999 y posteriores modificaciones.....	30
6.1.4. Real Decreto 1196/2003	31
6.1.5. Zonas de afección más importantes.....	31
6.2. Ámbito de la plataforma desarrollada.....	32
6.3. Modelos matemáticos utilizados	33
6.3.1. Método equivalente TNT.....	33
6.3.2. Fuga de Líquido Inflamable	36
6.4. Análisis de requisitos y casos de uso.....	38
6.4.1. Roles.....	38
6.4.2. Requisitos funcionales	39
6.4.3. Casos de uso	42
6.4.4. Requisitos no funcionales	68
6.5. Modelo del dominio	69
Capítulo 7. Diseño	71
7.1. Arquitectura básica de la plataforma	71
7.2. Tecnologías utilizadas	73
7.2.1. Servidor web.....	73
7.2.2. Cliente	77
7.2.3. Servidor BBDD	79
7.3. Patrones y otros aspectos de diseño.....	79
7.3.1. Servidor web.....	80

7.3.2. Cliente	81
7.4. Base de datos	83
7.4.1. Tabla TUserType.....	83
7.4.2. Tabla TUser	85
7.4.3. Tabla TSubstanceType	85
7.4.4. Tabla TSubstance	85
7.4.5. Tabla TAccidentType.....	86
7.4.6. Tabla TAccident	86
7.4.7. Tabla TMap	87
7.4.8. Tabla TResource.....	88
7.4.9. Tabla TMathematicalModel	88
7.4.10. Tabla RSubstanceProperties	88
7.4.11. Tabla RSubstanceValues	89
7.4.12. Tabla RScenarioResource.....	90
7.4.13. Tabla TAccidentTNT	91
7.4.14. Tabla TAccidentFLI	91
7.4.15. Tabla TZone	92
7.4.16. Tabla TSimulatedZone	92
7.4.17. Tabla TScenario.....	93
7.4.18. Tabla RScenarioAccident	93
Capítulo 8. Implementación	95
8.1. Cliente	95
8.1.1. Visor	95
8.1.2. Acceso al servidor	97
8.2. Servidor	99
8.2.1. ABC del Endpoint	99
8.2.2. Utilización del modelo FLI (Fuga de Líquido Inflamable).....	101
Capítulo 9. Conclusiones.....	105
Capítulo 10. Trabajos futuros	107
Capítulo 11. Bibliografía	109
Anexo I. Manual de usuario.....	113
A1.1. Acceso a la aplicación	113
A1.2. Registro de usuarios.....	113
A1.3. Login	114
A1.4. Modos	115
A1.4.1. Modo creación	115
A1.4.2. Modo simulación	115
A1.5. Crear un nuevo escenario.....	115
A1.5.1. Asociar un mapa	116
A1.5.2. Crear nuevo accidente.....	116
A1.5.3. Modificar accidente	118

A1.5.4. Eliminar accidente	119
A1.5.5. Asociar un nuevo recurso	119
A1.5.6. Eliminar un recurso.....	119
A1.5.7. Guardar escenario	119
A1.6. Simulación.....	119
A1.6.1. Añadir nueva zona para simular	119
A1.6.2. Eliminar una zona.....	120
A1.6.3. Simular zonas	121
A1.7. Añadir nuevo mapa/zona/recurso/sustancia.....	122
A1.8. Modificar/ Eliminar mapa/zona/recurso/sustancia	122
Anexo II. Valores umbrales en TNT	123

Capítulo 1. Introducción

Este proyecto surge de la propuesta de uno de los tutores del mismo: D. Blas J. Galván González (Departamento de Matemáticas de la ULPGC), para completar diversos estudios que se están llevando a cabo en el Instituto Universitario SIANI (Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería), dentro de la división CEANI (Computación Evolutiva y Aplicaciones), en el ámbito de los análisis de riesgos.

Según el Real Decreto 1196/2003 del 19 de Septiembre, donde se aprueba la “directriz básica de Protección Civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas”, se entiende riesgo como “la probabilidad de que se produzca un efecto dañino específico en un período de tiempo determinado o en circunstancias determinadas” [Gobierno de España - 1999]. Según esta directriz, las empresas que almacenen sustancias peligrosas* deben realizar un plan de autoprotección denominado: Plan de Emergencia, donde se realizará el análisis y evaluación de los riesgos además de otra serie de medidas que no competen a este proyecto.

El estudio de los riesgos normalmente se lleva a cabo a través de la participación de uno o varios expertos, sin embargo, siempre es conveniente disponer de alguna aplicación que facilite determinados cálculos. A pesar de no ser estrictamente necesario, el planteamiento sobre un mapa de los distintos riesgos existentes en una zona determinada, ayudaría a ver, de forma global, los posibles resultados en caso de accidente. El proyecto desarrollado, que consiste en una plataforma web, aúna estas dos características, ya que permite:

- la utilización de mapas de la zona donde existen los riesgos, pudiendo marcar puntos en aquellos sitios en los que existan riesgos
- realizar los cálculos exigidos para el desarrollo del plan de emergencia, en base a una serie de modelos matemáticos

Para la visualización de los mapas se hará uso de otro proyecto, desarrollado por: Samuel Díaz Cabrera, también alumno de la ULPGC, que lleva por título: “Desarrollo de un prototipo de componente web para la visualización e interacción con mapas de forma online”.

Además de estas características, la plataforma también ofrece otras ventajas que se irán comentando en posteriores secciones de la memoria.

1.1. Estructura de la memoria

En la primera parte de la memoria se realizará un análisis comparativo de las distintas aplicaciones existentes hoy en día que ofrecen similares características. Se intentará

explicar el nivel de utilización de cada una de estas principales características ofrecidas.

A continuación se desarrollarán, punto por punto, los distintos objetivos alcanzados con el proyecto, así como la metodología que se ha llevado a cabo para conseguirlo. Posteriormente, se indicarán los recursos que se han ido necesitando a lo largo del tiempo que se le ha dedicado y, por último, se encuentra el desarrollo de las conclusiones obtenidas y el planteamiento de una posible actualización de la plataforma.

Capítulo 2. Estado actual del tema

Actualmente no existen muchas herramientas que estén englobadas dentro del ámbito del análisis de riesgos. Por este motivo, la comparación no se ha centrado exclusivamente en las aplicaciones online, sino que también se han tenido en cuenta las aplicaciones de escritorio.

El caso que nos ocupa en este proyecto queda delimitado por los riesgos químicos. Dentro de este tipo de riesgos se han encontrado 2 aplicaciones similares a lo que se pretende desarrollar:

- PHAST
- ZEUS

2.1. PHAST

PHAST es un producto de la empresa DNV (Det Norske Veritas), que se dedica a examinar el progreso de un accidente grave basado en un proceso químico, desde que ocurre y se libera, hasta que se forma una nube tóxica o se termina dispersando. En ese proceso, calcula la concentración de la sustancia, la radiación, la toxicidad, la sobrepresión de explosión, etc.

Se trata de la herramienta más completa en el ámbito del análisis de riesgos, ya que cubre todas las áreas de estudio, obteniendo cálculos muy fiables basados en técnicas y modelos muy complejos, que han ido mejorando y ampliando a lo largo de los años.

El funcionamiento de esta aplicación es fácil de seguir a través de los siguientes pasos:

1. Definición de un estado inicial. Se configuran valores de condiciones climáticas, como pueden ser: velocidad y dirección del viento, humedad relativa, temperatura ambiente, etc.
2. Se definen los accidentes. Utilizando un mapa o a través de coordenadas, se establecen todos los accidentes que se desean plantear. Cada accidente podrá ser de distinto tipo y, su comportamiento durante la simulación quedará definido en base al modelo matemático que se haya establecido. Este modelo matemático normalmente viene predefinido con el tipo de accidente.

Dentro de la definición de los accidentes, también se incluye la inserción de todos aquellos datos requeridos para el correcto funcionamiento del modelo.

3. Por último, se realiza la simulación de cada uno de los accidentes, por separado o en conjunto, según se prefiera.

- Una vez realizada la simulación se obtendrán una serie de tablas y gráficas para cada uno de los accidentes simulados. En el mapa se podrán apreciar las áreas que se verían afectadas por cada accidente y en qué medida, ya que cada una tendrá una repercusión mayor o menor sobre la zona.

Para la visualización de los resultados, a la izquierda del mapa incluyen una leyenda indicando cada una de las zonas de afección mostradas, en cada una se indica el valor de presión que tendría todo el sector que se vea delimitado por ese círculo. En la *Figura 1* se observa como en el mapa están dibujadas todas las zonas de afección del accidente simulado.

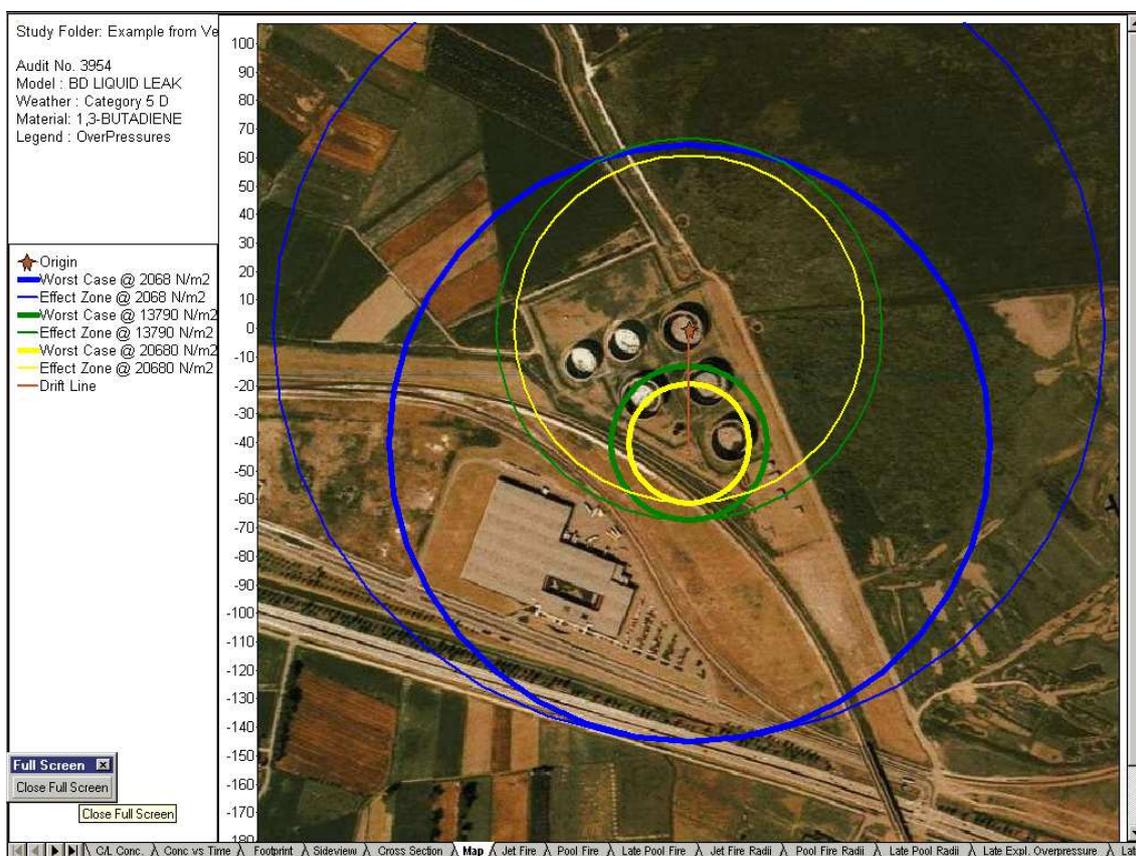


Fig. 1. Zonas de afección del accidente simulado

PHAST tiene la apariencia característica de una aplicación de escritorio (*Figura 2*), incluyendo en su cabecera un menú de herramientas principal. A su izquierda dispone de un menú de tipo árbol donde se pueden ver los distintos accidentes que han sido definidos, indicando en cada caso (a través de iconos preestablecidos) el tipo de accidente del que se trata.

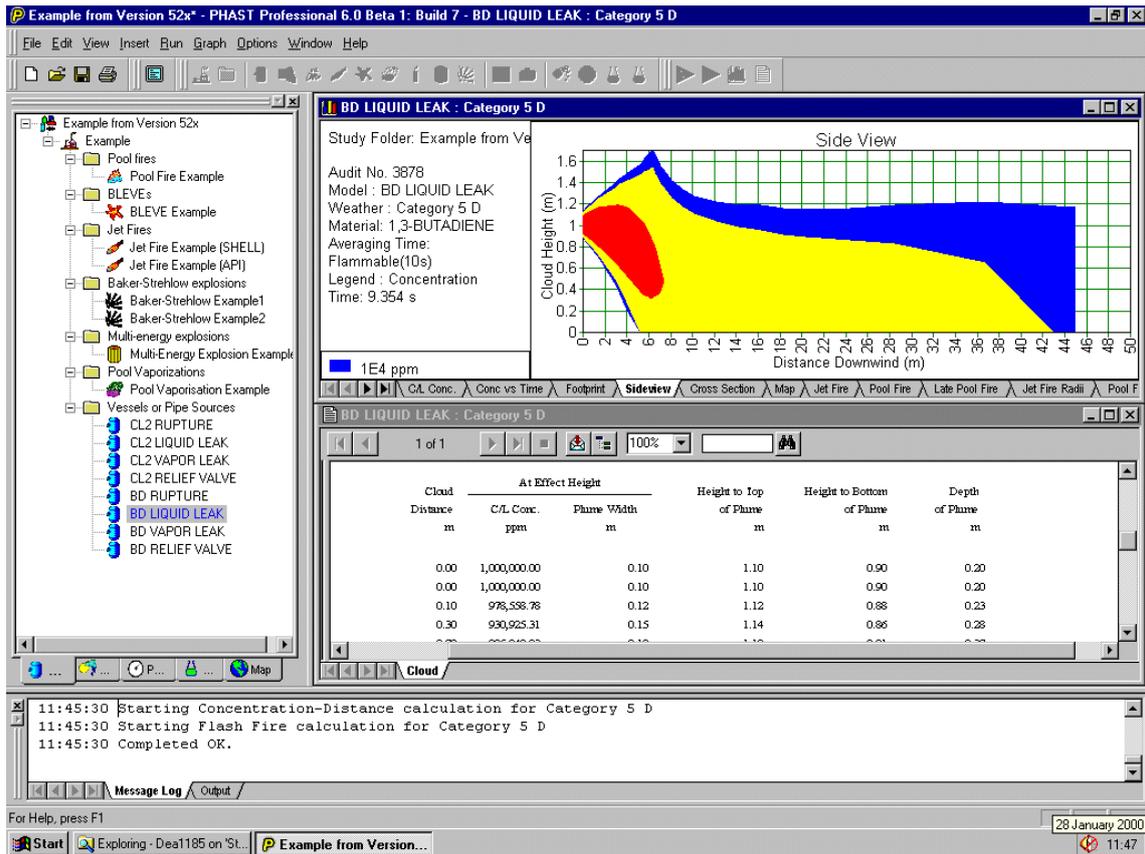


Fig. 2. Apariencia del PHAST

La sección derecha contiene dos ventanas:

- La primera muestra una gráfica donde quedan representados los valores obtenidos tras la simulación del accidente seleccionado (en el menú de árbol de la izquierda).
- La segunda, muestra los mismos valores pero representados en una tabla

Por último, en la sección inferior, la aplicación dispone de una ventana para indicar lo que va sucediendo a través de mensajes.

2.2. ZEUS

ZEUS, cuyas siglas significan Zonas de Planificación para Explosiones industriales, se trata de un programa para el cálculo de consecuencias y zonas de planificación para accidentes graves de tipo mecánico. Esta aplicación fue desarrollada en el marco de la tesis del Ingeniero Químico Fernando Díaz Alonso (Universidad de Murcia), y lleva por título: Análisis de consecuencias y zonas de planificación para explosiones industriales accidentales (en el ámbito de las directivas seveso).

La aplicación fue creada con el fin de facilitar todos los cálculos llevados a cabo para la obtención de las magnitudes peligrosas y las zonas de planificación expuestas en la tesis. Por este motivo no se puede hacer una comparación exhaustiva, dado que se trata de una aplicación desarrollada para un uso casi personal. A pesar de ello, abarca varios casos dentro de los fenómenos peligrosos* de tipo mecánico, siempre dentro de un entorno industrial, para ser más exactos, explosiones industriales accidentales producidas por:

- Explosivos
- Sustancias inestables o pirotécnicas
- Nubes de vapor no confinadas
- Estallido de recipientes

En la *Figura 3*, se puede ver la pantalla principal de la aplicación, concretamente la pantalla principal.

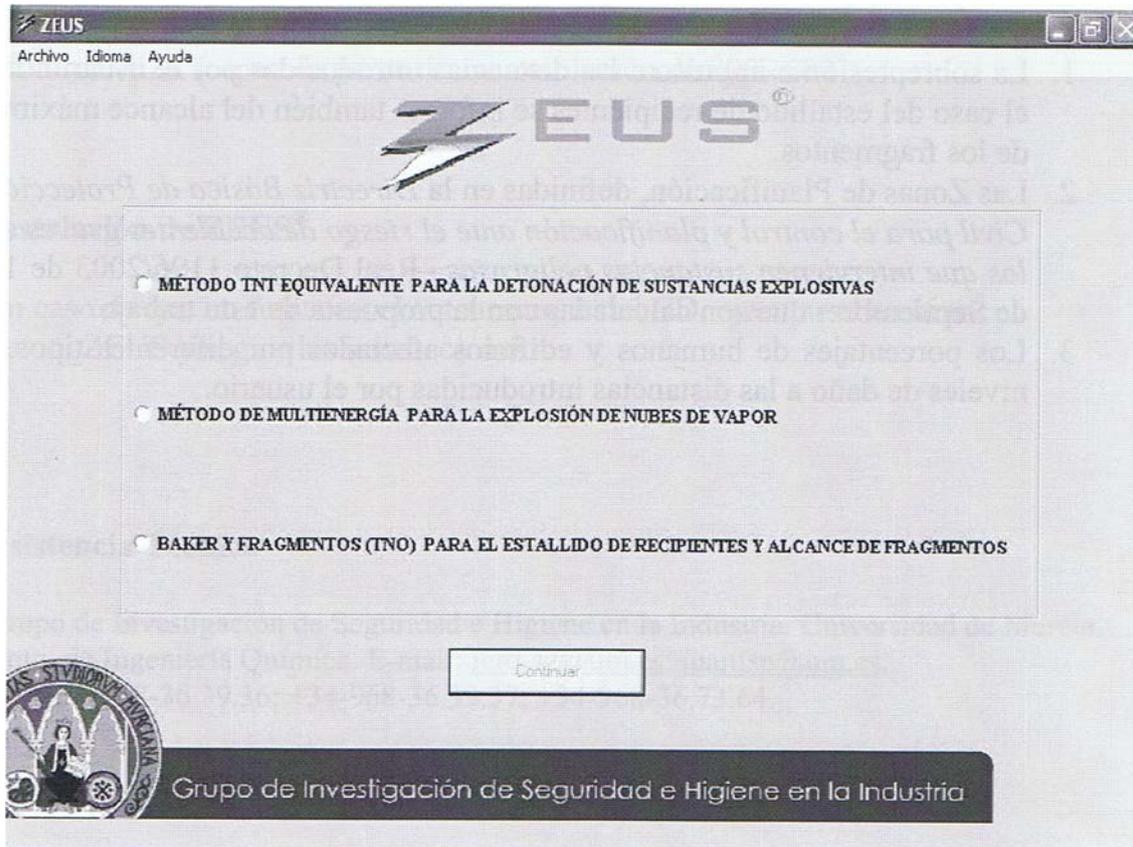


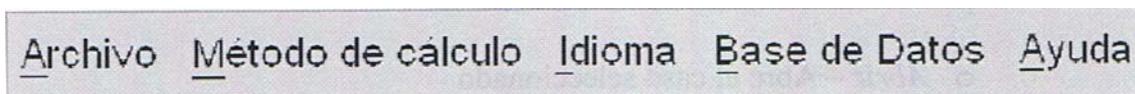
Fig. 3. Pantalla principal de ZEUS

La primera información que solicita la aplicación para empezar a trabajar consiste en la elección del modelo matemático que se empleará a la hora de realizar los cálculos. Una vez haya sido elegido el modelo, se mostrará una pantalla de trabajo donde solicitará la inserción de una serie de datos básicos para poder llevar a cabo la simulación del mismo (*Figura 4*).

Fig. 4. Pantalla de inserción de datos

Cada pantalla de trabajo será diferente en función de la explosión, pero todas tendrán varios elementos comunes, como son:

- **Barra de herramientas.** La barra de herramientas contiene las opciones básicas de cualquier aplicación, como por ejemplo: guardar, abrir, nuevo, salir, etc., así como algunas opciones adicionales de idioma y otras propias del tema que se está tratando (Figura 5).



- **Datos de entrada.** En esta sección será donde se solicitarán los datos necesarios, así como la elección de diversas opciones que determinarán el comportamiento del modelo (Figura 6).
- **Botón de calcular.** Presionando el botón “Calcular” que aparece en la sección inferior de la Figura 6, la aplicación llevará a cabo todos los cálculos necesarios para obtener los resultados.

TNT EQUIVALENTE
 Archivo Método de cálculo Idioma Base de Datos Ayuda

Datos de entrada

Datos de la sustancia

Sustancia

Cantidad kg ▶

Distancias a las que desea conocer los efectos

Distancia 1 m

Distancia 2 m

Distancia 3 m

Distancia-Sobrepresión-Impulso

Seleccione variable

Valor mínimo

Valor máximo

Universidad de Murcia Grupo de Investigación en Seguridad e Higiene en la Industria Copyright 2006

Fig. 6. Datos de entrada para el TNT

Tras calcular los resultados, la aplicación mostrará un conjunto de gráficas y tablas con los valores obtenidos (Figura 7).

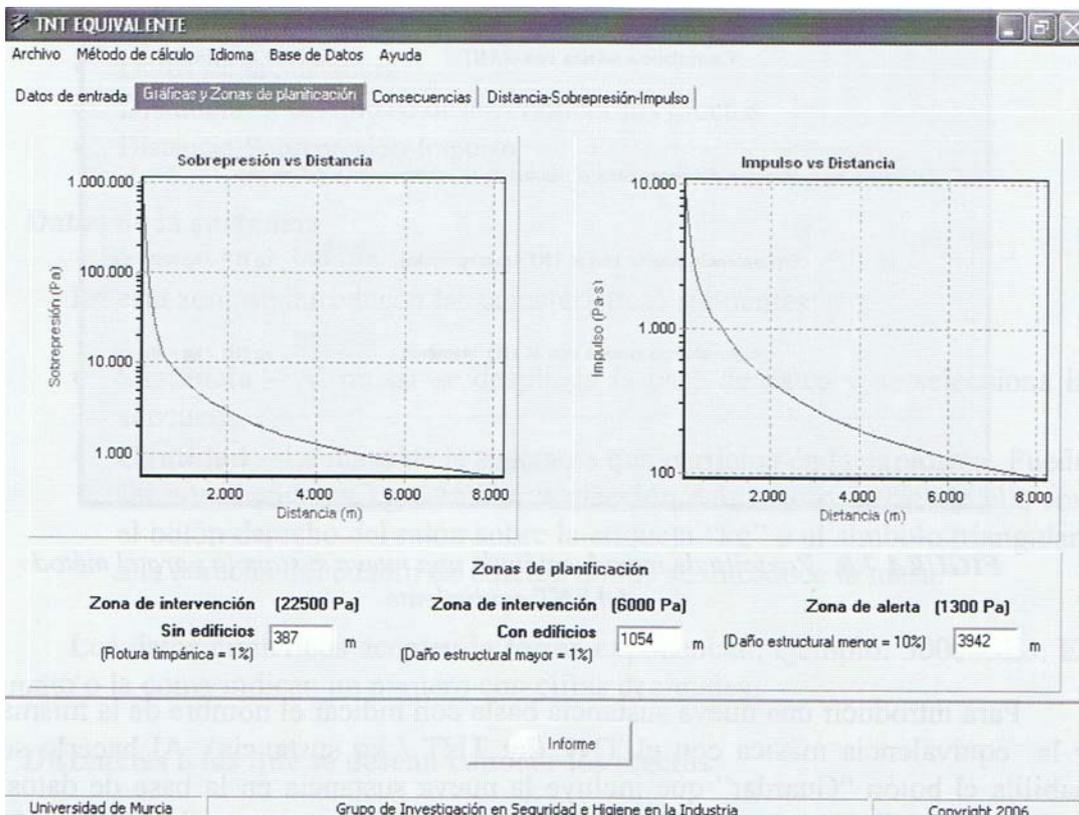


Fig. 7. Resultados de simulación

2.3. Propiedades Comunes

Tanto en el caso de PHAST como de ZEUS, se ofrece la opción de crear nuevas sustancias peligrosas, pero la forma de hacerlo es totalmente distinta (*Figura 8*).

- En el caso de **PHAST**, cuando se desee añadir una nueva sustancia, se tendrán que definir todas aquellas características propias de la sustancia, que puedan llegar a ser necesarias en el conjunto de modelos matemáticos que podrán utilizarla
- en el caso de **ZEUS**, cuando se pretenda añadir una nueva sustancia, también se deberá indicar el modelo matemático en el que se utilizará, de esta forma, sólo será necesario rellenar aquellas propiedades necesarias para el correcto funcionamiento del modelo.

The image shows a software window titled "NUEVA SUSTANCIA" with a close button in the top right corner. The window is divided into two main sections. The first section, "Información general", contains two input fields: "Sustancia" and "Peso molecular" (with "g/mol" as a unit). The second section, "Reacción de combustión", contains four input fields: "Número de moles estequiométricos de sustancia", "Número de moles estequiométricos de oxígeno", "Velocidad de combustión laminar" (with "m/s" as a unit), and "Entalpía de combustión" (with "J/kg" as a unit). At the bottom of the window is a "Guardar" button.

Fig. 8. Formulario de nueva sustancia

A pesar de haber realizado una búsqueda intensiva sobre las aplicaciones para el cálculo de análisis de riesgos, aquellas que se encuentran están obsoletas o no están a disposición del usuario común, con lo cual no se ha podido realizar un estudio comparativo como con PHAST y ZEUS. A pesar de ello, se muestra una breve descripción sobre su funcionamiento. Entre estas aplicaciones cabe destacar:

- **CIRMA:** la Dirección General de Protección Civil y Emergencias (Ministerio del Interior) realizó un análisis exhaustivo sobre la afección y vulnerabilidad del medio ambiente por accidentes graves ocurridos en establecimientos afectados por el Real Decreto 1254/1999. Además también elaboró una propuesta de metodología para el análisis de riesgos ambientales.

Este programa “gestiona el Cálculo del índice de riesgo medioambiental de los escenarios accidentales que se definan para un establecimiento, en base a los índices globales de consecuencias medioambientales obtenidos y a sus probabilidades/frecuencias.”

El Real Decreto 1254/1999 ya no se encuentra en vigor, por lo que como se comentaba anteriormente, esta aplicación a día de hoy está obsoleta.

- **SHELL FRED 5.0** “(Incendio, Fugas, Explosión, Dispersión, es un software que permite simular las consecuencias de las fugas de un producto, tanto accidentales como intencionadas.”
- **ALOHA 5.3 de la EPA:** “es un programa que se utiliza como modelo de dispersión atmosférica para evaluar escapes de vapores de sustancias químicas peligrosas.”
- **EFFECTS TNO 5.5:** “es un programa que calcula los efectos físicos debidos a escapes de sustancias peligrosas.
Calcula riesgos medioambientales
Contiene una base de datos de propiedades físicas de más de 100 sustancias peligrosas.”
- **RISKCURVES TOOLBOX 4.0.:** “se utiliza para realizar el análisis cuantitativo del riesgo en la industria química o petroquímica. Calcula todo tipo de riesgos, como por ejemplo riesgos de la salud o riesgos del transporte.
Análisis de los resultados del riesgo.
Indica si un sistema se encuentra balanceado con su medio.
Está basado en el Yellow Book, Green Book y el Purple Book.”

Capítulo 3. Objetivos

La motivación para realizar este proyecto surge de la necesidad de proporcionarle al CEANI una herramienta que ayude en el estudio de los análisis de riesgos, facilitándole el desarrollo de planes de emergencia en lo que al planteamiento de accidentes se refiere, pudiendo obtener las zonas afectadas en base a los modelos matemáticos desarrollados.

El objetivo principal de este proyecto consiste en desarrollar una plataforma web para el análisis de riesgos en entornos industriales. Para poder satisfacer estos objetivos, se plantea una plataforma compuesta por:

- **Base de datos:** en toda plataforma que almacene datos se necesita hacer uso de una base de datos. En este caso almacenará dos tipos de datos:
 - o los que facilite el sistema
 - o los que aporte el usuario en cada caso para poder resolver un accidente en concreto
- **Web Service:** este elemento de la plataforma proveerá un conjunto de servicios propios de los análisis de riesgos. En la mayoría de los casos el webservice se verá obligado a acceder a la base de datos. En unos casos para recuperar datos del usuario y en otros para utilizar los datos proporcionados por el sistema.
- **Aplicación web:** la parte visible para el cliente, que accederá a los servicios ofrecidos por el webservice para realizar la mayoría de sus funciones.

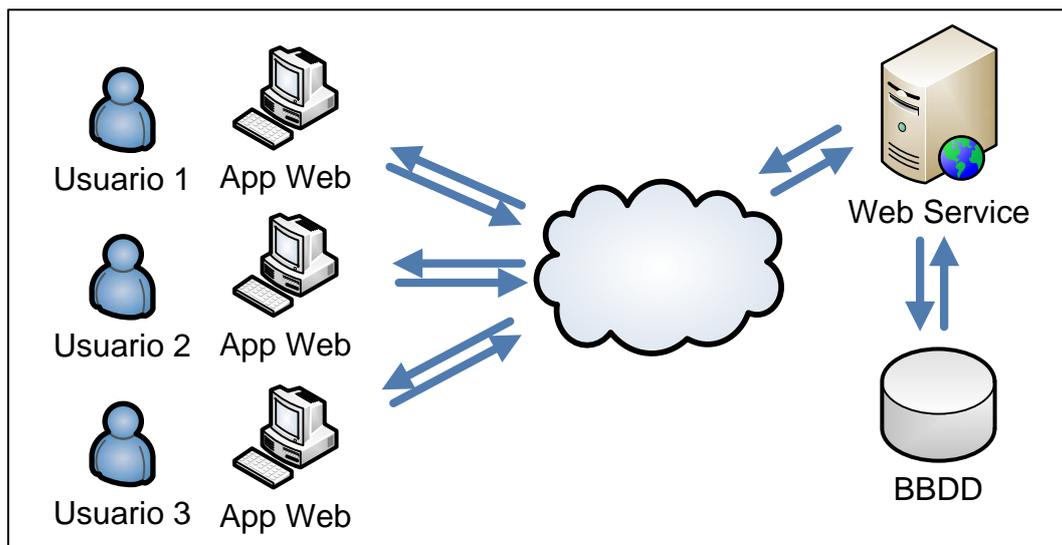


Fig. 9. Arquitectura de la plataforma

Objetivos

El principal objetivo a cubrir por esta plataforma es poder definir los escenarios donde suceden los accidentes, posicionarlos sobre un mapa y mostrar las zonas que se verían afectadas en caso de que se llegasen a producir. Aunque en el capítulo del análisis se hace un estudio más exhaustivo de todos los requerimientos de la aplicación, para satisfacer este objetivo principal, se han de cubrir los siguientes objetivos básicos:

- Definir nuevos escenarios para realizar las simulaciones
- Guardar los escenarios definidos para su uso posterior
- Poder modificar los escenarios definidos
- Crear nuevos accidentes asociados al escenario
- Posicionar los accidentes en el mapa
- Visualizar de forma gráfica los resultados obtenidos en las simulaciones
- Visualizar en forma de texto los resultados obtenidos en las simulaciones
- Definir nuevas zonas de afección
- Definir nuevas sustancias
- Definir nuevos recursos
- Cargar mapas del usuario

Capítulo 4. Recursos necesarios

Dada la naturaleza del proyecto, durante el estudio y desarrollo del mismo ha sido necesario utilizar diversos recursos, tanto hardware como software que se listarán a lo largo de esta sección, indicando en cada caso a quién pertenece.

La mayoría de los recursos utilizados para el desarrollo del proyecto han sido provistos por La División de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), perteneciente a la Universidad de Las Palmas de Gran Canaria; además de los recursos que se nombrarán a continuación, también han sido utilizadas las instalaciones del centro a nivel de una zona donde establecer los recursos prestados.

Recursos Hardware	Propietario
Ordenador con procesador Intel Pentium 4 a 3.00 GHz y 3 GB de memoria RAM	CEANI
Pantalla Dell 19''	CEANI
Pantalla Phillips 17''	CEANI
Ordenador portátil Dell Inspiron 1545 de 15,4'' (Intel Core Duo – 2GHz y 4GB de memoria RAM)	Alumno
Pantalla LG 17''	Alumno
Impresora láser blanco y negro	CEANI
Impresora multifunción Lexmark	Alumno
Pendrive 4 GB TDK	Alumno
Disco duro portátil WD 160 GB	Alumno

Tabla 1. Recursos utilizados a lo largo del desarrollo del proyecto

El desarrollo no ha sido orientado al contexto del software libre, sin embargo también se ha llegado a utilizar alguna de estas herramientas. A continuación se detalla el conjunto de programas que han sido utilizados.

Recursos Software	Propietario/Tipo de licencia
Sistema Operativo Windows Vista Enterprise	CEANI
Sistema Operativo Windows Vista Home Premium	Alumno (licencia propia)
Adobe Flex Builder	Alumno (licencia de estudiante)
Microsoft Visual Studio .NET 2008	Alumno (licencia de estudiante, convenio de la ULPGC)
Microsoft SQL Server 2008 Enterprise	Alumno (licencia de estudiante, convenio de la ULPGC)
Microsoft Office 2007	Alumno (licencia de estudiante, convenio de la ULPGC)
Microsoft Visio 2007	Alumno (licencia de estudiante, convenio de la ULPGC)
Internet Explorer 8	Gratuito
Mozilla Firefox 3.5	Gratuito

Tabla 2. Software utilizado para el desarrollo del proyecto

Capítulo 5. Metodología

La metodología que se ha seguido para el desarrollo del ciclo de vida de este proyecto consiste en el modelo evolutivo: iterativo e incremental. En este modelo se realizan iteraciones completas con el mismo número de fases. La primera fase será considerada como la fase inicial y el resto serán refinamientos de la misma.

Esta metodología posibilita una realimentación entre clientes y desarrolladores. Tras cada iteración, existe un prototipo del software que puede ser valorado por el cliente, lo que conlleva una pronta detección de errores, así como la inserción de nuevos requisitos que sean ajustables a los ya definidos.

Las fases que componen este ciclo de vida son (ver *Figura 10*):

5.1. Análisis

Durante el **análisis** se realizará la especificación de los requisitos de la aplicación, dividiéndolos entre requisitos funcionales y no funcionales. Cada requisito vendrá definido por una tabla compuesta de varios campos, donde se explicará la funcionalidad que se pretende cubrir. A continuación, unido a cada requisito vendrá un diagrama de casos de uso y un conjunto de tablas explicativas de los mismos.

Las herramientas utilizadas durante las fases de análisis han sido, principalmente, las proporcionadas por el lenguaje de modelado UML. Para la recogida de requisitos tanto funcionales como no funcionales, se hará uso de entrevistas, cuestionarios, etc.

5.2. Diseño

En la fase de **diseño**, como su propio nombre indica, se realizará el diseño de las distintas partes que componen la plataforma: cliente web, servicio web y base de datos.

Se presentarán un conjunto de diagramas de la aplicación web y del webservice, donde se pueden observar las distintas entidades que participan, así como la forma en que interactúan. Para la base de datos se realizará un diagrama de modelo de la base de datos que contiene el conjunto de tablas con las propiedades, cada tabla de la base de datos incorporará también una tabla explicativa.

En esta parte también se debe llevar a cabo un estudio de las distintas tecnologías existentes y que podrían llegar a utilizarse para la implementación de la plataforma. Una vez hecho se deberá decidir qué lenguaje se utilizará, el sistema gestor de la base de datos y cómo se comunicarán cada una de las partes de la plataforma.

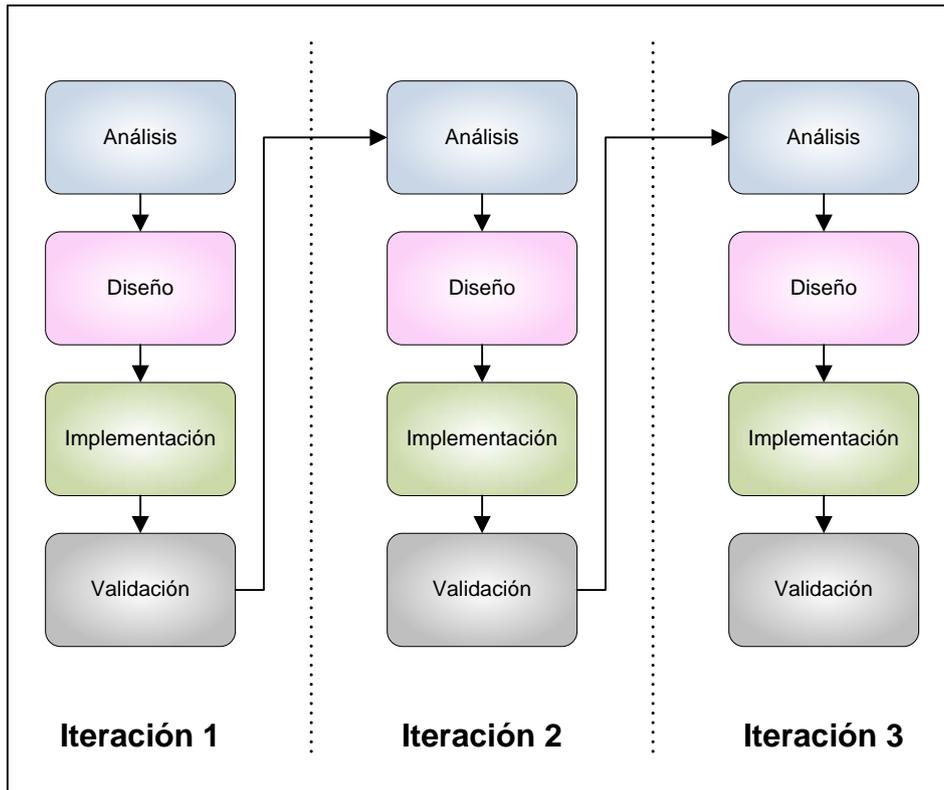


Fig. 10. Fases del ciclo de vida del modelo evolutivo: iterativo e incremental

5.3. Implementación

La fase de **implementación** consiste en llevar a cabo el desarrollo de los modelos obtenidos durante la fase de diseño, que a su vez han sido construidos en base a los requisitos obtenidos en la fase de análisis.

En la parte de diseño ya se habrá decidido qué lenguajes y herramientas se deberán utilizar, por lo que sólo restaría instalar cada una de ellas para poder hacer uso de las mismas.

5.4. Validación

Durante el periodo de **validación** se establecerán una serie de pruebas que aseguren el correcto funcionamiento de la plataforma. Para poder asegurar este hecho, no solo se ha de comprobar que la funcionalidad requerida esté cubierta, sino que, por otra parte, también habrá que comprobar que en caso de que alguna de las conexiones entre las distintas partes de la plataforma fallase, el sistema no se vendría abajo.

Para comprobar que se ha cubierto toda la funcionalidad definida en la parte del análisis, se plantearán una serie de escenarios haciendo pruebas de cada uno de los casos de uso. Y para la comprobación de las conexiones, se forzará el fallo de cada una de ellas revisando la reacción del sistema.

Capítulo 6. Análisis

En esta sección se realiza un análisis de la aplicación, realizando primeramente un estudio del problema planteado, así como las necesidades que se pretenden cubrir con su desarrollo. A continuación se presenta un estudio exhaustivo de ambos.

6.1. Descripción del problema / Contextualización de la plataforma

La plataforma desarrollada como proyecto está categorizada dentro de la temática de la Seguridad Industrial. En este ámbito, se presenta el Análisis del Riesgo como una herramienta fundamental para la planificación ante emergencias derivadas de posibles accidentes en la actividad industrial.

La Dirección General de Protección Civil y Emergencias, entidad que depende del Ministerio del Interior del Gobierno de España, hace una clasificación de los tipos de riesgos, diferenciando entre Riesgos Naturales y Riesgos Tecnológicos. Esta definición se hace extensible al resto de países europeos al igual que otros continentes.

Dentro de la clasificación establecida por el Gobierno, el análisis se ha centrado en la sección de Riesgos Tecnológicos. Estos riesgos se encuentran fuertemente relacionados con las actividades del transporte y de la industria, quedando englobados dentro del ámbito de la Seguridad Industrial, como ya se comentó anteriormente.

La Seguridad industrial se puede dividir en 3 secciones:

- Seguridad laboral
- Seguridad de los productos industriales
- Seguridad de los procesos e instalaciones industriales

Cualquier accidente de las citadas secciones puede llegar a ocasionar graves consecuencias sobre personas, bienes materiales o medioambiente. Actualmente el número de accidentes que se registran a nivel laboral es muy bajo, debido en mayor medida a los estudios que se realizan para poder evitarlos. Por este motivo se ha de seguir avanzando de forma que las nuevas tecnologías aplicadas en la industria no afecten negativamente en estos valores. Para lograr un equilibrio en este sentido, se ha de llevar un control sobre el riesgo de que se produzcan estos accidentes.

6.1.1. Accidentes graves

Dentro de los accidentes, se denomina accidente grave aquel que tiene repercusiones sobre las personas y el medioambiente, provocando la emisión de sustancias tóxicas o de energía por encima de lo que se considera normal. La emisión de energía suele ir acompañada de fenómenos peligrosos del tipo térmico (incendios) o mecánico (detonaciones, deflagraciones,...), favoreciendo en ocasiones la dispersión de los agentes tóxicos involucrados. En relación a los accidentes relativos a la actividad

industrial, la Dirección General de Protección Civil y Emergencias diferencia entre dos grupos de riesgos, los **riesgos químicos** por una parte y los **riesgos nucleares o radiológicos** por otra. Este proyecto se centra en la ayuda al estudio de los riesgos químicos, en concreto, en los accidentes declarados en el ámbito del Real Decreto 1254/1999, sobre medidas de control de los riesgos inherentes a los accidentes graves en los que intervengan sustancias peligrosas, quedando definidos los mismos como “Cualquier suceso, tal como una emisión en forma de fuga o vertido, incendio o explosión importantes, que sea consecuencia de un proceso no controlado durante el funcionamiento de cualquier establecimiento al que sea de aplicación el presente Real Decreto, que suponga una situación de grave riesgo, inmediato o diferido, para las personas, los bienes y el medio ambiente, bien en el interior o exterior del establecimiento, y en el que estén implicadas una o varias sustancias peligrosas”[Gobierno de España – 1999]. El citado Real Decreto se encuentra en el ámbito legislativo español, no obstante, este transpone la Directiva Europea 96/82/CE al régimen jurídico del país.

6.1.2. Legislación sobre “accidentes graves”

La Ley 21/1992 de Industria es la referencia para la legislación en materia de seguridad industrial en España. En ella se indica que las actividades relacionadas con la seguridad e higiene en el trabajo se rigen por normativa específica, diferente a la normativa relativa a lo que se ha denominado como “accidentes graves”.

La legislación específica sobre “riesgo químico” consiste principalmente en el Real Decreto 1254/1999, sobre medidas de control de los riesgos que van asociados a los accidentes graves producidos, en los que intervengan sustancias peligrosas. También son de aplicación los Reales Decretos 119/2005 y 948/2005, que modifican al anterior, y la Directriz Básica de Protección Civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas.

6.1.3. Real Decreto 1254/1999 y posteriores modificaciones

El Real Decreto 1254/1999 tiene como finalidad la prevención de accidentes graves en los que intervengan sustancias peligrosas, así como controlar el nivel de sus consecuencias teniendo por objeto la protección de las personas, los bienes y el medio ambiente.

Dentro del mismo se define al industrial como “cualquier persona física o jurídica que explote o posea el establecimiento o la instalación, o cualquier persona en la que se hubiera delegado, en relación con el funcionamiento técnico, un poder económico determinante”. Según se establece en las partes 1 y 2 del anexo I del Real Decreto, el industrial estará obligado a elaborar un informe de seguridad en función de la cantidad de sustancias peligrosas presentes y de los umbrales presentes en el propio anexo.

Siempre que el Real Decreto sea de aplicación en un establecimiento industrial, se obliga a la elaboración del Análisis del Riesgo. Dependiendo de la cantidad de sustancia peligrosa que haya en cada caso, el documento puede variar en su contenido, sin embargo, a nivel general son muy similares.

El Análisis del Riesgo consiste en una herramienta que se utiliza para controlar el riesgo. Esta herramienta permite, entre otros aspectos, la identificación de acciones encaminadas a la prevención de la ocurrencia de accidentes, así como la planificación de los pasos a seguir en caso de que este se produzca, con el propósito de minimizar las consecuencias del accidente.

6.1.4. Real Decreto 1196/2003

Este es el Real Decreto por medio del cual se aprueba la Directriz Básica de protección civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas. Por medio de la misma se realiza una revisión de la Directriz Básica anteriormente en vigor, bajo las consideraciones citadas en el Real Decreto 1254/1999 presentado anteriormente, en aspectos tan relevantes como son la adopción de políticas de prevención de accidentes graves, el establecimiento de sistemas de gestión de la seguridad, el desarrollo de planes de autoprotección, la elaboración de informes de seguridad, la consideración del efecto dominó, la realización de inspecciones, la ordenación territorial teniendo en cuenta el riesgo de accidentes graves y la información al público.

En el Artículo 4 de la citada Directriz, se expone el objetivo y contenido que ha de satisfacer al Análisis del Riesgo que debe de contener el Informe de Seguridad al que hace referencia el Artículo 9 del Real Decreto 1254/1999, exigido, como se comentó anteriormente, a aquellos industriales en cuyo establecimiento estén presentes sustancias peligrosas, en función de las cantidades expuestas en las partes 1 y 2 del anexo I del Real Decreto 1254/1999, y más concretamente en su columna 3. Por medio del Análisis del Riesgo se pretenden identificar los accidentes graves que pueden ocurrir en el establecimiento, así como el cálculo de las consecuencias y daños producidos por aquellos. El contenido de este Análisis del Riesgo es el que se expone a continuación:

- Identificación de peligros de accidentes graves.
- Cálculo de consecuencias. Zonas de riesgo según valores umbrales.
- Cálculo de vulnerabilidad.
- Relación de accidentes graves identificados.
- Medidas de prevención, control y mitigación.

No obstante, la legislación incluye lo que se denomina Análisis Cuantitativo de Riesgos (ACR). Este pudiera ser exigido por la autoridad competente en el caso de considerarlo oportuno.

6.1.5. Zonas de afección más importantes

Aunque se pueden definir distintas zonas, según se establece en la Directriz básica de protección civil citada en este documento, las zonas de afección más importantes que vienen definidas por el valor umbral establecido son:

- **Zona de intervención:** Es aquella en la que las consecuencias de los accidentes producen un nivel de daños que justifica la aplicación inmediata de medidas de protección.
- **Zona de alerta:** Es aquella en la que las consecuencias de los accidentes provocan efectos que, aunque perceptibles por la población, no justifican la intervención, excepto para los grupos críticos.
- **Efecto dominó:** la concatenación de efectos causantes de riesgo que multiplica las consecuencias, debido a que los fenómenos peligrosos pueden afectar, además de los elementos vulnerables exteriores, otros recipientes, tuberías o equipos del mismo establecimiento o de otros establecimientos próximos, de tal manera que se produzca una nueva fuga, incendio, estallido en ellos, que a su vez provoquen nuevos fenómenos peligrosos.

6.2. Ámbito de la plataforma desarrollada

Como se ha venido explicando hasta ahora, la plataforma desarrollada en este proyecto serviría como herramienta en el área del Análisis del riesgo. Una vez se hayan identificado los peligros de accidentes graves, se procede al cálculo de consecuencias de los mismos.

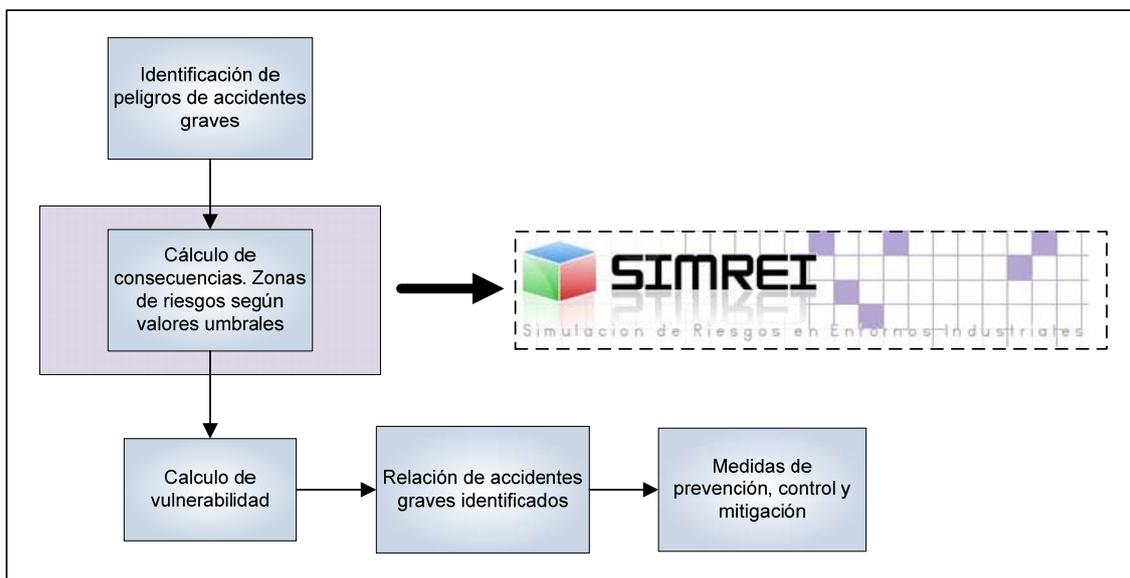


Fig. 11. Ubicación de la aplicación dentro del estudio de los riesgos

La fase de cálculo de consecuencias captura los datos necesarios de la fase anterior y, los resultados obtenidos, están calculados en base a un conjunto de modelos matemáticos que definen el comportamiento del accidente en caso de haberse producido.

Los resultados conseguidos por los modelos matemáticos definirán las distintas zonas que se verían afectadas por el accidente, quedando sometidas a una determinada presión que vendrá definida por el valor umbral establecido (por ejemplo: zona de alerta, zona de intervención ...). Cada valor umbral proporcionará una zona de afección distinta.

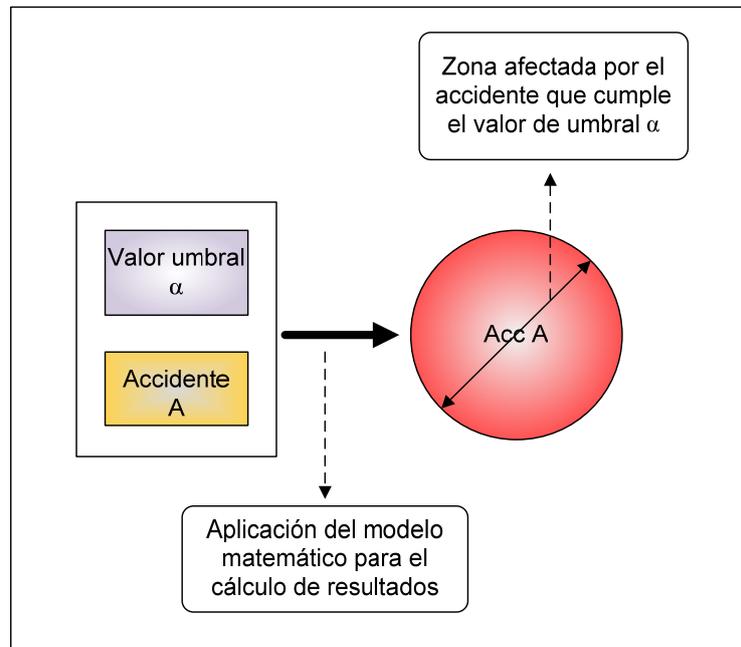


Fig. 12. Ejemplo

6.3. Modelos matemáticos utilizados

El conjunto de modelos matemáticos que se utilizan para el cálculo de consecuencias no es de dominio público, por tanto, se han desarrollado aquellos que estaban disponibles:

6.3.1. Método equivalente TNT

El TNT (trinitrotolueno) es un explosivo convencional. Ha sido uno de los explosivos más utilizados a nivel militar, lo cual ha permitido que sus efectos se hayan estudiado ampliamente. Este método permite predecir de una forma rápida y sencilla los daños ocasionados por la explosión de una nube de vapor no confinada, calculándolo a partir de la masa de TNT que equivaldría a la cantidad de hidrocarburo implicado, es decir, que produciría el mismo nivel de daños. Por ende, es uno de los métodos más utilizados.

La fórmula que se utiliza para el cálculo del resultado es:

$$W_{TNT} = W_{exp} \left(\frac{W_{TNT}}{W_{exp}} \right)_{explosivo}$$

Donde:

W_{TNT} : masa equivalente de TNT

W_{exp} : masa del explosivo

$(W_{TNT}/W_{exp})_{explosivo}$: equivalencia de TNT por unidad de masa de explosivo

Existe una tabla de referencia donde se indican los coeficientes para calcular las equivalencias de distintos hidrocarburos con respecto al TNT:

Explosivo	Equivalencia (sobrepresión)	Equivalencia (impulso)
Nitroglicerina	1.69	1.69
PETN	1.77	1.77
Ácido Pírico	1.06	1.06
RDX	1.57	1.57
Trinitrobenceno	1.11	1.11
2.4.-Dinitrotolueno	0.77	0.77
Nitrato Amónico	0.56	0.56
Peróxido de ciclohexanona	0.19	0.19
Clorato sódico	0.15	0.15
Nitrometano	1.34	1.34
Composición A-3	1.09	1.076
Composición C-4	1.37	1.19
Coclotol (70/30)	1.14	1.09
HBX-1	1.17	1.16
HBX-3	1.14	0.97
H-6	1.38	1.15
Minol II	1.20	1.11
Pentolita	1.38	1.14
Picratol	0.90	0.93
TNETB	1.36	1.10
Tritonal	1.07	0.96

Una vez se tiene la masa del explosivo se puede calcular la masa de TNT equivalente a la cantidad de sustancia que participa en la explosión:

$$W_{TNT} = W_{exp} * \text{coeficiente de la tabla}$$

A continuación se calcula la distancia escalada (z') a la que se origina una determinada presión, en base a las siguientes ecuaciones

$$1.13 \times 10^5 \geq P_s \geq 11000; P_s = 1.13 \times 10^5 z'^{(-2.01)}$$
$$11000 \geq P_s \geq 400; P_s = 1.83 \times 10^5 z'^{(-1.16)}$$

En función del valor umbral de la zona que se desee calcular, se utilizará la primera o la segunda ecuación. La primera será para valores comprendidos entre 1.13×10^5 y 11000 pascales y la segunda para valores comprendidos entre 11000 y 400 pascales.

Teniendo la z' calculada, sólo faltaría obtener el resultado del modelo, que será la distancia del área afectada partiendo desde el centro del accidente.

$$z' = \frac{z}{(W_{TNT})^{1/3}}$$

despejando el valor que falta por calcular:

$$z = z'(W_{TNT})^{1/3}$$

siendo z el resultado del modelo.

La Directriz básica de protección civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas, establece los valores umbrales para la estimación de las zonas de planificación ante emergencias. En cuanto al riesgo de impulsos y sobrepresiones se han establecido los valores umbrales siguientes:

6.3.1.1. Valores umbrales para la zona de intervención

- Valor local integrado del impulso debido a la onda de presión: 150 milibar segundo (15000 pascales)
- Sobrepresión local estática de la onda de presión de 125 milibar (12500 pascales)

6.3.1.2. Valores umbrales para la zona de alerta

- Valor local integrado del impulso debido a la onda de presión: 100 milibar segundo (10000 pascales)

- Sobrepresión local estática de la onda de presión de 50 milibar (5000 pascales)

6.3.1.3. Valores umbrales para la zona de efecto dominó

- Sobrepresión local estática de la onda de presión de 160 milibar (16000 pascales)

6.3.2. Fuga de Líquido Inflamable

Cuando la cisterna que contiene una sustancia líquida pierde su capacidad de retención, el líquido se fuga y se extiende ocupando una superficie que depende de la existencia de obstáculos que impidan su progresión. Si el líquido es inflamable, la parte que pudiera estar evaporando desde la superficie del charco generado (es dependiente de la volatilidad de la sustancia) es susceptible de ocasionar el incendio. Esto ocurre cuando los vapores emitidos se mezclan con el aire y cuando se sobrepasa el límite inferior de inflamabilidad y se aporta la energía de activación necesaria. Estos vapores son lo que arde realmente. Este incidente se denomina “incendio de charco”.

6.3.2.1. El efecto de los incendios

Son provocados por la oxidación rápida, no explosiva, de sustancias combustibles, produciendo llama, que puede ser estacionaria (incendio de charco, dardo de fuego) o progresiva (llamarada, bola de fuego), pero que en todos los casos disipa la energía de combustión mayoritariamente por radiación que puede afectar a seres vivos e instalaciones materiales.

Si la materia sobre la que incide el flujo de radiación térmica no puede disiparlo a la misma velocidad que lo recibe, éste provoca un incremento de su temperatura. Si este incremento no se limita, se producen alteraciones irreversibles y catastróficas, que pueden culminar en la combustión o fusión y volatilización de la materia expuesta.

En las proximidades del punto donde se desarrolla la llama, se tiene transmisión del calor tanto por convección como por radiación y conducción. Así pues, la única forma de evitar o mitigar sus efectos es la utilización de equipos de protección individual frente al calor o el fuego o protecciones adecuadas. En contraposición, a partir de una cierta distancia del foco del incendio, la transmisión del calor se efectúa exclusivamente por radiación, disminuyendo su intensidad al aumentar dicha distancia. Esto hace que cualquier pantalla opaca a la radiación térmica pueda constituir una medida de protección sumamente eficaz.

La variable por medio de la cual se describen los fenómenos peligrosos del tipo térmico es la Dosis de radiación, D , recibida por los seres humanos procedentes de las llamas o cuerpos incandescentes. Esta es expresada mediante:

$$D = I_m^{4/3} t_{exp}$$

Donde:

- I_m es la intensidad media recibida, en kW/m²
- t_{exp} el tiempo de exposición, en segundos.

Esta expresión es válida para intensidades superiores a 1.7 kW/m², para valores inferiores al anterior, el tiempo de exposición es prácticamente irrelevante, esto es, se considera que en dichas condiciones, la mayoría de la población puede estar expuesta durante dilatados períodos de tiempo sin sufrir daño.

Con fines de planificación, en los incendios de corta duración, inferiores a un minuto, el tiempo de exposición se hace coincidir con la duración de éstos; para los de mayor duración, se establece como tiempo de exposición el transcurrido hasta que los afectados alcancen una zona protegida frente a la radiación o donde la intensidad térmica sea inferior a 1.7 kW/m².

Para este último caso y con objeto de determinar las distancias que delimitan las zonas de intervención y alerta, se recomienda seguir el modelo de respuesta de la población ante la génesis de incendios, propuesto por TNO, en el que se establece un primer período de reacción de unos cinco segundos, donde la población permanece estática y a continuación se produce la huída, alejándose del incendio a una velocidad media de 4 m/s.

La Directriz básica de protección civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas, establece los valores umbrales para la estimación de las zonas de planificación ante emergencias. Estos se indican a continuación para el caso de ocurrencia de incendios:

6.3.2.2. Valores umbrales para la zona de intervención

Una dosis de radiación térmica de 250 (kW/m²)^{4/3} · s, equivalente a las combinaciones de intensidad térmica y tiempo de exposición que se indican a continuación.

I, kW/m ²	7	6	5	4	3
t _{exp} , s	20	25	30	40	60

6.3.2.3. Valores umbrales para la zona de alerta

Una dosis de radiación térmica de 115 (kW/m²)^{4/3} · s, equivalente a las combinaciones de intensidad térmica y tiempo de exposición que se indican a continuación.

I, kW/m ²	6	5	4	3	2
t _{exp} , s	11	15	20	30	45

6.3.2.4. Valores umbrales para la zona de efecto dominó

Radiación térmica: 8 kW/m².

6.3.2.5. Utilización del modelo

Debido a que los modelos de cálculo de estas magnitudes físicas resultan muy costosos computacionalmente, el equipo del CEANI, que es quien ha proporcionado este modelo, ha optado por el empleo de Redes Neuronales para aproximar los resultados a los aportados por herramientas software de reconocido prestigio internacional para las que se han implementado los citados modelos.

Aunque las redes neuronales requieren un importante esfuerzo de cómputo durante su entrenamiento (se han realizado más de 900 simulaciones y se han entrenado más de 3000 redes), una vez entrenada, el proceso de cálculo se realiza de forma casi inmediata.

6.4. Análisis de requisitos y casos de uso

Durante la fase de análisis se ha contado con 2 tipos de expertos, por un lado el analista de riesgos, que será el futuro usuario de la aplicación y, por otro lado, los expertos en tecnologías de la información. A través de entrevistas y cuestionarios se han obtenido el conjunto de requisitos funcionales y no funcionales que definen el comportamiento de la plataforma.

6.4.1. Roles

Se encuentran dos tipos de usuarios que harán uso de la plataforma a lo largo del ciclo de vida de la misma. Ambos se consideran actores principales, ya que harán uso directo.

- **Usuario de la aplicación:** será el encargado de hacer uso de la aplicación web. Utilizará toda la funcionalidad para realizar el cálculo de consecuencias, partiendo de los datos obtenidos en la identificación de los riesgos. El experto o analista de riesgos será este tipo de usuario.
- **Administrador de la BBDD:** será el encargado de actualizar los datos de la aplicación, modificándolos cuando sea necesario o incluyendo nuevos datos que podrán utilizar todos los usuarios. La actualización de los datos se hará directamente a través del Sistema Gestor de Base de Datos (SGBD).

6.4.2. Requisitos funcionales

Los requisitos funcionales describen las tareas específicas que el sistema debe ser capaz de llevar a cabo.

A continuación se presenta el conjunto de requisitos funcionales que satisfacen las necesidades descritas por el usuario experto a través de las entrevistas y los cuestionarios. Cada requisito se describirá en una tabla modelo cuyos campos son [Durán, A. & Bernárdez, B. - 2002]:

RF - X	<nombre descriptivo>
Autores	<autor de la versión actual>
Fuentes	<fuente de la versión actual>
Actores asociados	<nombre de los actores>
Requisitos asociados	Rx-y<nombre del requisito>
Descripción	El sistema deberá comportarse tal como se describe en los siguientes casos de uso.
Importancia	<Importancia del objetivo>
Estabilidad	<Estabilidad del objetivo>
Comentarios	<Comentarios adicionales sobre el objetivo>

Identificador y nombre descriptivo: cada requisito se debe identificar por un código único y un nombre descriptivo. Los identificadores de los requisitos funcionales comienzan con *RF*.

Autores y Fuentes: estos dos campos contienen el nombre de los autores (desarrolladores) y de las fuentes (clientes o usuarios), de la versión actual del requisito, de forma que la rastreabilidad sea efectiva, llegando hasta las personas que hicieron la propuesta de necesidad del requisito.

Actores asociados: este campo contiene el nombre del tipo de usuario (rol) que llevará a cabo la acción asociada al requisito funcional.

Requisitos asociados: en este campo se indican otros requisitos que están asociados por algún motivo con el requisito que se está describiendo, de esta forma se consigue tener una rastreabilidad horizontal, es decir, se podrán obtener todas las relaciones existentes entre el conjunto de requisitos.

Descripción: este campo, tal y como su nombre indica, recoge la descripción del requisito funcional, se explica el objetivo al que hace referencia.

Importancia: este campo indica la importancia del cumplimiento del requisito para los clientes y usuarios. Se puede asignar un valor numérico o alguna expresión que indique el grado de importancia, como: *vital*, *urgente*, *importante*, etc.

Estabilidad: este campo indica la estabilidad del requisito, es decir, una cuantificación de la probabilidad de que pueda sufrir cambios en el futuro. Esta estabilidad se puede

establecer mediante un valor numérico o mediante una expresión que indique el grado de estabilidad, como: *alta*, *media* o *baja*, o *PD* (*por determinar*) si es que aún no se ha determinado.

Comentarios: en este campo se puede añadir cualquier otra información sobre el requisito que no encaje en los campos establecidos en la tabla.

RF - 1	Gestionar usuarios
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos, Nayarit Santana Pacheco
Actores asociados	Usuario
Requisitos asociados	-
Descripción	Los usuarios tienen que poder loguearse en la aplicación para poder hacer uso de la misma. En caso de no tener acceso, deben poder crear un nuevo usuario
Importancia	Vital
Estabilidad	Media
Comentarios	Como campos obligatorios para el usuario se deben guardar: <ul style="list-style-type: none"> - Nombre de usuario - Contraseña - Email Como campos opcionales se deben guardar los datos personales.

RF - 2	Gestionar sustancias
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario
Requisitos asociados	-
Descripción	Los usuarios deben tener acceso al listado de sustancias disponibles, crear nuevas sustancias, modificarlas, eliminarlas y además poder ver sus propiedades.
Importancia	Vital
Estabilidad	Alta
Comentarios	

RF - 3	Gestionar recursos
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario

Requisitos asociados	-
Descripción	Los usuarios deben tener acceso al listado de recursos disponibles, crear nuevos recursos, modificarlos, eliminarlos y además poder ver sus propiedades.
Importancia	Media
Estabilidad	Alta
Comentarios	

RF - 4	Gestionar mapas
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario
Requisitos asociados	-
Descripción	Los usuarios deben tener acceso al listado de mapas disponibles, subir nuevos mapas y ver sus propiedades
Importancia	Vital
Estabilidad	Alta
Comentarios	Los mapas deben ser imágenes en formato: .jpg, .gif, .png, .tiff El usuario tiene que poder establecer las coordenadas del mapa

RF - 5	Gestionar zonas
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario
Requisitos asociados	-
Descripción	Los usuarios deben tener acceso al listado de zonas de riesgo disponibles, crear nuevas zonas, modificarlas y eliminarlas, además de ver sus propiedades
Importancia	Importante
Estabilidad	Alta
Comentarios	

RF - 6	Gestionar escenarios
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario
Requisitos asociados	RF2, RF3, RF4, RF5
Descripción	Los usuarios deben tener acceso al conjunto de escenarios disponibles, crear nuevos escenarios, modificarlos, eliminarlos y

	revisar sus propiedades
Importancia	Vital
Estabilidad	Media
Comentarios	Los usuarios sólo podrán ver e interactuar con los escenarios que sean de su propiedad. Los accidentes sólo se podrán posicionar en los mapas definidos.

RF - 7	Gestionar accidentes
Autores	Nayarit Santana Pacheco
Fuentes	Analista de Riesgos
Actores asociados	Usuario
Requisitos asociados	RF2, RF3, RF4, RF5, RF6
Descripción	Los usuarios deben tener acceso al conjunto de accidentes que existe dentro de cada escenario
Importancia	Vital
Estabilidad	Media
Comentarios	Los usuarios sólo podrán trabajar con los accidentes cuando estén dentro de un escenario. Los accidentes sólo podrán usar sustancias y zonas definidas.

6.4.3. Casos de uso

Una vez definidos los requisitos, se han realizado los casos de uso que los satisfacen. Además de los diagramas, se ha utilizando una plantilla para plasmar toda la información referente a cada caso de uso. Esta plantilla, al igual que la utilizada para la definición de requisitos, consiste en una tabla cuyos campos son [Durán, A. & Bernárdez, B. - 2002]:

CU-X	
Autor:	
Requisitos asociados:	
Actor Principal:	
Personal involucrado e intereses:	
Precondiciones:	
Garantías de éxito (Postcondiciones):	
Escenario principal de éxito (o Flujo Básico):	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

Identificador, Autor: estos dos campos tienen el mismo significado que en la tabla vista en la sección de requisitos, pero en este caso, haciendo referencia al caso de uso.

Requisitos asociados: en este campo se añaden los requisitos que quedarán cubiertos con el caso de uso que se está describiendo.

Actor principal: este campo contendrá el nombre del usuario que llevará a cabo la acción descrita en el caso de uso. En algunos casos, varios actores podrán realizar la acción, con lo cual, habrá más de un actor principal indicado.

Personal involucrado e intereses: en este campo se delimita qué es lo que debe hacer el sistema en función de los distintos actores que participan, de esta forma se establecen todos los objetivos que deben ser cumplidos con el caso de uso tratado.

Precondiciones: en este campo se expresan en lenguaje natural las condiciones necesarias para que se pueda realizar el caso de uso.

Garantías de éxito (Postcondiciones): en este campo se expresan en lenguaje natural las condiciones que se deben cumplir después de la terminación normal del caso de uso.

Escenario principal de éxito (o flujo básico): este campo describe el camino de éxito típico que satisface los intereses del personal involucrado. Normalmente no incluye ninguna condición o bifurcación, ya que es más inteligible postergar todo el manejo de caminos condicionales a la sección de extensiones.

Extensiones (o flujos alternativos): en este campo se indican todos los otros escenarios o bifurcaciones, tanto de éxito como de fracaso, que no se encuentran dentro del camino típico (escenario principal de éxito). El conjunto del apartado anterior y este apartado conformarán casi todos los intereses del personal involucrado.

Requisitos especiales: si un requisito no funcional, atributo de calidad o restricción se relaciona de manera específica con un caso de uso, se debe recoger dentro del caso de uso en este apartado.

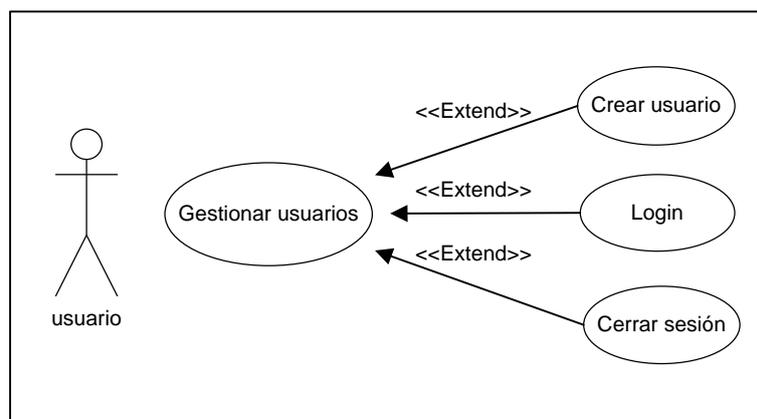


Fig. 13. Diagrama que cubre el Requisito Funcional 1 (RF-1)

CU-1	Crear usuario
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF1
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere acceder a la aplicación	
Precondiciones: El usuario no tiene una cuenta en la aplicación para poder acceder	
Garantías de éxito (Postcondiciones): El usuario obtiene su acceso a la aplicación con un nuevo nombre de usuario y contraseña	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario se conecta a la aplicación 2. El usuario selecciona la opción de crear nuevo usuario 3. Rellena los datos 4. Guarda los datos 5. La aplicación guarda el nuevo usuario y lanza un aviso 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1. El usuario en lugar de guardar los datos, cancela el registro 4.2. La aplicación no guarda los datos <p>-----</p> <ol style="list-style-type: none"> 5.1. La aplicación lanza un aviso indicando que no se han rellenado todos los campos obligatorios 5.2. El usuario rellena los campos que faltan 5.3. Guarda los datos 	
Requisitos especiales:	

CU-2	Login
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF1
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere acceder a la aplicación	
Precondiciones: El usuario tiene una cuenta de usuario en la aplicación	
Garantías de éxito (Postcondiciones): El usuario accede a la pantalla principal de la aplicación una vez ha introducido correctamente su usuario y contraseña	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario se conecta a la aplicación 2. Selecciona la opción de inicio de sesión 3. Introduce usuario y contraseña 4. Accede a la pantalla principal de la aplicación 	
Extensiones (o Flujos Alternativos):	

3.1. El usuario introduce un usuario y/o contraseña que no son correctos
3.2. El usuario recibe un mensaje de logueo incorrecto
Requisitos especiales:

CU-3	Cerrar sesión
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF1
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere salir de la aplicación	
Precondiciones: El usuario tiene una cuenta de usuario en la aplicación y está logueado en ella	
Garantías de éxito (Postcondiciones): El usuario sale de la aplicación y regresa a la pantalla de inicio	
Escenario principal de éxito (o Flujo Básico): 1. El usuario pica en el enlace de cerrar sesión	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

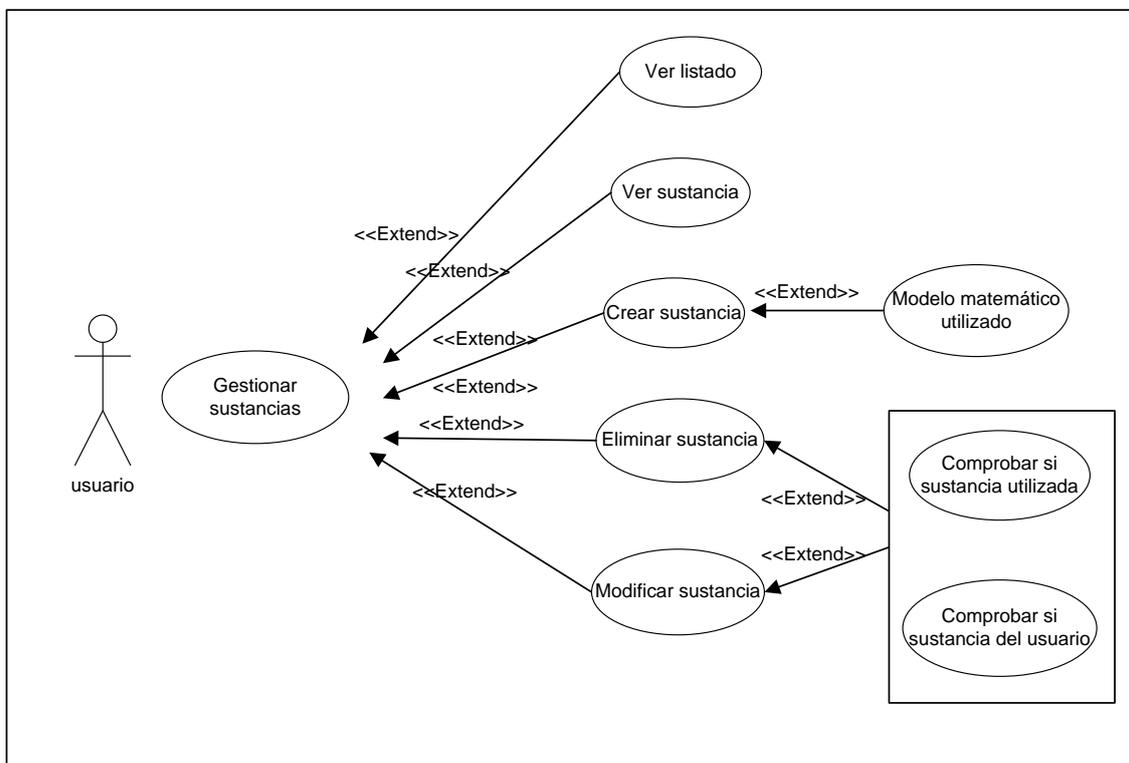


Fig. 14. Diagrama que cubre el Requisito Funcional 2 (RF-2)

CU-4	Ver listado
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF2
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de las sustancias disponibles	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza todas las sustancias	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de sustancias 2. La aplicación muestra el listado de sustancias disponibles 	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

CU-5	Ver sustancia
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF2
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver las propiedades de una sustancia	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza las propiedades de la sustancia	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 3. El usuario accede a la sección de sustancias 4. Selecciona la sustancia que quiere ver 5. La aplicación muestra las propiedades de la sustancia seleccionada 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El usuario selecciona la sustancia incorrecta 2.2. La aplicación muestra las propiedades de la sustancia no deseada 	
Requisitos especiales:	

CU-6	Crear sustancia
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF2
Actor Principal:	Usuario
Personal involucrado e intereses:	

Usuario: el usuario quiere definir una nueva sustancia
Precondiciones: El usuario está logueado
Garantías de éxito (Postcondiciones): El usuario puede utilizar la nueva sustancia definida
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de crear nueva sustancia 2. Rellena todos los campos obligatorios 3. Guarda los datos
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1.1. El usuario intenta guardar la sustancia sin haber rellenado los campos obligatorios 3.1.2. La aplicación lanza un mensaje de aviso indicando que faltan datos <p>-----</p> <ol style="list-style-type: none"> 5.2.1. El usuario no guarda los datos, en su lugar cancela 5.2.2. Los datos no se guardan en la aplicación
Requisitos especiales:

CU-7	Modificar sustancia
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF2
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere modificar alguna de las propiedades de una sustancia	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario puede utilizar la sustancia actualizada	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la sustancia que desea modificar 2. Selecciona la opción de modificar sustancia 3. Cambia los datos que quiere actualizar 4. Guarda los cambios 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1.1. El usuario intenta guardar los cambios dejando algún campo obligatorio vacío 4.1.2. La aplicación lanza un mensaje de alerta indicándolo <p>-----</p> <ol style="list-style-type: none"> 4.2.1. El usuario no guarda los datos, en su lugar cancela 4.2.2. La sustancia mantiene las propiedades que tenía 	
Requisitos especiales:	

CU-8	Eliminar sustancia
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF2
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar una sustancia	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): La sustancia es eliminada de la aplicación	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la sustancia que desea borrar 2. Selecciona la opción de borrar sustancia 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma que desea borrar la sustancia 5. La sustancia es eliminada de la aplicación 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 5.1.1. La sustancia está siendo utilizada en algún accidente 5.1.2. La aplicación lanza una alerta indicando que no se puede borrar la sustancia porque está siendo utilizada 5.1.2. La sustancia no se borra de la aplicación <p>-----</p> <ol style="list-style-type: none"> 5.2.1. El usuario no guarda los datos, en su lugar cancela 5.2.2. La sustancia no se borra de la aplicación 	
Requisitos especiales:	

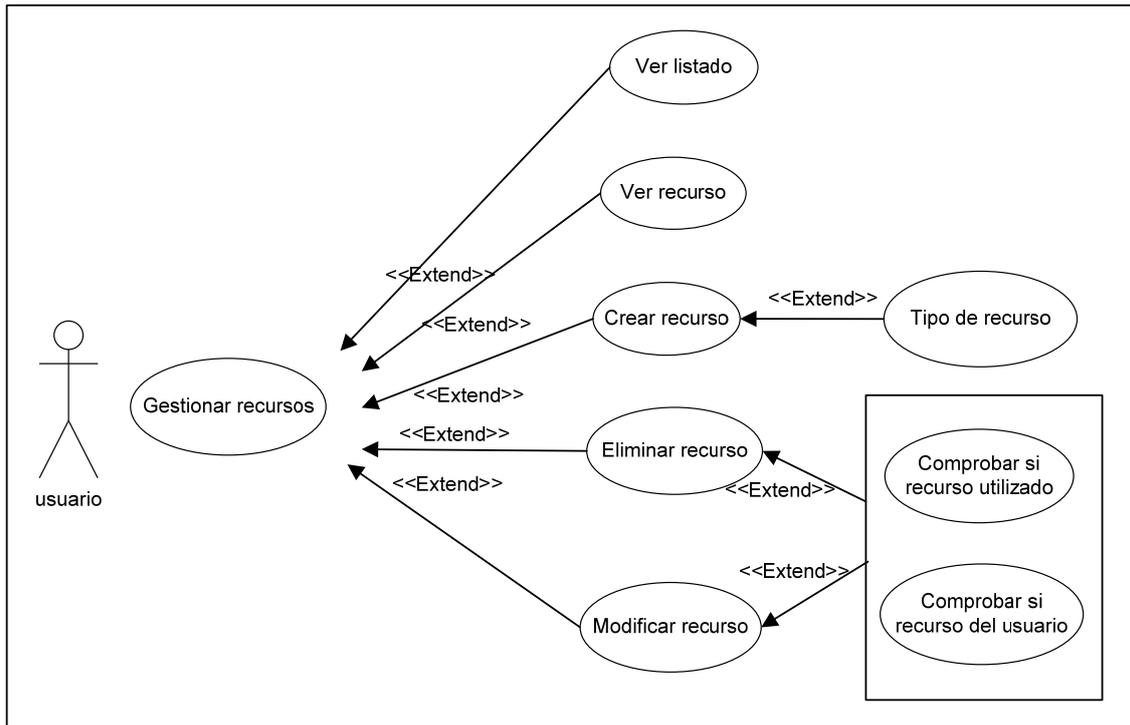


Fig. 15. Diagrama que cubre el Requisito funcional 3 (RF-3)

CU-9	Ver listado
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF3
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de los recursos disponibles	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza todos los recursos	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de recursos 2. La aplicación muestra el listado de recursos disponibles 	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

CU-10	Ver recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF3
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver las propiedades de un recurso	

Precondiciones: El usuario está logueado
Garantías de éxito (Postcondiciones): El usuario visualiza las propiedades del recurso
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de recursos 2. Selecciona el recurso que quiere ver 3. La aplicación muestra las propiedades del recurso seleccionado
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El usuario selecciona el recurso incorrecto 2.2. La aplicación muestra las propiedades del recurso no deseado
Requisitos especiales:

CU-11	Crear recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF3
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere definir un nuevo recurso	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario puede utilizar el nuevo recurso definido	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de crear nuevo recurso 2. Rellena todos los campos obligatorios 3. Guarda los datos 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1.1. El usuario intenta guardar el recurso sin haber rellenado los campos obligatorios 3.1.2. La aplicación lanza un mensaje de aviso indicando que faltan datos ----- 3.2.1. El usuario no guarda los datos, en su lugar cancela 3.2.2. Los datos no se guardan en la aplicación 	
Requisitos especiales:	

CU-12	Modificar recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF3
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere modificar alguna de las propiedades de un recurso	

<p>Precondiciones: El usuario está logueado</p>
<p>Garantías de éxito (Postcondiciones): El usuario puede utilizar el recurso actualizado</p>
<p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El usuario selecciona el recurso que desea modificar 2. Selecciona la opción de modificar recurso 3. Cambia los datos que quiere actualizar 4. Guarda los cambios
<p>Extensiones (o Flujos Alternativos):</p> <ol style="list-style-type: none"> 4.1.1. El usuario intenta guardar los cambios dejando algún campo obligatorio vacío 4.1.2. La aplicación lanza un mensaje de alerta indicándolo ----- 4.2.3. El usuario no guarda los datos, en su lugar cancela 4.2.4. El recurso mantiene las propiedades que tenía
<p>Requisitos especiales:</p>

CU-13	Eliminar recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF3
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar un recurso	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El recurso es eliminado de la aplicación	
Escenario principal de éxito (o Flujo Básico):	
<ol style="list-style-type: none"> 1. El usuario selecciona el recurso que desea borrar 2. Selecciona la opción de borrar recurso 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma que desea borrar el recurso 5. El recurso es eliminado de la aplicación 	
Extensiones (o Flujos Alternativos):	
<ol style="list-style-type: none"> 5.1.1. El recurso está siendo utilizado en algún accidente 5.1.2. La aplicación lanza una alerta indicando que no se puede borrar el recurso porque está siendo utilizado 5.1.3. El recurso no se borra de la aplicación ----- 5.2.1. El usuario no guarda los datos, en su lugar cancela 5.2.2. El recurso no se borra de la aplicación 	
Requisitos especiales:	

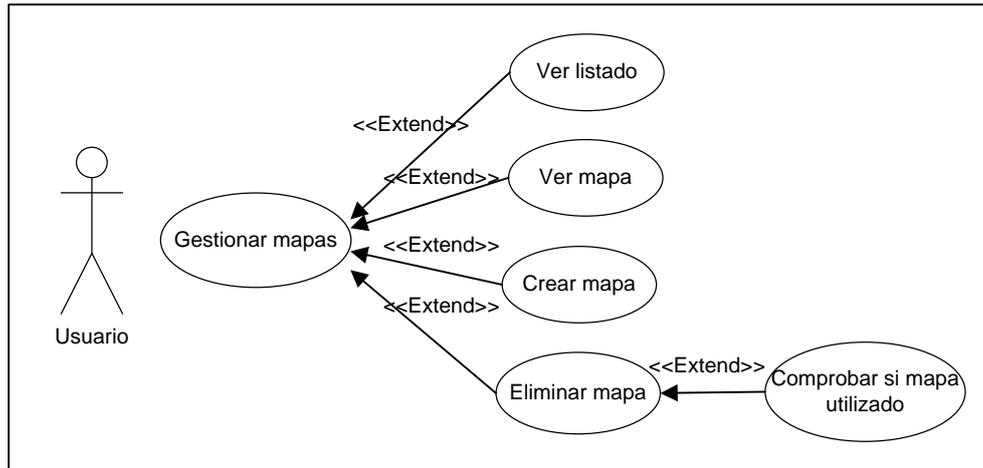


Fig. 16. Diagrama que cubre el Requisito Funcional 4 (RF-4)

CU-14	Ver listado
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF4
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de los mapas disponibles	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza todos los mapas	
Escenario principal de éxito (o Flujo Básico): 1. El usuario accede a la sección de mapas 2. La aplicación muestra el listado de mapas disponibles	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

CU-15	Ver mapa
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF4
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver las propiedades de un mapa	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza las propiedades del mapa	
Escenario principal de éxito (o Flujo Básico):	

<ol style="list-style-type: none"> 1. El usuario accede a la sección de mapas 2. Selecciona el mapa que quiere ver 3. La aplicación muestra las propiedades del mapa seleccionado
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El usuario selecciona el mapa incorrecto 2.2. La aplicación muestra las propiedades del mapa no deseado
Requisitos especiales:

CU-16	Crear mapa
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF4
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere definir un nuevo mapa	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario puede utilizar el nuevo mapa definido	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 4. El usuario selecciona la opción de crear nuevo mapa 5. Rellena todos los campos obligatorios 6. Guarda los datos 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 6.1.1. El usuario intenta guardar el mapa sin haber rellenado los campos obligatorios 6.1.2. La aplicación lanza un mensaje de aviso indicando que faltan datos ----- 3.2.1. El usuario no guarda los datos, en su lugar cancela 3.2.2. Los datos no se guardan en la aplicación 	
Requisitos especiales:	

CU-17	Eliminar mapa
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF4
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar un mapa	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El mapa es eliminado de la aplicación	
Escenario principal de éxito (o Flujo Básico):	

<ol style="list-style-type: none"> 1. El usuario selecciona el recurso que desea borrar 2. Selecciona la opción de borrar recurso 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma que desea borrar el recurso 5. El recurso es eliminado de la aplicación
<p>Extensiones (o Flujos Alternativos):</p> <ol style="list-style-type: none"> 5.1.1. El mapa está siendo utilizado en algún accidente 5.1.2. La aplicación lanza una alerta indicando que no se puede borrar el mapa porque está siendo utilizado 5.1.4. El mapa no se borra de la aplicación <p>-----</p> <ol style="list-style-type: none"> 5.2.1. El usuario no guarda los datos, en su lugar cancela 5.2.2. El mapa no se borra de la aplicación
<p>Requisitos especiales:</p>

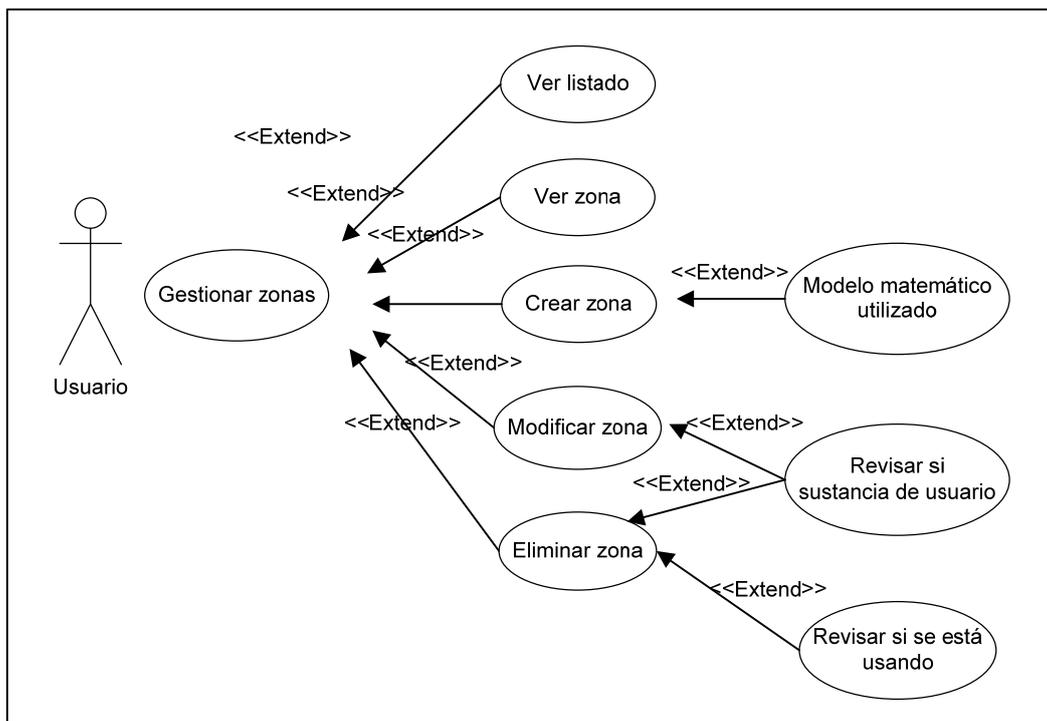


Fig. 17. Diagrama que cubre el requisito funcional 5 (RF-5)

CU-18	Ver listado
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF5
Actor Principal:	Usuario
Personal involucrado e intereses:	
Usuario: el usuario quiere ver el listado de las zonas disponibles	

Precondiciones: El usuario está logueado
Garantías de éxito (Postcondiciones): El usuario visualiza todas las zonas
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de zonas 2. La aplicación muestra el listado de zonas disponibles
Extensiones (o Flujos Alternativos):
Requisitos especiales:

CU-19	Ver zona
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF5
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver las propiedades de una zona	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza las propiedades de la zona	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de zonas 2. Selecciona la zona que quiere ver 3. La aplicación muestra las propiedades de la zona seleccionada 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El usuario selecciona la zona incorrecta 2.2. La aplicación muestra las propiedades de la zona no deseada 	
Requisitos especiales:	

CU-20	Crear zona
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF5
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere definir una nueva zona	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario puede utilizar la nueva zona definida	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de crear nueva zona 2. Rellena todos los campos obligatorios 	

3. Guarda los datos
Extensiones (o Flujos Alternativos):
3.1.1. El usuario intenta guardar la zona sin haber rellenado los campos obligatorios
3.1.2. La aplicación lanza un mensaje de aviso indicando que faltan datos

3.2.1. El usuario no guarda los datos, en su lugar cancela
3.2.2. Los datos no se guardan en la aplicación
Requisitos especiales:

CU-21	Modificar zona
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF5
Actor Principal:	Usuario
Personal involucrado e intereses:	
Usuario: el usuario quiere modificar alguna de las propiedades de una zona	
Precondiciones:	
El usuario está logueado	
Garantías de éxito (Postcondiciones):	
El usuario puede utilizar la zona actualizada	
Escenario principal de éxito (o Flujo Básico):	
1. El usuario selecciona la zona que desea modificar	
2. Selecciona la opción de modificar zona	
3. Cambia los datos que quiere actualizar	
4. Guarda los cambios	
Extensiones (o Flujos Alternativos):	
4.1.1. El usuario intenta guardar los cambios dejando algún campo obligatorio vacío	
4.1.2. La aplicación lanza un mensaje de alerta indicándolo	

4.2.1. El usuario no guarda los datos, en su lugar cancela	
4.2.2. La zona mantiene las propiedades que tenía	
Requisitos especiales:	

CU-22	Eliminar zona
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF5
Actor Principal:	Usuario
Personal involucrado e intereses:	
Usuario: el usuario quiere eliminar una zona	
Precondiciones:	
El usuario está logueado	

<p>Garantías de éxito (Postcondiciones): La zona es eliminada de la aplicación</p>
<p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El usuario selecciona la zona que desea borrar 2. Selecciona la opción de borrar zona 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma que desea borrar la zona 5. La zona es eliminada de la aplicación
<p>Extensiones (o Flujos Alternativos):</p> <ol style="list-style-type: none"> 5.1.1. La zona está siendo utilizada en algún accidente 5.1.2. La aplicación lanza una alerta indicando que no se puede borrar la zona porque está siendo utilizada 5.1.3. La zona no se borra de la aplicación <p>-----</p> <ol style="list-style-type: none"> 5.2.1. El usuario no guarda los datos, en su lugar cancela 5.2.2. La zona no se borra de la aplicación
<p>Requisitos especiales:</p>

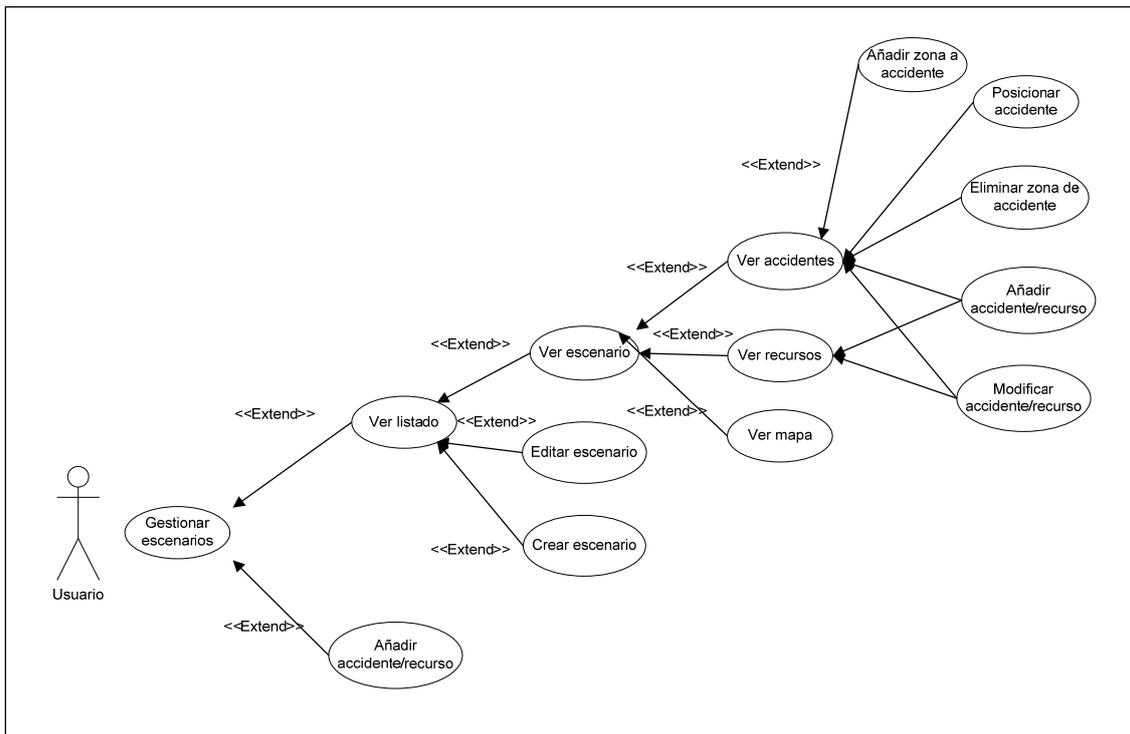


Fig. 18. Diagrama que cubre el requisito funcional 6 (RF-6)

CU-23	Ver listado
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario

Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de escenarios
Precondiciones: El usuario está logueado
Garantías de éxito (Postcondiciones): El usuario visualiza todas los escenarios
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de escenarios 2. La aplicación muestra el listado de escenarios del usuario
Extensiones (o Flujos Alternativos):
Requisitos especiales:

CU-24	Ver escenario
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver el contenido de un escenario	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza el escenario	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de escenarios 2. La aplicación muestra el listado de escenarios del usuario 3. El usuario selecciona el escenario que quiere visualizar 4. La aplicación muestra el contenido del escenario: mapa, accidentes y recursos asociados 	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

CU-25	Crear escenario
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere crear un nuevo escenario	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza el escenario creado	

<p>Escenario principal de éxito (o Flujo Básico):</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de crear nuevo escenario 2. La aplicación solicita los datos necesarios 3. El usuario rellena los datos necesarios <ol style="list-style-type: none"> 3.1. El usuario asocia un nuevo mapa al escenario <ol style="list-style-type: none"> 3.1.1. Selecciona el mapa que desea asociar 3.2. El usuario asocia un nuevo accidente al escenario <ol style="list-style-type: none"> 3.2.1. Rellena los datos del nuevo accidente 3.2.2. Posiciona el accidente en el mapa 3.2.3. Asocia una zona al accidente 3.2.4. Elimina una zona del accidente 3.3. El usuario asocia un nuevo recurso al accidente <ol style="list-style-type: none"> 3.3.1. Rellena los datos del nuevo recurso 3.3.2. Posiciona el recurso en el mapa 4. Guarda los datos
<p>Extensiones (o Flujos Alternativos):</p> <p>Requisitos especiales:</p> <ol style="list-style-type: none"> 3.1.1. El usuario no asocia mapa al escenario ----- 3.2.1. El usuario no asocia ningún accidente al escenario ----- 3.3.1. El usuario no asocia ningún recurso al escenario ----- 3.2.3.1. El usuario no asocia ninguna zona al accidente ----- 4.1. El usuario no guarda los datos, en su lugar cancela 4.2. Se pierden todos los datos sobre el nuevo escenario

CU-26	Editar escenario
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses:	
Usuario: el usuario quiere ver el contenido de un escenario	
Precondiciones:	
El usuario está logueado y tiene un escenario seleccionado	
Garantías de éxito (Postcondiciones):	
El usuario actualiza el escenario	
Escenario principal de éxito (o Flujo Básico):	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción modificar 2. Cambia los datos deseados <ol style="list-style-type: none"> 2.1. Añade nuevo accidente <ol style="list-style-type: none"> 2.1.1. Asocia una zona al accidente 2.2. Modifica accidente <ol style="list-style-type: none"> 2.2.1. Asocia una zona al accidente 	

<ul style="list-style-type: none"> 2.2.2. Elimina una zona del accidente 2.3. Elimina accidente existente 2.4. Añade nuevo recurso 2.5. Modifica recurso 2.6. Elimina recurso existente 2.7. Modifica el nombre del escenario 2.8. Modifica la descripción del escenario 3. Guarda los cambios 4. La aplicación muestra los datos actualizados
<p>Extensiones (o Flujos Alternativos):</p> <ul style="list-style-type: none"> 3.1. El usuario en lugar de guardar los cambios cancela 3.2. La aplicación no guarda los datos 3.3. La aplicación mantiene los datos antiguos <p>-----</p> <ul style="list-style-type: none"> 4.1. El usuario ha dejado campos obligatorios vacíos, la aplicación muestra el aviso 4.2. Guarda las modificaciones
<p>Requisitos especiales:</p>

CU-27	Ver accidentes
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
<p>Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de accidentes asociados al escenario seleccionado</p>	
<p>Precondiciones: El usuario está logueado y tiene un escenario seleccionado</p>	
<p>Garantías de éxito (Postcondiciones): El usuario visualiza el listado de accidentes</p>	
<p>Escenario principal de éxito (o Flujo Básico):</p> <ul style="list-style-type: none"> 1. El usuario accede a la sección de accidentes 2. La aplicación muestra el listado de accidentes asociados al escenario 	
<p>Extensiones (o Flujos Alternativos):</p> <ul style="list-style-type: none"> 2.1. El escenario no tiene ningún accidente asociado 2.2. La aplicación no muestra ningún accidente 	
<p>Requisitos especiales:</p>	

CU-28	Ver recursos
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
<p>Requisitos especiales:</p>	

Personal involucrado e intereses: Usuario: el usuario quiere ver el listado de recursos asociados al escenario seleccionado
Precondiciones: El usuario está logueado y tiene un escenario seleccionado
Garantías de éxito (Postcondiciones): El usuario visualiza el listado de recursos
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede a la sección de recursos 2. La aplicación muestra el listado de recursos asociados al escenario
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El escenario no tiene ningún recurso asociado 2.2. La aplicación no muestra ningún recurso
Requisitos especiales:

CU-29	Ver mapa
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere ver el mapa asociado al escenario	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): El usuario visualiza el mapa	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario accede al escenario 2. La aplicación muestra el mapa asociado al escenario 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 2.1. El escenario no tiene ningún mapa asociado 2.2. La aplicación no muestra ningún mapa 	
Requisitos especiales:	

CU-30	Añadir accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere añadir un accidente a un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario	

Garantías de éxito (Postcondiciones): El escenario tiene asociado un nuevo accidente
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de añadir nuevo accidente 2. Rellena los datos solicitados por la aplicación 3. Guarda los datos
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1.1. El usuario no guarda los datos, en su lugar cancela 3.1.2. El escenario conserva el número de accidentes inicial <p>-----</p> <ol style="list-style-type: none"> 3.2.1. La aplicación lanza un aviso indicando que no se han cubierto los campos obligatorios 3.2.2. El usuario rellena todos los campos obligatorios 3.2.3. Guarda los datos 3.2.4. El escenario pasa a tener un accidente más
Requisitos especiales:

CU-31	Añadir zona a accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere añadir una zona a un accidente	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario y dentro del escenario el accidente	
Garantías de éxito (Postcondiciones): El accidente tiene asociada una nueva zona	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de añadir nueva zona 2. Rellena los datos solicitados por la aplicación 3. Guarda los datos 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1. El usuario no guarda los datos, en su lugar cancela 3.2. El escenario conserva el número de zonas inicial 	
Requisitos especiales:	

CU-32	Eliminar zona de accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar una zona de un accidente	

Precondiciones: El usuario está logueado y tiene seleccionado el escenario y dentro del escenario el accidente
Garantías de éxito (Postcondiciones): El accidente tiene asociada una zona menos para simular
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la zona que quiere borrar 2. La aplicación lanza una advertencia de seguridad 3. El usuario confirma el borrado de la zona 4. La aplicación borra la zona del accidente
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1. El usuario no confirma el borrado de la zona 3.2. El escenario conserva el número de zonas inicial
Requisitos especiales:

CU-33	Eliminar accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar un accidente de un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario	
Garantías de éxito (Postcondiciones): El escenario tiene asociado un accidente menos	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el accidente que quiere borrar 2. Selecciona la opción borrar 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma el borrado del accidente 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1. El usuario no confirma el borrado del accidente 4.2. El escenario termina con el número de accidentes inicial 	
Requisitos especiales:	

CU-34	Modificar accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere modificar un accidente de un escenario	

Precondiciones: El usuario está logueado y tiene seleccionado el escenario
Garantías de éxito (Postcondiciones): El accidente es actualizado
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el accidente que quiere modificar 2. Selecciona la opción modificar 3. El usuario modifica los campos que quiere actualizar 4. El usuario guarda los cambios
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1.1. El usuario no guarda los cambios, en su lugar cancela 4.1.2. El accidente mantiene su forma original <p>-----</p> <ol style="list-style-type: none"> 4.2.1. La aplicación lanza un aviso indicando que se han dejado campos obligatorios sin rellenar 4.2.2. El usuario rellena los campos que faltan 4.2.3. Guarda los datos
Requisitos especiales:

CU-35	(Re) Posicionar accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere posicionar un accidente de un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario. El escenario tiene un mapa asociado	
Garantías de éxito (Postcondiciones): El accidente queda posicionado en el mapa	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el accidente que quiere posicionar 2. Selecciona la opción posicionar 3. El usuario selecciona la posición donde quiere ubicar el accidente 4. El usuario guarda los cambios 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1.1. El usuario no guarda los cambios, en su lugar cancela 4.1.2. El accidente mantiene su posición original 	
Requisitos especiales:	

CU-36	Añadir recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6

Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere añadir un recurso a un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario	
Garantías de éxito (Postcondiciones): El escenario tiene asociado un nuevo recurso	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona la opción de añadir nuevo recurso 2. Rellena los datos solicitados por la aplicación 3. Guarda los datos 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 3.1.1. El usuario no guarda los datos, en su lugar cancela 3.1.2. El escenario conserva el número de recursos inicial <p>-----</p> <ol style="list-style-type: none"> 3.2.1. La aplicación lanza un aviso indicando que no se han cubierto los campos obligatorios 3.2.2. El usuario rellena todos los campos obligatorios 3.2.3. Guarda los datos 3.2.4. El escenario pasa a tener un recurso más 	
Requisitos especiales:	

CU-37	Eliminar recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere eliminar un recurso de un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario	
Garantías de éxito (Postcondiciones): El escenario tiene asociado un recurso menos	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el recurso que quiere borrar 2. Selecciona la opción borrar 3. La aplicación lanza una advertencia de seguridad 4. El usuario confirma el borrado del recurso 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1. El usuario no confirma el borrado del recurso 4.2. El escenario termina con el número de recursos inicial 	
Requisitos especiales:	

CU-38	Modificar recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere modificar un recurso de un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario	
Garantías de éxito (Postcondiciones): El recurso es actualizado	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el recurso que quiere modificar 2. Selecciona la opción modificar 3. El usuario modifica los campos que quiere actualizar 4. El usuario guarda los cambios 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1.1. El usuario no guarda los cambios, en su lugar cancela 4.1.2. El recurso mantiene su forma original <hr/> <ol style="list-style-type: none"> 4.2.1. La aplicación lanza un aviso indicando que se han dejado campos obligatorios sin rellenar 4.2.2. El usuario rellena los campos que faltan 4.2.3. Guarda los datos 	
Requisitos especiales:	

CU-39	(Re) Posicionar recurso
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere posicionar un recurso de un escenario	
Precondiciones: El usuario está logueado y tiene seleccionado el escenario. El escenario tiene un mapa asociado	
Garantías de éxito (Postcondiciones): El recurso queda posicionado en el mapa	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el recurso que quiere posicionar 2. Selecciona la opción posicionar 3. El usuario selecciona la posición donde quiere ubicar el recurso 4. El usuario guarda los cambios 	
Extensiones (o Flujos Alternativos): <ol style="list-style-type: none"> 4.1.1. El usuario no guarda los cambios, en su lugar cancela 4.1.2. El recurso mantiene su posición original 	
Requisitos especiales:	

CU-40	Simular accidente
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere obtener los resultados de las zonas afectadas en un accidente	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): La aplicación genera los resultados solicitados	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el escenario 2. Selecciona el accidente 3. Selecciona la opción de simular 4. La aplicación muestra de forma gráfica y numérica las zonas afectadas en base a las zonas asociadas al accidente 	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

CU-41	Generar informe
Autor:	Nayarit Santana Pacheco
Requisitos asociados:	RF6
Actor Principal:	Usuario
Personal involucrado e intereses: Usuario: el usuario quiere obtener los resultados de las zonas afectadas en un documento, con valores numéricos	
Precondiciones: El usuario está logueado	
Garantías de éxito (Postcondiciones): La aplicación genera los resultados solicitados en un documento	
Escenario principal de éxito (o Flujo Básico): <ol style="list-style-type: none"> 1. El usuario selecciona el escenario 2. Selecciona la opción de generar informe 3. La aplicación genera un texto con los datos solicitados 4. El usuario almacena el texto 	
Extensiones (o Flujos Alternativos):	
Requisitos especiales:	

6.4.4. Requisitos no funcionales

Los requisitos no funcionales equivalen a las restricciones establecidas, limitando, de esta forma, las opciones para construir una solución. En las siguientes tablas se describen cada uno de los requisitos no funcionales definidos [Durán, A. & Bernárdez, B. – 2002]:

RNF-1	Servidor .NET
Autores	Nayarit Santana Pacheco
Fuentes	Expertos en TI
Actores asociados	---
Requisitos asociados	---
Descripción	El servidor web que ofrece los servicios que consumirá el cliente debe ser desarrollado en la plataforma .NET
Importancia	Alta
Estabilidad	Alta
Comentarios	---

RNF-2	SGBD SQL SERVER 2008
Autores	Nayarit Santana Pacheco
Fuentes	Expertos en TI
Actores asociados	---
Requisitos asociados	---
Descripción	El servidor web que ofrece los servicios que consumirá el cliente debe ser desarrollado en la plataforma .NET
Importancia	Alta
Estabilidad	Alta
Comentarios	---

RNF-3	Cliente RIA
Autores	Nayarit Santana Pacheco
Fuentes	Expertos en TI
Actores asociados	---
Requisitos asociados	---
Descripción	El cliente debe ser una Rich Internet Application
Importancia	Alta
Estabilidad	Alta
Comentarios	---

6.5. Modelo del dominio

Una vez expuestos los requisitos funcionales y no funcionales de la plataforma y los respectivos casos de uso, se ha realizado un diagrama del modelo del dominio. En este diagrama se exponen los conceptos más destacados que se obtienen como conclusión del análisis, así como las relaciones entre ellos.

El objetivo de este diagrama no es mostrar toda la funcionalidad, con él se pretende ayudar a visualizar los conceptos más destacados del problema y ver como se relacionan. Estas clases y relaciones, una vez llegados a la fase de diseño, pasarán a ser cada uno de ellos una clase. De igual forma, en la fase de diseño probablemente aparecerán nuevas clases y relaciones, además de modificar las planteadas en este modelo.

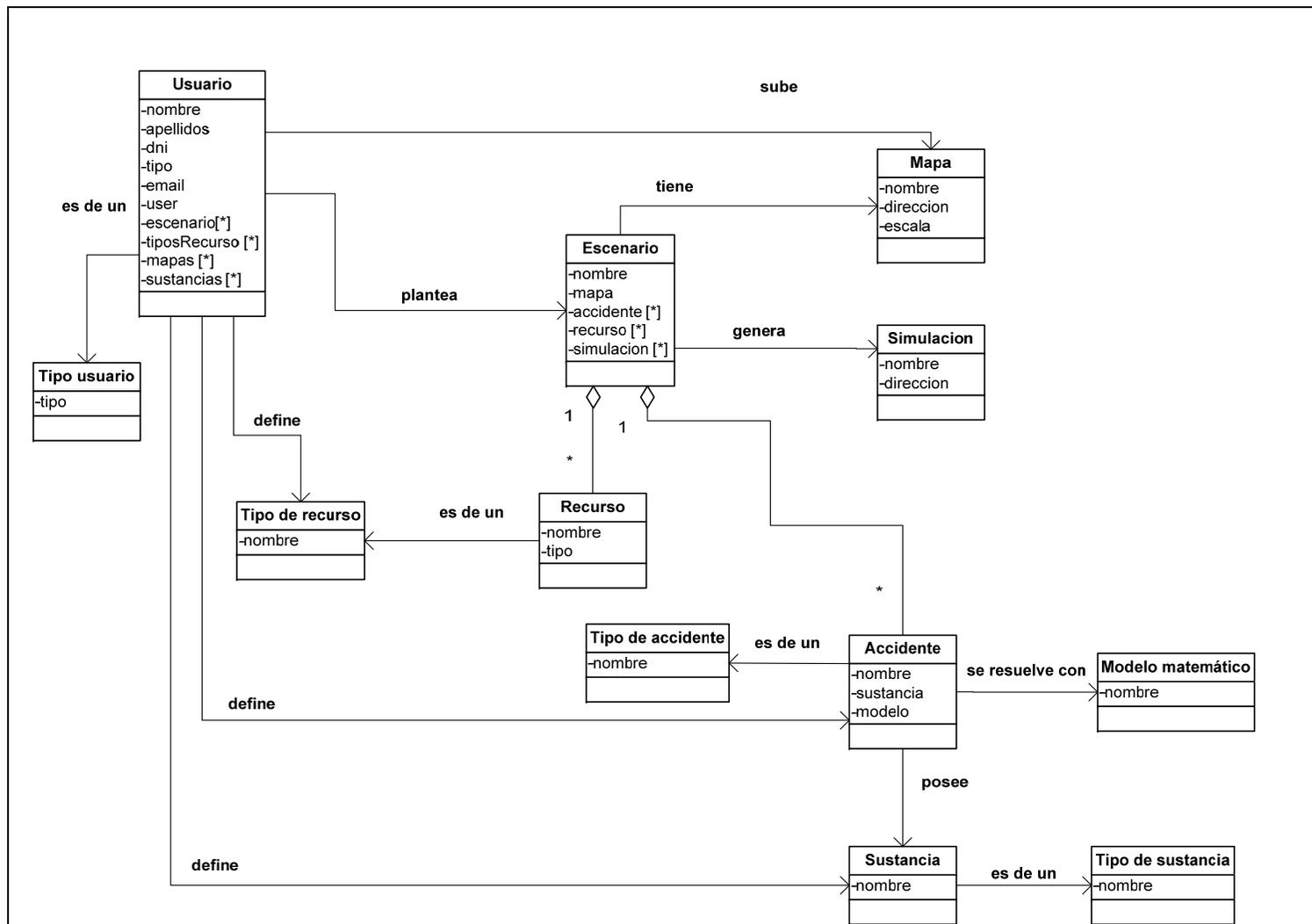


Fig. 19. Modelo del dominio

Capítulo 7. Diseño

Una vez realizado el análisis de la plataforma y de cada uno de sus componentes, se ha realizado el diseño de la misma. A continuación se encuentra una descripción de:

- Arquitectura utilizada
- Tecnologías elegidas
- Diseño de cada parte de la plataforma
 - o Patrones empleados
 - o Diagramas de clases

7.1. Arquitectura básica de la plataforma

El diseño arquitectónico representa la estructura de los datos y los componentes de la plataforma, sus propiedades visibles externamente, así como las relaciones entre ellos y todo asumiendo un estilo arquitectónico.

El estilo arquitectónico será el que establezca los componentes que formarán parte de la plataforma y cómo interactuarán entre ellos. En este caso, dada la naturaleza de la aplicación, se ha elegido una arquitectura cliente-servidor de 3 capas.

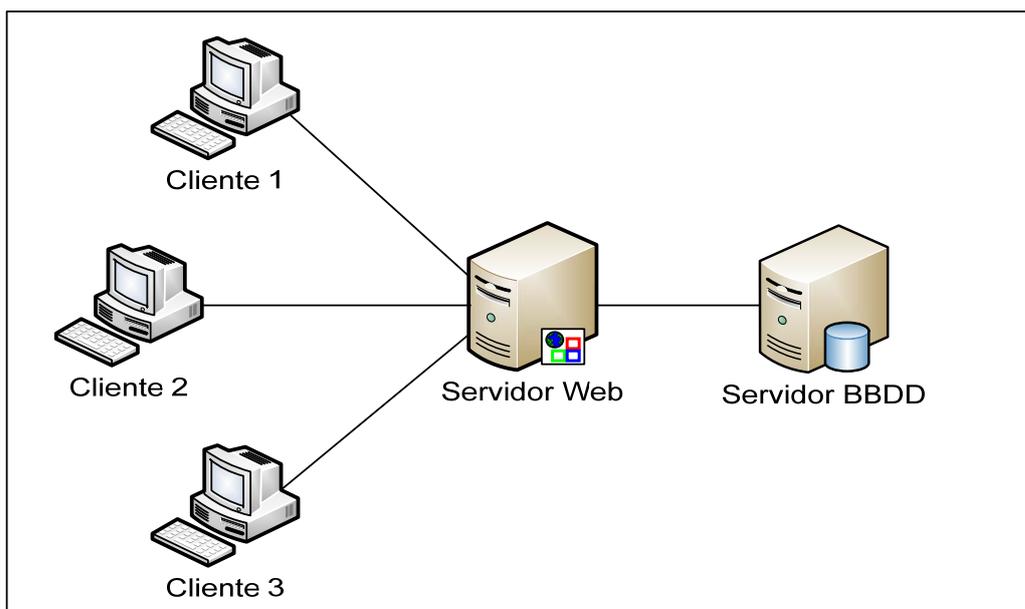


Fig. 20 Arquitectura cliente-servidor de 3 capas

La arquitectura cliente-servidor es una arquitectura distribuida donde los datos y el procesamiento se distribuyen entre varios procesadores. Los componentes principales son:

- Un conjunto de servidores independientes que proporcionan servicios a otros subsistemas
- Un conjunto de clientes que acceden a los servicios proporcionados por los servidores
- Una red que permite a los clientes comunicarse con los servidores y acceder a estos servicios

Para que la comunicación se pueda establecer correctamente, los clientes deberán conocer el *nombre* de los servidores disponibles, sin embargo, estos servidores no necesitarán conocer la identidad de los clientes.

En una arquitectura cliente-servidor sencilla, la funcionalidad del sistema recae principalmente en el servidor o por el contrario en el cliente, pero al dividir el sistema en 3 capas se consiguen distribuir las responsabilidades:

- Capa 1: clientes
- Capa 2: servidor web
- Capa 3: servidor de datos

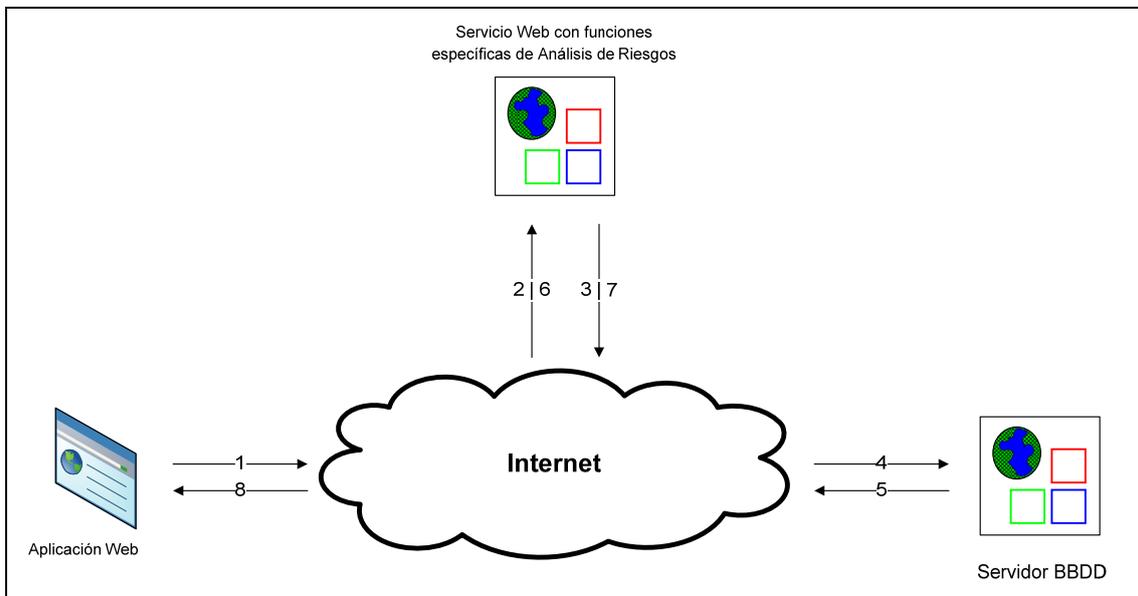


Fig. 21. Comunicación entre capas

A continuación se muestra como ejemplo, lo que podría ser una comunicación entre los distintos componentes de la arquitectura:

1. El cliente, en función de la petición del usuario, accederá a alguno de los servicios ofrecidos por el servicio web
2. Éste, a su vez, accederá al servidor de datos para obtener los datos requeridos.
3. Los datos serán devueltos al servidor tras una consulta.
4. El servidor hará el tratamiento necesario sobre esos datos, así como las comprobaciones pertinentes y devolverá la respuesta al cliente
5. El cliente mostrará la respuesta al usuario.

7.2. Tecnologías utilizadas

Dada la naturaleza del proyecto, se han utilizado diversas tecnologías para llevar a cabo el desarrollo de toda la plataforma. A continuación se describen cada una de ellas, indicando en que parte se utiliza junto con una descripción de la misma.

7.2.1. Servidor web

Basado en una Arquitectura Orientada a Servicios (SOA), se ha utilizado Windows Communication Foundation (WCF) para crear los servicios web. No se han estudiado otras tecnologías existentes dado que venía impuesta su utilización a través de los requisitos no funcionales descritos en el capítulo del análisis de la plataforma.

WCF, disponible desde la versión 3.0 de .NET, fue creada con el fin de permitir una programación rápida de sistemas distribuidos y el desarrollo de aplicaciones basadas en arquitecturas SOA, con una API simple. Se considera un framework para la creación de este tipo de aplicaciones. Permite enviar datos como mensajes asíncronos desde un extremo del servicio a otro. El servicio web puede estar hospedado en un servidor IIS o ser hospedado en una aplicación. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

7.2.1.1. Mensajería y extremos

WCF se basa en la noción de comunicación basada en mensajes, y cualquier cosa que se pueda modelar como un mensaje. Esto habilita una API unificada en todos los mecanismos de transporte diferentes. El modelo realiza una diferenciación entre clientes, que son las aplicaciones que inician la comunicación y servicios, que son las

aplicaciones que están a la espera de que los clientes se comuniquen con ellos y dan respuesta a esta comunicación. Una aplicación puede actuar como cliente, como servicio o como ambos a la vez.

Cada mensaje es enviado entre extremos, que son los lugares donde los mensajes se envían y/o reciben y, en ellos se define toda la información que se requiere para el intercambio de mensajes. Cada servicio puede exponer uno o más extremos y cada cliente debe generar un extremo que sea compatible con alguno de los extremos del servicio.

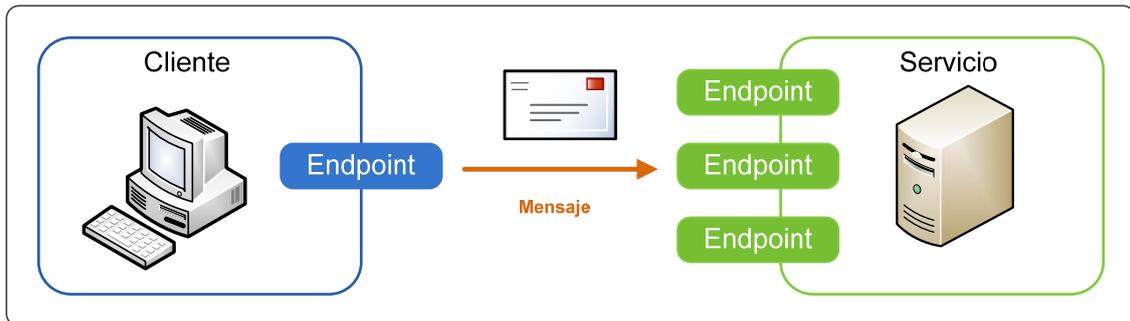


Fig. 22. Envío de mensajes entre Cliente y Servidor

Un *extremo* describe de una manera basada en estándar dónde se deberían enviar los mensajes, cómo se deberían enviar y qué aspecto deberían tener. Un servicio puede exponer esta información como metadatos que los clientes pueden procesar para generar clientes WCF adecuados y *pilas* de comunicación.

7.2.1.2. Protocolos de comunicación

Un elemento requerido de la pila de comunicación es el protocolo de transporte. Los mensajes se pueden enviar por intranets e internet utilizando protocolos de transporte comunes, como pueden ser: HTTP y TCP. También se admiten otros protocolos de transporte que permiten la comunicación con aplicaciones Message Que Server (MSMQ). Para agregar más mecanismos de transporte se pueden utilizar los puntos de extensión integrados en WCF.

Otra pieza necesaria en la pila de comunicación es la codificación que define cómo se da formato a los mensajes. WCF proporciona las siguientes codificaciones:

- Codificación de texto
- Codificación Mecanismo de optimización de transmisión de mensajes (MTOM)
- Codificación binaria para una transferencia eficaz

7.2.1.3. Patrones de mensajes

WCF admite varios patrones de mensajería, incluida la comunicación de solicitud-respuesta unidireccional y dúplex.

7.2.1.4. Endpoint

Cada endpoint se define como un portal para la comunicación con todo el mundo. Es un recurso en la red al que se pueden enviar mensajes. Está compuesto por 3 partes comúnmente denominada como el “ABC” de un endpoint:

- Address (Dirección)
- Binding (Canal)
- Contract (Contrato)

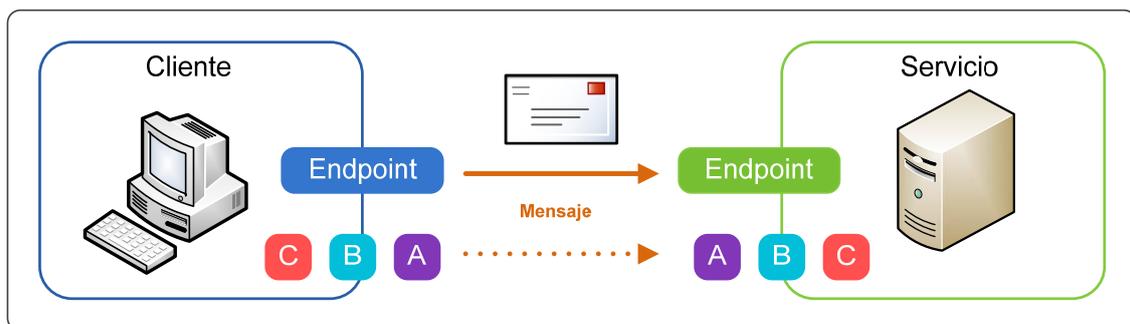


Fig. 23. Endpoint

El **ABC** de un Endpoint, tal y como se ha descrito anteriormente, se define por:

- **Address:** dirección de la red donde está alojado el Endpoint del servicio
- **Binding:** define el canal y los parámetros de comunicación entre endpoints. Consta de:
 - o Protocolos: TCP, HTTP, MSMQ
 - o Codificación: Texto o binario
 - o Seguridad: SSL, Seguridad SOAP
 - o Referencia al servicio a exponer

Existen los siguientes tipos de bindings:

- o BasicHttpBinding
- o WSHttpBinding
- o WSDualHttpBinding
- o WSFederationHttpBinding
- o NetTcpBinding

- NetNamePipeBinding
 - NetMsmqBinding
 - NetPeerTcpBinding
- **Contract:** es el contrato del servicio o interfaz. Define las capacidades ofrecidas por el Endpoint. Existen varios tipos:
- Contrato de servicios (ServiceContract): une varias operaciones relacionadas en una unidad funcional única, además, define ajustes de servicio.
 - Contrato de operación (OperationContract): define los parámetros y el tipo de valor devuelto de una operación. Cuando se crea una interfaz que define el contrato de servicio se está creando un contrato de operación
 - Contrato de mensaje (MessageContract): describe el formato del mensaje. Se puede definir la cabecera y cuerpo del mensaje, nivel de seguridad a aplicar y en qué elementos, etc.
 - Contrato de datos (DataContract): define los tipos de datos complejos que usa un servicio. Los tipos de datos se pueden usar en cualquier parte del mensaje, tanto como parámetros como en los tipos de valores devueltos. Si el servicio sólo utiliza tipos simples en los mensajes, no es necesario hacer uso de este tipo de contratos
 - Contrato de error: puede estar asociado a una operación de servicio para indicar errores que se pueden devolver al cliente. Son errores SOAP que se configuran como excepciones en el modelo de programación.

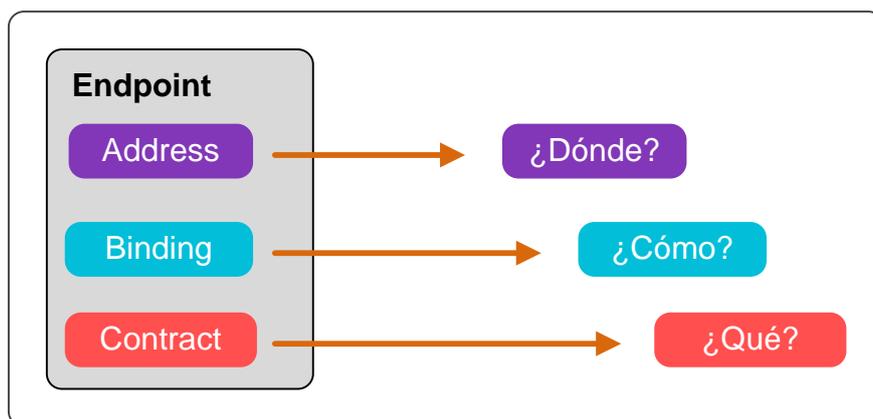


Fig. 24. El ABC del Endpoint

7.2.2. Cliente

El cliente, tal como se especifica en los requisitos no funcionales comentados durante el análisis, debe ser desarrollado en una tecnología que permita la creación de aplicaciones RIA. Dentro de las posibilidades existentes se han barajado: Silverlight y Adobe Flex.



Fig. 25. Adobe Flex y Microsoft Silverlight

Silverlight es una tecnología perteneciente a Microsoft, frente a **Flex**, tecnología que al inicio de este proyecto pertenecía a Adobe, pero que posteriormente ha sido absorbida por Apache.

7.2.2.1. Silverlight



Microsoft Silverlight es una estructura para aplicaciones web que surge como la competencia de Flash. Se trata de una versión reducida del framework de .NET WPF (Windows Presentation Foundation) que está enfocada al tratamiento de elementos multimedia dentro del navegador. Agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad.

Su principal ventaja frente a Flash reside en ser un subconjunto de XAML (eXtensible Application Markup Language) especialmente diseñado para la construcción de aplicaciones multimedia sobre la web. Está basado en XML, lo que le permite que la indexación en los buscadores sea más eficiente.

Pensado para ser multiplataforma, es necesaria la instalación de un plugin en cada navegador. Para los sistemas operativos basados en UNIX, Microsoft lanzó en conjunto con Novell el proyecto de código abierto Moonlight, pero actualmente este proyecto ha sido abandonado por la falta de proyección de Silverlight.

7.2.2.2. Adobe Flex



Flex es un framework de presentación de Adobe basado en la tecnología Flash, utilizado para construir aplicaciones RIA (Rich Internet Applications). Permite lograr por medio de desarrollo orientado a objetos

y/o declarativo los mismos resultados con la calidad visual proporcionada por Flash y la interacción propia de una aplicación de escritorio.

Su objetivo principal es proporcionar a los desarrolladores una forma rápida y sencilla de construir aplicaciones RIA. Siendo Flash un producto orientado principalmente a diseñadores, Flex se presenta como la solución indicada para programadores, que permite la construcción de la capa de presentación en aplicaciones de internet basadas en arquitecturas MVC (Model View Controller).

Al igual que Silverlight, Flex es multiplataforma y necesita la instalación de un plugin en el navegador para su funcionamiento. Según algunos estudios, el 98% de los usuarios tienen instalado ese plugin, teniendo además el 90% el plugin actualizado. Este es el principal motivo por el cual se ha decidido utilizar Flex para el desarrollo del cliente. Flash es una herramienta completamente extendida, con lo cual va a llegar a más usuarios.

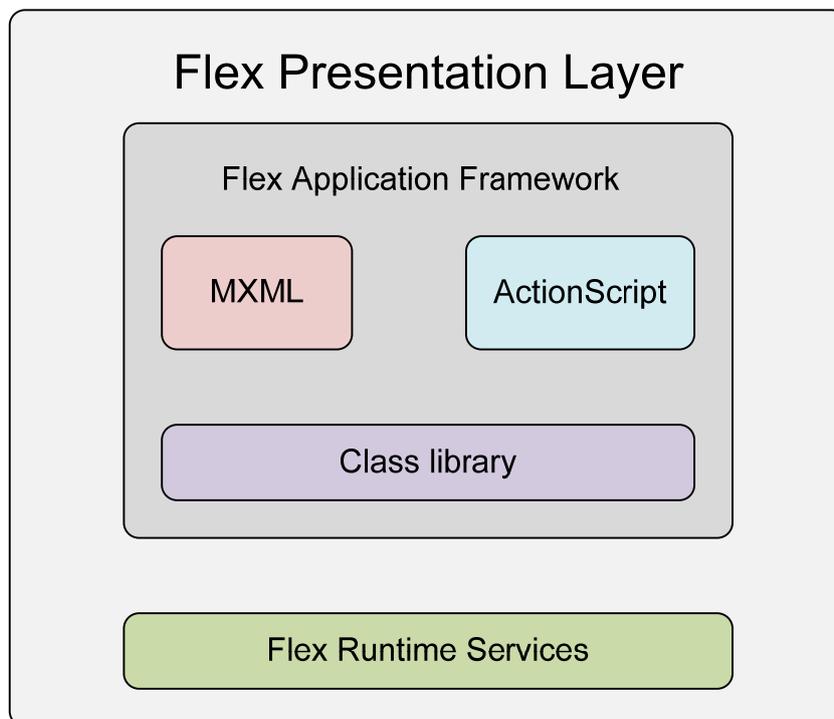


Fig. 26

Existen 2 formas de programar en Flex:

- Código mxml
- ActionScript 3

MXML es un lenguaje declarativo basado en XML que significa: Macromedia eXtensible Markup Language. Su estructura se basa en etiquetas que se utilizan de forma similar a

como se hace en HTML. Proporciona una gran variedad de etiquetas además, permite extenderlas y crear nuevos componentes por el desarrollador.

ActionScript 3 (AS3) es el lenguaje que siempre se ha utilizado para el desarrollo de aplicaciones en Flash. Se compila en bytecode y es interpretado por el Flash Player (al igual que ocurre con la Java Virtual Machine que interpreta el Java bytecode). AS3 dispone de características propias de un lenguaje orientado a objetos. Tiene: manejo de excepciones, procesamiento de eventos, clases, interfaces, etc.

A pesar de nombrarse como dos formas de programar en Flex, también se pueden utilizar de forma conjunta. Dentro de los archivos .mxml, a través de las etiquetas correspondientes, se puede utilizar código ActionScript, de esa forma se puede interactuar con los objetos que han sido declarados con mxml.

El framework de flex es gratuito, así que cualquier desarrollador tiene acceso al sdk para crear sus aplicaciones. No ocurre lo mismo con el IDE que proporciona Adobe, conocido como Flex Builder. Este entorno de desarrollo está basado en Eclipse y provee una interfaz WYSIWYG (What You See Is What You Get) para la creación de la interfaz, y permite arrastrar los componentes que se van a utilizar desde una paleta de componentes. Adobe concede 60 días de prueba para Adobe Flex Builder, aunque también proporciona licencias de estudiantes.

7.2.3. Servidor BBDD

Al igual que en el caso del servidor web, para la elección de la BBDD no se ha hecho ningún estudio de tecnologías, dado que viene marcada por un requisito no funcional de la plataforma.

Microsoft SQL Server es un sistema gestor de BBDD de Microsoft que está basado en el modelo relacional. Los lenguajes de consulta utilizados son:

- Transact – SQL (T-SQL)
- ANSI SQL

Este sistema constituye la alternativa por parte de Microsoft a otros SGBD también muy utilizados, como son: Oracle, PostgreSQL o MySQL.

7.3. Patrones y otros aspectos de diseño

Los patrones de diseño proporcionan soluciones a problemas que se encuentran frecuentemente en el desarrollo del software y otros ámbitos referentes al diseño de interacción o interfaces. Estos patrones, que ofrecen soluciones ya probadas y documentadas a problemas comunes en el ámbito del desarrollo del software, están

definidos por: su nombre, el problema que resuelven, la solución que proporcionan y las consecuencias de su uso/aplicación.

Los patrones usualmente se dividen en 3 tipos, clasificándolos en función del tipo de problema que resuelven: creacionales, estructurales y de comportamiento. Con el paso del tiempo y su uso extensivo, han ido surgiendo nuevos patrones específicos de una determinada tecnología, como puede ser J2EE, pero que también pueden llegar a ser aplicados en otras tecnologías.

A continuación se detallan los patrones utilizados en cada uno de los componentes de la plataforma. Para ello se ofrecerá una breve descripción del patrón utilizado y un diagrama de clases explicativo.

7.3.1. Servidor web

En el servidor web se han utilizado los siguientes patrones:

7.3.1.1. Data Access Object (DAO)

En el diseño del servidor web se ha utilizado el patrón Data Access Object (DAO). El DAO, surgido dentro del ámbito del desarrollo J2EE (Java 2 Platform, Enterprise Edition), proporciona un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos.

A nivel lógico, el siguiente diagrama representa la forma en que se ha aplicado el patrón sobre el servidor.

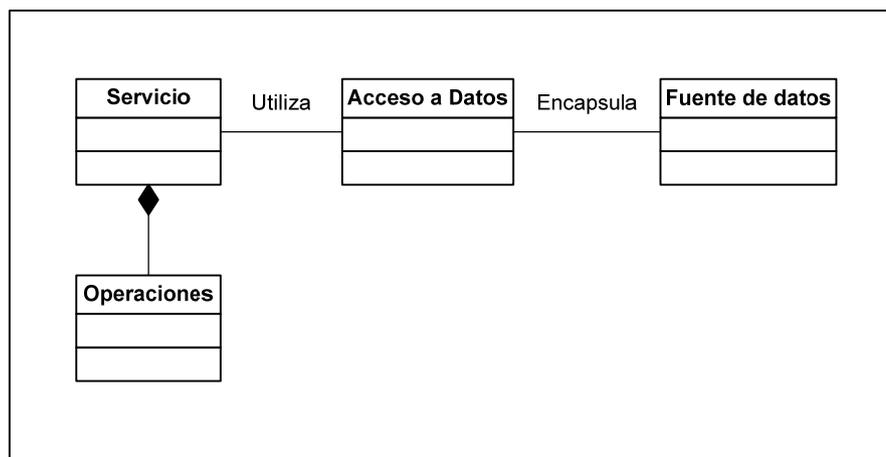


Fig. 27. Aplicación del DAO sobre el servidor

El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos, de esta forma, el DAO consigue aislar la aplicación de la tecnología utilizada para el almacenamiento de los datos.

7.3.1.2. Data Transfer Object (DTO)

Para el paso de datos entre el servidor y la aplicación web también se ha utilizado un patrón perteneciente al ámbito de desarrollo J2EE, se trata del patrón Data Transfer Object (DTO).

El DTO se aplica para conseguir que un objeto sea serializable para la transferencia de datos sobre la red. Para satisfacer este punto, se ha creado una clase por cada tipo de dato complejo que se utiliza tanto en la entrada, como en la salida de datos de las operaciones proporcionadas por el servicio. De esta forma se consigue transferir sólo los datos necesarios. En la siguiente figura se puede ver un ejemplo del mismo.

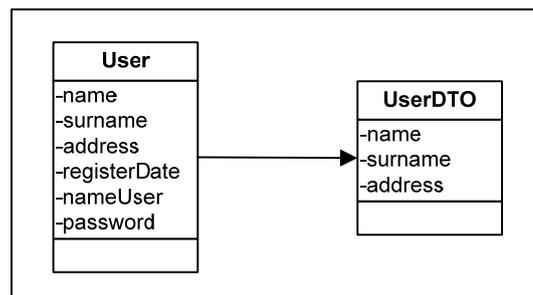


Fig. 28. Ejemplo del uso del patrón DTO

7.3.2. Cliente

En el cliente se han utilizado los siguientes patrones:

7.3.2.1. Data Transfer Object (DTO)

Como se explicó anteriormente en la sección de patrones utilizados en el servidor, el patrón Data Transfer Object se utiliza para el paso de datos entre el servidor y la aplicación web, es decir, el cliente. Existe una serie de datos que no es necesario almacenar en la BBDD, porque son para el uso interno del cliente, con lo cual carece de sentido enviarlos al servidor, por este motivo se utilizan los DTO en esta parte.

7.3.2.2. Model View Controller (MVC)

El Model View Controller (MVC) es un patrón arquitectónico que separa los datos y la lógica de la interfaz de usuario, dentro de la aplicación. El Modelo es el objeto de aplicación, donde estarían almacenados los datos, la Vista es la representación del modelo en pantalla y el Controlador define la forma en la que la interfaz reacciona a la entrada del usuario.

Al desacoplar el modelo de la vista, pueden realizarse diferentes representaciones del mismo modelo. Estas vistas se actualizarán siempre que haya un cambio en el modelo. Esta separación evitará tener que realizar cambios en cada vista cada vez que se

produzca un cambio en el modelo. Para gestionar estas actualizaciones está el controlador, que será el encargado de comunicarse tanto con la vista como con el modelo. En algunos casos la vista se actualiza cuando se produce un cambio en el modelo, sin que llegue a intervenir el controlador.

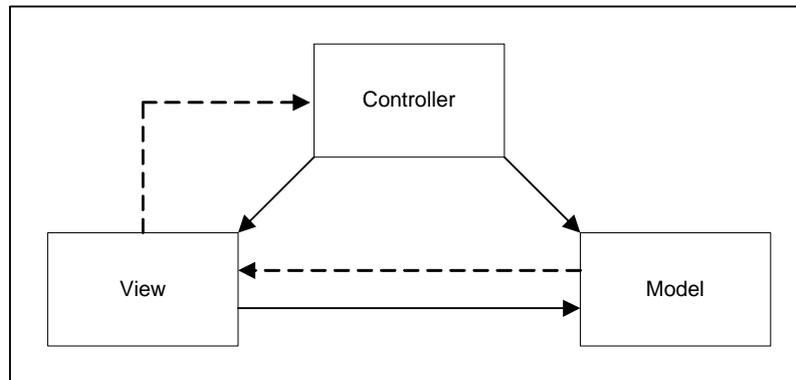


Fig. 29. Patrón Modelo Vista Controlador

7.3.2.3. Singleton

El patrón Singleton se utiliza para garantizar que la clase sobre la que se aplique sólo tendrá una instancia, lo que proporcionará un punto de acceso global a ella. Al utilizar el patrón Modelo Vista Controlador, el uso del patrón Singleton se hace obligatorio, dado que es necesario asegurar la existencia de una única instancia para el modelo.

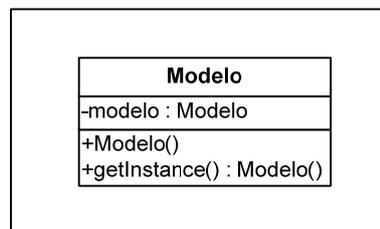


Fig. 30. Patrón Singleton aplicado a la clase Modelo

7.3.2.4. Capa de servicios

No se trata de un patrón de diseño, sin embargo, se ha considerado necesario hacer una separación lógica dentro del cliente, aislando en una clase la comunicación con el servidor. De esta forma, cada vez que alguno de los controladores del cliente tiene que interactuar de alguna forma con el servidor web, necesita pasar por la capa de servicios y entonces será ésta la encargada de la comunicación en ambos sentidos. Cuando la capa de servicios recibe la respuesta del servidor, lanza un aviso al controlador que ha hecho la llamada.

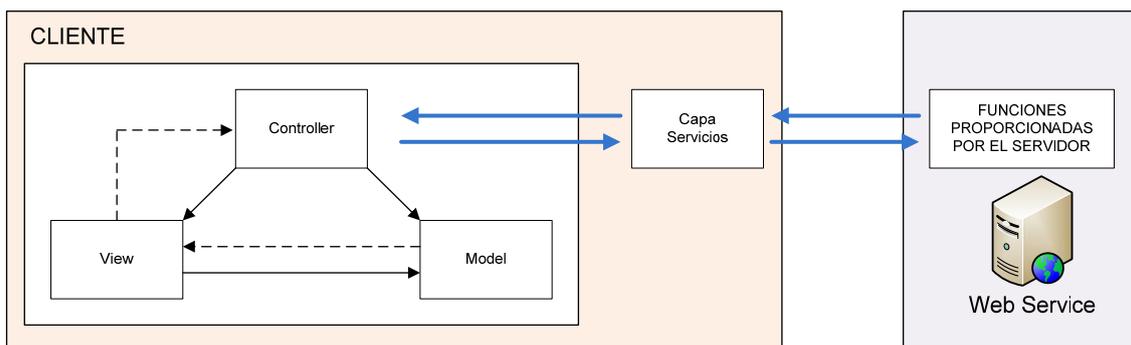


Fig. 31. Capa de Servicios

7.4. Base de datos

Tras realizar un estudio exhaustivo sobre los riesgos en entornos industriales, junto con la ayuda y conocimiento proporcionado por el experto en la materia, y posterior al análisis de la plataforma, se ha obtenido el siguiente diseño para la base de datos. Aquí están almacenados, en primer lugar, los datos proporcionados desde un inicio por la plataforma para todos los usuarios y, a continuación, los datos que vaya almacenando cada usuario con el uso de la misma (siguiente página).

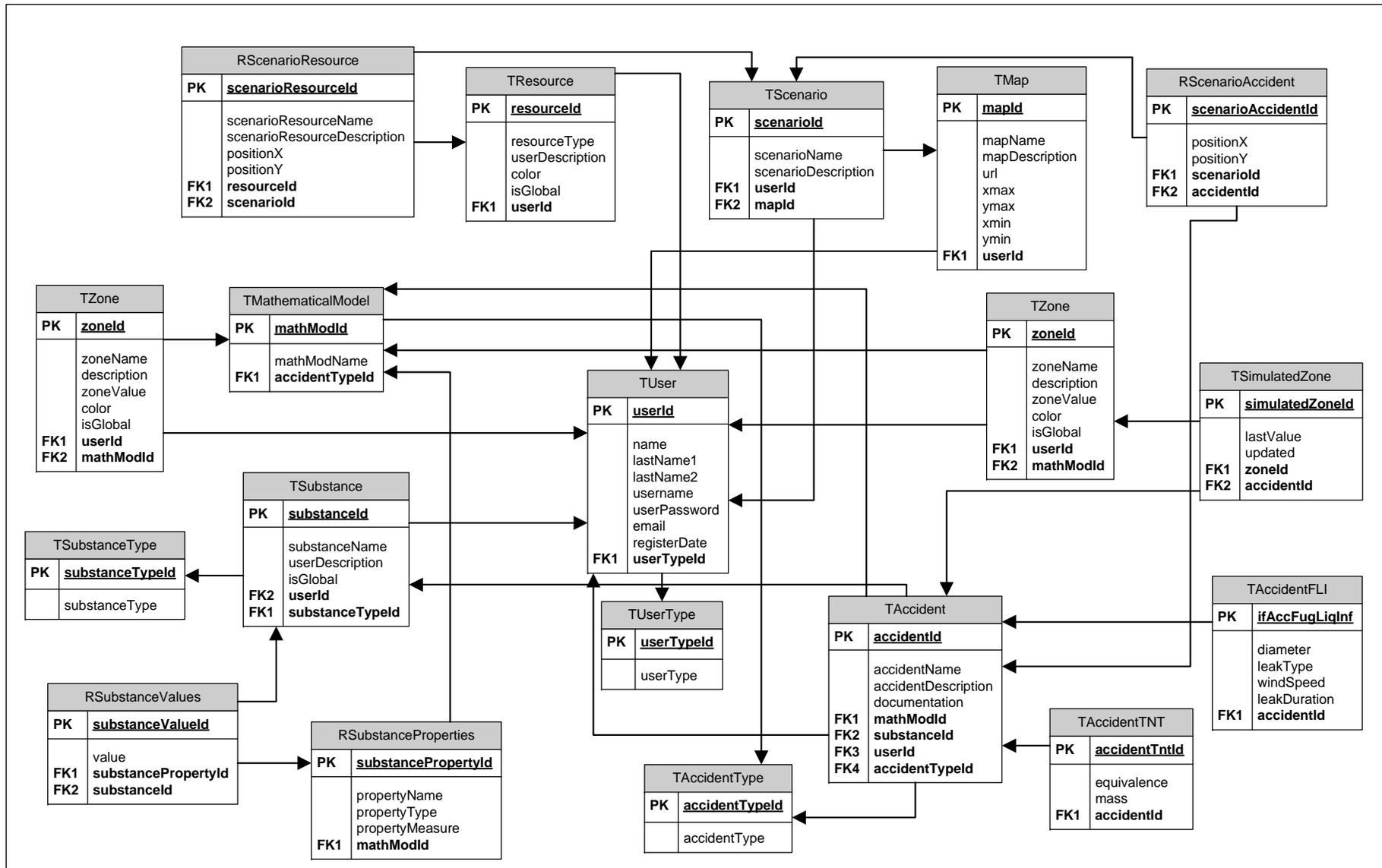
A continuación se detalla el contenido de cada tabla y cada uno de sus campos, indicando en cada caso el tipo de dato que representa y para qué resulta necesario.

7.4.1. Tabla TUserType

Almacena los distintos tipos de usuario existentes. En caso de crearse nuevos tipos de usuarios bastaría con actualizar la tabla.

Los campos que componen la tabla son:

Nombre	Tipo	NULL	FK	Descripción
userId	int (auto)		---	Identificador de la tabla
userType	varchar (20)		---	nombre del tipo de usuario



7.4.2. Tabla TUser

Almacena todos los datos referentes al usuario de la aplicación, tanto datos personales como datos del registro. Cada vez que un usuario se da de alta en la aplicación se crea un nuevo registro en la tabla.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
userId	int (auto)		---	Identificador de la tabla
name	varchar (20)	x	---	Nombre real del usuario
lastName1	varchar (20)	x	---	Primer apellido del usuario
lastName2	varchar (20)	x	---	Segundo apellido del usuario
userName	varchar (10)		---	Nombre por el que el usuario será identificado dentro de la aplicación
userPassword	varchar (32)		---	Contraseña del usuario que será necesaria para loguearse en la aplicación y así hacer uso de todas las funcionalidades de la misma. La contraseña está almacenada codificada con MD5.
email	varchar (100)		---	Dirección de correo (email) del usuario
registerDate	date		---	Fecha en la que el usuario realizó el registro en la aplicación
userId	int		TUserType	Tipo de usuario al que pertenece

7.4.3. Tabla TSubstanceType

Almacena los distintos tipos de sustancias existentes.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
substanceTypeId	int (auto)		---	Identificador de la tabla
substanceType	varchar (20)		---	nombre característico que identifica al tipo de sustancia

7.4.4. Tabla TSubstance

Almacena el conjunto de sustancias proporcionadas por la aplicación, así como las sustancias creadas por el usuario. Se clasifican en función de:

- El tipo de sustancia al que pertenecen
- El usuario que la ha creado

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
substancelid	int (auto)		---	Identificador de la tabla
substanceName	varchar (20)		---	Nombre propio de la sustancia por el que puede ser identificada
isGlobal	bit		---	Indicativo de si es una sustancia creada por un usuario concreto o es global a todos los usuarios
userDescription	text	x		descripción de la sustancia
substanceTypeld	int		TSubstanceType	Tipo de sustancia al que pertenece
userId	int		TUser	Usuario propietario de la sustancia
mathModld	int		TMathematicalModel	Modelo matemático para el que se ha almacenado la sustancia. Solo se podrá utilizar la sustancia para este modelo matemático

7.4.5. Tabla TAccidentType

Almacena cada uno de los tipos de accidente que se pueden llegar a dar.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
accidentTypeld	int (auto)	---	---	Identificador de la tabla
accidentType	varchar (20)	---	---	Nombre que identifica al tipo de accidente

7.4.6. Tabla TAccident

Almacena todos los accidentes que han sido generados por la aplicación, independientemente de quién haya sido el creador de los mismos. Estarán clasificados por el modelo matemático al que pertenecen. Dado que se conoce el modelo matemático y el modelo matemático sólo pertenecerá a un tipo de accidente, no se necesitará hacer referencia al tipo de accidente al que pertenece el accidente.

Se hace una referencia a la tabla donde se almacenan los modelos para saber a qué modelo pertenece el accidente y en otra tabla poder establecer las propiedades que se tendrán que rellenar.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
accidentId	int (auto)	---	---	Identificador de la tabla
accidentName	varchar (20)	---	---	Nombre por el que se identifica al accidente
accidentDescription	text	x	---	Descripción asociada al accidente
documentation	text	x	---	Campo utilizado para documentar el accidente de forma técnica
mathModId	int	---	TMathematicalModel	Identificador del modelo matemático al que pertenece este accidente
substancelId	int	---	TSubstance	Identificador de la sustancia que se está utilizando en el accidente
userId	int	---	TUser	Referencia al usuario creador del accidente

7.4.7. Tabla TMap

Almacena todos los datos necesarios para poder hacer uso del mapa en la aplicación.

Se hace referencia a:

- TUser: para saber a qué usuario pertenece el mapa

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
mapId	int (auto)	---	---	Identificador de la tabla
mapName	varchar (20)	---	---	Nombre por el que se identifica al mapa
mapDescription	text	x	---	Descripción asociada al mapa
url	varchar (max)	---	---	Url donde está guardado el mapa
xmax	float	---	---	Coordenada x máxima del mapa
ymax	float	---	---	Coordenada y máxima del mapa
xmin	float	---	---	Coordenada x mínima del mapa
ymin	float	---	---	Coordenada y mínima del mapa
userId	int	---	TUser	Usuario al que pertenece el mapa

7.4.8. Tabla TResource

Almacena los datos necesarios para identificar un tipo de recurso. Los recursos son conceptos que están dentro del ámbito de los riesgos, siendo por ejemplo: extintores, mangueras, etc.

Se hace referencia a la tabla *TUser*, para saber a qué usuario pertenece el tipo de recurso.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
resourceId	int (auto)	---	---	Identificador de la tabla
resourceType	varchar (20)	---	---	Nombre con el que se identifica este tipo de recurso
isGlobal	bit	---	---	Indicativo de si es un recurso creado por un usuario concreto o es global a todos los usuarios
userDescription	text	x	---	Descripción asociada al tipo de recurso
color	int	---	---	Color asociado al tipo de recurso, todos los recursos de este tipo se pintarán en el mapa con este color
userId	int	---	TUser	Usuario al que pertenece este tipo de recurso

7.4.9. Tabla TMathematicalModel

Almacena el conjunto de modelos matemáticos ofrecidos por la aplicación para realizar las simulaciones.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
mathModId	int (auto)	---	---	Identificador de la tabla
mathModName	varchar (20)	---	---	Nombre que identifica al modelo matemático
accidentTypeId	int	---	TAccidentType	Tipo de accidente para el que se puede aplicar el modelo

7.4.10. Tabla RSubstanceProperties

Almacena el nombre de las propiedades que se necesitan conocer acerca de una sustancia para poder utilizarla dentro de un modelo matemático. Se almacena de esta

forma y no directamente como columnas de la tabla, para no tener que rellenar propiedades sobre una sustancia que posteriormente no serán de utilidad, de esta forma sólo se tendrán aquellas propiedades que realmente se utilicen, y se irán añadiendo más, a medida que también se vayan añadiendo nuevos modelos matemáticos.

Se hace referencia a la tabla donde se almacenan los modelos matemáticos para saber qué propiedades se necesitan conocer sobre una sustancia en función del modelo en el que vayan a ser utilizadas.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
substancePropertyId	int (auto)	---	---	Identificador de la tabla
propertyName	varchar (10)	---	---	Nombre que identifica a la propiedad
propertyType	varchar (10)	---	---	Tipo de la variable que almacenará el valor de esta propiedad
propertyMeasure	varchar (10)	---	---	Unidad en la que se mide esta propiedad
mathModId	int	---	TMathematicalModel	Identificador del modelo matemático donde se va a hacer uso de esta propiedad

7.4.11. Tabla RSubstanceValues

Almacena los valores de cada una de las propiedades de una sustancia específica. Como se vio en la tabla anterior (*RSubstanceProperties*), el modelo matemático es el que establece las propiedades de una sustancia que serán necesarias para poder ejecutarlo y, por otro lado, se tienen las sustancias ofrecidas por la aplicación, así que la tabla *RSubstanceValues* recoge los valores propios de la sustancia para cada una de esas propiedades. El valor se guardará en un campo de texto porque no se puede conseguir que cada registro de esta tabla tenga un tipo distinto en cada columna, así que ayudados por la tabla *RSubstanceProperties*, que almacena el tipo de dato de esta propiedad, a la hora de programar se podrá hacer una conversión desde texto hasta el tipo específico.

Se hace referencia a:

- TSubstance: almacena las sustancias, para saber de qué sustancia se trata
- RSubstanceProperties: almacena las propiedades de los modelos matemáticos, para saber cuántos y qué valores hay que almacenar en la tabla actual

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
substanceValueId	int	---		Identificador de la tabla
value	varchar (100)	---		Valor de la propiedad
substancePropertyId	int	---	RSubstanceProperties	Propiedad de la sustancia para la que se está almacenando el valor
substancelId	int	---	TSubstance	Sustancia para la que se está almacenando el valor de la propiedad

7.4.12. Tabla RScenarioResource

Almacena la relación existente entre los recursos creados por un usuario en un determinado escenario. Se hace referencia a:

- TResource: para saber el tipo de recurso que se está creando
- TScenario: para saber en qué escenario se ha creado el recurso

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
scenarioResourceId	int (auto)	---	---	Identificador de la tabla
positionX	float	X	---	Coordenada X donde está situado el recurso en el mapa
positionY	float	X	---	Coordenada Y donde está situado el recurso en el mapa
scenarioResourceName	nvarchar (50)	---	---	Nombre por el que se identifica al recurso que se ha creado en el escenario
scenarioResourceDescription	text	X	---	Descripción asociada al recurso creado en el escenario
resourceId	int	---	TResource	Referencia al tipo de recurso que se está

				creando
scenariold	int	---	TScenario	Referencia al escenario donde se está creando el recurso

7.4.13. Tabla TAccidentTNT

Almacena los datos propios del modelo TNT. Estos datos están ligados al accidente guardado en la tabla TAccident, donde sólo se ha guardado la información genérica a cualquier tipo de accidente.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
accidentTntId	int (auto)	---	---	Identificador de la tabla
equivalence	varchar (20)	---	---	Tipo de equivalencia: sobrepresión/impulso
mass	decimal(18,2)	---	---	Cantidad de masa del explosivo que está en el accidente
accidentId	int	---	TAccident	Referencia al información general del accidente

7.4.14. Tabla TAccidentFLI

Almacena los datos propios del modelo FLI. Estos datos están ligados al accidente guardado en la tabla TAccident, donde sólo se ha guardado la información genérica a cualquier tipo de accidente.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
idAccFugLiqInf	int (auto)	---	---	Identificador de la tabla
diameter	float	---	---	Diámetro de la fuga
leakType	int	---	---	Tipo de fuga: nocturna/diurna
windSpeed	float	---	---	Velocidad del viento que sopla en el momento de la fuga
leakDuration	float	---	---	Duración de la fuga
accidentId	int	---	TAccident	Referencia al información general del accidente

7.4.15. Tabla TZone

Almacena las zonas que se pueden calcular para un determinado modelo matemático. Estas zonas definen un valor de presión que se establecería en caso de producirse un accidente. Una vez ocurrido el accidente se calcula el área que se vería afectada por esa presión, denominándose entonces zona x, siendo x el nombre de la zona.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
zoneld	int (auto)	---	---	Identificador de la tabla
zoneName	varchar (20)	---	---	Nombre por el que se identifica a una determinada zona
description	varchar (200)	x	---	Descripción asociada a la zona
isGlobal	bit	---	---	Indicativo de si es una zona creada por un usuario concreto o es global a todos los usuarios
zoneValue	float	---	---	Valor de presión asignado a una determinada zona
color	int	---	---	Color con el que se representarán en el visor todas las áreas afectadas por el accidente con un valor de presión como el que marca esta zona
userId	int	---	TUser	Identificador del usuario que ha creado la zona
mathModId	int	---	TMathematicalModel	Modelo matemático para el que se utiliza la zona

7.4.16. Tabla TSimulatedZone

Almacena las áreas afectadas por las zonas, simuladas para un determinado modelo matemático.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
simulatedZoneld	int (auto)	---	---	Identificador de la tabla
lastValue	float	---	---	Último valor de zona calculado
updated	bit	---	---	Indicativo de si ha habido algún cambio sobre el accidente que afecte al resultado lastValue

zoneld	int	---	TZone	Rererencia a la zona que se está calculando
accidentId	int	---	TAccident	Referencia al accidente sobre el que se hacen los cálculos de las zonas

7.4.17. Tabla TScenario

Almacena los datos básicos del escenario así como el mapa asociado al mismo, en caso de existir alguno.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
scenarioid	int (auto)	---	---	Identificador de la tabla
scenarioName	varchar (50)	---	---	Nombre por el que se identifica el escenario
scenarioDescription	text	x	---	Descripción del escenario
userId	int	---	TUser	Rerefencia al usuario creador del escenario
mapId	int	x	TMap	Referencia al mapa que está asociado al escenario. Como hay escenarios sin mapas puede ser nulo

7.4.18. Tabla RScenarioAccident

Almacena la información referente al posicionamiento del accidente dentro del mapa asociado al escenario.

La tabla está compuesta por los siguientes campos:

Nombre	Tipo	NULL	FK	Descripción
scenarioAccidentId	int (auto)	---	---	Identificador de la tabla
positionX	float	x	---	Coordenada x del accidente dentro del mapa asociado al escenario. Al no ser obligatorio tener un mapa asociado a cada escenario, este campo puede ser nulo
positionY	float	x	---	Coordenada y del accidente dentro del mapa asociado al escenario. Al no ser obligatorio tener un mapa asociado a cada escenario, este campo puede ser nulo
scenarioid	int	---	TScenario	Referencia al escenario en el que está asociado este accidente

Diseño

accidentId	int	---	TAccident	Referencia al accidente sobre el que se está definiendo una posición en el mapa
------------	-----	-----	-----------	---

Capítulo 8. Implementación

En esta fase del proyecto se ha de desarrollar todo lo expuesto en las anteriores fases. En el análisis se captan los requisitos del problema, posteriormente se diseña una solución y a continuación pasa a implementarse.

Añadir el código implementado en esta sección no aportaría nada de valor a esta memoria, por este motivo a continuación se mostrarán sólo algunos aspectos relevantes del desarrollo.

8.1. Cliente

Dentro de la aplicación cliente se considera destacable el uso del visor, componente desarrollado fuera de este proyecto, así como la forma en que se establece la conexión con el web Service y el manejo de la respuesta ofrecida por el mismo.

8.1.1. Visor

Como se explicó anteriormente, el componente visual de la plataforma ha sido desarrollado en otro proyecto (Desarrollo de un prototipo de componente web para la visualización e interacción con mapas de forma online, desarrollado por Samuel Díaz Cabrera). El componente se ha integrado en la interfaz de la plataforma, en la sección principal de la misma. Para poder hacer uso de este visor se ha utilizado la api que proporciona.

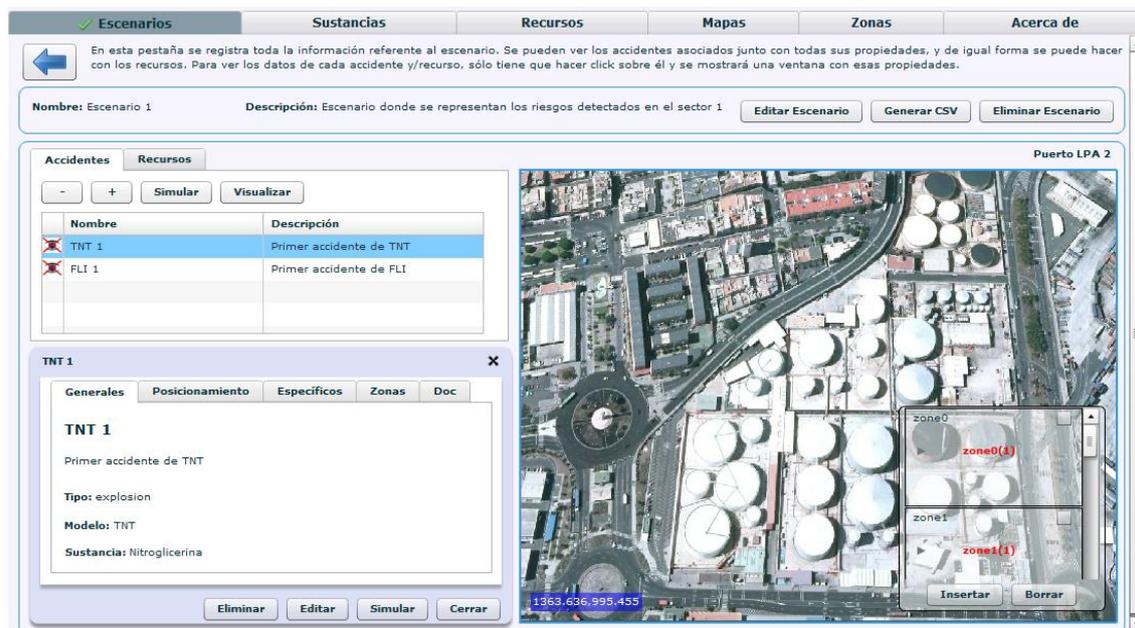


Fig. 32. Interfaz donde se hace uso del visor

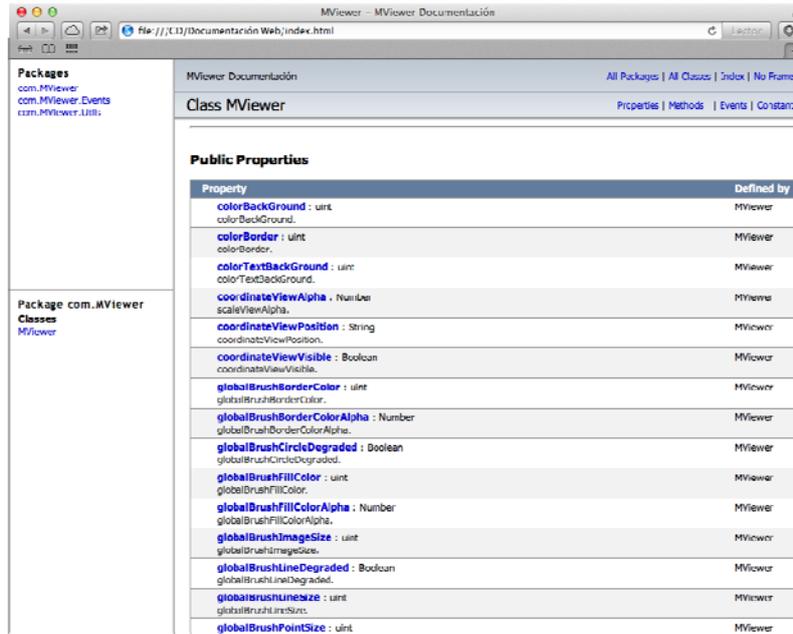


Fig. 33. Documentación de la API del visor

El visor ofrece la siguiente funcionalidad:

- Mostrar puntos/cuadrados, que en este caso serán utilizados para mostrar:
 - o Accidentes
 - o Recursos
- Mostrar áreas en forma de círculo y rellenarlas, en este caso se ha utilizado para mostrar las áreas afectadas en el supuesto caso de que sucediese el accidente declarado. Al permitir rellenar la zona, se utiliza un color para poder diferenciarlas
- Las formas que se hayan declarado en el visor podrán ser mostradas u ocultadas

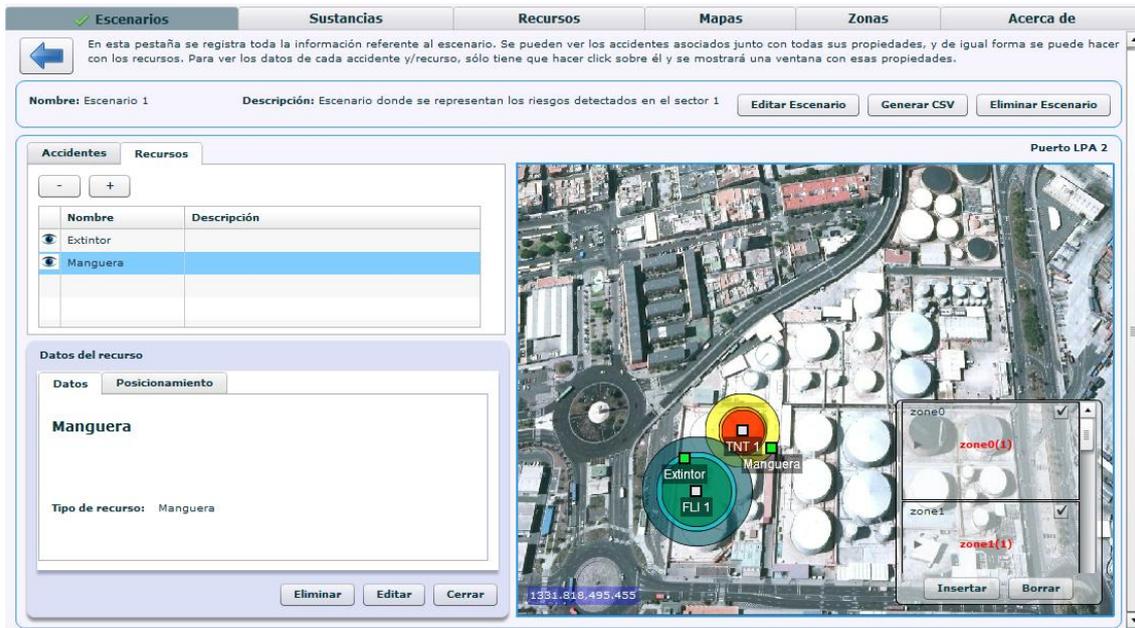


Fig. 34. Plataforma utilizando la funcionalidad del visor

Tanto en la figura 32 como en la figura 34 se puede observar un cuadro en la sección inferior derecha, dentro del visor. Este cuadro no existe en la versión final de la plataforma, pero se trata de un depurador ofrecido por la api del visor. Con esta opción se puede revisar en tiempo de ejecución si la parte gráfica se está creando y/o eliminando de forma correcta.

8.1.2. Acceso al servidor

Como se comentó en la fase de diseño, para acceder al servidor se ha creado una capa de servicios. Esta clase es la encargada de conectar con el servidor para hacer las peticiones.

La clase *Services* tiene 2 atributos principales:

```

public class Services extends EventDispatcher
{

    private var webService:WebService;
    private var wsdl:String;
    ...
}
    
```

- La variable **webService**, de tipo *WebService*, define la conexión con el servidor. Esta clase provee acceso a web services remotos basados en SOAP.
- La variable **wsdl** almacena la dirección wsdl del web service, necesaria para realizar la conexión con el mismo.

Para el tratamiento del resultado devuelto por el servidor, la clase *Service* define un evento para lanzar de tipo *ServiceResultEvent*.

```
[ Event (name="ServiceResultEvent",  
type="events.ServiceResultEvent.SERVICE_RESULT_EVENT" ) ]
```

Este tipo de evento se ha creado específicamente para lanzar el resultado devuelto por el servidor. La clase que lo define tiene 2 atributos principales:

```
public var message:String;  
public var resultEvent:ResultEvent;
```

En *message* se puede almacenar el mensaje deseado y *resultEvent* será el valor devuelto desde el servidor.

La clase que hace la llamada a *Services* se suscribe al evento en concreto que le interesa de este tipo. A continuación se presenta un ejemplo de cómo una clase se suscribe a este tipo de evento, para posteriormente poder tratarlo:

```
services.addEventListener(EventsConstant.SET_SCENARIO_EVENT, setScenarioServiceEventHandler);  
  
services.setScenarioService(modelLocator.scenarioAux, modelLocator.user.userId);
```

En la primera línea la clase llamadora se suscribe al evento: *SET_SCENARIO_EVENT*, que es de tipo *ServiceResultEvent*. En esta misma instrucción define la función que se hará cargo del tratamiento del evento: *setScenarioServiceEventHandler*. En la segunda línea hace el llamamiento a una función concreta de *Services* pasándole los parámetros necesarios.

Por otro lado, dentro de la clase *Services*, la llamada al servidor se establece de la siguiente forma:

```
public function setScenarioService(scenario:Scenario, userId:int):void  
{  
    var object:Object=converter.scenarioToObject(scenario);  
    webService.setScenario.addEventListener(ResultEvent.RESULT,  
setScenarioServiceResultHandler);
```

```

        webService.setScenario.addListener(FaultEvent.FAULT,
setScenarioServiceFaultHandler);
        webService.setScenario.resultFormat="object";
        webService.setScenario.send(object,userId);
    }

private function setScenarioServiceResultHandler(event:ResultEvent):void
{
    var serviceResultEvent:ServiceResultEvent = new
ServiceResultEvent("",event,EventsConstant.SET_SCENARIO_EVENT);
    dispatchEvent(serviceResultEvent);
}

```

En primer lugar se suscribe al evento *RESULT* y al evento *FAULT* del servidor, para saber en qué momento se ha obtenido una respuesta positiva o negativa del mismo. A continuación se establece la forma en cómo se quiere recibir la respuesta, en este caso será un objeto. Por último se hace la llamada a la función del web Service.

La función que se ha definido para tratar el evento *RESULT* del servidor, en este caso *setScenarioServiceResultHandler*, será la encargada de lanzar el evento de tipo *ServiceResultEvent*, para que la clase llamadora haga el tratamiento necesario con el resultado devuelto por el servidor.

8.2. Servidor

Dentro del servidor se considera relevante la propia configuración del servidor y la forma en cómo se comunica con la aplicación cliente. Para poder enviar y recibir datos se deben definir una serie de estructuras que se explicarán a continuación.

8.2.1. ABC del Endpoint

Como se explicó anteriormente, en el capítulo 7 – sección: *Tecnologías utilizadas*, para la comunicación entre cliente y servidor se deben establecer los endpoints o extremos. Cada nuevo endpoint definirá un nuevo canal de comunicación y una forma distinta de comunicarse entre cliente y servidor. Éste vendrá definido por el comúnmente conocido como ABC del endpoint: Address, Binding y Contract.

8.2.1.1. Address y Binding

La configuración del web Service se define en el archivo Web.config del servidor. Se debe establecer la dirección donde está alojado (*baseAddress*), el endpoint y el tipo de binding.

```

<system.serviceModel>

    <services>

```

```

    <service behaviorConfiguration="SimulatorService"
name="Servicios.Simulador">
    <host>
        <baseAddresses>
            <add baseAddress="http://193.145.154.10:8082/Servicios/" />
        </baseAddresses>
    </host>

    <endpoint address="" binding="basicHttpBinding"
bindingConfiguration="" contract="Servicios.ISimulador" />

    </service>
</services>

<behaviors>
    <serviceBehaviors>
        <behavior name="SimulatorService">
            <serviceMetadata httpGetEnabled="true" />
        </behavior>
    </serviceBehaviors>
</behaviors>

</system.serviceModel>

```

8.2.1.2. Contract

Existen diversos tipos de contratos, como ya se explicó en el capítulo anterior. En el siguiente cuadro se presenta un ejemplo de los que se han utilizado en este proyecto. Los contratos utilizados han sido: *ServiceContract*, que define el contrato del servicio, *OperationContract*, que define los parámetros de entrada y el tipo del valor de la salida de las operaciones del web Service y, *DataContract*, que se utiliza para definir los tipos de datos complejos que usa el servicio.

```

namespace Servicios
{
    [ServiceContract]
    public interface ISimulador
    {
        [OperationContract]
        List<Substance> getAllSubstances(int userId);
        ...
        ...
    }
}

namespace Servicios
{
    [DataContract]
    public class Accident
    {
        ...
        ...
    }
}

```

8.2.2. Utilización del modelo FLI (Fuga de Líquido Inflamable)

En el capítulo 6, referente al análisis de la plataforma, a la hora de exponer los distintos modelos desarrollados en la misma, se indicó la forma en que se iba a usar este modelo. Dada su complejidad a nivel computacional, el equipo del CEANI, que fue quien proporcionó la forma de calcular los resultados, optó por la elaboración de una red neuronal que entrenó en base a un elevado número de muestras proporcionado por un software profesional utilizado ampliamente en este campo.

Para poder hacer uso de la red neuronal fue necesario cargar las distintas sustancias sobre las que se haría el cálculo, que en este caso han sido:

- Acetona
- Gasoil
- Gasolina
- Tolueno

Es decir, los cálculos que el programa ofrece siempre han de ser para fenómenos peligrosos donde participen alguna de estas sustancias. Esto se debe a que la red neuronal fue entrenada para cada una de estas sustancias en base a los valores de muestra indicados anteriormente.

A continuación se muestra el ejemplo de cómo se hizo uso de las sustancias dentro del servidor, para el cálculo de las zonas afectadas por el derrame:

```
public double fli(float zoneValue, int substanceId, float diameter,
float windSpeed, int leakType, float leakDuration)
{
    BasicNetwork AcetonaNetwork = null;
    BasicNetwork GasoilNetwork = null;
    BasicNetwork GasolinaNetwork = null;
    BasicNetwork ToluenoNetwork = null;

    double radioOutput = 0.0;

    double diameterNorm = Normalize(diameter, 400, 10, 0.9, 0.1);
    double WindNorm = Normalize(windSpeed, 8, 1, 0.9, 0.1);
    double DiaNocheNorm = leakType==0 ? 0.1 : 0.9;
    double DurationNorm = Normalize(leakDuration, 120, 30, 0.9,
0.1);
    double zonaNorm = Normalize(zoneValue, 8, 4, 0.9, 0.1);

    //Primero: se cargan las redes, en función de la sustancia
    switch (substanceId)
    {
        case 19:
```

```

        string AcetonaPath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Substances/acetona.eg");
        AcetonaNetwork =
(BasicNetwork)EncogDirectoryPersistence.LoadObject(new
FileInfo(AcetonaPath));

        double[] radioNorm = new double[1];
        AcetonaNetwork.Compute(new double[] { diameterNorm,
WindNorm, DiaNocheNorm, DurationNorm, zonaNorm }, radioNorm);

        radioOutput = Denormalize(radioNorm[0], 230, 25,
0.9, 0.1);
        break;

    case 20:
        string GasoilPath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Substances/gasoil.eg");
        GasoilNetwork =
(BasicNetwork)EncogDirectoryPersistence.LoadObject(new
FileInfo(GasoilPath));

        DurationNorm = Normalize(leakDuration, 180, 30,
0.9, 0.1);

        double[] radioNorm1 = new double[1];
        GasoilNetwork.Compute(new double[] { diameterNorm,
WindNorm, DiaNocheNorm, DurationNorm, zonaNorm }, radioNorm1);

        radioOutput = Denormalize(radioNorm1[0], 130, 20,
0.9, 0.1);
        break;

    case 21:
        string GasolinaPath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Substances/gasolina.eg");
        GasolinaNetwork =
(BasicNetwork)EncogDirectoryPersistence.LoadObject(new
FileInfo(GasolinaPath));

        double[] radioNorm2 = new double[1];
        GasolinaNetwork.Compute(new double[] {
diameterNorm, WindNorm, DiaNocheNorm, DurationNorm, zonaNorm },
radioNorm2);

        radioOutput = Denormalize(radioNorm2[0], 160, 15,
0.9, 0.1);
        break;

    case 22:
        string ToluenoPath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Substances/tolueno.eg");
        ToluenoNetwork =
(BasicNetwork)EncogDirectoryPersistence.LoadObject(new
FileInfo(ToluenoPath));

        double[] radioNorm3 = new double[1];

```

```

        ToluenoNetwork.Compute(new double[] { diameterNorm,
WindNorm, DiaNocheNorm, DurationNorm, zonaNorm }, radioNorm3);

        radioOutput = Denormalize(radioNorm3[0], 140, 10,
0.9, 0.1);
        break;
    }

    //Se calcula el valor
    return radioOutput;
}

...

private double Normalize(double input, double MaxInput, double
MinInput, double MaxOutput, double MinOutput)
{
    return ((input - MinInput) / (MaxInput - MinInput)) * (MaxOutput
- MinOutput) + MinOutput;
}

private double Denormalize(double input, double MaxInput, double
MinInput, double MaxOutput, double MinOutput)
{
    return ((input - MinOutput) / (MaxOutput - MinOutput)) *
(MaxInput - MinInput) + MinInput;
}

```


Capítulo 9. Conclusiones

En primer lugar se considera importante destacar la experiencia que ha aportado este proyecto. Desde el inicio de los estudios de la Ingeniería Informática, se van planteando distintos problemas que se han de solucionar, sin embargo, normalmente se trata de problemas acotados, secciones de un proyecto mayor o problemas relativamente pequeños. Este proyecto ha brindado la oportunidad de enfrentar directamente un proyecto mayor, partiendo desde el inicio del proceso hasta su consecución, obteniendo el producto requerido y, de esta forma, obligando a pasar por todo el proceso que requiere elaborar un proyecto de principio a fin.

A pesar de ser un proyecto individual, se ha utilizado un componente desarrollado en otro proyecto. El hecho de haber tenido que integrar otro producto también aporta una mayor dificultad, siendo necesario estudiar y trabajar con la documentación generada para ese componente. Al ser necesaria esa comunicación entre los dos proyectos, se ha tenido que trabajar en conjunto para obtener las especificaciones de entrada y de salida. Esta comunicación también ha ofrecido la oportunidad de trabajar en grupo y conocer todo lo que ello implica.

Para el desarrollo del proyecto se han utilizado lenguajes de programación desconocidos, al igual que el uso de una arquitectura cliente servidor. Todo lo que se ha tenido que leer y estudiar, todo lo que se ha tenido que investigar para poder utilizarlo, se considera que ha aportado un enriquecimiento a nivel curricular. Quizá no sea una experiencia demostrable, pero independientemente de ese detalle, es algo que se ha podido aprender para utilizar en un futuro para el desarrollo profesional.

Además de lo que este proyecto ha aportado personal y profesionalmente, también sería conveniente destacar lo que este proyecto puede aportar a la sociedad, concretamente al ámbito que le compete, el estudio de los riesgos de accidentes en entornos industriales. Como se ha ido explicando a lo largo de toda la memoria, el estudio de los riesgos en entornos industriales es muy importante para conseguir que se reduzcan los accidentes laborales. Esta plataforma por sí sola no puede hacer el estudio, pero sí puede contribuir en el mismo aportando los resultados de los cálculos basados en los modelos matemáticos.

A lo largo de los años, los estudios sobre prevención de riesgos laborales han ido ganando la importancia que merecen y actualmente es uno de los campos más relevantes. Por este motivo existe la obligatoriedad de realizar los planes de emergencia interior en las empresas, incluyendo el estudio de riesgos de accidentes graves. Esta plataforma ofrece a los expertos la posibilidad de realizar todos los cálculos necesarios (para los modelos que están desarrollados) y además posicionarlos en un mapa del sitio sobre el que se esté realizando el estudio. Al poder posicionar los accidentes y al obtener las áreas que se verían afectadas por los mismos, entre otras cosas, se podría:

Conclusiones

- Reposicionar los recursos utilizados para calmar los efectos producidos por un determinado accidente, como podría ser un extintor en caso de incendio
- Tener situados en lugares estratégicos botiquines específicos para cada tipo de accidente
- Obligar al uso de equipos de protección individual (EPIS) en determinadas áreas, para evitar fatales consecuencias en caso de producirse el accidente
- Etc.

El cliente es bastante flexible, permitiendo al experto definir distintas situaciones dentro de un mismo escenario. Puede mostrar u ocultar aquellos accidentes que desee. Además, tras realizar el estudio puede generar un archivo csv. Los archivos csv son utilizados comúnmente en este ámbito para posteriormente generar tablas y trabajar con ellas. Además otra característica importante es que se trata de una aplicación online, esto le permite al experto poder desplazarse a la zona e ir haciendo el estudio in situ. Lo que se pretende destacar es la validez de la plataforma en caso de un uso profesional de la misma.

Para terminar, resulta altamente gratificante poder aportar una herramienta dentro de este ámbito. El hecho de que la aplicación desarrollada íntegramente durante este proyecto pueda ayudar de alguna forma a salvar vidas, aunque sea de forma indirecta, ha resultado muy satisfactorio

Capítulo 10. Trabajos futuros

La plataforma desarrollada en este proyecto funciona completamente, pero a pesar de ello, siempre existirán formas de mejorarla. Dependiendo de los cambios y/o actualizaciones que se realicen, la plataforma variará en mayor o menor medida. A continuación se detallan una serie de propuestas de mejora:

- Crear una aplicación que le permita al administrador no tener que tocar directamente la BBDD, es decir, que cuando se tuviesen que actualizar datos podría hacerlo a través de esa aplicación.
- Crear una estructura de permisos para manejar la aplicación, de esta forma se podrían crear usuarios que puedan modificar y usuarios que sólo puedan consultar los datos.
- Cuando el usuario intenta borrar: una sustancia, un recurso, es decir, cualquier entidad básica, si está siendo utilizada en algún accidente o escenario le denegará la opción. Como actualización se propone darle al usuario la opción de borrar todo lo que esté asociado a esta entidad básica o cancelar el borrado.
- Añadir una opción en la pestaña de Mapas que le permita al usuario revisar en que escenarios está siendo utilizado el mapa seleccionado.
- Poder borrar varios escenarios y/o varios accidentes de forma simultánea
- Permitir al usuario subir sus mapas a un servidor donde pueda almacenarlos

Capítulo 11. Bibliografía

- Kazoun, C. & Lott, J.**
2008 *Adobe Flex 3*. United States of America, O'Reilly Media, Inc. 2008. 636p.
- Lott, J. & Patterson, D.**
2007 *ActionScript 3 with Design Patterns*. United States of America, Peachpit Press. 2007. 286p.
- Mook, C.**
2008 *ActionScript 3.0*. Traducción en Madrid. Anaya Multimedia, S.A. 2008. 1088p.
- Nielsen, J.**
2000 *Usabilidad. Diseño de sitios web*. Traducción en Madrid. Pearson Educación S.A. 2000. 432p.
- Larman, Craig**
2003 *UML y patrones: una introducción al análisis y diseño orientado a objetos*. Editorial Alhambra S. A. 2003. 590p.
- Shupe, R. & Rosser, Z.**
2008 *Learning ActionScript 3.0: a beginner's guide*. Canadá. 2008. 363p.
- Gamma, E. & Helm, R. & Johnson, R. & Vlissides, J.**
2003 *Patrones de diseño: Elementos de software orientado a objetos*. Traducción en Madrid. Pearson Educación, S.A. 2003. 384p.
- Cole, A.**
2008 *Learning Flex 3: Getting up to Speed with Rich Internet Application*. United States of América, O'Reilly Media, Inc. 2008. 304p
- Durán, A. & Bernárdez, B.**
2002 *Metodología para la Elicitación de Requisitos de Sistemas Software*. Sevilla 2002, Edición 2.3 en <http://www.lsi.us.es/~beat/>. 82p.
- Microsoft**
Obtenida durante el 2013 *Microsoft Developer Network*
<http://msdn.microsoft.com/es-es>

Fernando Díaz Alonso

2004

Análisis de consecuencias y zonas de planificación para explosiones industriales accidentales (en el ámbito de las directivas SEVESO)

<http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/ProyectoSubvencionados/2004/2004%20Informe-tesis%20doctoral%20PIGAQ.pdf>

Gobierno de Canarias

Obtenida durante el 2013

Consejería de Economía, Hacienda y Seguridad

<http://www.gobcan.es/dgse/>

Gobierno de España

Obtenida durante el 2013 a

NTP 321: Explosiones de nubes de vapor no confinadas: evaluación de la sobrepresión. Ministerio de Trabajo y Asuntos Sociales. Instituto Nacional de Seguridad e Higiene en el Trabajo.

http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/301a400/ntp_321.pdf

Obtenida durante el 2013 b

Guía para la realización del análisis del Riesgo medioambiental (en el ámbito del Real Decreto 1254/99 [Seveso II]). Ministerio del Interior. Dirección General de Protección Civil y Emergencias.

http://www.proteccioncivil.org/catalogo/carpeta02/carpetta22/g_rarm_presen.htm

1992

LEY 21/1992, de 16 de julio, de Industria. Ministerio de Industria, Energía y Turismo.

http://www.minetur.gob.es/industria/RII/legislacion/Legislacion/LeyIndustria_1992.pdf

1999

Real Decreto 1254/1999, de 16 de julio, por el que se aprueban medidas de control de los riesgos inherentes a los accidentes graves en los que intervengan sustancias peligrosas. Ministerio de la Presidencia. Agencia Estatal Boletín Oficial del Estado

<http://www.boe.es/buscar/doc.php?id=BOE-A-1999-15798>

2003

Directriz Básica de Protección Civil para el control y planificación ante el riesgo de accidentes graves en los que intervienen sustancias peligrosas. Cuadernos de Legislación de Protección Civil. Cuaderno VII (Nuevo).

<http://www.interior.gob.es/file/55/55653/55653.pdf>

- 2005 a *Real Decreto 119/2005*. Ministerio de la Presidencia
<http://www.boe.es/boe/dias/2005/02/11/pdfs/A04873-04874.pdf>
- 2005 b *Real Decreto 948/2005*. Ministerio de la Presidencia
<http://www.boe.es/buscar/doc.php?id=BOE-A-2005-13121>

Anexo I. Manual de usuario

A continuación se indican los pasos básicos para poder crear un escenario y realizar una simulación.

A1.1. Acceso a la aplicación

Para tener acceso a la aplicación hay que acceder a la siguiente dirección:

<http://193.145.154.10:8082/CLIENTE/General.html>

Una vez se accede a esta web se presenta la pantalla principal de la aplicación, que muestra:

- la cabecera,
- un pequeño resumen sobre para que es utilizada y
- el acceso al registro o al inicio de sesión



Fig. 35. Pantalla principal

A1.2. Registro de usuarios

El registro de un nuevo usuario se consigue accediendo al enlace proporcionado en la pantalla principal: *Realizar Registro*. Para completarlo serán necesarios al menos los campos marcados como obligatorios, es decir, los que están señalados con un asterisco.

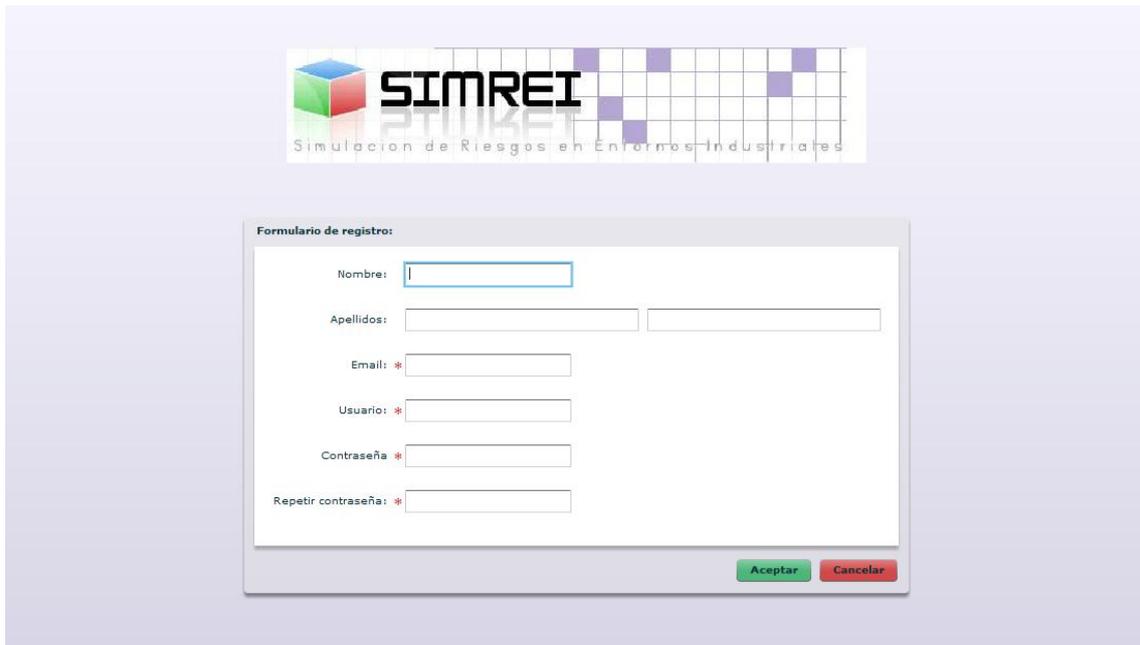


Fig. 36. Pantalla de registro de usuario

A1.3. Login

Para acceder a la aplicación y a todos los datos que ha cargado el usuario, se debe elegir la opción de login en la pantalla principal e ingresar los valores solicitados: usuario y contraseña. La aplicación comprobará la veracidad de los datos, una vez se ha comprobado que el usuario existe, se cargarán todos los datos.

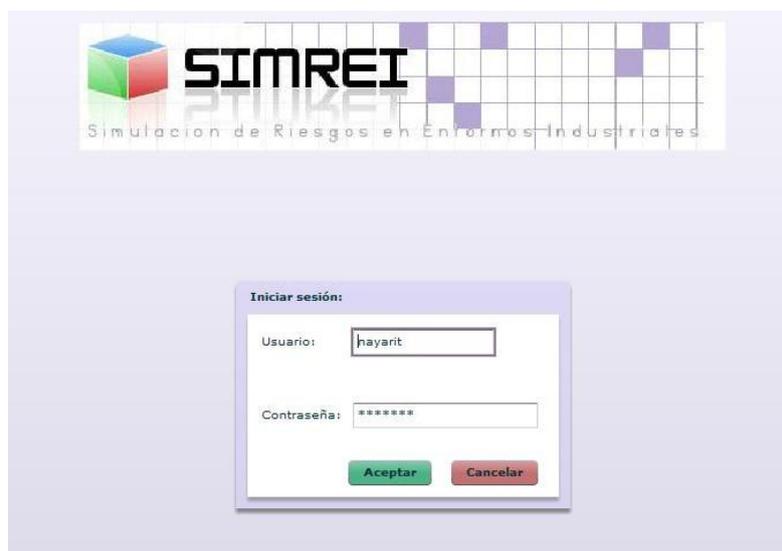


Fig. 37. Inicio de sesión

En primer lugar se debe indicar un nombre y una descripción, siendo la segunda opcional.

Introduzca todos los datos necesarios para crear el nuevo escenario. El mapa sólo podrá cambiarse si previamente no se han posicionado los accidentes.

Nombre: No utilizar mapa **Mapa seleccionado:** Todavía no hay un mapa seleccionado ...

Descripción:

Accidentes Recursos

- +

Nombre	Descripción

Imagen no Disponible

Fig. 39. Formulario para crear un nuevo escenario

A1.5.1. Asociar un mapa

En caso de querer posicionar los accidentes y/o recursos sobre un mapa, se tendrá que haber asociado uno previamente. Para asociar un mapa al escenario que se está creando, se ha de seleccionar el botón “seleccionar mapa”. En la pantalla se verá el listado de mapas que tiene el usuario y entonces podrá elegir el mapa deseado para el escenario en creación. En caso de que no se quiera asociar ningún mapa, el usuario debe seleccionar el checkbox indicado para tal fin.

A1.5.2. Crear nuevo accidente

A continuación, para crear un nuevo accidente se ha de picar sobre el botón señalado con un “+” dentro de la pestaña Accidentes, dentro de Escenarios.

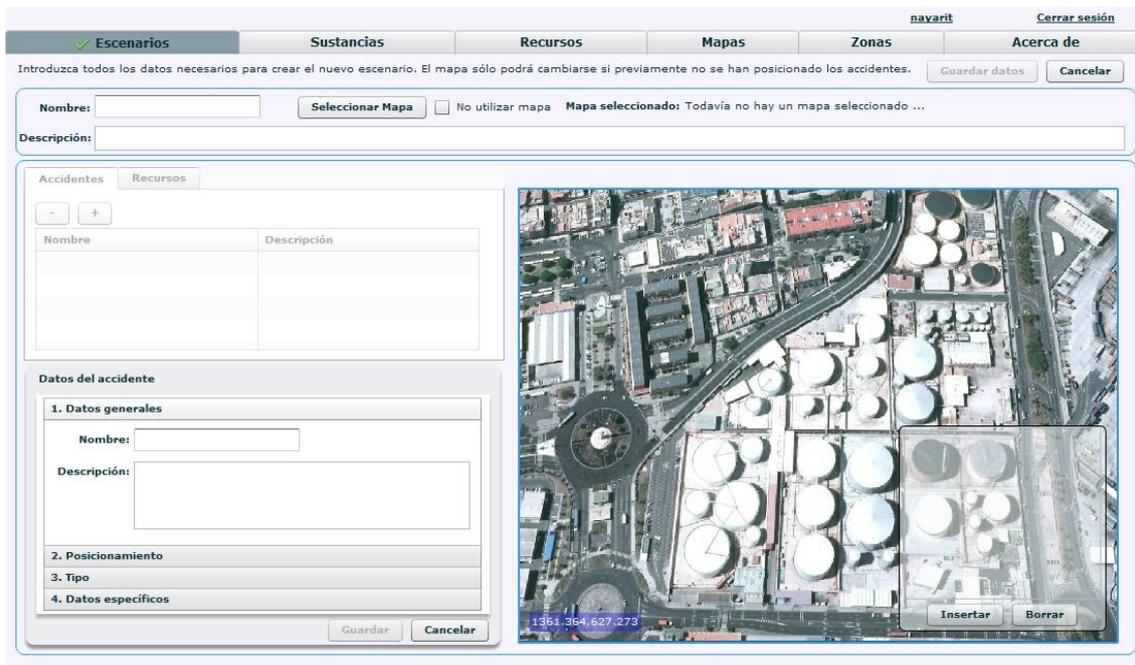


Fig. 40. Panel de nuevo accidente

Como se puede observar en la figura anterior, una vez se ha seleccionado la opción de nuevo accidente se despliega una nueva ventana. En esta ventana se solicitan los datos básicos para crear el nuevo accidente.

Cuando se está creando un nuevo accidente se deshabilitan el resto de opciones hasta que no se termina y se guarda o se cancela.

La ventana de nuevo accidente está dividida en 4 secciones:

1. Datos generales: aquí se piden datos que no afectan a las simulaciones, tales como el nombre y la descripción del accidente
2. Posicionamiento: se da la opción de posicionar el accidente sobre el mapa desde el momento de creación. A pesar de contar con un mapa asociado no es obligatorio posicionar los accidentes.

Existen 2 formas de posicionar el accidente:

- Posicionamiento manual: el usuario puede introducir las coordenadas a mano. La posición indicada debe estar dentro de las coordenadas del mapa que fueron establecidas por el usuario cuando lo guardó
- Posicionamiento a través del mapa: el usuario pincha sobre la posición donde desea marcar el accidente y la aplicación recoge la posición exacta en coordenadas.

Datos del accidente

1. Datos generales

2. Posicionamiento

Para posicionar el accidente en el mapa, debe seguir uno de estos pasos:

1. Indicar el punto exacto donde se encuentra
2. Posicionarlo en el mapa

x: y:

3. Tipo

4. Datos específicos

Fig. 41. Posicionamiento del accidente

3. Tipo: en esta sección se deben seleccionar las propiedades que definirán al accidente. Se debe indicar el tipo de accidente, el modelo matemático que se utilizará y la sustancia correspondiente.
4. Datos específicos: estos datos son, como su propio nombre indica, específicos del tipo de accidente. La aplicación pedirá los datos que necesita para poder realizar la simulación posteriormente. En el caso del modelo TNT, se solicita la masa de explosivo y si el cálculo es por sobrepresión o impulso.

Una vez se ha terminado de rellenar los datos solicitados se le da a guardar. Si todos están completos los datos obligatorios y no hay ningún valor incorrecto, que exceda algún parámetro marcado, el accidente se guardará y pasará a verse dentro de la tabla de accidentes que hay dentro del panel.

A1.5.3. Modificar accidente

Para modificar un accidente se debe seleccionar el accidente y pinchar sobre el botón "Modificar" que aparece en la ventana de información del mismo. Hay algunas modificaciones que no están permitidas, son aquellas que no dan opción al cambio cuando se selecciona la opción modificar. Por ejemplo, se considera que si se quiere cambiar el modelo matemático es mejor crear un nuevo accidente, en lugar de modificar el existente.

Cuando se modifican datos que afectan al resultado de una simulación, en caso de existir datos simulados, se marcan como no actualizados, de esta forma el usuario sabrá cuando los datos que está visualizando no son correctos, por no basarse en los valores indicados en el accidente.

A1.5.4. Eliminar accidente

Para eliminar un accidente basta con seleccionarlo y pulsar sobre el botón “-” o el botón eliminar que se visualiza en la ventana de información del accidente.

A1.5.5. Asociar un nuevo recurso

Para crear un nuevo recurso dentro del escenario que se está creando basta con acceder a la pestaña de recursos y pinchar sobre el botón “+”. A continuación saldrá una nueva ventana que solicitará los datos necesarios. Los datos solicitados estarán separados en 2 grupos y serán:

1. Tipo de recurso, nombre y descripción
2. Posicionamiento. Al igual que los accidentes, los recursos también se pueden posicionar. La forma de posicionar los recursos es idéntica a la forma de posicionar los accidentes.

A1.5.6. Eliminar un recurso

Basta con seleccionar el recurso que se desea eliminar y pinchar sobre el botón “-”. La otra opción es pinchar sobre el botón “Eliminar” que proporciona la ventana de información del recurso

A1.5.7. Guardar escenario

Una vez se han rellenado todos los datos necesarios sobre mapas, accidentes y recursos, el usuario debe pinchar sobre el botón “Guardar datos”. Si existiese algún problema con alguno de los valores insertados, la aplicación lanzará un aviso indicándolo.

A1.6. Simulación

Para poder simular los datos de los accidentes nos debemos encontrar en modo simulación. Como se explicó en la sección correspondiente, el modo de simulación se establece una vez ya se ha guardado el escenario y se accede de nuevo a él.

A1.6.1. Añadir nueva zona para simular

Para poder realizar la simulación, el usuario debe seleccionar las zonas sobre las que quiere obtener un resultado, para ello debe seleccionar el accidente que desea simular y posicionarse en la pestaña zonas. Para añadir una zona debe pinchar sobre el botón “+”, entonces la aplicación lanzará otra ventana donde puede seleccionar tantas zonas como desee (Fig. 42).

A1.6.2. Eliminar una zona

Para eliminar una zona que ya no interesa tener asociada a un accidente, basta con pulsar sobre el botón eliminar que se encuentra en cada registro de la tabla de zonas (Fig. 43).

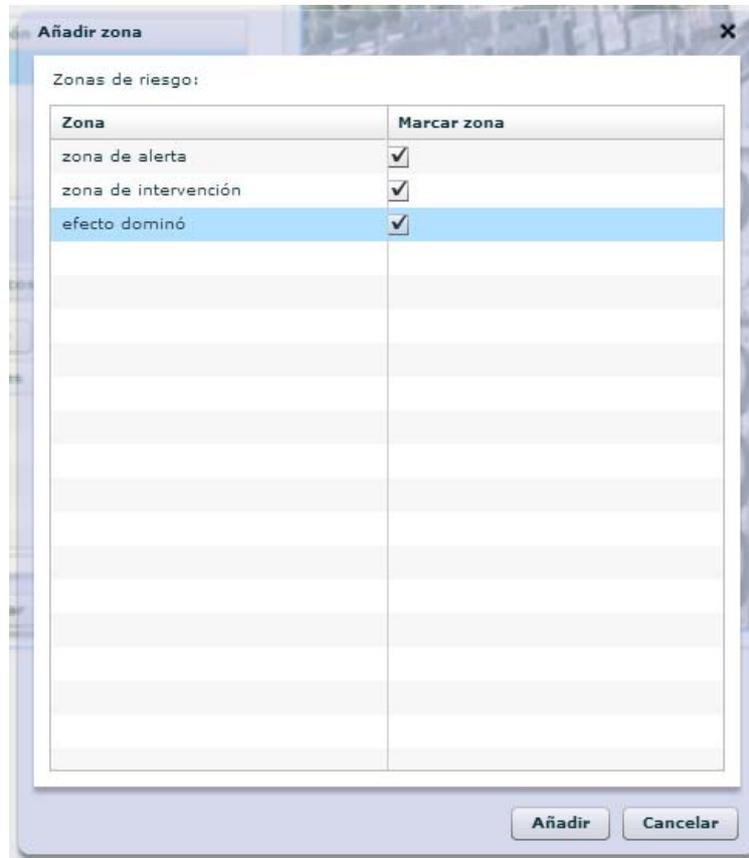


Fig. 42. Añadir zona al accidente para la simulación



Fig. 43. Botón eliminar zona

A1.6.3. Simular zonas

Para simular una zona existen 2 formas distintas de hacerlo:

- Por grupo: si existen varias zonas que se desean simular, se debe pinchar sobre el botón que se encuentra en la parte inferior derecha de la ventana "Simular", de esta forma se simularán todas las zonas asociadas al accidente.
- De forma individual: si por el contrario, sólo se quiere simular una zona concreta obviando el resto de zonas, se debe pulsar sobre el botón "Simular" que se encuentra en la misma fila de la tabla donde aparece el listado de zonas. Cada registro de la tabla tiene un botón de simulación.

Una vez se simulan las zonas, se muestran en el mapa los resultados en caso de estar posicionados.



Fig. 44. Representación de los valores simulados sobre el mapa

Los datos se pueden mostrar y ocultar pinchando sobre el botón que aparece en cada registro de la tabla y que tiene como indicativo un ojo (cuando el accidente se está mostrando) o un ojo con una cruz roja (cuando el accidente está oculto). Si se oculta un accidente, se ocultan todas sus zonas

Al igual que se puede mostrar y ocultar un accidente, también se pueden mostrar u ocultar de forma individual cada zona de cada accidente. El sistema utilizado es el mismo que para los accidente.

A1.7. Añadir nuevo mapa/zona/recurso/sustancia

En todos los casos se deben seguir los mismos pasos. Para crear un nuevo objeto se debe en primer lugar acceder a la pestaña correspondiente, a continuación se debe pinchar sobre el botón “+”. La aplicación mostrará una ventana donde se piden los datos necesarios en cada caso, siendo algunos de ellos obligatorios. Una vez se hayan ingresado los datos se debe pinchar sobre “Guardar”, en caso de que hubiese algún dato incorrecto o incompleto, la aplicación lanzará un aviso indicándolo.

A1.8. Modificar/ Eliminar mapa/zona/recurso/sustancia

Siempre y cuando el usuario sea propietario del objeto podrá modificarlo y/o eliminarlo, para ello debe acceder a la pestaña correspondiente y seleccionar el objeto concreto. Una vez esté seleccionado debe pinchar sobre “Modificar” o sobre “Eliminar”.

En el caso de que desee modificar el objeto, una vez haya terminado de cambiar los datos, debe guardar los cambios, si no, no tendrán validez.

Anexo II. Valores umbrales en TNT

Para evaluar las consecuencias de una explosión se puede partir de tablas comparativas de daños y sobrepresiones, donde se reflejan los valores umbrales utilizados en el caso del modelo TNT. A continuación se indican algunos otros valores umbrales, que se pueden sumar a los ya indicados en el capítulo 6 de esta memoria:

Daños Personales	Sobrepresiones
Umbral de muerte por lesiones de pulmón	70000 Pa (0,70 bar)
Umbral de rotura de tímpano	35000 Pa (0,35 bar)

Daños Estructurales	Sobrepresiones
Demolición total	80000 (0,80 bar)
Daños irrecuperables	40000 (0,40 bar)
Daños estructurales importantes	18000 (0,18 bar)
Daños graves reparables	15000 (0,15 bar)
Daños estructurales menores	4700 (0,047 bar)
Cristales rotos al 90%	4000 (0,040 bar)