



# Proyecto Fin de Carrera

## Anexos

Autor

Óscar Cardiel Enguita

Director

Sergio Ilarri Artigas

Escuela de Ingeniería y Arquitectura

Año 2014



# ANEXO A: Manual de Usuario

---

Este anexo contiene un manual de usuario de la aplicación desarrollada. En él se explican todas las capacidades del programa tanto para la monitorización de flujos de datos como de la simulación de ellos.

## A1 Introducción

La aplicación tiene dos funcionalidades principales. La primera monitorizar flujos de datos y la segunda simular estos flujos. La primera permite crear consultas, crear alarmas sobre tipos de sensores y visualizar los resultados obtenidos. La segunda permite obtener flujos de datos de tres tipos diferentes: flujos de GPS, flujos de cotizaciones y flujos genéricos basados en distribuciones estadísticas. También es posible importar trazas de ficheros GPS con diversos formatos y visualizarlos por pantalla, transformándolos al formato reconocido por el sistema de procesamiento.

En los sucesivos apartados de este manual se explicará la instalación del sistema, la ejecución del mismo y las cómo realizar las operaciones del mismo.

## A2 Inicio de la aplicación

En este punto se va a presentar la forma de compilar y ejecutar la aplicación, así como la pantalla inicial de la aplicación.

### A2.1 Compilación y ejecución

La aplicación funciona bajo Java lo que hace necesario que esté instalado el JDK de esta plataforma para poder compilarlo, y el JRE para ejecutarlo. En concreto la aplicación está desarrollada con el JDK 1.7, con lo cual será necesario tener instalada esta versión o superior. La aplicación funciona en cualquier plataforma que pueda ejecutar JDK 1.7, si bien esta optimizada para sistemas Windows, puede ejecutarse en entornos Linux.

Para realizar la compilación de todo el proyecto, habrá que tener el directorio del JDK dentro de las variables de entorno del sistema de elegido. En el caso de sistemas Windows habrá que ir a la pantalla PROPIEDADES DEL SISTEMA, VARIABLES DE ENTORNO y añadir en la variable PATH el directorio del JDK, es decir, si el JDK de java se encuentra en la carpeta C:\JDK1.7\, habrá que incluir la

línea en la variable PATH: C:\JDK1.7\bin. Una vez que esté este dato incluido, bastará con ejecutar el archivo COMPIALAR.BAT para que se generen los archivos compilados. Para ejecutar la aplicación bastará con ejecutar el archivo EJECUTAR.BAT.

Para realizar la misma operación en sistemas LINUX habrá que realizar la siguiente operación en la línea de comandos suponiendo que jdk se encuentre en el directorio /usr/java/jdk1.7/bin

```
export PATH=$PATH/usr/java/jdk1.7/bin
```

Una vez incluido, se compilará el proyecto ejecutando el script compilar.sh y se ejecutara con el script ejecutar.sh.

También hay dos scripts (uno para cada plataforma) que permite borrar todos los archivos de compilación generados, para poder proceder a otra compilación nueva.

## A2.2 Pantalla Inicial

Al iniciar la aplicación aparecerá la pantalla principal que consta de:

- A) Barra de Menú
- B) Menú Lateral en forma de Árbol
- C) Pantalla Principal
- D) Barra de mensajes.

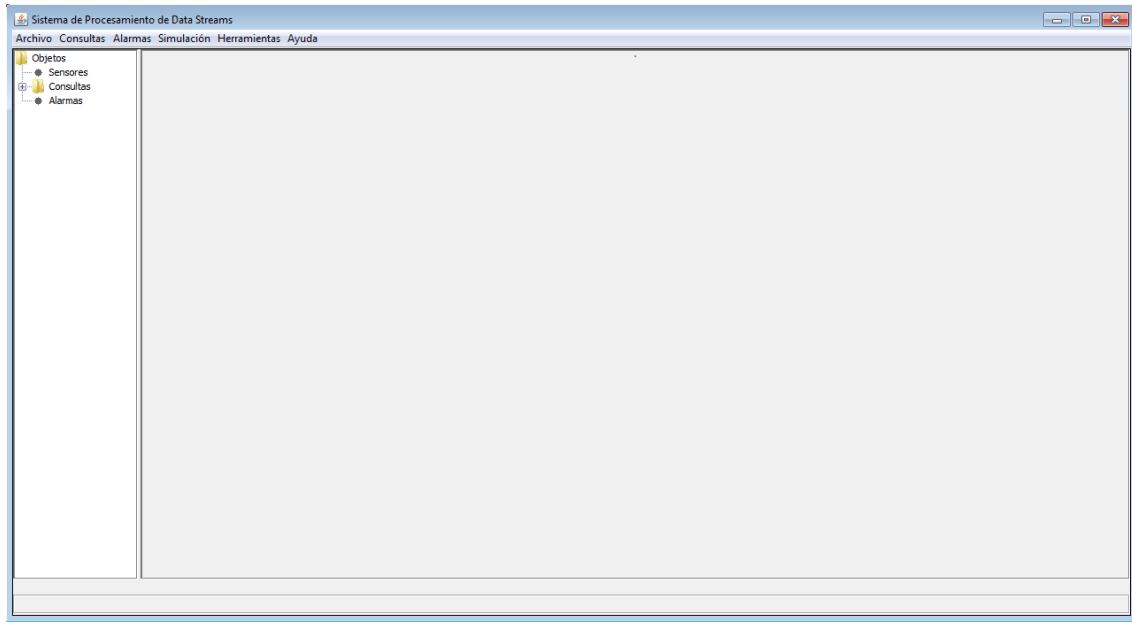


Figura A1 Pantalla Principal

Esta disposición permanece constante durante toda la ejecución del programa, solo variando la pantalla principal que mostrará la opción ejecutada por el usuario.

En esta pantalla se encuentra la barra superior con todas las opciones del programa, los menús de los que consta son:

ARCHIVO  
CONSULTAS  
ALARMAS  
SIMULACIÓN  
HERRAMIENTAS  
AYUDA



Figura A2 Barra de Menú

Menú Archivo - Desde él se puede salir de la aplicación y terminar la ejecución

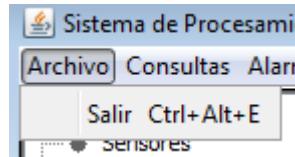


Figura A3 Menú Archivo

Consultas: Contiene el submenú de manejo de consultas:

*Abrir Consulta:* Permite abrir un fichero .dsm que contiene información de una consulta guardada previamente por el usuario.

*Guardar Resultados:* Permite guardar un fichero .dsm que contiene la información de una consulta (se guardarán todos los resultados que se hayan obtenido hasta el momento del guardado).

*Crear Consulta:* Muestra un menú para definir los valores de una consulta.

*Cancelar Consulta:* Permite cancelar una consulta que se está ejecutando en el sistema.

*Ver Consultas Activas:* Muestra un listado de las consultas que están activas en este momento.

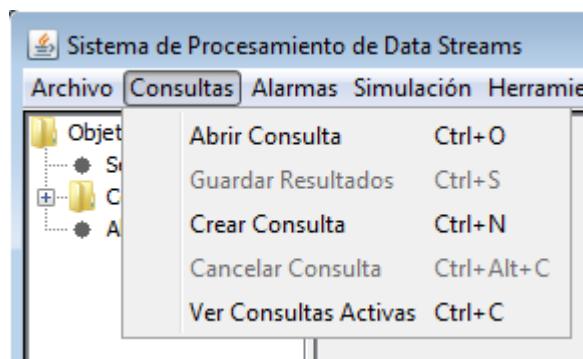


Figura A4 Menú Consultas

Alarmas: Contiene todas las operaciones de manejo de Alarmas sobre consultas.

*Crear Alarma:* Permite crear una alarma sobre una consulta ya definida, o bien crear una consulta a la que asociar la alarma.

*Lista de Alarmas:* Muestra todas las alarmas definidas en el sistema.

*Cancelar Alarma:* Muestra un listado de las alarmas para cancelar una de ellas (aunque la consulta seguirá ejecutándose).

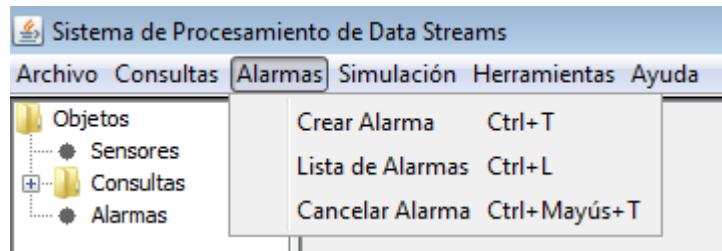


Figura A5 Menú Alarmas

*Simulación:* Contiene todas las operaciones referidas a la simulación de sensores.

*Importar:* Permite importar trazas GPS, o ficheros de simulación .sim.

*Exportar:* Permite exportar una simulación que se muestra por pantalla.

*Simuladores:* Este menú permite acceder a todos los tipos de sensores que puede simular el programa.

*GPS:* Simula una traza GPS.

*Cotizaciones:* Simula la evolución de una cotización respecto al tiempo

*Genérico:* Se trata de un simulador genérico basado en distribuciones estadísticas.

*Redes de Sensores:* Este menú permite acceder al menú de redes de sensores, conteniendo redes de los tipos de sensores anteriores.

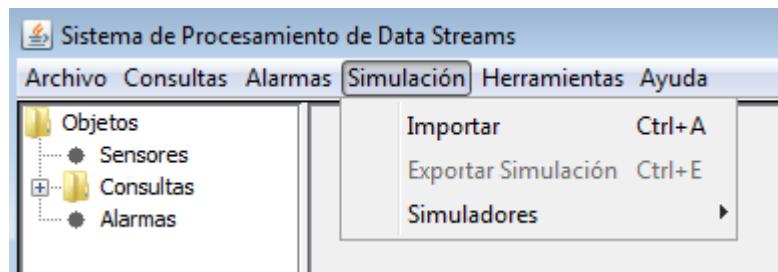


Figura A6 Menú Simulación

Herramientas: Desde este menú se accede a la configuración general de la aplicación.

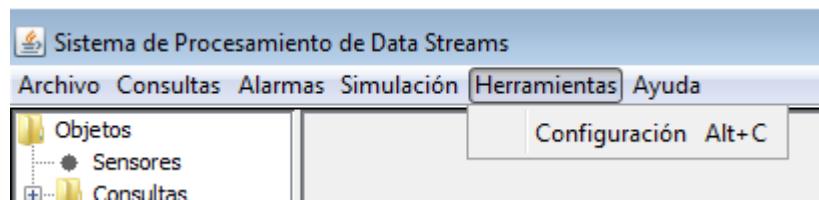


Figura A7 Menú Herramientas

Ayuda: Permite acceder a este manual de usuario y a la información de la aplicación.

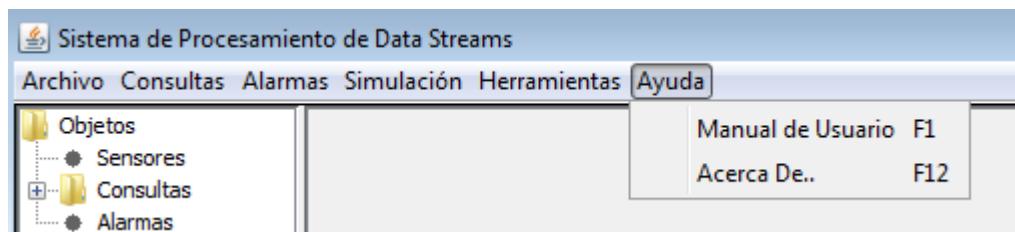


Figura A8 Menú Ayuda

### Menú lateral

El menú lateral muestra un acceso rápido a todos los objetos manejados por el sistema. Estos son:

SENSORES

CONSULTAS

ALARMAS.

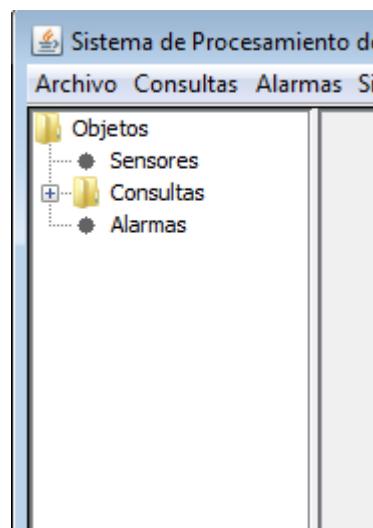


Figura A9 Menú Lateral

Sensores: Muestra todos los sensores asociados ejecutados por el simulador agrupados por tipo. Tanto en el tipo y en el nombre del sensor se puede acceder a un menú contextual con el botón derecho del ratón para realizar diferentes operaciones.

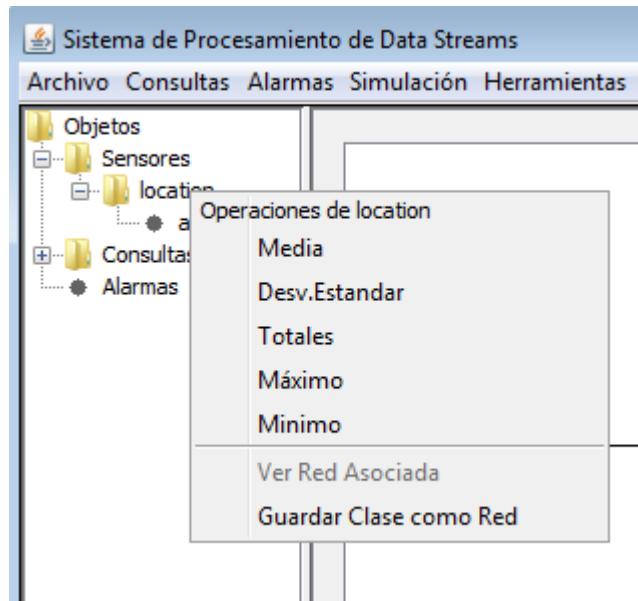


Figura A10 Menú Contextual Sensores

*Operaciones:* Permite ejecutar una operación de agregación sobre la clase de sensores o sobre el sensor para obtener datos como la MEDIA, DESVIACIÓN ESTANDAR, para más información ver apartado A7 FUNCIONES DE AGREGACIÓN

*Ver Red Asociada* (Solo sobre un sensor) Si el sensor pertenece a una red, permite ver la pantalla de configuración de toda la red.

*Guardar Clase como Red* (Solo sobre la clase de sensor): Permite guardar toda una clase como si fuera una red de sensores.

*Consultas:* Muestra todas las consultas definidas en la aplicación, también es posible acceder a un menú contextual pulsando con el botón derecho.

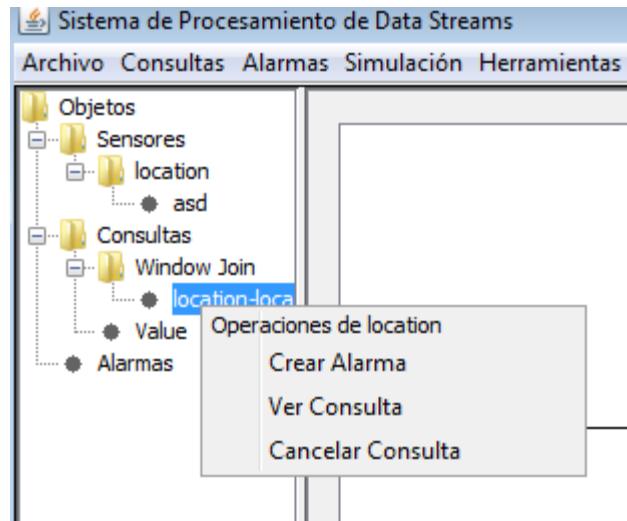


Figura A11 Menú Contextual Alarmas

*Crear Alarma* - Permite definir una alarma de la consulta seleccionada.

*Ver Consulta* - Permite ver los datos y resultados de una consulta, este efecto se consigue también haciendo doble click sobre la consulta.

*Cancelar Consulta* - Cancela la consulta seleccionada.

Alarmas: Muestra todas las alarmas dadas de alta en la aplicación, también consta de un menú contextual con diversas opciones:

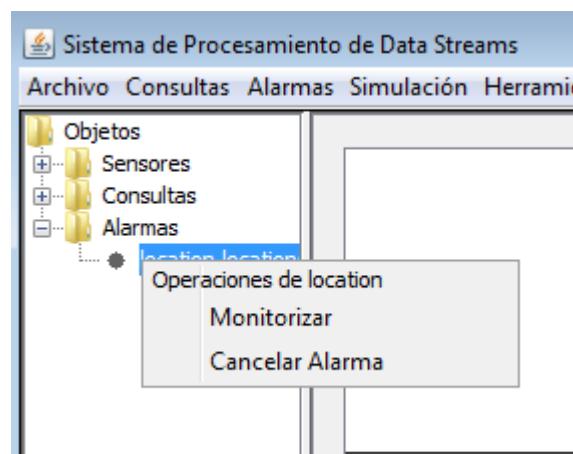


Figura A12 Menú Contextual Alarmas

*Monitorizar:* Muestra la consulta asociada a la alarma, este efecto se consigue también haciendo doble click sobre la alarma.

*Cancelar Alarma:* Cancela la Alarma seleccionada.

### A3 Combinación de teclas de acceso rápido.

Para agilizar la navegación en el programa, están definidas las siguientes teclas de acceso rápido.

CTRL-ATL-E Salir de la aplicación

CTRL-O Abrir Consulta

CTRL-S Guardar resultados de consulta

CTRL-N Crear nueva consulta

CTRL-ALT-C Cancelar Consulta

CTRL-C Ver consultas activas

CTRL-T Crear Alarma

CTRL-L Ver Listado de Alarmas

CTRL-MAYUS-T- Cancelar Alarma

CTRL-A Importar Simulación

CTRL-E Exportar Simulación

G- Simulación GPS

C-Simulación de Cotizaciones

S-Simulación Genérica

Mayus-G Simulación de Red GPS

Mayus-C Simulación de red de cotizaciones

Mayus-S Simulación de red genérica.

## Alt-C Menú de Configuración

## F1 Manual de Usuario

## F12 Acerca de.

**A4 Simulación**

En este apartado se recoge toda la información para realizar simulación de sensores de diferentes tipos, así como la carga y guardado de información referente a dichos sensores.

**A4.1 Realizar una simulación GPS**

Para realizar una simulación de un sensor GPS habrá que seleccionar la opción dentro del menú Simulación, submenú Simuladores, opción GPS para acceder a la pantalla de configuración del simulador, o bien pulsar G para acceder directamente.

Esta pantalla permite definir todos los parámetros necesarios para realizar una simulación:

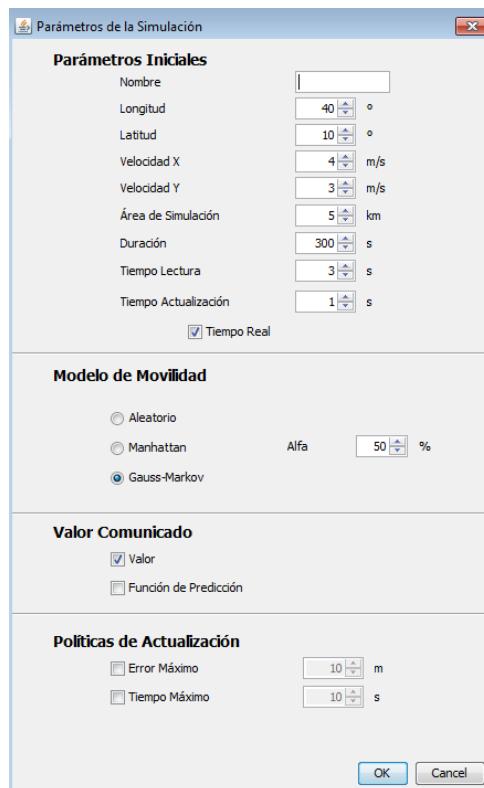


Figura A13 Configuración GPS

## PARÁMETROS INICIALES

Estos parámetros sirven para configurar los valores iniciales de la simulación, así como frecuencias de muestreo y otras opciones que afectan al desarrollo de la simulación

*Nombre:* Nombre del sensor GPS, (si se introduce uno que ya existe en el sistema lo creará con el nombre elegido añadiendo un 1 o número que corresponda hasta que no coincida con otro sensor del mismo tipo).

*Longitud:* Dato de la longitud inicial dado en grados decimales, este dato puede tomar valores desde -180º hasta 180º.

*Latitud:* Dato de la latitud inicial dado en grados decimales, este dato puede tomar valores desde -90º hasta 90º.

*Velocidad X:* Velocidad en m/s del móvil a simular en el eje X.

*Velocidad Y:* Velocidad en m/s del móvil a simular en el eje Y.

*Área de Simulación:* Área del cuadrado donde se realiza la simulación, si el móvil se acerca a los límites cambia la dirección para quedar dentro de los límites.

*Duración:* Duración de la simulación en segundos.

*Tiempo de Lectura:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado A6 CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el valor del tiempo de muestreo del sensor.

*Tiempo de Actualización:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado A6 CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el tiempo que tarda el actualizador en realizar el cambio en la posición del móvil a simular.

*Tiempo Real:* Si está activado, el simulador se ejecutará en tiempo real, si no se obtendrá toda la simulación directamente.

## MODELO DE MOVILIDAD

En esta sección se define el modelo de movilidad que seguirá el actualizador para realizar la simulación del elemento móvil. Estas opciones son:

**ALEATORIO:** El móvil variará  $+10^\circ$ ,  $-10^\circ$  o seguirá la misma trayectoria según dos sorteos aleatorios, uno determinara si hay cambio de trayectoria con el valor de la opción PROBABILIDAD que se seleccione, y si se produce un cambio se realizará otro sorteo para determinar si se incrementa o disminuye la trayectoria.

**MANHATTAN:** El elemento móvil seguirá una trayectoria simulando el modelo MANHATTAN, éste se moverá con un 25 de probabilidad en cada uno de los puntos cardinales.

**GAUSS-MARKOV:** El elemento móvil cambiara su trayectoria siguiendo el modelo GAUSS MARKOV con un valor de ALFA definido por el usuario.

Para mayor información sobre estos modelos consultar la sección MODELOS DE MOVILIDAD de este anexo.

#### VALOR COMUNICADO

En esta sección se permite seleccionar el dato que comunicará el sensor al sistema de procesamiento.

**VALOR:** El sensor solo comunicará la posición del elemento móvil.

**FUNCIÓN DE PREDICCIÓN:** El sensor comunicará una función de predicción que permite calcular la posición según el valor del tiempo.

#### POLITICAS DE ACTUALIZACIÓN

En esta sección se pueden definir los parámetros que provocaran un envío del nuevo dato al sistema de procesamiento.

**ERROR MÁXIMO:** El sensor enviará un dato siempre que la distancia entre el anterior dato comunicado y el actual sea mayor que el valor definido en Error Máximo.

**TIEMPO MÁXIMO:** El sensor enviará un dato siempre que el último dato se haya comunicado tantos segundos antes como se haya definido en *Tiempo Máximo*, este dato se enviará en el muestreo inmediatamente siguiente al vencimiento de este tiempo.

Si no se selecciona ninguno de estas políticas, el sensor enviará siempre el dato cada tantos segundos como se haya definido en el tiempo de lectura.

Una vez configurado todos los parámetros del sensor, pulsando aceptar accederemos a la pantalla de simulación.

Esta pantalla muestra de forma gráfica un eje cartesiano donde la línea roja muestra la trayectoria real del elemento móviles y los círculos azules muestran datos que se han mandado al sistema de procesamiento. Desde esta pantalla, si estamos ejecutando una simulación en tiempo real podremos pausar la simulación o detenerla definitivamente. También será posible realizar ZOOM para ampliar el tamaño de la trayectoria.

En el recuadro de texto de abajo se obtendrán todas las comunicaciones que realiza la simulación, con el formato de la tuplas de comunicación.

También será posible guardar la simulación, para ello ver el apartado GUARDAR SIMULACIÓN DE SENSORES

#### **A4.2 Realizar una simulación de cotizaciones**

Para realizar una simulación de un sensor de cotizaciones habrá que seleccionar la opción dentro del menú Simulación, submenú Simuladores, opción Cotizaciones para acceder a la pantalla de configuración del simulador, o bien pulsar C directamente.

Esta pantalla permite definir todos los parámetros necesarios para realizar una simulación:

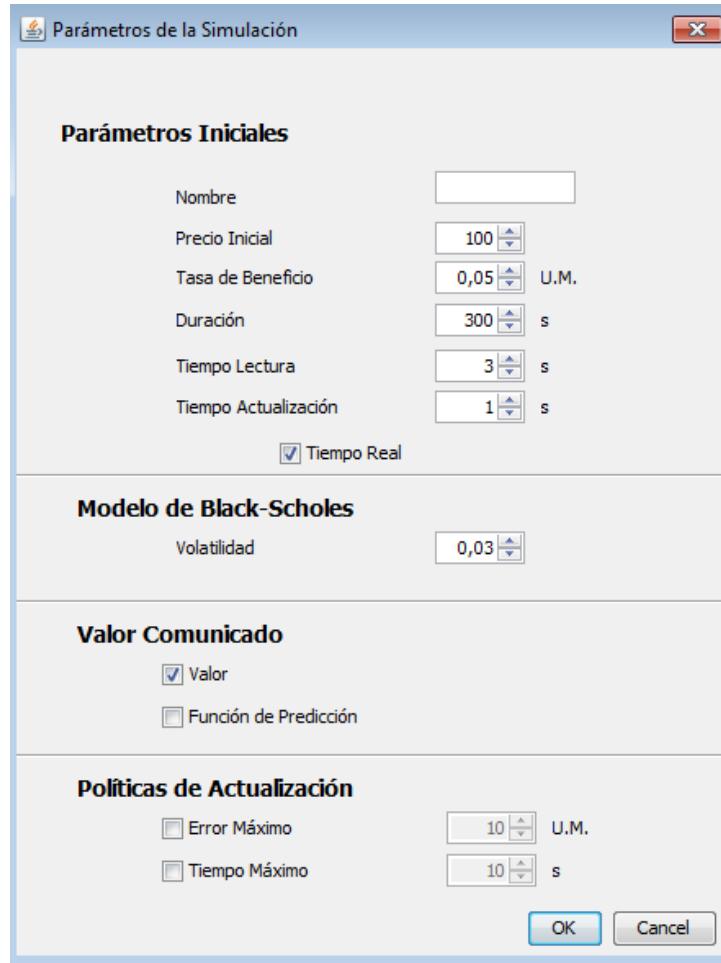


Figura A14 Configuración Cotizaciones

## PARÁMETROS INICIALES

Estos parámetros sirven para configurar los valores iniciales de la simulación, así como frecuencias de muestreo y otras opciones que afectan al desarrollo de la simulación

*Nombre:* Nombre del sensor de cotizaciones, (si se introduce uno que ya existe en el sistema lo creará con el nombre elegido añadiendo un 1 o número que corresponda hasta que no coincida con otro sensor del mismo tipo).

*Precio Inicial:* Establece el valor del precio inicial de la simulación

*Tasa de Beneficio:* Indica el crecimiento que sufrirá la acción.

*Duración:* Duración de la simulación en segundos.

*Tiempo de Lectura:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el valor del tiempo de muestreo del sensor.

*Tiempo de Actualización:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el tiempo que tarda el actualizador en realizar el cambio en la posición del móvil a simular

*Tiempo Real:* Si está activado, el simulador se ejecutará en tiempo real, si no se obtendrá toda la simulación directamente.

### MODELO DE BLACK-SCHOLES

En este apartado se puede configurar el valor de la volatilidad que sufrirá la acción. Para más información de cómo funciona este modelo consultar el anexo F: Modelo de Black-Scholes.

### VALOR COMUNICADO

En esta sección se permite seleccionar el dato que comunicará el sensor al sistema de procesamiento.

*Valor:* El sensor solo comunicará la posición del elemento móvil.

*Función de predicción:* El sensor comunicará una función de predicción que permite calcular la posición según el valor del tiempo.

### POLITICAS DE ACTUALIZACIÓN

En esta sección se pueden definir los parámetros que provocaran un envío del nuevo dato al sistema de procesamiento.

*Error Máximo:* El sensor enviará un dato siempre que la distancia entre el anterior dato comunicado y el actual sea mayor que el valor definido en Error Máximo.

*Tiempo Máximo:* El sensor enviará un dato siempre que el último dato se haya comunicado tantos segundos antes como se haya definido en *Tiempo Máximo*, este dato se enviará en el muestreo inmediatamente siguiente al vencimiento de este tiempo.

Una vez configurado, pulsando aceptar se accederá a la pantalla de simulación. En ella se mostrará en el eje central el valor inicial de la cotización, y se podrá observar como esta fluctúa según transcurre la simulación, desde aquí podremos pausar o detener la simulación y hacer zoom en el recuadro de visualización.

En el recuadro de texto de abajo se obtendrán todas las comunicaciones que realiza la simulación, con el formato de la tuplas de comunicación.

También será posible guardar la simulación, para ello ver el apartado GUARDAR SIMULACIÓN DE SENSORES.

#### A4.3 Realizar una simulación genérica

Para realizar una simulación de un sensor genérico (que no sea ni de tipo GPS ni de cotizaciones) habrá que seleccionar la opción dentro del menú Simulación, submenú Simuladores, opción Genérico para acceder a la pantalla de configuración del simulador, o bien pulsar G directamente.

Este tipo de simulador permite crear un tipo de sensor que se adapte a nuestras necesidades, ya que el valor del mismo cambiará siguiendo un valor aleatorio de una distribución de probabilidad definida.

Esta pantalla permite definir todos los parámetros necesarios para realizar una simulación:

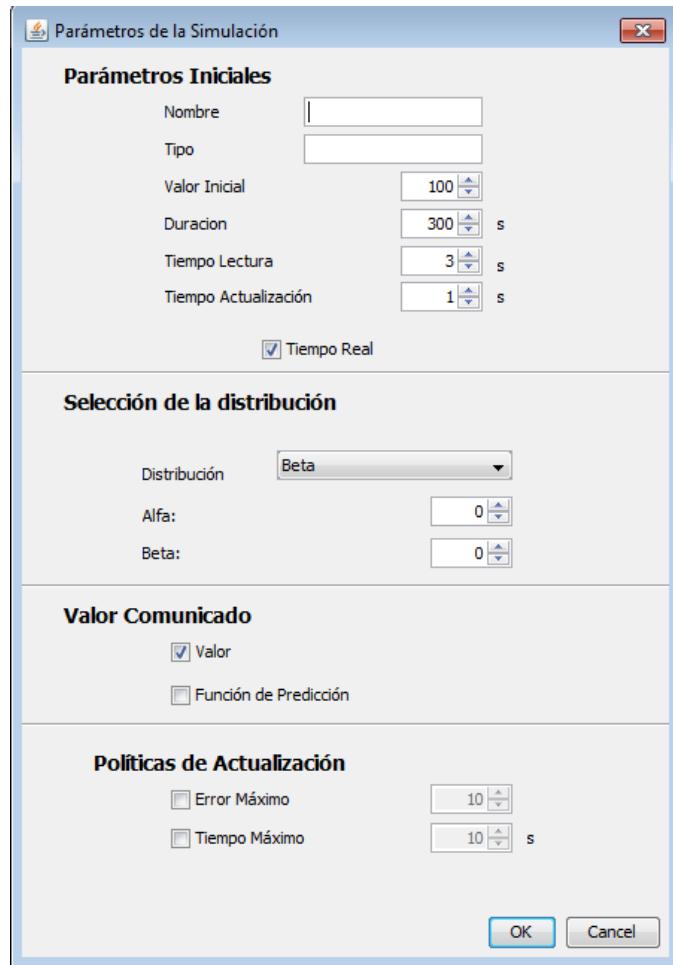


Figura A15 Configuración Genérico

## PARÁMETROS INICIALES

Estos parámetros sirven para configurar los valores iniciales de la simulación, así como frecuencias de muestreo y otras opciones que afectan al desarrollo de la simulación.

*Nombre:* Nombre del sensor de cotizaciones, (si se introduce uno que ya existe en el sistema lo creará con el nombre elegido añadiendo un 1 o número que corresponda hasta que no coincida con otro sensor del mismo tipo).

*Tipo:* Ya que se trata de un simulador genérico, en este punto podremos darle un nombre al tipo de sensor que estamos creando (por ejemplo “GASES”)

*Valor Inicial:* Establece el valor inicial de la simulación.

*Duración:* Duración de la simulación en segundos.

*Tiempo de Lectura:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el valor del tiempo de muestreo del sensor.

*Tiempo de Actualización:* Por defecto muestra el valor definido en la pantalla general de configuración (ver apartado CONFIGURACIÓN GENERAL DE LA APLICACIÓN). Modifica el tiempo que tarda el actualizador en realizar el cambio en la posición del móvil a simular.

*Tiempo Real:* Si está activado, el simulador se ejecutará en tiempo real, si no se obtendrá toda la simulación directamente.

### SELECCIÓN DE LA DISTRIBUCIÓN

En esta sección se puede configurar tanto la distribución de probabilidad que utilizará el actualizador, como los valores para configurarla. Según que distribución se elija los parámetros a configurar cambiarán.

### VALOR COMUNICADO

En esta sección se permite seleccionar el dato que comunicará el sensor al sistema de procesamiento.

*Valor:* El sensor solo comunicará la posición del elemento móvil.

*Función de predicción:* El sensor comunicará una función de predicción que permite calcular la posición según el valor del tiempo.

### POLITICAS DE ACTUALIZACIÓN

En esta sección se pueden definir los parámetros que provocaran un envío del nuevo dato al sistema de procesamiento.

*Error Máximo:* El sensor enviara un dato siempre que la distancia entre el anterior dato comunicado y el actual sea mayor que el valor definido en *Error Máximo*.

*Tiempo Máximo:* El sensor enviará un dato siempre que el último dato se haya comunicado tantos segundos antes como se haya definido en *Tiempo Máximo*, este

dato se enviará en el muestreo inmediatamente siguiente al vencimiento de este tiempo.

Una vez configurado, pulsando aceptar se accederá a la pantalla de simulación. En ella se mostrara en el eje central el valor inicial del sensor, y se podrá observar como este valor fluctúa según transcurre la simulación, desde aquí se podrá pausar o detener la simulación y hacer zoom en el recuadro de visualización.

En el recuadro de texto de abajo se obtendrán todas las comunicaciones que realiza la simulación, con el formato de la tuplas de comunicación.

También será posible guardar la simulación, para ello ver el apartado GUARDAR SIMULACIÓN DE SENsoRES.

#### A4.4 Realizar una simulación de una red de sensores

Para realizar una simulación de una red de sensores habrá que seleccionar la opción dentro del menú Simulación, submenú Simuladores, submenú Redes de Sensores y seleccionar el tipo de sensor del que queremos simular la red.

Una vez seleccionado aparecerá una pantalla como ésta:

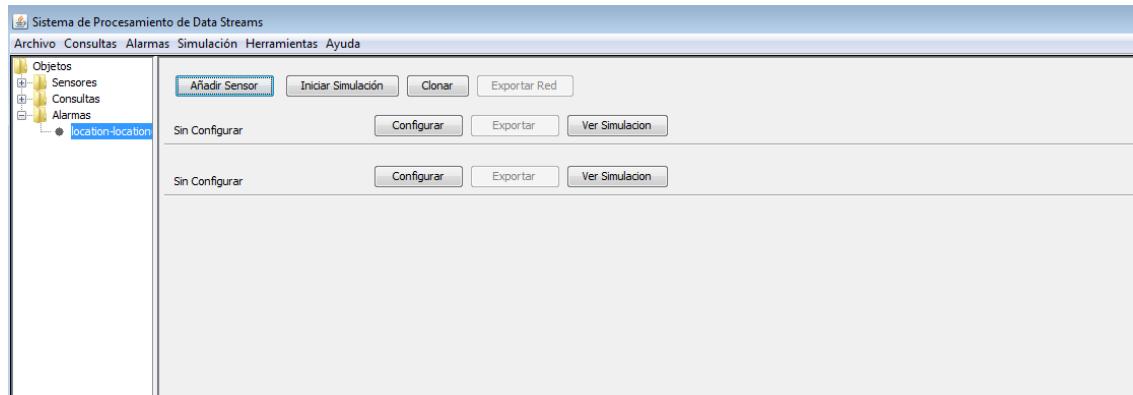


Figura A16 Red de Sensores

#### CONFIGURACIÓN GENERAL

En la parte superior de la pantalla aparece una serie de botones que permite realizar las siguientes acciones:

*Añadir Sensor:* Permite añadir un simulador a la red, añadiendo información a la pantalla de configuración.

*Iniciar Simulación:* Una vez configurado todos los sensores, permite iniciar la simulación.

*Clonar:* Pulsando en él, pedirá el número de clones del primer sensor que se quieren realizar. Este sensor deberá estar configurado para poder copiar en todos ellos la configuración de este (cambiando el nombre de cada uno).

*Exportar Red:* Una vez realizada la simulación permitirá exportar la red. Para más información consultar el apartado GUARDAR SIMULACIÓN DE SENSORES.

#### CONFIGURACIÓN DE CADA SENSOR

*Configurar:* Permite configurar el sensor, al pulsar obtendremos la pantalla de configuración del tipo del sensor de la red. Para más información sobre como configurar un simulador consultar la sección REALIZAR UNA SIMULACIÓN del tipo del que deseamos realizar la red.

*Exportar:* Una vez realizada la simulación permitirá exportar solo este sensor. Para más información consultar el apartado GUARDAR SIMULACIÓN DE SENSORES.

*Ver Simulación:* Permite crear una nueva ventana con la pantalla de simulación del sensor tal y como se explica en el apartado RELIZAR UNA SIMULACIÓN del tipo del que se está creando la red.

#### **A4.5 Guardar simulación de sensores**

Una vez que se haya realizado una simulación, será posible guardar los datos comunicados en un fichero .sim para posteriormente recuperarlos y visualizar el sensor o red de sensores asociadas e incluir estos datos en el sistema de procesamiento.

Para guardar una simulación finalizada accederemos al menú Simulación > Exportar Simulación., o pulsaremos CTRL+E. Una vez pulsado accederemos a una pantalla que nos permitirá escribir el nombre del archivo donde no será necesario incluir la extensión .sim.

Para guardar una red de sensores, se hará desde la pantalla de simulación de redes de sensores con el botón Exportar Red. En el caso de que se esté en otra pantalla, se podrá volver a la pantalla de simulación de redes haciendo clic en un sensor de la parte izquierda que pertenezca a esa red y pulsar VER RED ASOCIADA.

#### A4.6 Importar ficheros .sim

Los ficheros .sim son ficheros que genera la propia aplicación donde almacena las comunicaciones efectuadas por un sensor o red de sensores. Para importar bastará con ejecutar la opción IMPORTAR dentro de simulación o pulsar CTRL+A.

Si en el fichero hay almacenado solo un sensor, el programa mostrará solamente la pantalla de simulación con la gráfica y los datos comunicados. En este caso la línea roja será la trayectoria simulada y los puntos azules los puntos donde se efectúa comunicación.

En el caso de que haya una red de sensores, el programa mostrará la pantalla de configuración de red de sensores donde podremos acceder individualmente a cada simulación.

Todos estos simuladores aparecerán en el menú lateral de la aplicación.

Si se desea más información consultar el apartado DEFINICION DE FICHEROS-FICHERO DE SIMULACIÓN.

#### A4.7 Importar Trazas GPS

El sistema de simulación permite la carga de los siguientes tipos de trazas GPS.

GPX- Ficheros XML con formato definido que incluye una traza GPS.

PLT-Ficheros de Texto de la plataforma GEOLIFE de Microsoft.

CSV-Ficheros separados por comas de la plataforma MOVEBANK.

Para ver una descripción detallada de estos ficheros ver el anexo B FORMATO DE FICHEROS.

Para importar un fichero de estas características se hace de forma análoga a un fichero .sim. Una vez el programa detecta que se trata de uno de estos ficheros mostrará una pantalla donde seleccionar diversos parámetros.

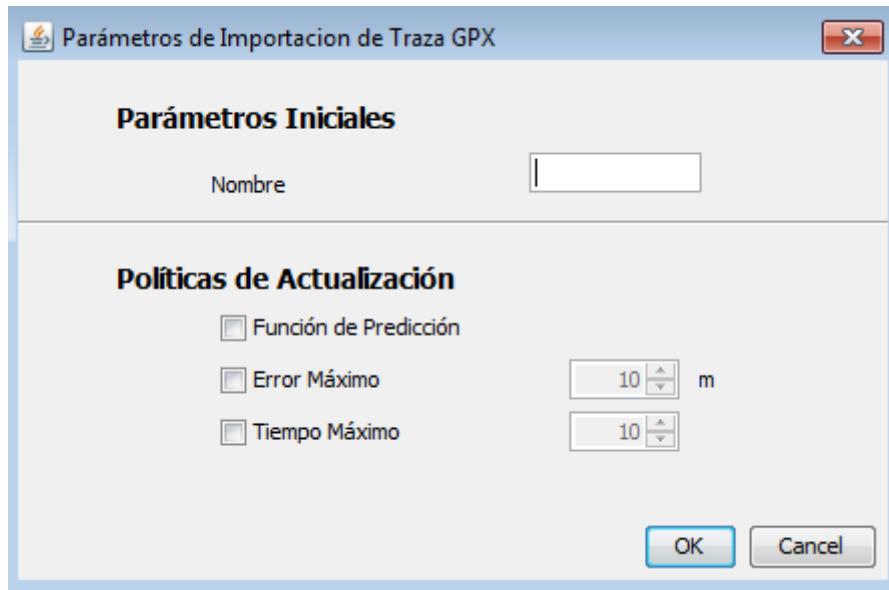


Figura A17 Importar Traza GPS

*Nombre:* Elegir un nombre para el sensor.

*Políticas de Actualización:* Establecerá que datos se importan siguiendo una política de actualización:

*Función de predicción:* Con los datos que se encuentran en el fichero se calcularán una serie de funciones de predicción que serán las que se comunicarán. En caso contrario se comunicará el valor del fichero.

*Error MÁXIMO:* Permite definir un valor para que se importe un dato si la distancia con el anterior es mayor a este error.

*Tiempo MÁXIMO:* Permite definir un valor para que se importe un dato si el dato anterior se emitió hace una cantidad de tiempo superior a tiempo máximo.

Como se ven estas son las opciones generales a todos los simuladores que se pueden aplicar a la hora de simular una traza.

Una vez realizado estos cálculos, mostrará el resultado por pantalla y se tomará como si fuera una simulación estándar realizada por el sistema.

## A5 Procesamiento de *data streams*

En este apartado están incluidas todas las funcionalidades para el manejo de los flujos generados por el apartado de simulación, desde cómo crear una consulta, como asociarle una alarma, visualizar resultados etc...

### A5.1 Crear una consulta

Para crear una consulta hay que ir al menú CONSULTAS > Crear Consulta.

Aparecerá una pantalla de configuración de una consulta WINDOW JOIN donde podremos definir los siguientes parámetros.

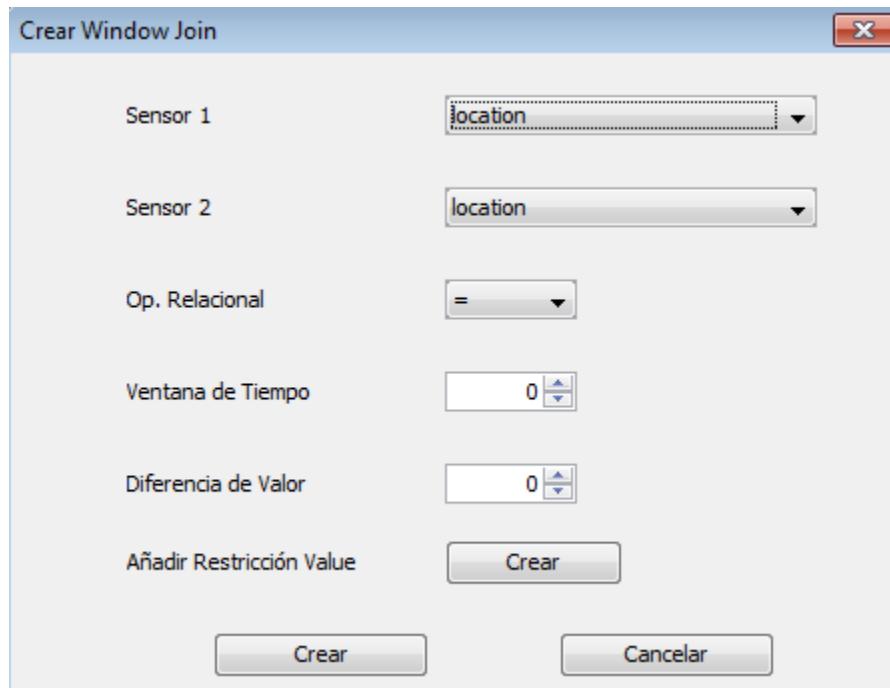


Figura A18 Crear Consulta

*Clase de Sensor 1:* Clase de sensores 1 donde se va a ejecutar la consulta

*Clase de Sensor 2:* Clase de sensores 2 donde se va a ejecutar la consulta

*Op. Relacional:* Operador relacional a aplicar en la consulta:

*W:* Valor de la ventana de tiempo donde tiene validez la consulta

*Diferencia de Valor:* Permite establecer el valor que tiene que cumplir la consulta.

*Añadir Restricción Value:* Permite añadir una restricción de valor a la consulta, desde aquí se accede a otra pantalla similar a esta donde introducir las siguientes propiedades:

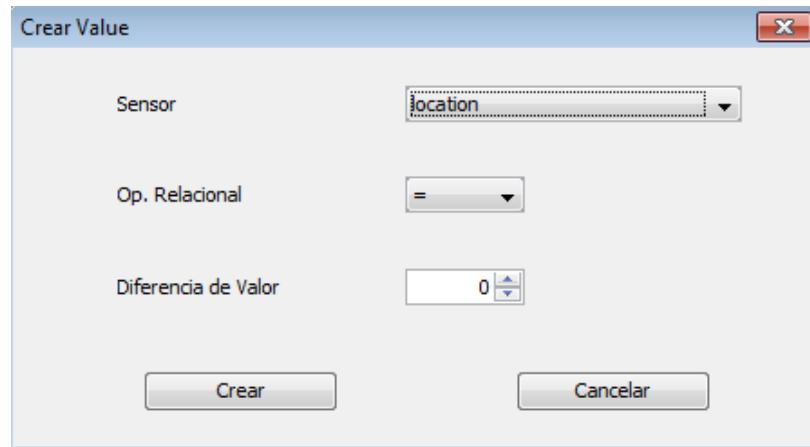


Figura A19 Crear Restricción

*Clase de Sensor:* Clase de sensor donde se va a ejecutar la consulta

*Op. Relacional:* Operador relacional a aplicar en la consulta:

*Diferencia de Valor:* Permite establecer el valor que tiene que cumplir la consulta.

Una vez definidos todos los valores se podrá guardar la consulta pulsando CREAR y esta se añadirá a la lista de consultas del menú de la izquierda, y desde este momento se empezará la monitorización del sistema.

## A5.2 Visualizar una consulta del sistema

Para visualizar una consulta definida se puede acceder de diferentes formas:

1- En la opción CONSULTAS>VER CONSULTAS ACTIVAS y seleccionando la consulta deseada

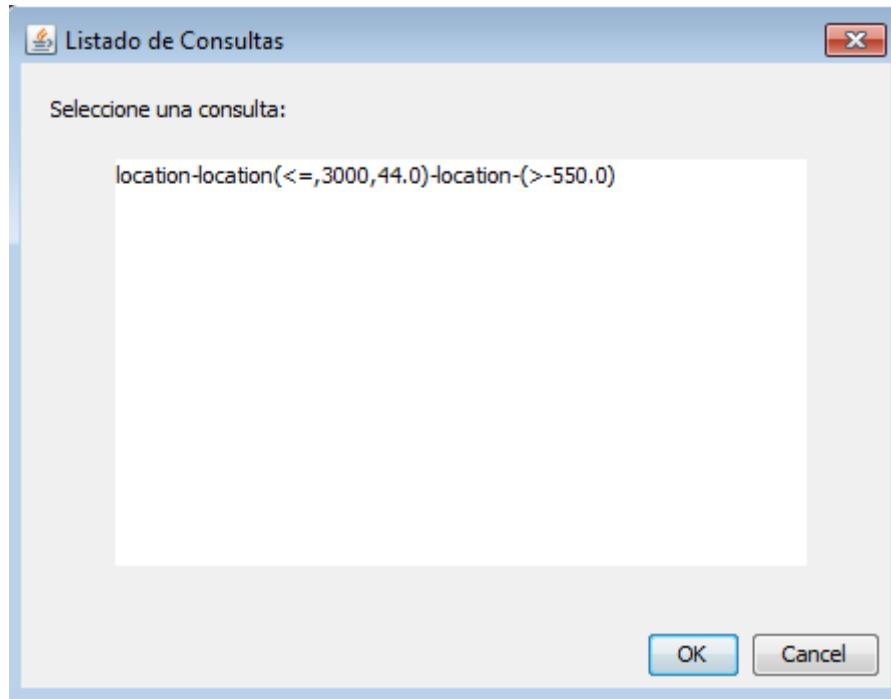


Figura A20 Listado de Consultas

2-Doble clic en el menú de la izquierda en la consulta deseada

3-Click derecho en el menú de la izquierda en la consulta deseada y pulsar Ver Consulta.

Una vez pulsado aparecerá la pantalla de monitorización de consultas. En ella se muestra la siguiente información:

En la parte superior de la pantalla se muestra la información de la consulta (clase de sensores, valor de la ventana temporal, operador relacional, restricción de valor etc...).

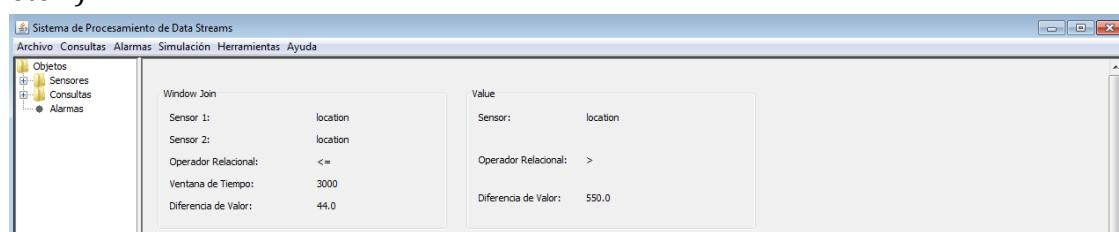


Figura A21 Consulta Parte Superior

En la parte media de la pantalla se muestra el estado de la consulta (ACTIVA, INACTIVA, ALARMA o FICHERO), y la hora de creación de la misma, así como botones de creación de alarma, cancelación y visualización de snapshot.

Propiedades			
Estado:	ACTIVA	<a href="#">Crear Alarma</a>	<a href="#">Cancelar Consulta</a>
Fecha de Creación:	<dynamic>	<a href="#">Snapshot</a>	

Figura A22 Consulta parte media

En la parte inferior se muestra una tabla con los resultados de la consulta. Conforme se van produciendo estos resultados, los datos se pueden visualizar en tiempo real. Haciendo doble clic en cualquier registro de la tabla se obtiene una pantalla como ésta donde se visualiza un resultado concreto de la consulta:

Id 1	Clase 1	Id 2	Clase 2	Predicción 1	Timestamp 1	Predicción 2	Timestamp 2	Marca
asdd	location	Asad	location	{4.0*t-5.6318608431...}	1409078405656	{4.0*t-5.6318608431...}	1409078405653	ValidityMark -> Interv.
Asad	location	asdd	location	{3.854392528533935...}	1409078408658	{4.0*t-5.6318608431...}	1409078405656	ValidityMark -> Interv.
asdd	location	Asad	location	{4.938167095184326...}	1409078411658	{3.854392528533935...}	1409078408658	ValidityMark -> Interv.
Asad	location	Asad	location	{4.938167095184326...}	1409078411658	{4.0*t-5.6318608431...}	1409078405653	ValidityMark -> Interv.
asdd	location	asdd	location	{5.934607028961182...}	1409078411658	{4.938167095184326...}	1409078411658	ValidityMark -> Interv.
Asad	location	asdd	location	{5.934607028961182...}	1409078411658	{4.0*t-5.6318608431...}	1409078405656	ValidityMark -> Interv.
asdd	location	Asad	location	{3.360336303710937...}	1409078414658	{5.934607028961182...}	1409078411658	ValidityMark -> Interv.
Asad	location	asdd	location	{3.360336303710937...}	1409078414658	{3.854392528533935...}	1409078408658	ValidityMark -> Interv.
Asad	location	asdd	location	{1.-485933780670166...}	1409078414658	{3.360336303710937...}	1409078414658	ValidityMark -> Interv.

Figura A23 Consulta Resultados

Desde esta pantalla podremos crear una alarma asociada pulsando el botón crear alarma. Una vez creada aparecerá con estado ALARMA y los botones que aparecerán serán CANCELAR ALARMA en vez de CANCELAR CONSULTA.

Propiedades			
Estado:	ALARMA CREADA	<a href="#">Crear Alarma</a>	<a href="#">Cancelar Alarma</a>

Figura A24 Consulta con Alarma

Si pulsamos cancelar consulta, ésta dejará de calcularse, aunque se mantendrán los datos y no se generaran nuevos resultados. Para guardar estos resultados deberemos ejecutar GUARDAR CONSULTA desde el menú Consulta para obtener un fichero .dsm que almacene toda la información generada por esta consulta.

#### **A5.3 Visualizar una consulta desde fichero**

Pulsando Consultas > Abrir Consulta se tendrá la posibilidad de seleccionar un fichero .dsm que almacene la información de una consulta. Una vez abierto se obtiene la misma pantalla de visualización de consultas de sistema aunque el estado de la consulta será FICHERO y no se podrán crear alarmas ya que no se trata de una consulta activa. Salvo por esta particularidad, el manejo es similar al apartado anterior.

#### **A5.4 Cancelar consulta**

Una consulta que esté activa en el sistema puede cancelarse para dejar de calcular resultados. Esta operación es irreversible y provocará el vaciamiento de buffers y puede provocar valores erróneos en las operaciones de agregación. Para cancelar una consulta bastará con pulsar CANCELAR CONSULTA en la pantalla de visualización de consultas, o bien seleccionar la consulta deseada en el menú izquierdo, pulsar botón derecho y seleccionar cancelar consulta, o desde la operación CANCELAR CONSULTA desde el menú CONSULTAS, que cancelará la consulta visualizada.

#### **A5.5 Crear una alarma**

Una alarma monitoriza una consulta para avisar al usuario si se producen resultados. Para crear una alarma podremos hacerlo desde la pantalla de visualización de una consulta, o en el menú contextual de la consulta. Una vez creada aparecerá en el menú izquierdo de la pantalla. En ese momento, en la pantalla de visualización de la consulta aparecerá como estado ALARMA indicando que la consulta tiene una alarma activada.

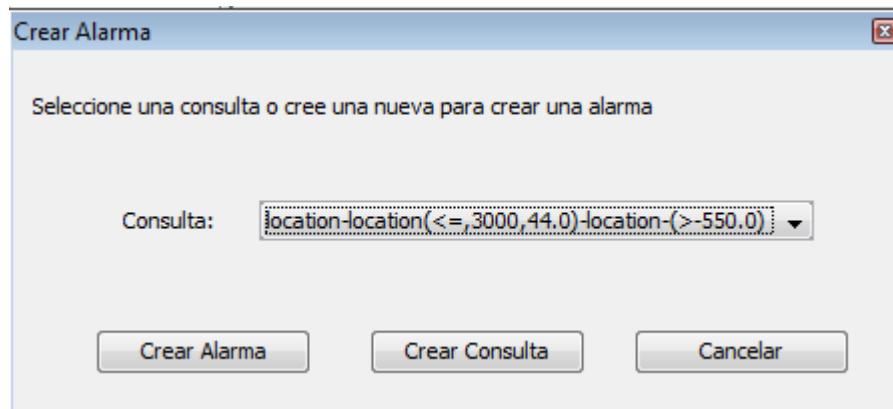


Figura A25 Crear Alarma

#### A5.6 Cancelar una alarma

Para dejar de monitorizar una alarma bastará con cancelarla, esta operación no cancelará la consulta a la que está asociada, con lo que se seguirán produciendo resultados, aunque estos no se notificarán directamente al usuario pudiéndose ver aún en la pantalla de monitorización de la consulta.

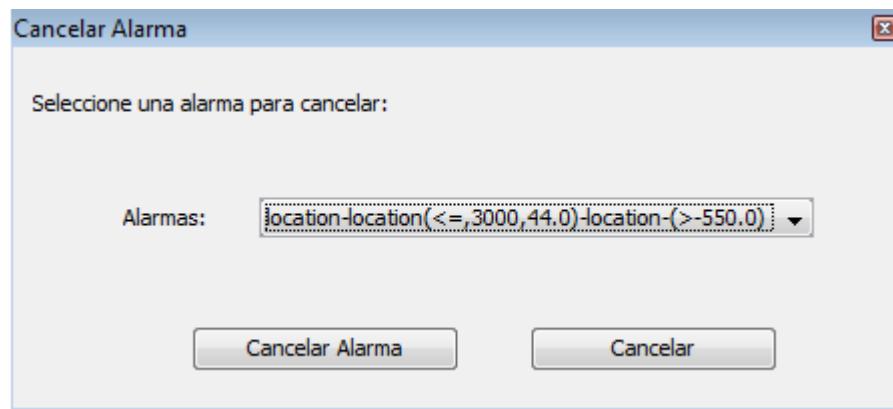


Figura A26 Cancelar Alarma

## A5.7 Funciones de Agregación

Las funciones de agregación que puede realizar la aplicación son:

*MEDIA*: Calcula la media de los datos almacenados en el buffer.

*DESVIACIÓN ESTÁNDAR*: Calcula la desviación estándar de los datos almacenados en buffer.

*TOTALES*: Calcula el número de tuplas almacenados en el buffer.

*MÍNIMO*: Calcula los valores mínimos almacenados en el buffer.

*MÁXIMO*: Calcula los valores máximos almacenados en el buffer.

Estos datos se pueden obtener tanto como de una clase de sensores en conjunto como de un sensor particular. Para ello bastará con abrir el menú contextual pulsando encima con el botón derecho de la clase del sensor o del sensor en el menú de la izquierda.

Una vez elegida la función deseada, si existen datos en el buffer, se mostrará el resultado actual de la operación en la siguiente pantalla:

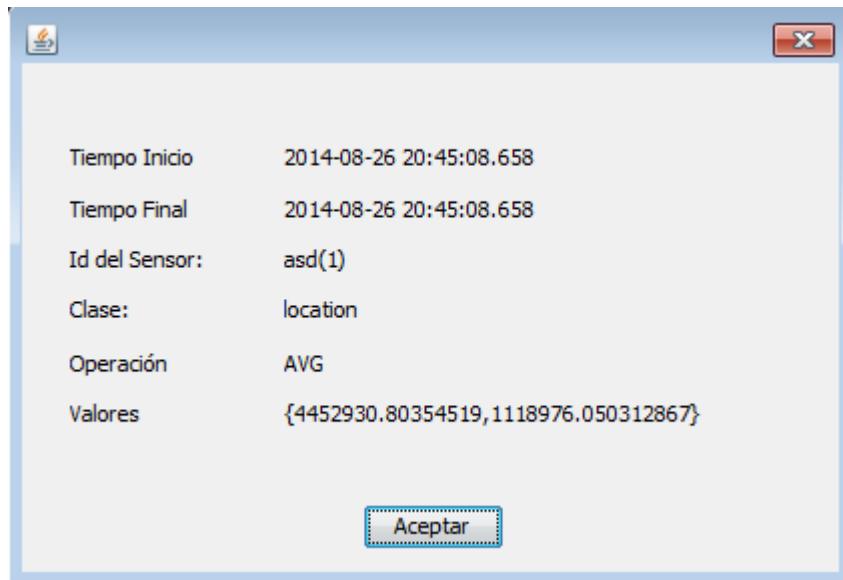
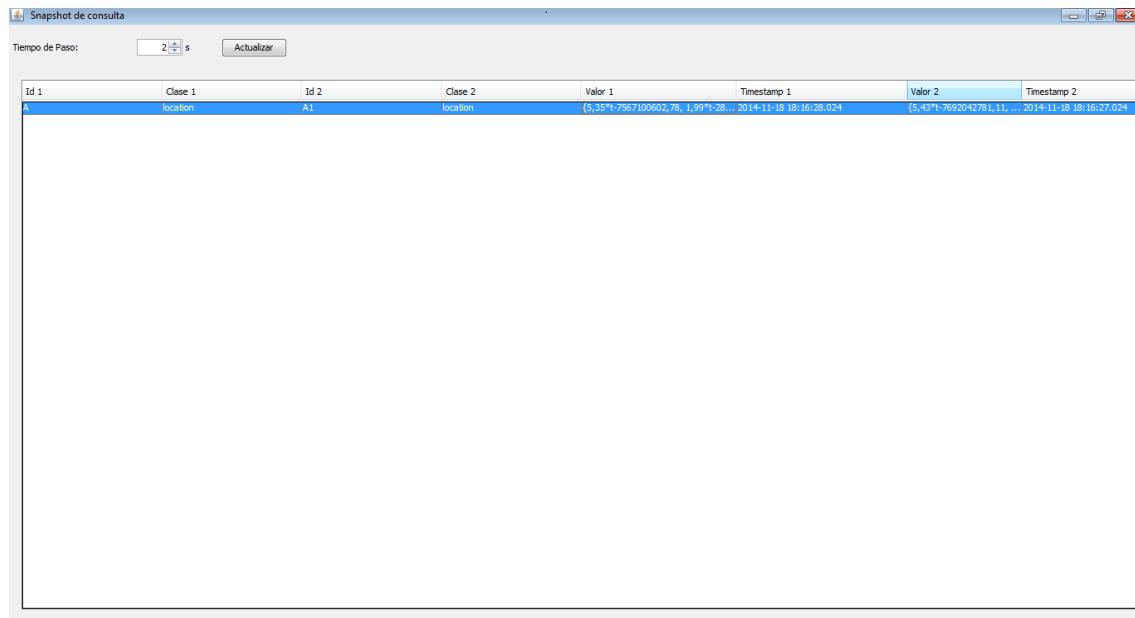


Figura A27 Resultado Agregación

## A5.8 Snapshots

Para poder visualizar un *snapshot* de una consulta que tenga resultados, bastará con pulsar el botón *Snapshot* de la pantalla de visualización de una consulta, y nos mostrará una pantalla como la siguiente.



The screenshot shows a window titled 'Snapshot de consulta'. At the top, there is a text input field 'Tiempo de Paso:' with the value '2' and a dropdown menu, followed by a button 'Actualizar'. The main area is a table with the following data:

Id 1	Clase 1	Id 2	Clase 2	Valor 1	Timestamp 1	Valor 2	Timestamp 2
A	location	A1	location	[5,35°t-7567100602,78, 1,99°t-28]	2014-11-18 18:16:28.024	[5,43°t-7692042781,11, ...]	2014-11-18 18:16:27.024

Figura A28 Visualización de Snapshot

En esta pantalla se irán mostrando las tuplas que se generen como respuesta en forma de flujo de datos, es decir, se irá actualizando para mostrar la respuesta activa en el instante actual de tiempo. Para cambiar el refresco bastará con cambiar el valor del tiempo de actualización y pulsar actualizar.

## A6 Configuración general de la aplicación

En esta sección se explicará la pantalla de configuración general de la aplicación. Cada pestaña recoge las opciones de cada uno de los aspectos del programa:

### A6.1 Predicción

En esta sección podremos determinar si la aplicación usa funciones de predicción o no, o si se mantiene toda la infraestructura, aunque no se vayan a utilizar.

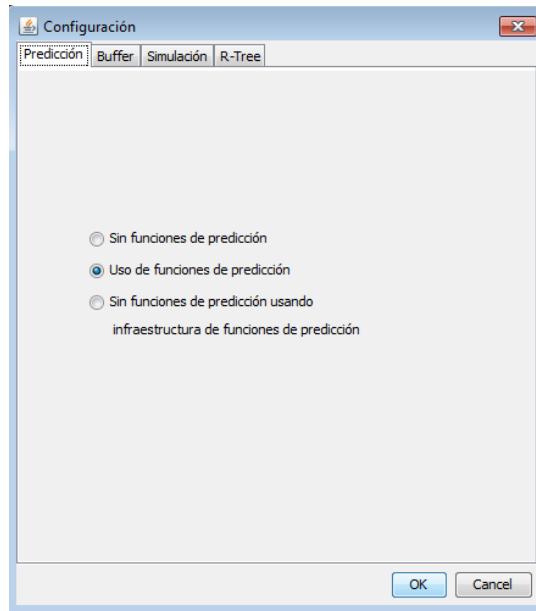


Figura A29 Configuración Predicción

### A6.2 Buffer

En esta sección se seleccionará la política que seguirá el buffer, esta puede ser:

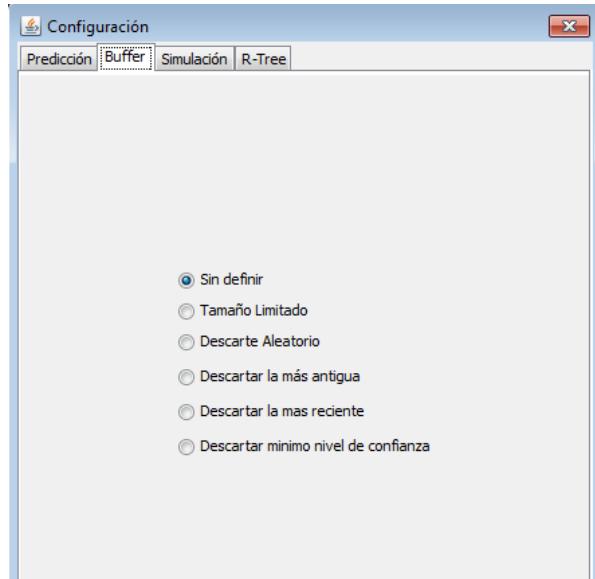


Figura A30 Configuración Buffer

*Sin definir:* No hay una política definida para descartar tuplas.

*Tamaño Limitado:* Se establece un tamaño límite para el buffer.

*Descarte Aleatorio:* Se realiza un sorteo aleatorio para determinar que tupla se descarta.

*Descartar la más antigua:* Descarta las tuplas más antiguas almacenadas en el sistema

*Descartar la más moderna:* Descarta la tupla más nueva almacenada en el buffer.

*Descartar la de menor contribución:* Cada tupla tiene una contribución al total de las consultas, se descartará aquella que tenga la menor contribución al global del sistema.

### A6.3 Simulación

En esta sección se establecen los parámetros globales de simulación (aunque el tiempo de lectura y el tiempo de actualización pueden modificarse en cada sensor). También es posible establecer si se producen retrasos, y la probabilidad de los mismos.

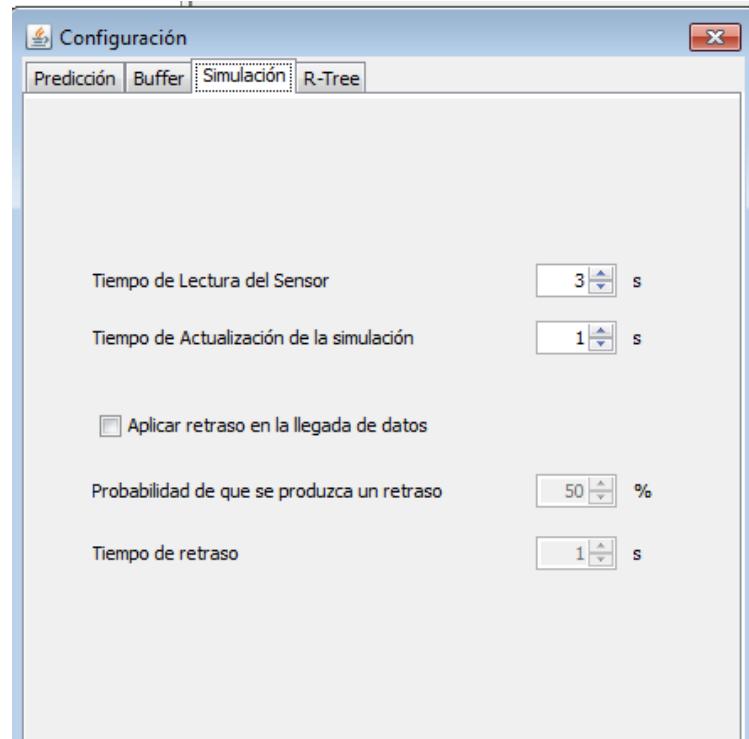


Figura A31 Configuración Sensor

#### A6.4 R-tree

Permite establecer si se usa R-tree y en el caso de que se utilice, que biblioteca se usa JSI o SPATIAL INDEX.

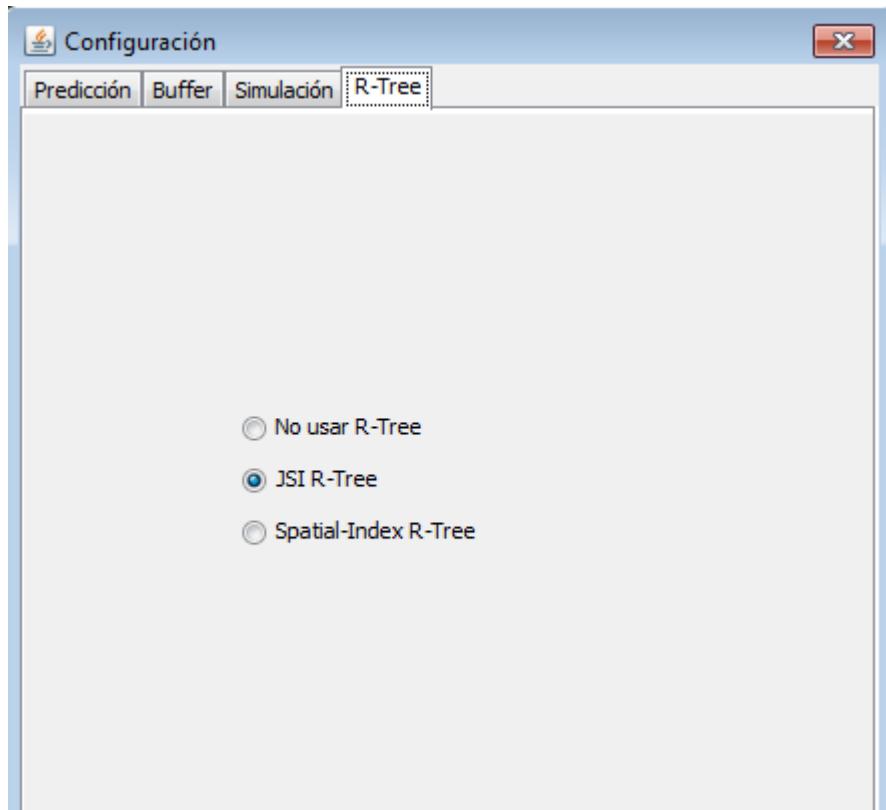


Figura A32 Configuración R-TREE

## A7 Tutorial

En este punto del manual de usuario se va a realizar un pequeño tutorial con el manejo básico de la aplicación para después, presentar un escenario real de análisis.

### A7.1 Manejo básico de la aplicación

En este punto del manual se va a realizar un pequeño ejemplo que recorra transversalmente el manejo de la aplicación. Se empezará creando una simulación de una red de sensores se definirán una par de consultas para visualizar resultados y se trabajará con ellos.

En primer lugar se va a definir una consulta con los siguientes parámetros. Para ello bastará con entrar al menú CONSULTAS y entrar a la opción CREAR CONSULTA. Se definirá con los siguientes parámetros:

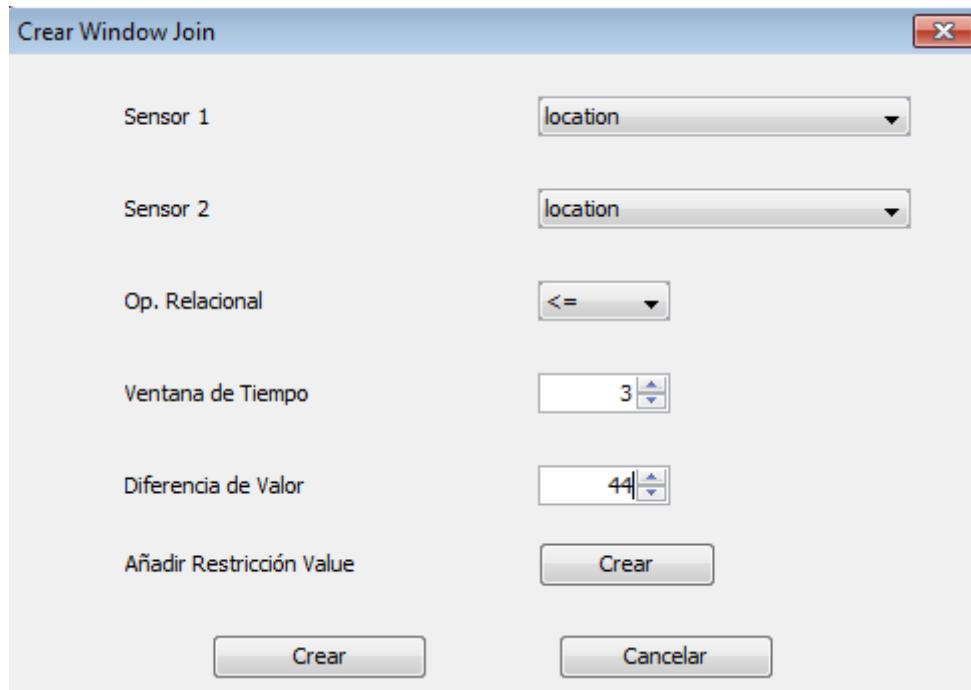


Figura A33 Parámetros de la consulta

En esta pantalla se define que se trata de una consulta que afecta a los tipos de sensor de “location” y que se activará cuando en una ventana de 3 segundos los elementos móviles tengan una distancia menor o igual que 44 metros.

Una vez definida la consulta, se procederá a definir una pareja de sensores dentro de una red de sensores GPS. Para ello solo habrá que ir al menú SIMULACIÓN, una vez dentro pulsar en SIMULADORES, RED DE SENsoRES y por último RED DE SENsoRES GPS. En esta pantalla habrá que pulsar el botón AÑADIR SENSOR para obtener la simulación de dos sensores. El primero de ellos, que simulará un coche tendrá la siguiente configuración:

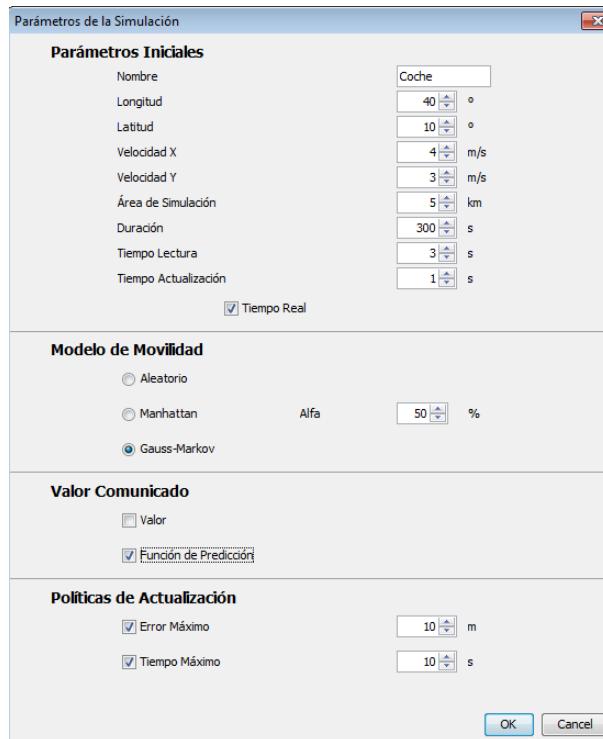


Figura A34 Configuración del sensor "COCHE"

Se trata de un sensor con función de predicción cuya política de actualización será por un error máximo de 10 metros y un periodo de validez de 10 segundos.

El segundo sensor tiene la siguiente configuración:

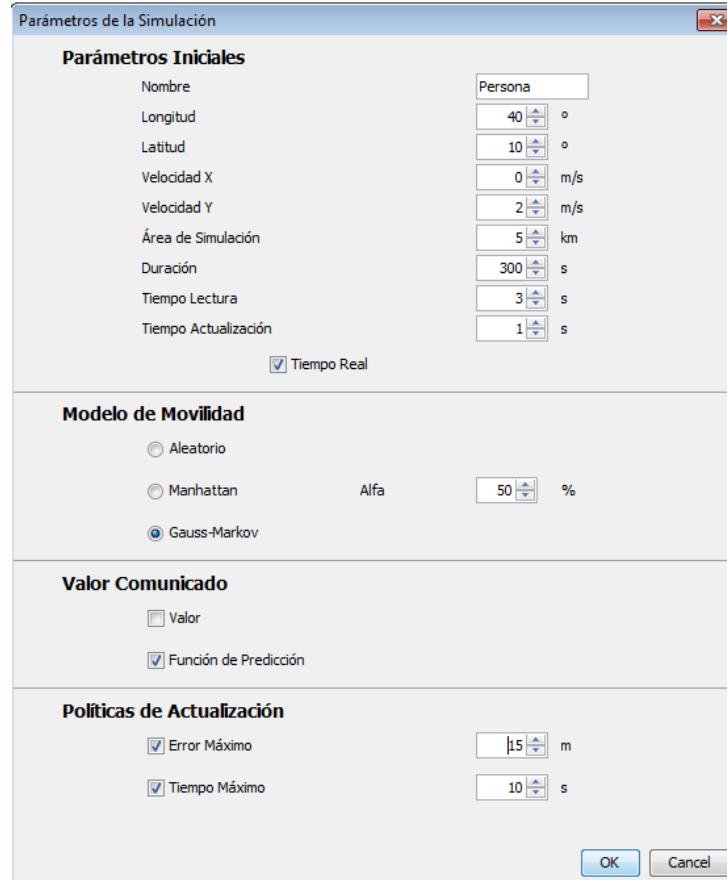


Figura A35 Configuración sensor "PERSONA"

En este caso se trata de un sensor que empieza en el mismo punto que el anterior, con una velocidad menor y con una política de actualización con un error máximo de 15 metros y un tiempo máximo de 10 s.

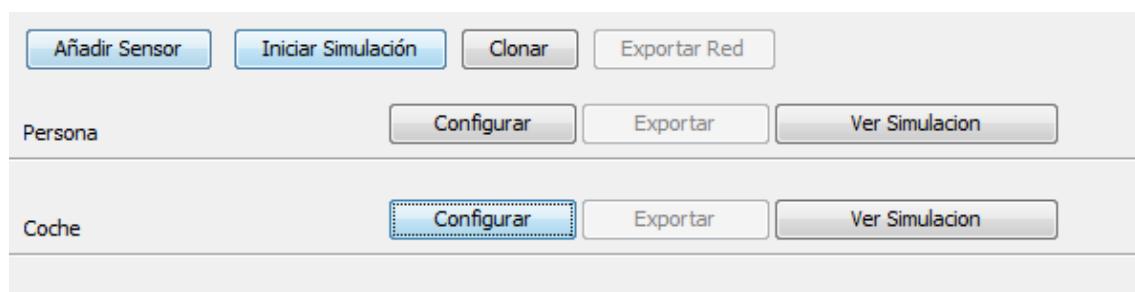


Figura A36 Estado de la pantalla de la red

Una vez terminado de definir los dos sensores, la pantalla de simulación de red tendrá la forma mostrada en la figura A37. Para iniciar la simulación bastará con pulsar el botón INICIAR SIMULACIÓN.

Se ha definido que ambos sensores son en tiempo real, luego pulsando el botón VER SIMULACIÓN se seguirá el transcurso de la simulación por pantalla:

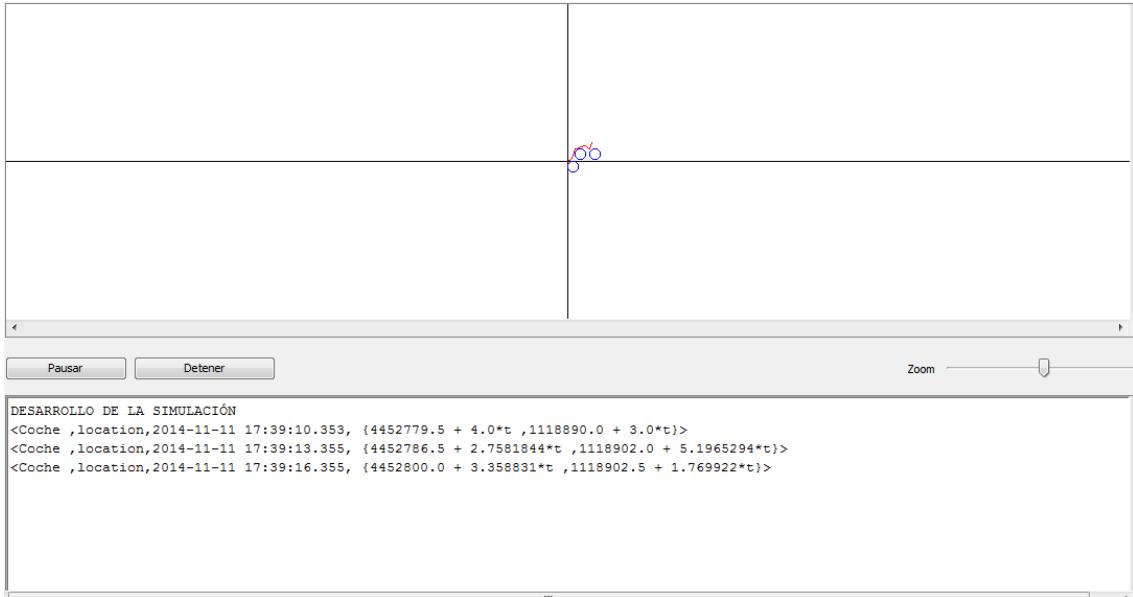


Figura A37 Simulación en tiempo real

En este punto la consulta ya habrá provocado resultados, para verlos habrá que hacer doble click en la consulta que aparece en el menú lateral de la aplicación.

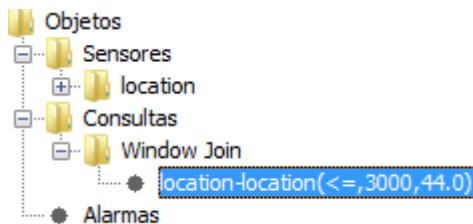


Figura A38 Menú lateral para acceder a la consulta

Una vez en la pantalla de visualización de la consulta aparecerán los resultados de la misma. En este punto se va a definir una alarma para que el sistema avise cada vez que se produzca un resultado. Para ello bastará pulsar el botón CREAR ALARMA.

Window Join	location
Sensor 1:	location
Sensor 2:	location
Operador Relacional:	<=
Ventana de Tiempo:	3000
Diferencia de Valor:	44.0
Propiedades	
Estado:	ACTIVA
Fecha de Creación:	Tue Nov 11 17:18:21 CET 2014
	<input type="button" value="Crear Alarma"/>
	<input type="button" value="Cancelar Consulta"/>
	<input type="button" value="Snapshot"/>

Id 1	Clase 1	Id 2	Clase 2	Prediccion 1	Timestamp 1	Prediccion 2	Timestamp 2	Marca
Coche	location	Persona	location	{4452779.5, 11188...}	2014-11-11 17:24:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..
Coche	location	Persona	location	{4452789.0, 11188...}	2014-11-11 17:24:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..
Coche	location	Persona	location	{4452797.0, 11188...}	2014-11-11 17:24:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..
Coche	location	Persona	location	{4452807.5, 11189...}	2014-11-11 17:24:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..
Coche	location	Persona	location	{4452807.0, 11189...}	2014-11-11 17:24:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..
Persona	location	Coche	location	{-1.500016450881...}	2014-11-11 17:24:...	{4452807.5, 11189...}	2014-11-11 17:24:...	ValidityMark -> Int..
Persona	location	Coche	location	{-1.500016450881...}	2014-11-11 17:24:...	{4452797.0, 11188...}	2014-11-11 17:24:...	ValidityMark -> Int..
Persona	location	Coche	location	{-1.500016450881...}	2014-11-11 17:24:...	{4452807.0, 11189...}	2014-11-11 17:24:...	ValidityMark -> Int..
Coche	location	Persona	location	{4452820.0, 11189...}	2014-11-11 17:25:...	{4452779.5, 2.0*t...}	2014-11-11 17:24:...	ValidityMark -> Int..

Figura A39 Resultado de Consulta

Para que el sistema avise habrá que salir de la pantalla de visualización de la consulta, para ello bastará con volver a la pantalla de visualización de red. Para ello habrá que realizar un click derecho con el ratón sobre uno de los dos sensores y pulsar la opción VER RED ASOCIADA. Una vez en este punto, si se produce un resultado en la consulta saltará un mensaje advirtiéndonos.

Si se quiere obtener algún valor estadístico de un sensor, bastará con realizar un click derecho con el ratón encima del nombre del sensor en el menú lateral y pulsar la función deseada.

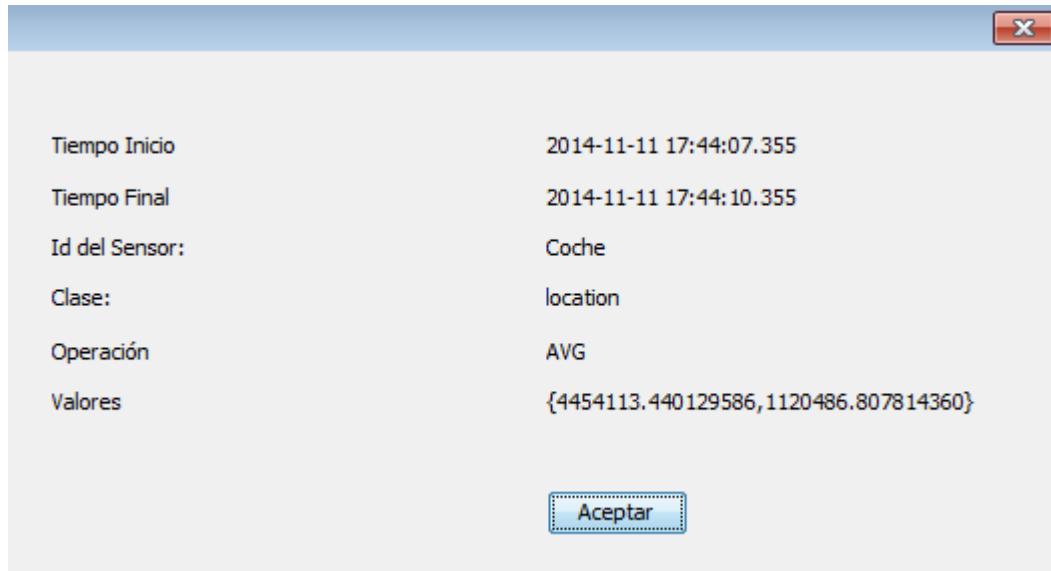


Figura A40 Estadístico

En este punto del tutorial, se va a realizar la exportación de datos de simulación y consulta (una vez terminada la simulación). En la pantalla de simulación de red habrá que pulsar la opción EXPORTAR RED y darle el nombre deseado, en este caso COCHEPERSONA.SIM

Durante la ejecución de la consulta, si se desea ver el valor del flujo de salida actual, bastará con pulsar el botón SNAPSHOT que proporcionara las tuplas en el momento que se produzca la salida, e indicará su validez en el momento actual.

Volviendo a la pantalla de resultado de la consulta, se podrá guardar para estudios futuros, bastará con acceder al menú CONSULTA y GUARDAR CONSULTA.

### A7.2 Ejemplo con escenario real

En este ejemplo se va a utilizar los ficheros de la plataforma Geolife [28], en concreto se van a utilizar 7 trazas reales recogidas durante una semana (una cada día) para responder a la siguiente pregunta: desde que se inicia el recorrido, ¿qué posiciones están a menos de un kilómetro dentro de una ventana temporal de una hora?

En primer lugar habrá que importar las trazas de los días que van del día 23 de octubre de 2008 a 29 de octubre (el día 25 no hay traza y el día 29 hay dos), a la aplicación de procesamiento. En todas ellas se configurará como nombre del sensor GEOLIFE y un número del 1 al 7.

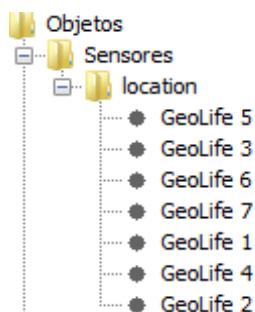


Figura A41 Árbol con los sensores importados

En cada importación, el sistema preguntará si se desea que se importe siguiendo un sistema temporal relativo. Habrá que hacerlo de este modo para poder comparar las trazas independientemente del día en que fueron obtenidas.

Para convertir las trazas reales a un sistema de funciones de predicción, se configurará de tal forma siguiendo los parámetros de la imagen:

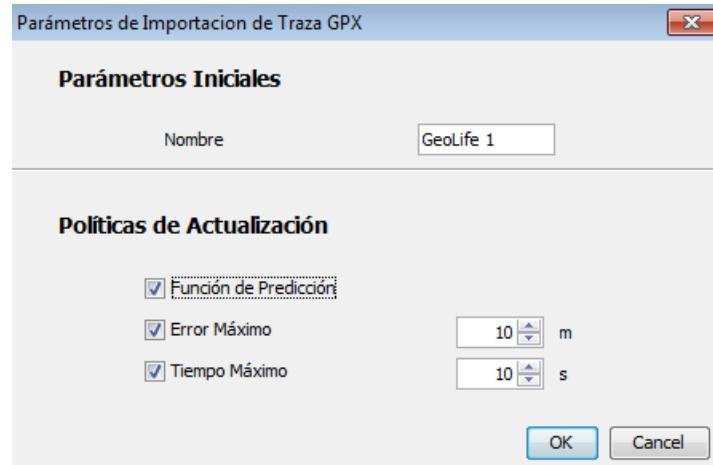


Figura A41 Parámetros de transformación

Esto permitirá a la aplicación calcular funciones de predicción para transformarlo al sistema de funciones de predicción de la aplicación.

Una vez realizado todas las importaciones, habrá que exportar la clase de sensores "location" como una red de sensores para tratarlos en su conjunto.

Con el fichero ya exportado, se procederá a crear la consulta. Los parámetros para responder a la pregunta sobre la que se basa este ejemplo serían los siguientes:

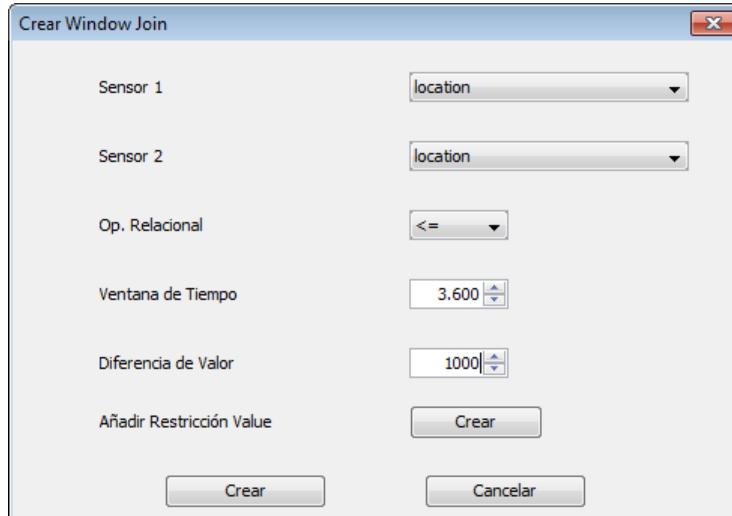


Figura A42 Parámetros de la consulta

Tras la definición de la consulta, habrá que importar el fichero resultado de exportar los ficheros de GeoLife como red de sensores.

Una vez importada la red, en la pantalla de visualización de la consulta se podrán observar los resultados obtenidos.

Id 1	Clase 1	Id 2	Clase 2	Prediccion 1	Timestamp 1	Prediccion 2	Timestamp 2	Marca
GeoLife 5	location	GeoLife 3	location	(1.0*t+1.294653...	1970-01-01 02:47...	{-1.0*t+1.29566...	1970-01-01 01:52...	ValidityMark -> In...
GeoLife 3	location	GeoLife 5	location	{-1.0*t+1.29566...	1970-01-01 01:52...	{1.0*t+1.294656...	1970-01-01 02:48...	ValidityMark -> In...
GeoLife 3	location	GeoLife 5	location	{-1.0*t+1.29566...	1970-01-01 01:52...	{1.0*t+1.294660...	1970-01-01 02:51...	ValidityMark -> In...

*Figura A43 Resultados del ejemplo*

Una vez obtenidos los resultados se podrá trabajar con ellos de la forma que se desee.

# ANEXO B: Formato de Ficheros

---

La aplicación puede leer 6 tipos de ficheros, 2 propios de la aplicación (.SIM, .DSM), 3 formatos predefinidos (.GPX, .PLT, .CSV), y un fichero XML que mantiene la configuración general de la aplicación.

## B1 Ficheros GPX

Los ficheros GPX (21) (GPS eXchange Format) son esquemas XML que contienen la información de una traza GPS. Estos ficheros son ampliamente utilizados por sistemas de almacenamiento de rutas en aplicaciones deportivas, turísticas y otras que utilicen geo posicionamiento.

Estos archivos pueden mantener información sobre puntos (waypoints), recorridos (tracks), y rutas (routes). En el contexto de la aplicación solo interesan los archivos que almacenen recorridos. En este caso el esquema que sigue el fichero es el siguiente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<gpx ...>
<!--Definicion de metadatos-->
<metadata> ... </metadata>
<!--Datos de un Track>
<trk>
  <trkseg>
    <trkpt lat="50.90110500" lon="5.652347000">
      <ele>69.092000</ele>
      <time>2010-01-01T00:00:00Z</time>
    </trkpt>
    <trkpt lat="50.90115400" lon="5.652281000">
      <ele>68.272000</ele>
      <time>2010-01-01T00:00:02Z</time>
    </trkpt>
    <trkpt lat="50.90118700" lon="5.652269000">
      <ele>68.962000</ele>
      <time>2010-01-01T00:00:03Z</time>
    </trkpt>  </trkseg>
  ...
</trk>
<!--Fin Fichero-->
</gpx>
```

En este ejemplo la ruta viene definida por el nodo `<trkseg>` dentro de `<trk>`. Dentro de este nodo se tiene la información de cada punto dentro del nodo `<trkpt>` cuyos atributos son la latitud (lat) y la longitud (lon) y tiene como nodos hijos la

elevación (ele) y el tiempo de paso por el punto (time). Para el contexto de la aplicación la elevación es indiferente.

## B2 Ficheros GeoLife

Geolife es una red social propiedad de Microsoft, creada con el objetivo de compartir experiencias y mantener conexiones entre personas a través del geoposicionamiento. El proyecto transcurrió durante los años 2007 a 2012 y generó 17,621 trayectorias diferentes recorriendo una cantidad de 1.2 millones de kilómetros con una duración aproximada de 48000 horas.

Estos ficheros tienen formato texto y siguen la siguiente estructura:

```
Geolife trajectory
WGS 84
Altitude is in Feet
Reserved 3
0,2,255,My Track,0,0,2,8421376
0
39.984702,116.318417,0,492,39744.1201851852,2008-10-23,02:53:04
39.984683,116.31845,0,492,39744.1202546296,2008-10-23,02:53:10
39.984686,116.318417,0,492,39744.1203125,2008-10-23,02:53:15
39.984688,116.318385,0,492,39744.1203703704,2008-10-23,02:53:20
...
...
```

Las primeros 5 líneas son informativas y contienen datos como el sistema de coordenadas usado por los ficheros, la unidad en la que se mide la altitud y demás información. A partir de estas líneas, cada línea contiene una posición con el siguiente formato:

*Campo 1:* Latitud en grados decimales.

*Campo 2:* Longitud en grados decimales.

*Campo 3:* 1 si hay discontinuidad, 0 en otro caso (no se utiliza en esta aplicación).

*Campo 4:* Altitud en el formato definido en las líneas superiores (si es -777 es un dato no válido).

*Campo 5:* Fecha en formato codificado.

*Campo 6:* Fecha de toma de datos.

*Campo 7:* Hora de toma de datos.

Como puede verse los campos 6 y 7 contienen el mismo dato codificado de diferente forma.

Leyendo línea a línea el fichero se obtendrá la traza deseada.

### B3 Ficheros CSV

Los ficheros .csv (ficheros de texto separados por comas que incluyen un registro en cada línea) son usados por la plataforma Movebank. Esta plataforma es una comunidad investigadora hospedada por el instituto Max Planck de Ornitología que contiene diversa información sobre trayectorias seguidas por animales. Estos ficheros tienen varios formatos dependiendo del tipo de sensor que están utilizando, solo interesarán aquellos que almacenen trazas GPS.

Estos archivos no tienen un formato fijo, En la primera fila contiene una cabecera con todos los datos que almacenan y solo nos interesará si almacena alguna columna con el dato lat (latitud), lon (longitud) y time (tiempo). Estos pueden aparecer en cualquier posición con lo cual la primera operación a realizar será obtener las columnas en las que se encuentran estos datos.

Una vez obtenido, bastará con leer línea a línea en la posición anteriormente calculada para obtener los datos longitud latitud y tiempo para reconstruir la traza.

Un ejemplo de los datos contenidos en un archivo de estas características seria:

```
event-id,visible,timestamp,location-long,location-lat,external-temperature,height-above-ellipsoid,sensor-type,individual-taxon-canonical-name,tag-local-identifier,individual-local-identifier,study-name,utm-easting,utm-northing,utm-zone,study-timezone,study-local-timestamp

329723310,true,2008-03-22 17:00:00.000,-2.1601667,15.653505,38.0,,"gps","Loxodonta africana","267","Salif Keita","Elliptical Time-Density Model (Wall et al. 2014) African Elephant Dataset (Source: Save the Elephants)",590011.0731667372,1730786.6485357105,"30N","Greenwich Mean Time",2008-03-22 17:00:00.000
```

Como se puede observar, en la primera fila aparecen todos los datos que almacena el archivo, siendo solo útiles para el programa timestamp, location-long y location-lat, además almacena información como la temperatura, identificación del animal, formato horario, altitud, tipo de sensor, taxón del animal etc.

## B4 Ficheros SIM

Los ficheros SIM son los ficheros que utiliza la aplicación para guardar una simulación de un sensor o red de sensores. Estos ficheros tienen formato XML y siguen el siguiente esquema:

```

<data streams>
  <datastream>
    <id>Person20</id>
    <type>location</type>
    <threshold>10.0</threshold>
    <timeReference>2014-08-20 20:45:11.931</timeReference>
    <info>
      <send>
        <time>1970-01-01 00:00:00.000</time>
        <function>4452779.5 + 4.0*t ,1118890.0 +
          3.0*t</function>
      </send>
      ...
    </info>
  </datastream>
  <datastream>
  ...
  </datastream>
</data streams>

```

Cada sensor está almacenado bajo el nodo *datastream*, dentro de él, el nodo *id* almacena el identificador del sensor, *type* almacena la clase del sensor , *threshold* almacena el umbral aplicado a la simulación y *timeReference* almacena el momento en el que se inician los datos. Después aparece el nodo *info* que contiene todas las comunicaciones que realiza el sensor, cada una de ellas dentro del nodo *send*, que consta de *time*, el tiempo relativo en el que se manda la información, y *function* que contiene el valor comunicado. La suma entre *timeReference* y *time* de cada información proporciona el valor real temporal en que fue enviado el dato.

Existirán tantos nodos *send* como comunicaciones haya realizado el sensor y tantos nodos *datastream* como sensores formen la red.

## B6 Ficheros DSM

Los ficheros DSM son los ficheros que almacenan los resultados de una consulta realizada, así como toda la información necesaria para recuperarla y visualizarla por pantalla. Estos ficheros siguen un esquema XML y tienen la siguiente descripción:

```

<outputstream>
    <windowjoin>
        <sensorClass1>location</sensorClass1>
        <sensorClass2>location</sensorClass2>
        <relOp>2</relOp>
        <w>3000</w>
        <valDif>44.0</valDif>
    </windowjoin>
    <value1>
        <sensorClass>location</sensorClass>
        <relOp>3</relOp>
        <valDif>550.0</valDif>
    </value1>
    <value2/>
    <time>Wed Aug 20 19:22:57 CEST 2014</time>
    <output>
        <tuple>
            <sensorID1>asdasd</sensorID1>
            <sensorID2>asd</sensorID2>
            <sensorClass1>location</sensorClass1>
            <sensorClass2>location</sensorClass2>
            <pf1>{4.0*t-5.629768787944E9, 3.0*t-4.224547285583E9}</pf1>
            <pf2>{4.0*t-5.629768787888E9, 3.0*t-
4.2245472855409994E9}</pf2>
            <time1>1408555391861</time1>
            <time2>1408555391847</time2>
            <validityMark>ValidityMark -&gt; Interval=Validity
interval: [1.408555391E9, 1.408555396E9], Timestamp-
matching Boundary=</validityMark>
        </tuple>
        ...
    </output>
</outputstream>

```

El archivo tiene de 2 a 4 nodos principales dependiendo de la consulta guardada. En primer lugar aparecerá el nodo *windowjoin* donde se encontrarán todos los parámetros de la consulta:

*SensorClass1*: Almacena el tipo de sensor 1 de la consulta.

*SensorClass2*: Almacena el tipo de sensor 2 de la consulta.

*relOp*: Almacena el operador relacional codificado con un número.

*W*: almacena el valor de la ventana temporal de la consulta.

*Valdif*: Almacena el valor de diferencia de la consulta.

Después en el caso de que se almacene una restricción de valor, esta aparecerá en su lugar correspondiente (si en la consulta aparece en primer lugar o segundo lo almacenará en el nodo correspondiente). La definición de los valores es idéntica a sus equivalentes en *window join*.

Tras esto aparece el nodo <output> que contienen las tuplas (<tuples>) que han surgido como respuesta a la consulta durante la monitorización:

*sensorID1*: Id del primer sensor.

*sensorClass1*: clase del primer sensor.

*sensorID2*: id del segundo sensor.

*sensorClass2*: clase del segundo sensor.

*pf1*: función de predicción o valor del primer dato que cumple la consulta.

*pd2*: función de predicción o valor del segundo sensor que cumple la consulta.

*time1*: valor de tiempo que hace cumplir la consulta del sensor 1.

*time2*: valor de tiempo que hace cumplir la consulta del sensor 2.

*ValiditiMark*: Información que contiene la marca de validez de la consulta.

## B2 Fichero de configuración

El fichero de configuración (config.xml) es donde se guardan todas las opciones de la configuración general de la aplicación para que esta sea persistente de una ejecución a otra, este fichero se trata de un fichero XML donde cada nodo representa una opción del menú de configuración y su formato es el siguiente:

```
<config>
    <sensorTime>1</sensorTime>
    <simulatorTime>3</simulatorTime>
    <bufferPolicy>0</bufferPolicy>
    <simulateDelay>false</simulateDelay>
    <probDelay>50.0</probDelay>
    <maxDelay>1</maxDelay>
    <R-tree>1</R-tree>
    <prediction>1</prediction>
</config>
```

*sensorTime*: Tiempo de lectura del sensor.

*simulatorTime*: tiempo de actualización del sensor.

*bufferPolicy*: Guarda la política de buffer a aplicar en la monitorización.

*simulateDelay*: indica si se aplica retardo a las tuplas.

*probDelay*: Indica la probabilidad de que sufra un retardo si *simulateDelay* es true.

*maxDelay*: Indica el máximo tiempo de Delay admitido por la aplicación.

*R-tree*: Indica la estructura de almacenamiento de las tuplas.

*Prediction*: indica el uso o no de funciones de predicción.

Todas estas opciones indican la opción seleccionada en el menú de configuración tal y como se ha mostrado en *A6 Configuración general de la aplicación*.



# ANEXO C: Análisis del sistema

---

En este anexo se va a presentar el análisis de sistema. Se van a enumerar los requisitos de la aplicación, los casos de uso, el prototipado y el diagrama de navegación de ventanas.

## C1 Requisitos de la Aplicación

**Identificador:** RF1.

**Descripción:** La aplicación deberá realizar simulaciones para incorporarlas al sistema de procesamiento. Estas simulaciones serán de tipo GPS, cotizaciones y un simulador genérico para simular otros tipos.

**Tipo:** Funcional.

**Identificador:** RF2.

**Descripción:** En la simulación GPS se deberá poder aplicar modelos de movilidad, tales como Manhattan y Gauss-Markov para realizar diferentes simulaciones.

**Tipo:** Funcional.

**Identificador:** RF3.

**Descripción:** En la simulación de aplicaciones se aplicará el modelo de Black-Scholes aplicado a la simulación informática.

**Tipo:** Funcional.

**Identificador:** RF4.

**Descripción:** En las simulaciones se podrá elegir la política de comunicación deseada, ya sea un valor o una función de predicción.

**Tipo:** Funcional.

**Identificador:** RF5.

**Descripción:** En la simulación se deberá poder elegir que política de actualización seguir en el sensor, ya sea por error máximo o por tiempo de la última comunicación.

**Tipo:** Funcional.

**Identificador:** RF6.

**Descripción:** Deberá haber un tipo genérico de simulación, basado en distribuciones estadísticas para poder simular diferentes fenómenos no contemplados en los dos tipos anteriores.

**Tipo:** Funcional.

**Identificador:** RF7.

**Descripción:** Se deberán poder realizar simulaciones de redes de sensores de los tipos anteriores.

**Tipo:** Funcional.

**Identificador:** RF8

**Descripción:** Se podrá guardar una simulación, ya sea de un sensor o de una red.

**Tipo:** Funcional.

**Identificador:** RF9.

**Descripción:** Se deberá poder importar una simulación guardada anteriormente.

**Tipo:** Funcional.

**Identificador:** RF10.

**Descripción:** La aplicación deberá importar trazas GPS de diversos formatos (GPX, PLT y CSV) para su posterior tratamiento para adaptarlo al sistema de simulación de la aplicación.

**Tipo:** Funcional.

**Identificador:** RF10.

**Descripción:** Se deberá poder visualizar una simulación, tanto gráficamente como las tuplas obtenidas que se comunicarán al sistema de procesamiento.

**Tipo:** Funcional.

**Identificador:** RF11.

**Descripción:** La simulaciones se podrán realizar tanto en tiempo real, como obtener una simulación inmediatamente.

**Tipo:** Funcional.

**Identificador:** RF12

**Descripción:** En el caso de una simulación en tiempo real, ésta se podrá pausar y detener a elección del usuario.

**Tipo:** Funcional.

**Identificador:** RF13.

**Descripción:** Los sensores de una misma clase, aunque hayan sido simulados de forma independiente, podrán guardarse como si de una red se tratase.

**Tipo:** Funcional.

**Identificador:** RF14.

**Descripción:** La interfaz gráfica deberá integrar la herramienta de simulación con el sistema de procesamiento, siendo lo más homogéneo posible.

**Tipo:** Funcional.

**Identificador:** RF15.

**Descripción:** La aplicación mostrará en todo momento todos los objetos que están creados en ella (denominamos objeto a un sensor, una consulta o una alarma) para poder consultarlos en cualquier momento, en forma de árbol desplegable.

**Tipo:** Funcional.

**Identificador:** RF16.

**Descripción:** Deberá haber una pantalla para crear consultas *Window Join y Value* de forma sencilla e integrada con la interfaz.

**Tipo:** Funcional.

**Identificador:** RF17.

**Descripción:** Se podrán visualizar en tiempo real el resultado de las consultas conforme se vayan produciendo.

**Tipo:** Funcional.

**Identificador:** RF18.

**Descripción:** Los resultados de una consulta deberán ser guardados en fichero para poder ser recuperados posteriormente.

**Tipo:** Funcional.

**Identificador:** RF19.

**Descripción:** La aplicación tendrá un sistema de alarmas que monitorice una consulta y avise al usuario inmediatamente cuando se produzca algún resultado.

**Tipo:** Funcional.

**Identificador:** RF20.

**Descripción:** Tanto las consultas como las alarmas se deberán poder cancelar para no producir más resultados.

**Tipo:** Funcional.

**Identificador:** RF21.

**Descripción:** Se ampliará el número de operaciones existentes con operaciones de agregación que constará de media (AVG), desviación típica (DESVEST), número de tuplas (COUNT), valor máximo (MAX) y valor mínimo (MIN). Estos valores se podrán aplicar tanto a una clase de sensores como a un sensor en concreto.

**Tipo:** Funcional.

**Identificador:** RF22.

**Descripción:** Todas las operaciones posibles deberán poder hacerse desde el menú superior de la pantalla, salvo las agregaciones que se harán por un menú contextual con el clic derecho.

**Tipo:** Funcional.

**Identificador:** RF23.

**Descripción:** El usuario podrá clonar la configuración de un sensor para poder repetir la configuración del sensor en N sensores.

**Tipo:** Funcional.

**Identificador:** RF24.

**Descripción:** Los datos generados por el simulador se comunicaran al sistema de procesamiento para ser tratados como información procedente de sensores.

**Tipo:** Funcional.

**Identificador:** RF25.

**Descripción:** Se podrá visualizar el estado concreto de una consulta para diferentes valores temporales.

**Tipo:** Funcional.

**Identificador:** RNF1.

**Descripción:** La aplicación se desarrollará en el lenguaje Java.

**Tipo:** No Funcional.

**Identificador:** RNF2.

**Descripción:** La aplicación tendrá un fichero de configuración (denominado config.xml) que estará en la carpeta raíz de la aplicación y permitirá dar persistencia de una ejecución a otra de la configuración básica de la aplicación.

**Tipo:** No Funcional.

**Identificador:** RNF3.

**Descripción:** La aplicación usará la biblioteca gráfica Swing.

**Tipo:** No Funcional.

**Identificador:** RNF4.

**Descripción:** La aplicación usará la biblioteca COLT para la obtención de números aleatorios siguiendo distribuciones estadísticas.

**Tipo:** No Funcional.

**Identificador:** RNF5.

**Descripción:** El usuario tendrá un asistente para poder elegir los archivos a importar.

**Tipo:** No Funcional.

**Identificador:** RNF6.

**Descripción:** El usuario podrá consultar un manual de usuario durante la ejecución.

**Tipo:** No Funcional.

## C2 Casos de Uso

En el siguiente apartado se van a enumerar los casos de uso de la aplicación. En este caso se va a suponer la existencia de un único actor con nivel medio de carácter usuario o investigador.

### C2.1 Casos de uso de simulación

**Nombre:** Realizar simulación GPS.

**Descripción:** El usuario aportará por pantalla mediante un diálogo una serie de parámetros tales como latitud, longitud, velocidad inicial, duración de la simulación, modelo de movilidad, política de comunicación y política de actualización para realizar una simulación en tiempo real o de forma instantánea.

**Nombre:** Realizar simulación de cotizaciones.

**Descripción:** El usuario aportará por pantalla mediante un diálogo una serie de parámetros tales como tasa de beneficio, volatilidad, precio inicial, duración de la simulación, política de comunicación y política de actualización para realizar una simulación en tiempo real o de forma instantánea.

**Nombre:** Realizar simulación genérica.

**Descripción:** El usuario aportará por pantalla mediante un diálogo una serie de parámetros tales como valor inicial, distribución de probabilidad, valores propios de la distribución, duración de la simulación, modelo de movilidad, política de comunicación y política de actualización para realizar una simulación en tiempo real o de forma instantánea.

**Nombre:** Realizar simulación de redes de sensores.

**Descripción:** El usuario podrá realizar una simulación de redes de sensores.

**Nombre:** Exportar simulación.

**Descripción:** Una vez realizada la simulación, el usuario podrá elegir exportarla a un fichero, eligiendo el nombre y ubicación del mismo

**Nombre:** Importar simulación.

**Descripción:** Una vez realizada la simulación, el usuario podrá elegir exportarla a un fichero, eligiendo el nombre y ubicación del mismo

**Nombre:** Zoom en Simulación.

**Descripción:** En la gráfica de simulación el usuario podrá realizar zoom en la gráfica para poder ver los puntos de forma más clara.

**Nombre:** Pausar/Detener Simulación

**Descripción:** Una simulación en tiempo real debe poder ser parada y detenida por el usuario.

## C2.2 Casos de uso de procesamiento

**Nombre:** Crear Consulta Windows Join.

**Descripción:** El usuario podrá crear una consulta Windows Join , proporcionando la clase de sensores asociadas, la ventana temporal, el operador relacional y el valor a comprobar.

**Nombre:** Crear Consulta Value.

**Descripción:** El usuario podrá crear una consulta Valué. Asociada a una Windows Join, proporcionando la clase del sensor asociadas, el operador relacional y el valor a comprobar.

**Nombre:** Realizar Operación de Agregación.

**Descripción:** El usuario podrá realizar una operación de agregación consultando el valor de la media, desviación típica, máximo y mínimo de un sensor o una clase de sensores en su conjunto.

**Nombre:** Visualizar resultados de una consulta.

**Descripción:** Una consulta en ejecución deberá mostrar los resultados por pantalla, conforme se van produciendo cuando el usuario así lo solicite y además podréseles aplicar valores temporales para determinar su valor concreto.

**Nombre:** Crear una alarma sobre una consulta existente.

**Descripción:** Una consulta en ejecución deberá podérsele asociar una alarma para que en el caso de que se produzcan resultados se avise inmediatamente al usuario, pudiendo comprobar los resultados de la misma.

**Nombre:** Cancelar alarma / consulta.

**Descripción:** Se podrá cancelar una alarma o consulta a petición del usuario.

**Nombre:** Guardar resultados de Consulta.

**Descripción:** El usuario podrá guardar los resultados de una consulta en un fichero para posteriormente recuperarlos.

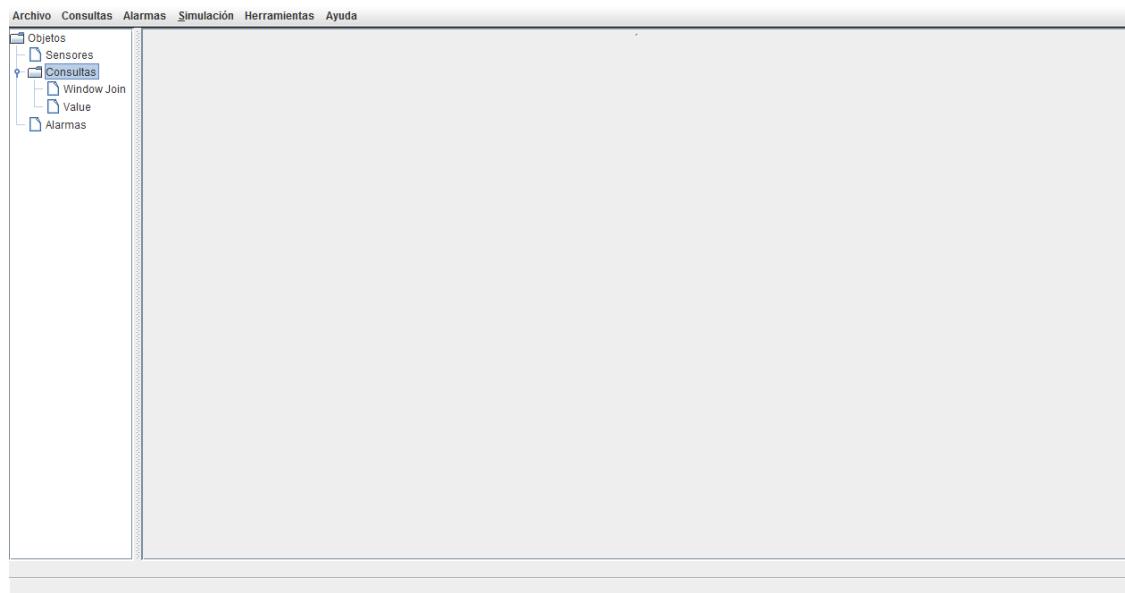
**Nombre:** Importar consulta.

**Descripción:** Una consulta guardada con anterioridad se podrá visualizar en la aplicación.

### C3 Prototipado de Ventanas

En este punto se va a presentar el diseño prototipado de las ventanas de la interfaz gráfica que se implementará la aplicación. Cada pantalla está numerada para que en el diagrama de navegación del punto siguiente sea más fácil su identificación.

## Pantalla Principal (1)



## Configuración GPS (2)

**Parámetros Iniciales**

Nombre	<input type="text"/>
Longitud	<input type="text"/> 40 °
Latitud	<input type="text"/> 10 °
Velocidad X	<input type="text"/> 4 m/s
Velocidad Y	<input type="text"/> 3 m/s
Área de Simulación	<input type="text"/> 5 km
Duración	<input type="text"/> 300 s
Tiempo Lectura	<input type="text"/> 3 s
Tiempo Actualización	<input type="text"/> 1 s

Tiempo Real

**Modelo de Movilidad**

Aleatorio

Manhattan

Gauss-Markov

Alfa  50 %

**Valor Comunicado**

Valor

Función de Predicción

**Políticas de Actualización**

Error Máximo  10 m

Tiempo Máximo  10 s

OK  Cancel

## Configuración Cotizaciones (3)

**Parámetros Iniciales**

Nombre	<input type="text"/>	
Precio Inicial	<input type="text"/> 100	
Tasa de Beneficio	<input type="text"/> 0,05	U.M.
Duración	<input type="text"/> 300	s
Tiempo Lectura	<input type="text"/> 3	s
Tiempo Actualización	<input type="text"/> 1	s

Tiempo Real

---

**Modelo de Black-Scholes**

Volatilidad	<input type="text"/> 0,03
-------------	---------------------------

---

**Valor Comunicado**

Valor

Función de Predicción

---

**Políticas de Actualización**

Error Máximo	<input type="text"/> 10	U.M.
Tiempo Máximo	<input type="text"/> 10	s

**OK** **Cancel**

## Configuración Genérica (4)

**Parámetros Iniciales**

Nombre	<input type="text"/>	
Tipo	<input type="text"/>	
Valor Inicial	<input type="text"/> 100	
Duración	<input type="text"/> 300	s
Tiempo Lectura	<input type="text"/> 3	s
Tiempo Actualización	<input type="text"/> 1	s

Tiempo Real

---

**Selección de la distribución**

Distribución	<input type="text" value="Beta"/>
Alfa:	<input type="text"/> 0
Beta:	<input type="text"/> 0

---

**Valor Comunicado**

Valor

Función de Predicción

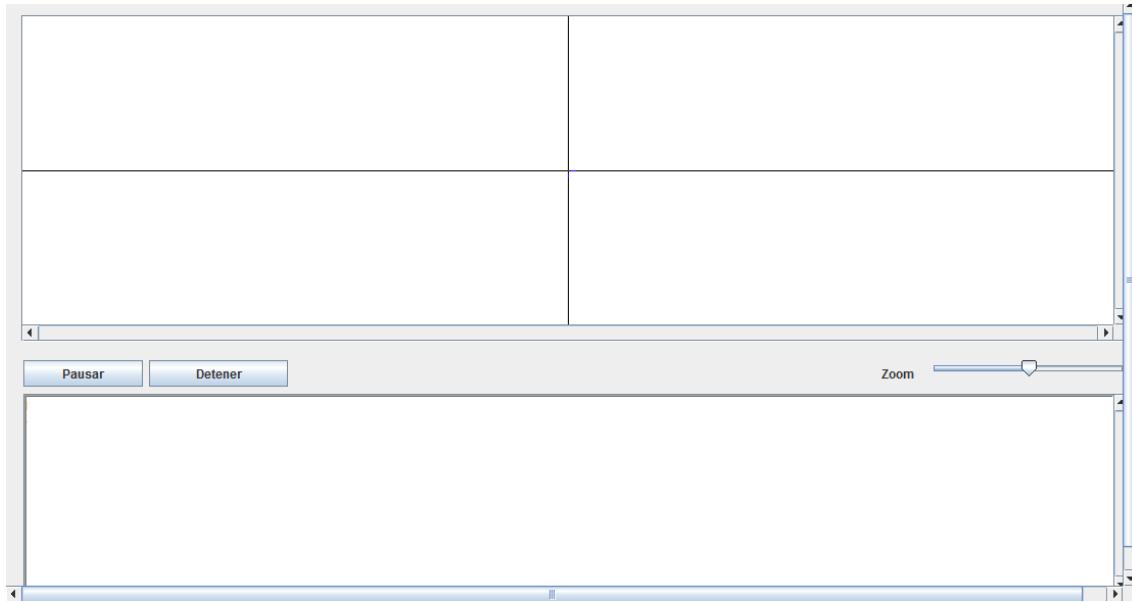
---

**Políticas de Actualización**

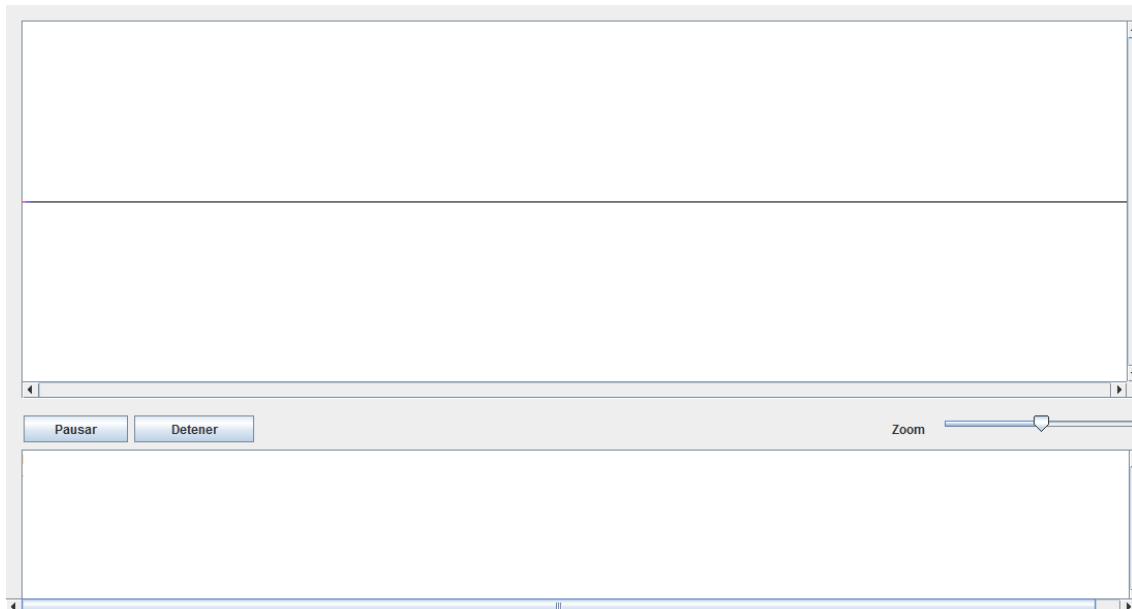
Error Máximo	<input type="text"/> 10	
Tiempo Máximo	<input type="text"/> 10	s

**OK** **Cancel**

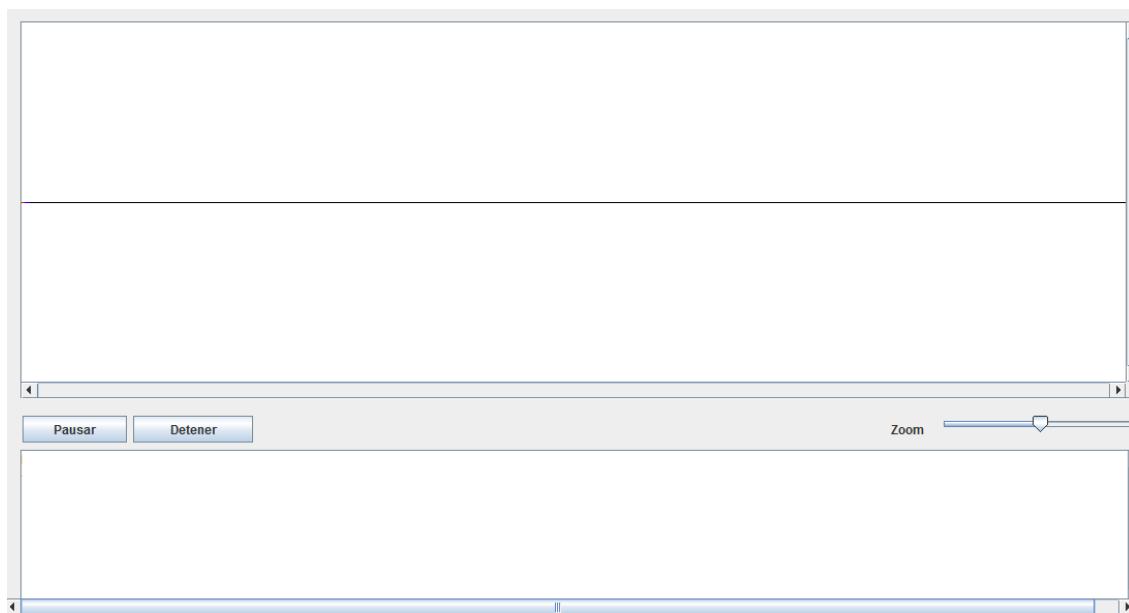
Simulación GPS (5)



Simulación Cotizaciones (6)



Simulación Genérica (7)



Red de Sensores (8)



## Creación de Consulta (9)

Sensor 1	<input type="text" value="location"/> ▼
Sensor 2	<input type="text" value="location"/> ▼
Op. Relacional	= ▼
Ventana de Tiempo	<input type="text" value="0"/> ▲ ▼
Diferencia de Valor	<input type="text" value="0"/> ▲ ▼
Añadir Restricción Value	<input type="button" value="Crear"/>
<input type="button" value="Crear"/>	<input type="button" value="Cancelar"/>

## Añadir Value (10)

Sensor	<input type="text"/> ▼
Op. Relacional	<input type="text"/> ▼
Diferencia de Valor	<input type="text"/> ▲ ▼
<input type="button" value="Crear"/>	<input type="button" value="Cancelar"/>

## Visualización de consulta (11)

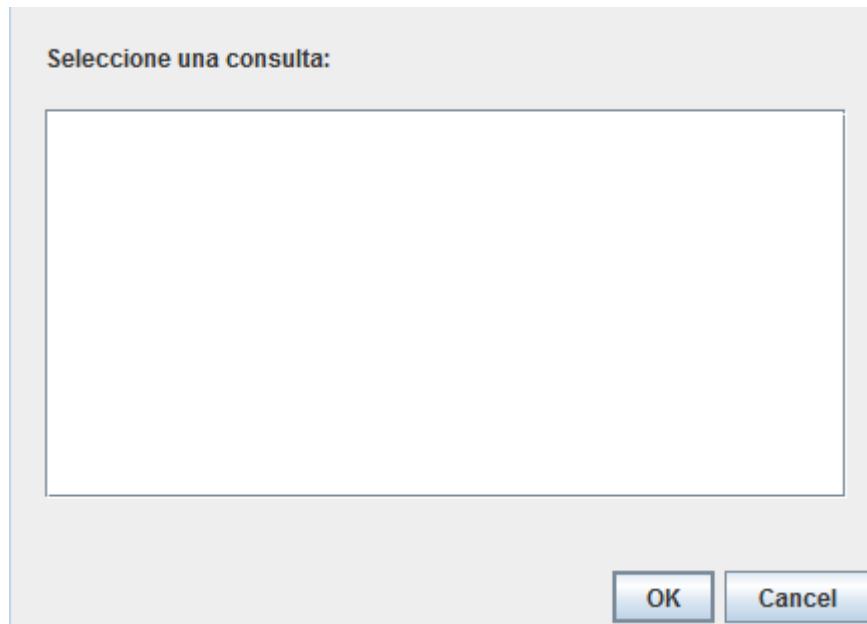
Window Join	Value																		
Sensor 1:	New label																		
Sensor 2:	New label																		
Operador Relacional:	New label																		
Ventana de Tiempo:	New label																		
Diferencia de Valor:	New label																		
<b>Propiedades</b>																			
Estado:	ACTIVA																		
Fecha de Creación:	<dynamic>																		
<b>Buttons:</b>																			
<b>Crear Alarma</b>																			
<b>Cancelar Consulta</b>																			
<b>Snapshot</b>																			
<table border="1"><thead><tr><th>Id 1</th><th>Clase 1</th><th>Id 2</th><th>Clase 2</th><th>Predictor 1</th><th>Timestamp 1</th><th>Predictor 2</th><th>Timestamp 2</th><th>Marca</th></tr></thead><tbody><tr><td colspan="9"> </td></tr></tbody></table>		Id 1	Clase 1	Id 2	Clase 2	Predictor 1	Timestamp 1	Predictor 2	Timestamp 2	Marca									
Id 1	Clase 1	Id 2	Clase 2	Predictor 1	Timestamp 1	Predictor 2	Timestamp 2	Marca											

## Crear Alarma (12)

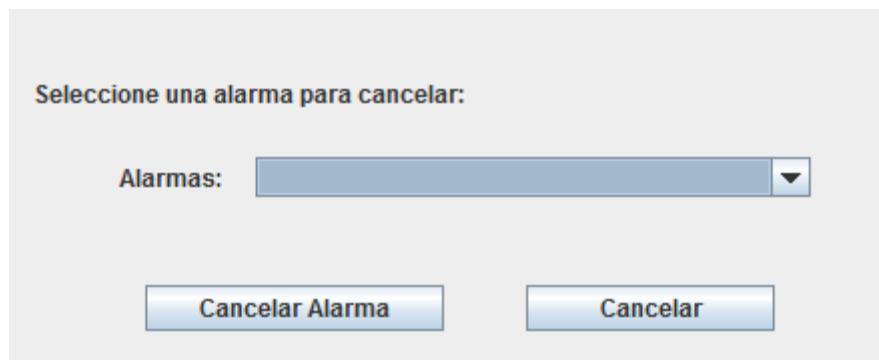
Seleccione una consulta o cree una nueva para crear una alarma:

Consulta:

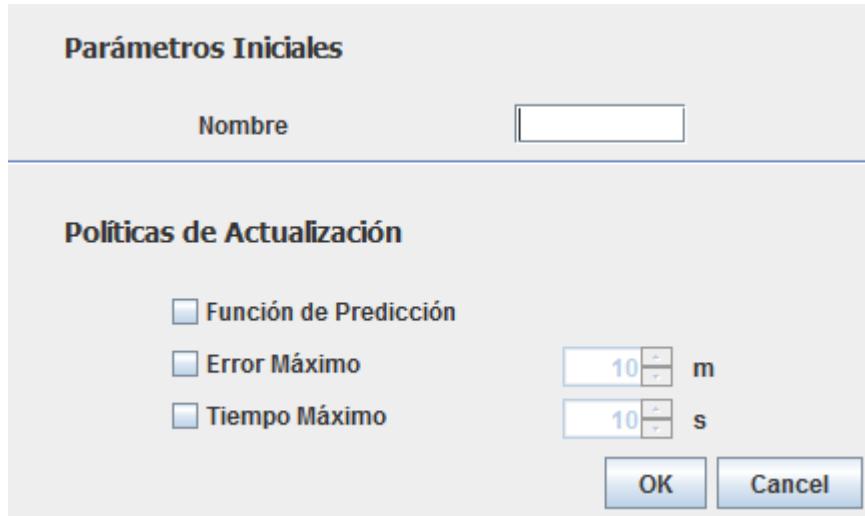
Ver Listado de Consultas (13)



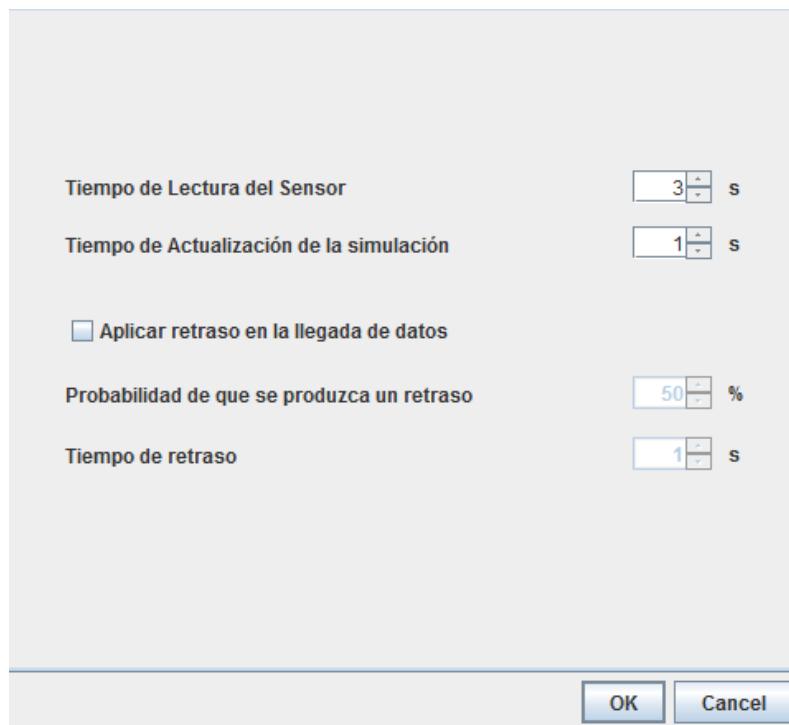
Cancelar Alarma (14)



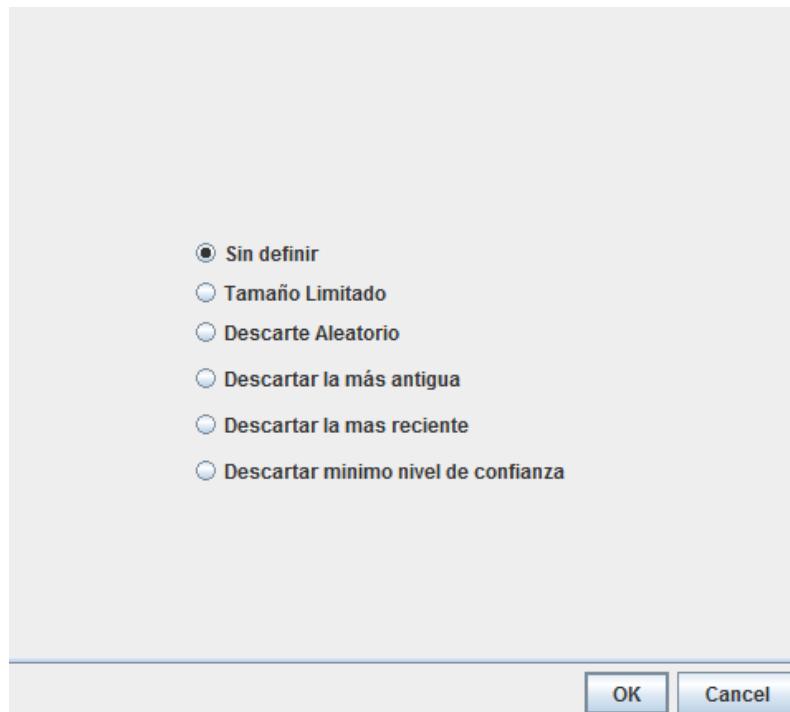
## Importar traza GPS (16)



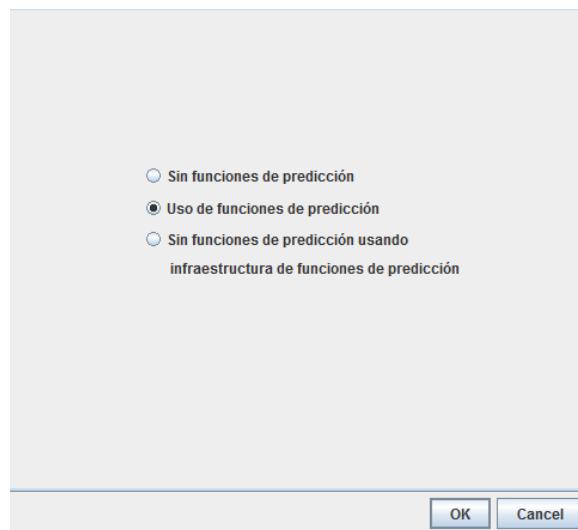
## Configuración – Simulación (17)



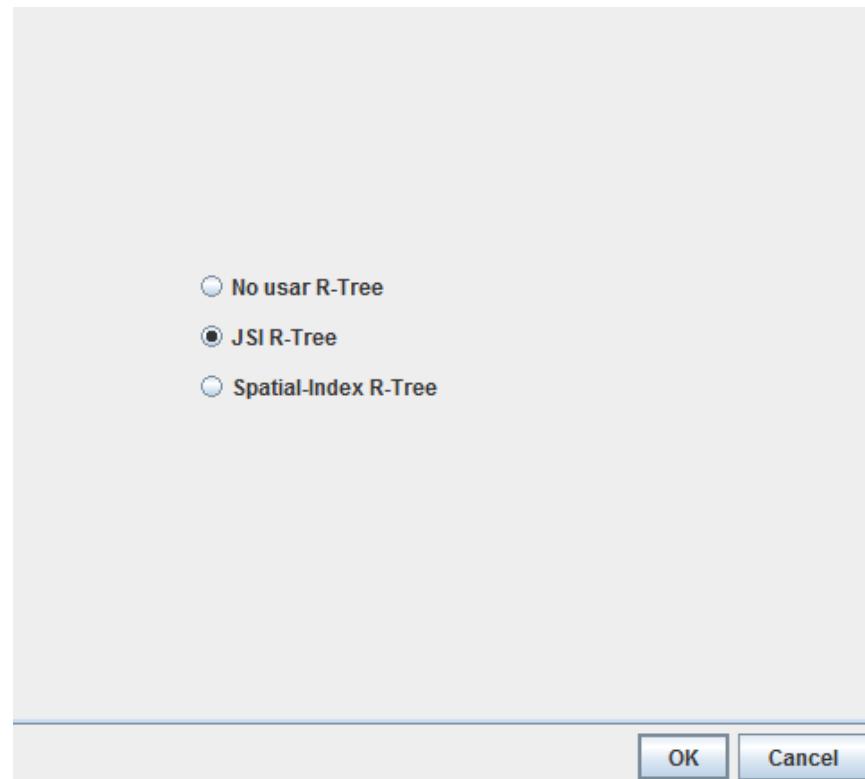
## Configuración – Buffer (18)



## Configuración –Funciones de Predicción (19)



## Configuración - R-tree (20)



## Visualización de un Snapshot (21)

Tiempo de Paso: <input type="text" value="2"/> s <input type="button" value="Actualizar"/>							
Id 1	Clase 1	Id 2	Clase 2	Timestamp 1	Valor 1	Timestamp 2	Valor 2

## C4 Navegación de Ventanas

En la página siguiente se encuentra un diagrama que muestra el flujo de navegación de las ventanas de la aplicación. En este diagrama se muestra la dirección del sentido de la navegación, así como el nombre y número de la pantalla según el punto anterior para que puedan distinguirse entre ellas.

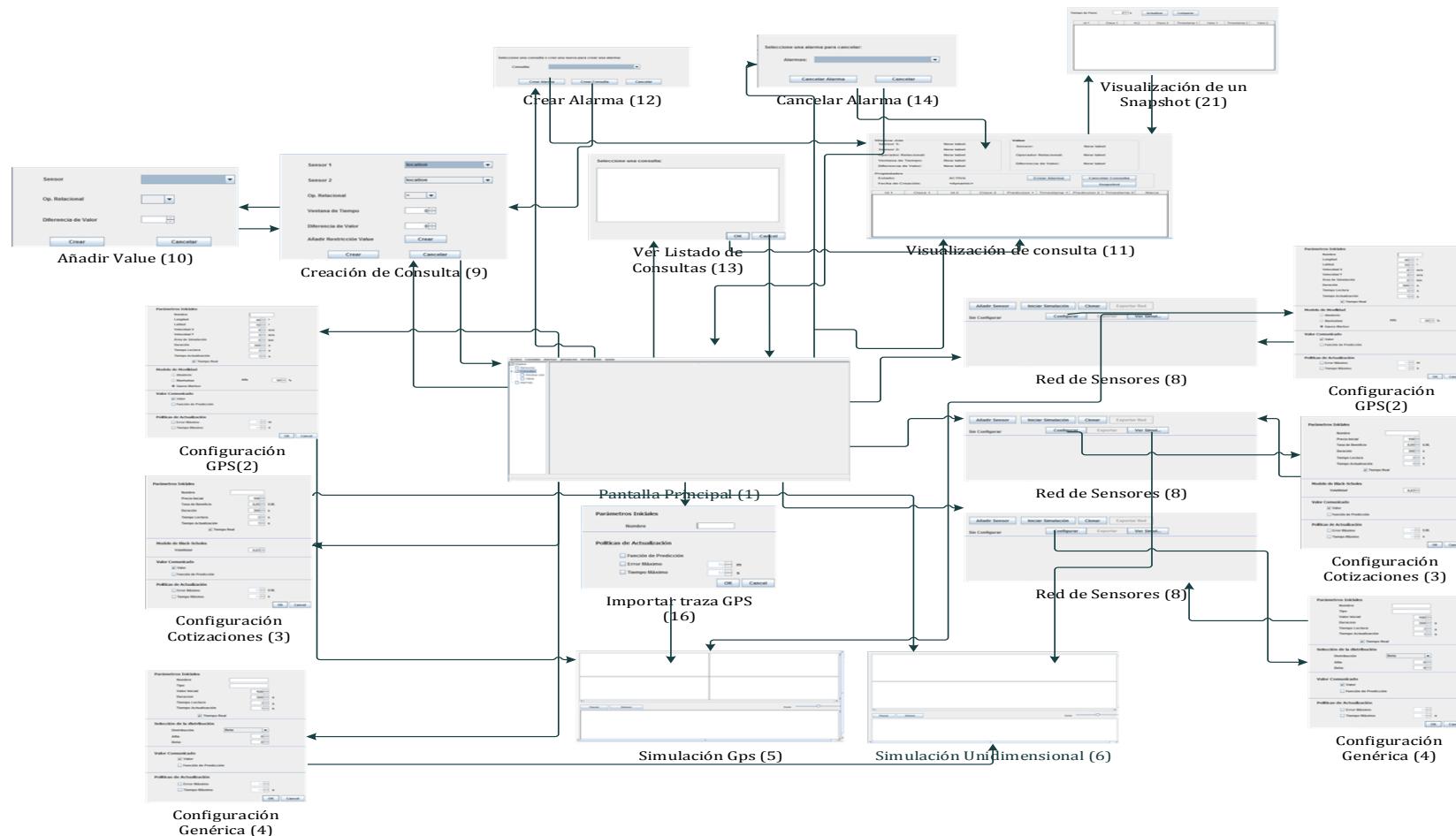


Figura C1 Diagrama de Navegación



# ANEXO D: Diseño e implementación del sistema

---

En este apartado se va a presentar el diseño y la implementación del proyecto. Se va a distinguir entre el código existente previamente y las clases nuevas desarrolladas (simulación e interfaz gráfica). Todo el código se ha separado en paquetes para una mayor facilidad de localización.

## D1 Paquete DSMS

El paquete DSMS contiene toda la infraestructura del procesador de data streams que existía previamente. Como el objetivo del proyecto ha sido ampliarlo y mejorar la interfaz, se va a presentar un pequeño resumen de la estructura y las clases de las que se compone el mismo:

### D1.1 Buffers

Este paquete dentro de DSMS contiene todas las clases referentes a los buffer y las diferentes políticas que se pueden implementar. En concreto contiene las clases:

- Buffer: Implementación de un buffer de una clase de sensores.
- BufferManager: Gestiona toda la arquitectura de buffers.
- BufferManagerPolicy: Interfaz para las diferentes políticas de buffer.
- BufferManagerPolicyDiscardOldest: Implementa la política de descartar la tupla mas antigua.
- BufferManagerPolicyDiscardRandomly: Implementa la política de descarte aleatorio.
- BufferManagerPolicyLimitSize: Descarta si hay un límite de tamaño.
- BufferManagerPolicyMinimizeUncertainty: Descarta aquella con el mínimo nivel de incertidumbre.
- Buffers: Almacena los buffers del sistema.

### D1.2 Constants

En este paquete se encuentra la clase *Constants*, en la que se recogen todos los parámetros de configuración para la ejecución de la aplicación.

### D1.3 Constraints

En este paquete se recogen las restricciones que soporta el programa contenidoo las siguientes clases

- ValidityMark: Contiene la marca de validez de una tupla de salida.
- ValueConstraint: Crea una restricción Value.
- WindowJoinConstraint: Crea una restricción Window Join.

### D1.3 Delayer

Este paquete contiene una única clase denominada *TuplesDelayer* que se encarga de añadir un retardo a la propagación de las tuplas a la hora de realizar la simulación.

### D1.4 Geo

En este paquete aparecen varias clases para el manejo de elementos geográficos, contenidoo las siguientes:

- Distance: Clase abstracta para todas los diferentes cálculos de distancias del paquete.
- DistanceInSeparateDimensions: Calcula la máxima distancia en una dimensión.
- EuclideanDistance: Calcula la distancia euclídea.
- ManhattanDistance: Calcula la distancia Manhattan.

### D1.5 Joins

Este paquete contiene la clase Joiner utilizada para calcular la unión de dos tuplas segúun una restricción, es una de las clases principales de la aplicación ya que se encarga de proporcionar resultados a las consultas.

### D1.6 Logs

Este paquete contiene la clase Logger que se encarga de todos los logs presentes en la aplicación.

## D1.7 Maths

Este paquete se compone de una serie de utilidades matemáticas para el manejo de inecuaciones, colas de prioridad, matrices, etc. Consta de las siguientes clases y paquetes internos.

- Constraint: Modela una restricción lineal.
- ConstraintSet: Conjunto de Restricciones.
- IntegralComputation: Realiza una integración definida.
- Intersection: Intersección entre dos restricciones.
- LinearFunction: Modela una función lineal.
- LinearInequality: Modela una inecuación lineal.
- RelationalOperator: Contiene los operadores relacionales, esta es una de las clases principales ya que interviene en la definición de las consultas.
- SimplePriorityQueue: Modela una cola de prioridad.
- TimestampMatchingBound: Establece una frontera temporal.
- ValidityInterval: Proporciona el intervalo de validez de una función.

## Maths.Matrix

- GaussianElimination: Realiza la eliminación Gaussiana de una matriz.
- MatrixOperation: Contiene diversas operaciones con matrices.
- UpperTriangle: Contiene una matriz triangular superior.

## D1.8 Output

Este paquete contiene las clases que modelan la salida proporcionada por las consultas:

- OutputRefresherWithArbitraryPredictionModels: Refresca la salida cuando se producen funciones lineales arbitrarias.
- OutputStreamer: Realiza un flujo de salida para las tuplas de salida.

## D1.9 Policies

En este paquete se establecen las políticas de actualización

- ThresholdUpdatePolicy: Política de umbral.
- UpdatePolicy: Interfaz para las políticas.

### D1.10 Predictions

En este paquete se proporcionaban los tipos de función de predicción que se usaban en el prototipo. Muchas de estas funciones están ya integradas en el simulador con lo cual se presentarán posteriormente las actuales.

### D1.11 Qp (Query Processor)

Este es el paquete principal de la aplicación ya que contiene todas las clases para la monitorización y para la creación de consultas. Consta de:

- Monitor: Es el procesador de consultas, es la clase más importante de la aplicación ya que se encarga de recoger las tuplas, las consultas, evaluar las nuevas tuplas, borrarlas del sistema y obtener los resultados.
- OutputDataStream: Es el flujo de salida que proporciona como resultado una consulta
- Query: Se trata de la clase que se usa para crear una consulta, esta se añade al monitor y ofrece como salida un flujo de tuplas de salida.
- ValueConstraintEvaluator: Sirve para evaluar restricciones de Valor.

### D1.12 Statics

Este paquete contiene una serie de clases para control puramente estadístico en la fase de investigación de la aplicación.

### D1.13 Test

En este paquete aparecen diferentes para probar el programa, es donde se lanzaba la primera versión de la aplicación por línea de comandos.

### D1.14 Util

Este paquete contiene diversas utilidades para el formateo de textos y numéricos. Hay que destacar las clases que se encargan de los Árboles R-tree de los que hace uso la aplicación.

- R-treeManager: Interfaz para manejar los R-tree.
- R-treeManagerJSI: Sirve para manejar los arboles R-tree de la biblioteca JSI. [27]
- R-treeManagerSpatialIndex: Sirve para manejar los arboles de la biblioteca Spatial Index. [28]
- R-treeVisitor: Recorre un árbol R-tree.

## D2 Paquete GUI

Este paquete contiene todas las clases referentes a la interfaz gráfica. Como su único cometido es mostrar la información por pantalla que le es proporcionada por la clase *Controller*, no se van a presentar los métodos que aparecen en ellas, ya que son todos referidos a la gestión de eventos de Java.

Además este paquete contiene las clases *Controller*, que gestiona todas las operaciones que se realizan en el programa, y el paquete *Configuration* que contiene todas las opciones de configuración. La forma de estas clases es la siguiente:

### CLASE CONTROLLER

Esta es la clase principal de la aplicación, su cometido es comunicar la interfaz gráfica con las clases que proporcionan funcionalidad a la aplicación. Con esto se consigue una separación de la parte gráfica de la parte computacional. Este modo de trabajo está inspirado en el modelo MVC (modelo, vista, controlador) usado para separar las partes de interfaz, lógica de negocio y acceso a datos. Como en este caso no existe acceso a datos, se sustituye por generación de datos (simulación y resultado de consultas).

Esta clase está formada por los siguientes atributos:

- cache: mantiene una cache de la equivalencia entre nombre de sensores y la clase sensor.
- dialog: es el diálogo actual en pantalla.
- mainframe: pantalla principal de la aplicación.
- monitor: Instancia de Monitor del procesador de consultas.
- MonitoringQuery: Consulta que se muestra por pantalla.
- Queries: Listado de queries almacenadas en el sistema.
- theAlarm: alarma monitorizada por pantalla actualmente.
- theQuery: query monitorizada por pantalla actualmente.
- theSims: simuladores que forman parte de una red.
- theSimss: simuladores almacenados en el sistema.
- triggers: clase que maneja las alarmas de la aplicación.
- multiSim: panel de multisimulación
- network: booleano que indica que estamos en una red.
- theConfigurations: configuraciones usadas en la red de sensores actual.
- theReports: estadísticas.

- theSim: Simulador actual que se muestra por pantalla.
- trigger\_t: Thread de ejecución de las alarmas.

Y los siguientes métodos:

- Controller(MainFrame) Constructor de la aplicación, necesita de la pantalla principal de la aplicación.
- actionConfiguration(int,MultiSimConfigPanel): Lanza una pantalla de configuración de un sensor en una red.
- actionExporter(int,MultiSimConfigPanel): Lanza una pantalla de exportación de un sensor en una red.
- actionViewSim(int,MultiSimConfigPanel) Lanza una pantalla de visualización de un sensor en una red.
- addReport(Report): Añade una estadística a la lista de estadísticas.
- addSensorToTree(AbstractSimulator): añadir sensor al árbol
- addSimMulti(int): Añade un sensor a un simulador de red.
- aggregation(int,string,string): Realiza una operación de agregación.
- assistantAddQuery(): Lanza la pantalla de creación de consulta.
- cloneSim(int,int): Clona un Simulador.
- exit(): Salir.
- exitSimulation(): Termina una simulación.
- GenericSimulation(): Crea una simulación genérica.
- getQueryResult(): Mostrar los resultados de una consulta.
- GpsSimulator(): Crear una simulación GPS.
- Importer(): Lanza el importador de ficheros.
- multiSim(int): Crea una red de sensores.
- openQuery(): Abre una consulta.
- PriceSimulator(): Crea un simulador de red de sensores.
- runMultiSim(MultiSimConfigPanel,int): lanza una simulación de red de sensores.
- saveQuery(): Guarda una consulta.
- setFrame(MainFrame): Cambia la pantalla mostrada por pantalla.
- setText(String): Establece un texto en la parte inferior de la pantalla principal.
- startStopSimulator(). Pausar/Relanzar una simulación.
- viewSimFromTree() Muestra una simulación desde el árbol lateral.

## D3 Paquete Communication

En este paquete se recogen las clases usadas para la comunicación de datos:

### D3.1 Clase AbstractCommunication

Es una clase abstracta padre de todas las comunicaciones de la aplicación.

Atributos:

- sensor: es el sensor sobre el que recae la política de comunicación.
- value : valor medido por el sensor.

Métodos

- getValue(): devuelve el valor medido por el sensor.
- send(long): envía un mensaje al procesador de consultas en el instante de simulación pasado como parámetro.
- setValue(AbstractValue) cambia el valor medido por el sensor.

### D3.1 Clase Message

Esta clase modela la información enviada al procesador:

Atributos:

- function: función de predicción enviada por el sensor.
- id: id del sensor.
- timestamp: fecha y hora de la comunicación.
- type: tipo del sensor.

Estos atributos tienen sus correspondientes métodos get para consultarlos.

### D3.1 Clase PredictionComunication

Hereda de AbstractCommunication, modela una política de comunicación que envía una función de predicción. También se usa para enviar un valor, considerando una función de predicción con variable independiente 0.

Atributos:

- maxUpdate: Umbral temporal de comunicación.
- prediction: función de predicción usada.
- Threshold: Umbral espacial de comunicación.

Estos atributos contienen sus correspondientes funciones get y set además de la función Send, empleada para enviar un mensaje.

## D4 Paquete GenericModel

Este paquete contiene la clase usada para actualizar un sensor genérico. Esta clase implementa la interfaz Runnable para poder ser lanzado como un Thread.

Atributos

- Eng: Motor usado para la creación de números aleatorios con una distribución de probabilidad. Usa la librería COLT.
- desviación: valor de la desviación para las distribuciones de probabilidad.
- dist: distribución de probabilidad elegida.
- I: valor de la iteración para simulaciones instantáneas.
- media: valor de la media.
- pause: booleano para indicar que está en pausa.
- running: booleano para indicar que está en funcionamiento.
- type: tipo del sensor.
- TIEMPO\_RESPUESTA\_ACTUALIZADOR: tiempo en el que actúa el actualizador.

Métodos:

- getNext(); obtiene el siguiente número aleatorio.
- pauseToogle() pausa o relanza la simulación.
- run(): inicia la simulación.
- stop() termina una simulación.

## D5 Paquete Mobility

En este paquete se recogen las diferentes clases para los modelos de movilidad. Todas heredan de la clase AbstractMobility que tiene la siguiente forma:

**Atributos:**

- boundary\_down\_right; punto inferior derecha del área de simulación.
- boundary\_up\_left: punto superior izquierda del área de simulación.
- gps: el sensor sobre el que actua el actualizador.
- pause: booleano para indicar que está en pausa.
- running: booleano para indica que está en funcionamiento.
- TIEMPO\_RESPUESTA\_ACTUALIZADOR: tiempo en el que actúa el actualizador.

**Métodos:**

- getNext(); obtiene el siguiente número aleatorio.
- pauseToogle() pausa o relanza la simulación.
- run(): inicia la simulación.
- stop() termina una simulación.
- Set de los puntos inferior y superior del área de simulación.

El resto de clases (GaussMarkov, Aleatory y Manhattan) heredan de esta última, incluyendo los datos propios de cada modelo.

## D6 Paquete PriceModel

Este paquete contiene la clase usada para actualizar un sensor de cotizaciones. Esta clase implementa la interfaz Runnable para poder ser lanzado como un Thread.

**Atributos**

- Eng: Motor usado para la creación de números aleatorios con una distribución de probabilidad. Usa la librería COLT. [29]
- b: tasa de beneficio.
- dist: distribución de probabilidad gaussiana.
- N: periodos de simulación.
- p: precio inicial.
- Sigma: volatilidad.
- I: valor de la iteración para simulaciones instantáneas.

- pause: booleano para indicar que está en pausa.
- running: booleano para indica que está en funcionamiento.
- type: tipo del sensor.
- TIEMPO\_RESPUESTA\_ACTUALIZADOR: tiempo en el que actúa el actualizador.

Métodos:

- getNext(); obtiene el siguiente número aleatorio.
- pauseToogle() pausa o relanza la simulación.
- run(): inicia la simulación.
- stop() termina una simulación.

## D7 Paquete Sensor

Este paquete contiene los diferentes tipos de sensores y las funciones de predicción.

### D7.1 AbstractPredictionFunction

Contiene toda la información de una función de predicción. De esta clase heredan tanto Vector2DPredictionFunction como FloatPredictionValue, usadas para funciones bidimensionales y unidimensionales respectivamente.

Métodos:

- clone(): crea una copia de la función de predicción.
- predictedValue(float) proporciona el valor para el instante proporcionado como parámetro.
- toLinearFunction(long,long) función que realiza la sincronización temporal entre el sistema de referencia del procesador y del simulador.

Las funciones hijas de esta contienen además las siguientes funciones:

changeFunction(): cambia la función de predicción.

setNoPrediction(): indica que el valor que multiplica a  $t$  es 0 para que no se actualice.

Y cada una además contiene el tipo de dato al que hace referencia.

## D7.2 AbstractSensor

Se trata de la entidad que modela un sensor, todos los tipos de sensores heredan de esta clase.

Atributos:

- communicationModel: Contiene el modelo de comunicación elegido.
- id : Identificador del sensor.
- timestamp : Hora de la última actualización.
- type : Clase del sensor.

Los métodos de esta clase son métodos get y set para los atributos anteriores.

## D7.3 GPSPredictionSensor

Esta clase modela un sensor de GPS que utiliza funciones de predicción. Su estructura es la siguiente.

Atributos:

- realPosition: Posición real almacenada por el sensor.
- realSpeed : Velocidad real almacenada por el sensor.

Métodos

- changeTrajectory(float, float, float, float): Cambia la trayectoria del elemento móvil.
- getAngle(): Devuelve el ángulo de la trayectoria.
- getModSpeed(): Devuelve el módulo de la velocidad.
- getPredictedPosition(float): Devuelve la posición según la función de predicción.
- setPredictedSpeed(Vector2D): Inicializa la velocidad en la función de predicción.

Además contiene funciones get y set para los atributos

#### D7.4 PricePredictionSensor

Esta función modela un sensor de cotizaciones con función de predicción.

##### Atributos

- value : Valor real del sensor.

##### Métodos

- getPredictedPrice(float): proporciona el valor predicho por la función.
- setPredictedValue(FloatValue) Inicializa el valor de la función de predicción.

#### D7.5 GenericPredictionSensor

Esta función modela un sensor genérico con función de predicción.

##### Atributos

- value : Valor real del sensor

##### Métodos

- getPredictedValue(float): proporciona el valor predicho por la función.
- setPredictedValue(FloatValue) Inicializa el valor de la función de predicción.

### D8 Paquete Importer

Este paquete contiene clases para importar diferentes tipos de ficheros.

La clase GPXReader actúa como superclase de los demás formatos ya que tiene la misma estructura y fue la primera que se creó. Esta clase tiene la siguiente forma:

##### Atributos

- file: nombre del fichero.
- name: nombre del sensor.
- positions: lista de posiciones leidas del fichero.
- speeds: lista de velocidades calculadas a partir del fichero.
- timestamps: lista de tiempos leidas por el fichero.

## Métodos

- `parse()`: lee el fichero y rellena los datos de posición y tiempo.
- `toNormal()`: transforma a una simulación sin ninguna política.
- `toPrediction()`: transforma a una simulación con función de predicción.

Las clases `CsvImporter` y `GeoLifeImporter` tienen la misma estructura, solo cambia la implementación de la función `parse()`.

Este paquete también tiene las clases `DsmCustomFilter` y `SimCustomFilter` que simplemente actúa de filtros a la hora de buscar ficheros.

## D9 Paquete Simulator

Este paquete contiene las clases que modelan los simuladores que utilizan todas las clases definidas anteriormente para realizar la simulación de datos.

Todos los sensores heredan de la clase `AbstractSimulator` que contiene toda la infraestructura común para el resto de sensores:

### ABSTRACTSIMULATOR

Esta es la clase padre de todas las clases de simulación. Su forma es la siguiente:

Atributos:

- `duration`: duración de la simulación.
- `id`: identificador del sensor.
- `messages` : Lista de mensajes de simulación.
- `pause` : booleano para indicar la pausa.
- `realNodes` : lista de nodos reales visitados por la simulación.
- `report` : Estadísticas.
- `run` : booleano para indicar que está en funcionamiento.
- `sendedNodes` : lista de nodos enviados al procesador.
- `theSensor` : Almacena el sensor utilizado en la simulación.
- `TIEMPO_RESPUESTA_ACTUALIZADOR`: Tiempo en el que se actualiza el valor del sensor.
- `TIEMPO_RESPUESTA_SENSOR`: Tiempo de lectura del sensor.
- `type`: Tipo de sensor.

Métodos:

- `export(String)`: Crea un fichero y exporta un sensor.
- `exportMulti(String)`: Guarda en un fichero existente la información del sensor.
- `initSimulator()`: inicializa el simulador.
- `pause()`: pausa el simulador.
- `runSimulator()`: lanza el simulador.
- `setTimes(int, int)`: Actualiza los tiempos de actualización y lectura si se desean modificar los datos globales.
- `stopSimulator()`: detiene el simulador.

Las clases GPSSimulator, PriceSimulator y GenericSimulation heredan de esta clase y mantienen idéntica la estructura, solo cambiando el actualizador y el valor (bidimensional o unidimensional según proceda).

## D10 Paquete Operations

Este paquete contiene la clase Aggregation que contiene las operaciones nuevas que se pueden realizar con el sistema de procesamiento. Consta de la clase agregación que consta de las siguientes operaciones

- `AVG()`: Calcula la media de una clase o sensor.
- `DESVEST()`: Calcula la desviación estándar.
- `COUNT()`: Calcula el número de tuplas almacenadas en el buffer de un tipo concreto.
- `MIN()`: Calcula el valor mínimo almacenado por el buffer.
- `MAX()`: Calcula el valor máximo almacenado por el buffer.

## D11 Paquete Triggers

Este paquete contiene toda la infraestructura de alarmas desarrollada para la aplicación. Consta de la clase Triggers que realiza la consulta si se ha activado alarmas y Alarm que define la alarma.

## D12 Paquete Snapshots

Este paquete contiene la clase Snapshot que es la encargada de realizar una “fotografía” de los resultados de la consulta aplicando a cada tupla obtenida como resultado un valor temporal proporcionado por el usuario.

## D13 Paquete Utils

Esta es un paquete con diferentes clases útiles a lo largo de la implementación de la aplicación, las clases más importantes son: ExportMessageList e ImportMessageList que contienen las funciones necesarias para importar y exportar las simulaciones. EL resto de clases solo se encargan de definir conjuntos de datos para simplificar el resto de la aplicación.

## D14 Diagramas de Clases

En este punto se presentan dos diagramas de clases resumidos (solo se presenta el nombre de la clase y sus relaciones debido a su gran extensión) que muestran la arquitectura del sistema de simulación creado para la aplicación, y otro que muestra la comunicación del mismo con todo el sistema de procesamiento de data streams.

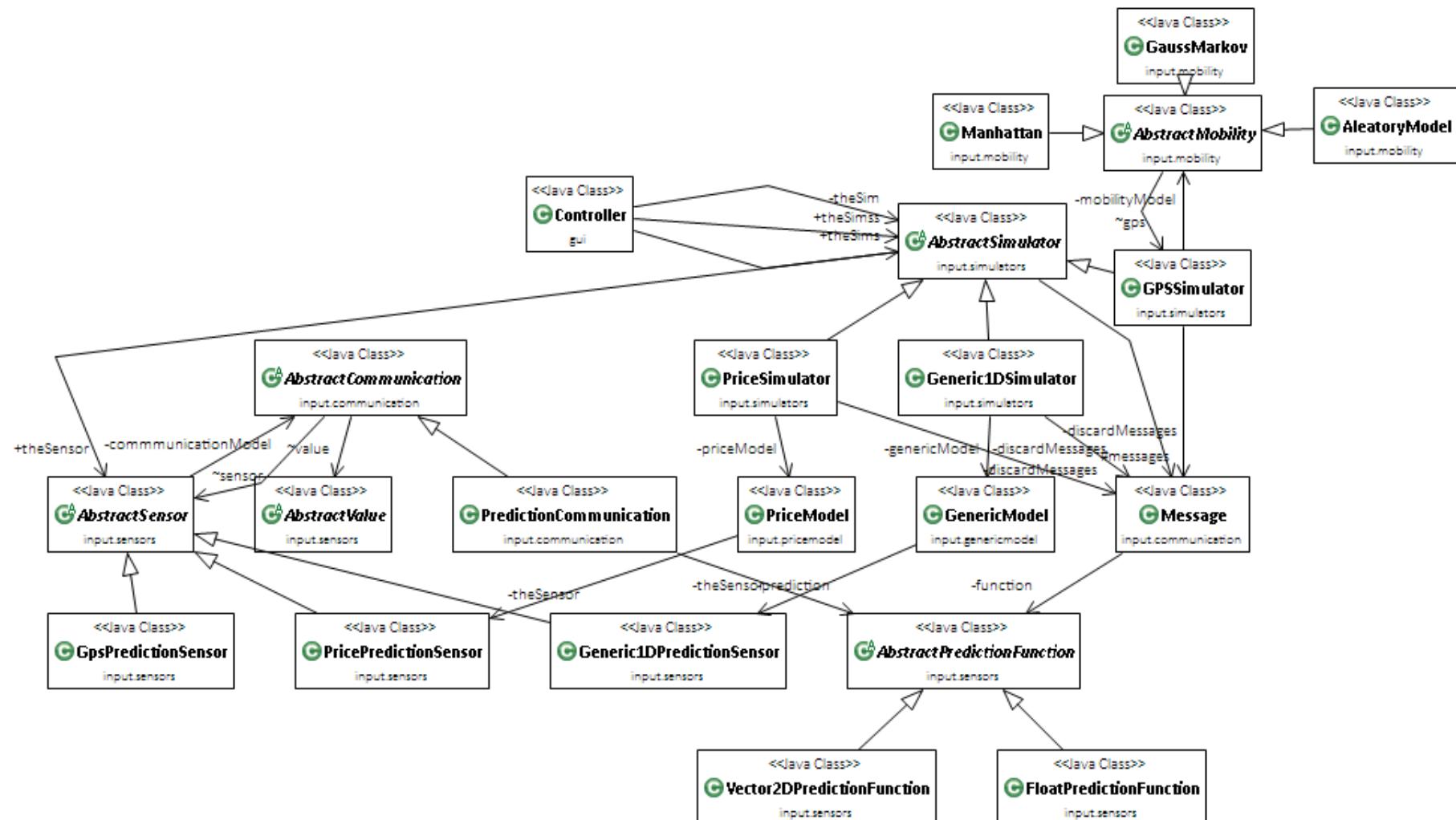


Figura D1 Diagrama de clases del simulador

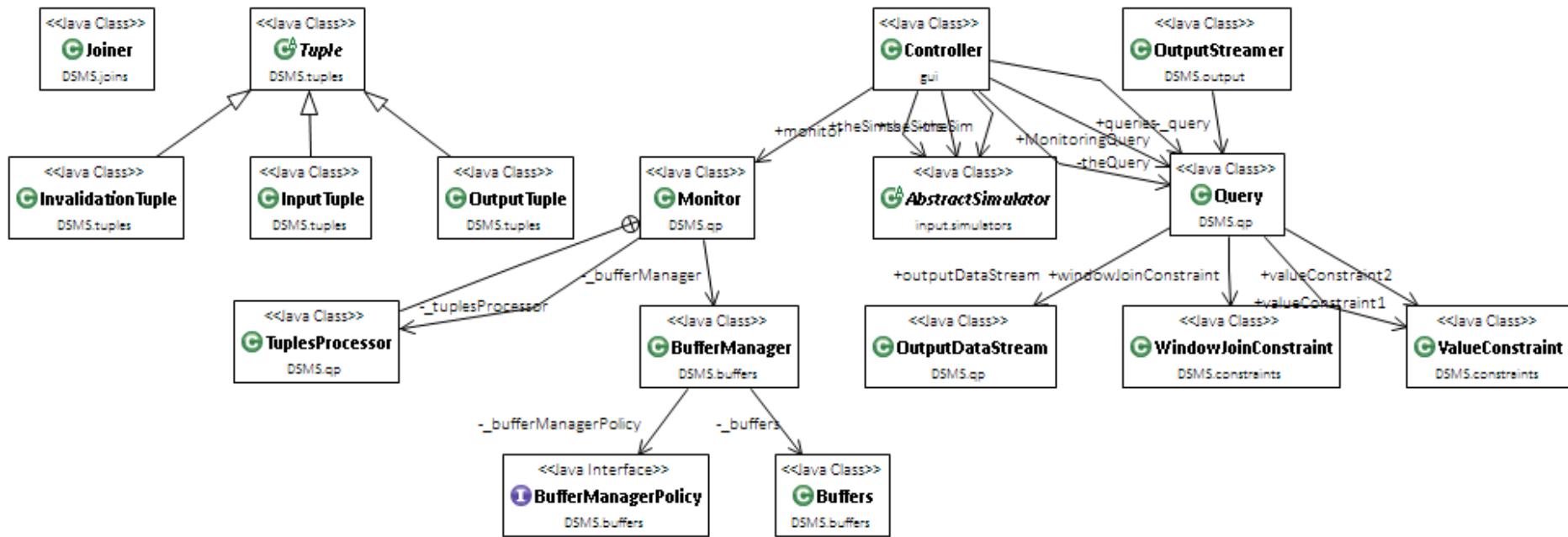


Figura D2 Diagrama de clases resumido del sistema de procesamiento con el simulador



# ANEXO E: Estrategias de movilidad

---

En este anexo se van a explicar los fundamentos matemáticos de las estrategias de movilidad adaptadas a la simulación de GPS que se han desarrollado para la aplicación de simulación. En concreto se van a explicar el modelo aleatorio, el modelo Manhattan y el modelo de Gauss-Markov.

## E1 Modelo Aleatorio

El modelo aleatorio fue la primera estrategia de movilidad que se implantó en el sistema. Es un sistema muy sencillo aunque bastante alejado de la realidad, pero que, para realizar el prototipo del simulador, fue muy útil ya que permitía realizar toda la arquitectura de la simulación y marcando las bases que seguirían los demás modelos de movilidad.

En este modelo se realiza un sorteo aleatorio con una probabilidad proporcionada por el usuario. Se obtiene un número aleatorio obtenido por la máquina del 0 a 1, si la probabilidad dada por el usuario es mayor que el valor obtenido por el número aleatorio el elemento móvil variará su trayectoria  $10^\circ$ , si no permanecerá en la misma dirección. Para determinar si se variará  $10^\circ$  o  $-10^\circ$  se volver a realizar otro sorteo aleatorio, si en este sorteo el valor es menor que 0,5 el móvil variará  $+10^\circ$ , si no  $-10^\circ$ .

En el siguiente gráfico se muestra una simulación con este modelo de movilidad:

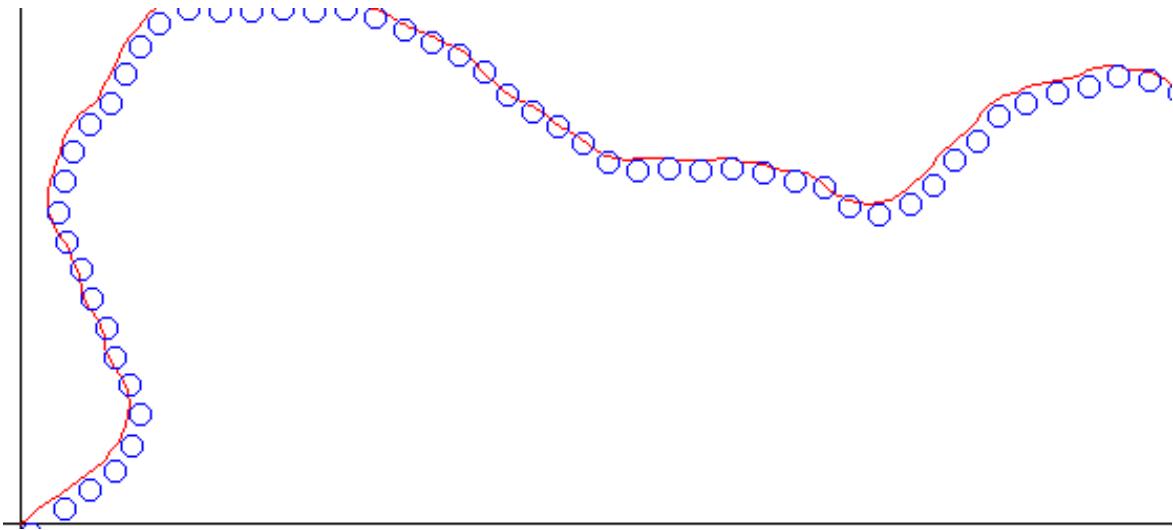


Figura E1 Simulación aleatoria con probabilidad 50%

## E2 Modelo Manhattan

La modelo Manhattan toma su nombre de la isla de Manhattan debido a que esta estrategia está especializada en ciudades con la topografía de Manhattan. Las calles de la isla de Manhattan tienen una topografía muy característica y es que tienen forma de malla o cuadrícula, es decir, las calles son paralelas o perpendiculares entre sí.

Para simular este modelo, en nuestra aplicación, se realiza un sorteo aleatorio para determinar en qué dirección se va a mover el elemento móvil. El elemento cambiará de dirección según estas probabilidades 25% Norte, 25% Sur, 25% Este y 25% Oeste, es decir, tendrá una posibilidad entre cuatro de que el elemento no cambie de dirección. Con esto se obtienen trayectorias en forma de cuadrícula, simulando el movimiento en una ciudad con estructura cuadrangular.

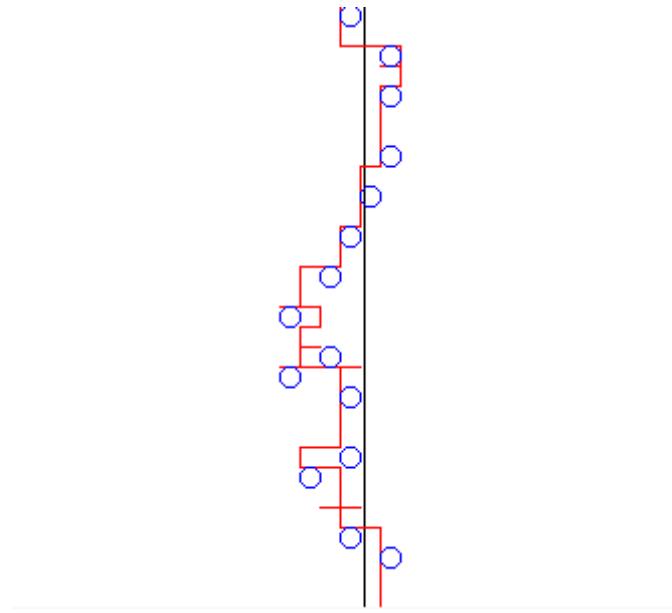


Figura E2 Simulación con Modelo Manhattan

### E3 Modelo Gauss-Markov

El modelo Gauss-Markov está pensada para que el elemento móvil se mueva en cualquier dirección sin restricciones de movimiento. Este modelo tiene diferentes niveles de aleatoriedad que se determinan a través de un parámetro denominado alfa ( $\alpha$ ). En cada paso de la simulación se actualizan la velocidad y dirección en base al instante anterior y el valor de este parámetro.

Las ecuaciones que se usan en este modelo son:

$$s_n = \alpha s_{n-1} + (1 - \alpha) \bar{s} + \sqrt{(1 - \alpha^2)} s_{x_{n-1}}$$

$$d_n = \alpha d_{n-1} + (1 - \alpha) \bar{d} + \sqrt{(1 - \alpha^2)} d_{x_{n-1}}$$

En estas ecuaciones  $s_n$  y  $d_n$  indican la velocidad y dirección respectivamente en el instante  $n$  de tiempo,  $\alpha$  es un parámetro que va de 0 a 1 y establece el valor de aleatoriedad y  $s$  y  $d$  son constantes que representan el valor medio la velocidad y la dirección cuando  $n \rightarrow \infty$ ,  $s_{xn-1}$  y  $d_{xn-1}$  son variables aleatorias que una distribución normal con media 0 y desviación típica 1.

En cada intervalo de tiempo, la siguiente posición se calcula en base a la posición actual y la velocidad y dirección calculadas con las anteriores ecuaciones. Para ello se utilizan estas ecuaciones:

$$x_n = x_{n-1} + s_n \cos d_n$$

$$y_n = y_{n-1} + s_n \sin d_n$$

Donde  $(x_n, y_n)$  es la posición obtenida en base a la posición anterior y los cálculos de  $s_n$  y  $d_n$  de las ecuaciones calculadas anteriormente.

Con todo ello se puede obtener una simulación como esta:

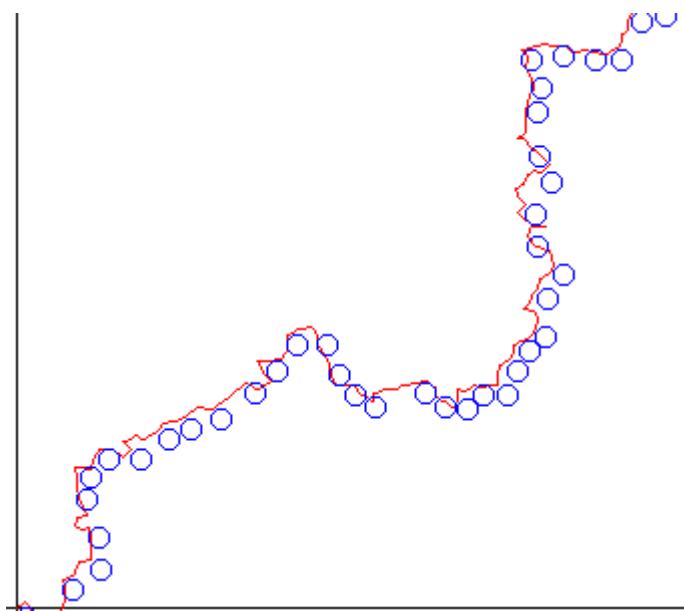


Figura E3 Simulación con modelo Gauss-Markov

## ANEXO F: Modelo de Black-Scholes

---

En este anexo se va a explicar el fundamento matemático para el modelo de simulación de cotizaciones.

A pesar de que el modelo de Black-Scholes en bolsa conceptualmente es muy complejo ya que es un modelo continuo que debe de ser simulado en infinitos puntos de tiempos, existe un algoritmo sencillo para su simulación.

El precio de una acción en el instante de tiempo  $t$ ,  $P_1(t)$ , en un modelo de Black-Scholes tiene la siguiente forma matemática:

$$P_1 = p_1 e^{(b - \frac{1}{2}\sigma^2)t + \sigma W(t)}$$

Siendo  $b$  la tasa de beneficio,  $\sigma$  la volatilidad de la acción y  $W(t)$  un proceso estocástico denominado Movimiento Browniano que cumple que:

$$W(t) \sim N(0, t)$$

Sí únicamente queremos simular el precio de la acción en un punto fijado de tiempo  $t$ , solo necesitamos que el número aleatorio  $X$ , siga una distribución normal estándar. Tras multiplicar este número por  $t$  se cumple que

$$\sqrt{t} \cdot X \sim N(0, t)$$

Sustituyendo  $W(t)$  por el valor aleatorio  $\sqrt{t} \cdot X$  obtenemos el precio simulado de la acción en el tiempo  $t$ . Sin embargo si se quiere simular toda la evolución del precio de  $P_1(t)$  para  $t[0, T]$ , se selecciona una partición fina y adecuada del intervalo  $[0, T]$ . El siguiente algoritmo muestra cómo realizar este modelo de simulación:

## Algoritmo 1: Modelo de Black-Scholes

Sea  $P_1(0)=p_1$ ,  $W(0)=0$ ,  $t_0=0$ ,  $i=0$ .

Desde  $i=1$  hasta a  $N$  se tiene que:

$$t_i = i \cdot \frac{T}{N}$$

Se genera el número aleatorio  $X_i$  que sigue una distribución normal estándar, que es independiente de los números creados anteriormente y se calcula:

$$W(t_i) = W(t_{i-1}) + \sqrt{\frac{T}{N}} X_i$$

$$P_1(t_i) = p_1 e^{(b - \frac{1}{2}\sigma^2)t + \sigma W(t)}$$

Se conectan los puntos para obtener la traza

Con este algoritmo aplicado a la simulación se obtienen resultados como este:

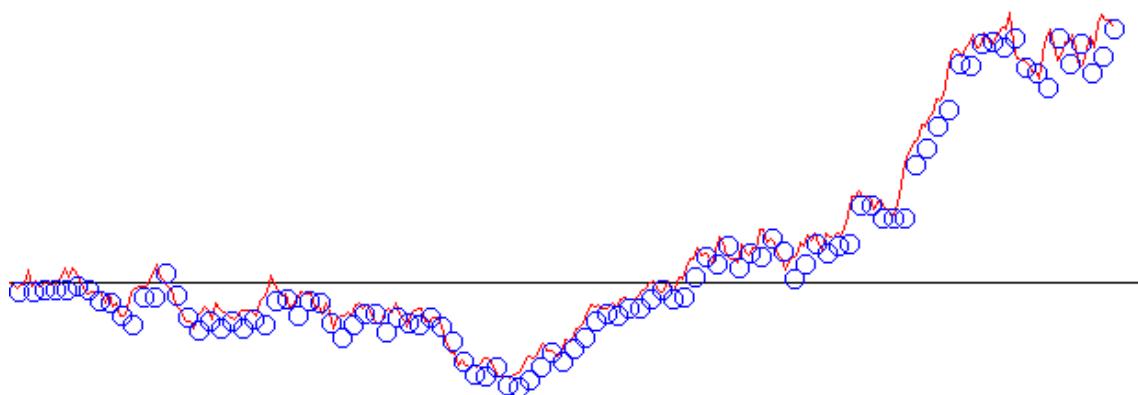


Figura F1 Simulación modelo Black-Scholes



# BIBLIOGRAFÍA

- [1] S. Ilarri, O. Wolfson, E. Mena, A. Illaramendi and N. Rishe, "Processing of Data Streams with Prediction Functions," *39th Hawaii International Conference on System Sciences (HICSS'06), Kauai (Hawaii), IEEE Computer Society, ISBN 0-7695-2507-5, ISSN 1530-1605*, pp. 237A-247A, January 2006.
- [2] S. Ilari, O. Wolfson, E. Mena, A. Illaramendi and P. Sistla, "A Query Processor for Prediction-Based Monitoring of Data Streams," *12th International Conference on Extending Database Technology (EDBT'09), Saint-Petersburg (Russia), ACM Press*, pp. 415-426, Marzo 2009.
- [3] "Java.com," [Online]. Available: <https://www.java.com/es/download/>. [Accessed 28 Agosto 2013].
- [4] "Eclipse," [Online]. Available: <https://www.eclipse.org/home/index.php>. [Accessed 31 Agosto 2014].
- [5] "Git," [Online]. Available: <http://git-scm.com/>. [Accessed 30 Agosto 2014].
- [6] "W3C: XML," [Online]. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>. [Accessed 31 Agosto 2014].
- [7] "Wikipedia: GPX," [Online]. Available: <http://es.wikipedia.org/wiki/GPX>. [Accessed 31 Agosto 2014].
- [8] "GeoLife," [Online]. Available: <http://research.microsoft.com/en-us/projects/geolife/>. [Accessed 20 Agosto 2014].

- [9] "Movebank," [Online]. Available: <https://www.movebank.org/>. [Accessed 20 Agosto 2014].
- [10] "Wikipedia: Manhattan mobility model," [Online]. Available: [http://en.wikipedia.org/wiki/Manhattan\\_mobility\\_model](http://en.wikipedia.org/wiki/Manhattan_mobility_model). [Accessed 24 Febrero 2014].
- [11] "Wikipedia: Black-Scholes," [Online]. Available: [http://en.wikipedia.org/wiki/Black%E2%80%93Scholes\\_model](http://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model). [Accessed 1 Marzo 2014].
- [12] "Gauss-Markov Mobility Model," [Online]. Available: <http://www-public.it-sudparis.eu/~gauthier/MobilityModel/mobilitymodel.html#Gauss>. [Accessed 24 Agosto 2014].
- [13] "Java (lenguaje de programación)," [Online]. Available: [http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). [Accessed 28 Agosto 2014].
- [14] "Java Swing," [Online]. Available: <http://docs.oracle.com/javase/tutorial/uiswing/>. [Accessed 20 Octubre 2013].
- [15] "Git," [Online]. Available: <http://es.wikipedia.org/wiki/Git>. [Accessed 2014 Agosto 2014].
- [16] "Data stream management system," [Online]. Available: [http://en.wikipedia.org/wiki/Data\\_stream\\_management\\_system](http://en.wikipedia.org/wiki/Data_stream_management_system). [Accessed 3 Noviembre 2014].
- [17] "SQLStream," [Online]. Available: <http://www.sqlstream.com/products/>. [Accessed 03 Noviembre 2014].

- [18] "Stanford Stream Data Manager," [Online]. Available: <http://infolab.stanford.edu/stream/>. [Accessed 3 Noviembre 2014].
- [19] "The Aurora Project," [Online]. Available: <http://cs.brown.edu/research/aurora/>. [Accessed 3 Noviembre 2014].
- [20] "Hadoop," [Online]. Available: <http://hadoop.apache.org/>. [Accessed 31 Agosto 2014].
- [21] "Hadoop Streaming," [Online]. Available: <http://hadoop.apache.org/docs/r1.2.1/streaming.html>. [Accessed 31 Agosto 2014].
- [22] "Introduction to Hadoop Streaming," [Online]. Available: <http://www.devx.com/opensource/introduction-to-hadoop-streaming.html>. [Accessed 03 Noviembre 2014].
- [23] A. Foundation, "Storm," [Online]. [Accessed 31 Agosto 2014].
- [24] E. Korn and R. Korn, "Capítulo 5-Modelo para el precio de las acciones," in *Modelo para el precio de las acciones*, Kaiserslautern, Management Mathematics for European Schools, pp. 152-187.
- [25] "Wikipedia: Interpolación Lineal," [Online]. Available: [http://es.wikipedia.org/wiki/Interpolaci%C3%B3n\\_lineal](http://es.wikipedia.org/wiki/Interpolaci%C3%B3n_lineal). [Accessed 30 Agosto 2014].
- [26] "Conversion de Coordenadas," [Online]. Available: <http://stackoverflow.com/questions/16080225/convert-lat-long-to-x-y-coordinates-c>. [Accessed 31 Enero 2014].

- [27] "JSI," [Online]. Available: <http://jsi.sourceforge.net/>. [Accessed 10 Julio 2014].
- [28] "Wikipedia: R-Tree," [Online]. Available: <http://en.wikipedia.org/wiki/R-tree>. [Accessed 10 Enero 2014].
- [29] C.E.R.N, "COLT," [Online]. Available: <http://acs.lbl.gov/ACSSoftware/colt/>. [Accessed 30 Diciembre 2013].
- [30] Y. Zheng, X. Xie and W.-Y. Ma, "GeoLife: A Collaborative Social Networking Service among User," *IEEE Data(base) Engineering Bulletin*.
- [31] "The GPS Exchange Format," [Online]. Available: <http://www.topografix.com/gpx.asp>. [Accessed 31 Agosto 2014].
- [32] "Wikipedia: Extensible Markup Language," [Online]. Available: [http://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language). [Accessed 31 Agosto 2014].