

Sistema de ayuda para la prevención e intervención temprana de situaciones de acoso escolar



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Maxim Betin

Tutor/es:

Antonio Manuel Jimeno Morenilla y

María Luisa Pertegal Felices



Universitat d'Alacant
Universidad de Alicante

Septiembre 2018

Justificación y objetivos

El acoso escolar o también conocido como *bullying*, es uno de los grandes problemas a los que un niño puede enfrentarse en su etapa de educación infantil ya que está en pleno desarrollo de su personalidad.

Es por ello que es de crucial importancia prevenir estas posibles situaciones en cuanto antes para que no dejen cicatrices tanto emocionales como físicas en el individuo durante sus años de escolarización obligatoria. El acoso escolar se puede dar muchas formas, algunas de ellas son la persecución, y el aislamiento de un individuo frente al resto.

El **objetivo principal** del proyecto consiste en la realización de un sistema automático que pretende detectar las situaciones de persecución y el aislamiento. Esto se pretende conseguir analizando las posiciones de un grupo de niños en un espacio abierto.

La posición de cada niño en cada instante es almacenada y a partir de esta información, mediante técnicas de trigonometría, estadística e inteligencia artificial podemos obtener una aproximación acerca de lo que está ocurriendo a un niño en concreto. De este modo, el profesor responsable del grupo de niños puede recibir una alerta acerca de un posible caso de acoso del sistema para poder juzgarlo y actuar como corresponda.

Las motivaciones que me han conducido a escoger este proyecto son varias; una de ellas es que, de primera mano, yo mismo he sido víctima del acoso escolar durante muchos años en mi educación en España. Tengo que admitir que a día de hoy aún sufro las consecuencias de ello y, por tanto, quiero dentro de lo posible, ayudar a prevenir estas prácticas utilizando mis conocimientos adquiridos como estudiante de ingeniería informática.

Agradecimientos

Se agradece a Antonio Manuel Jimeno Morenilla y Maria Luisa Pertegal Felices en apoyarme con el proyecto, prestar atención a los problemas y aportar soluciones cuando yo no las tenía.

Citas

“Si no hay héroes que te salven, te tienes que convertir en héroe”

Denpa Kyoshi

Contenido

1	Introducción.....	9
1.1	El acoso escolar o bullying.....	9
1.2	El acoso escolar en etapas tempranas.....	9
1.3	La detección del acoso escolar.....	10
1.4	Contexto de la detección en el proyecto.....	11
1.4.1	Limitaciones del sistema de detección.....	12
2	Marco teórico-tecnológico.....	14
2.1	Java.....	14
2.2	MySQL.....	14
2.3	La función Gaussiana.....	14
3	Objetivos.....	17
3.1	Objetivos generales.....	17
3.2	Objetivos específicos.....	17
4	Diseño e Implementación.....	18
4.1	El simulador.....	18
4.1.2	La aleatoriedad.....	21
4.2	La base de datos.....	23
4.2.1	Playground.....	24
4.2.2	Pursuers.....	25
4.2.3	Pursued.....	25
4.2.4	Isolated.....	26
4.3	La detección de la persecución.....	27
4.3.1	Requisitos del seguimiento.....	27
4.3.2	Fases de detección.....	28

4.3.2.1	La detección débil.....	28
4.3.2.2	La detección fuerte	33
4.4	La detección de aislamiento.....	40
5	Prueba del sistema	43
5.1	La simulación y detección.....	44
5.2	Los resultados	46
5.3	Base de datos.....	54
5.4	Tests del detector débil.....	55
6	Conclusiones	56
6.1	Trabajo futuro	56
7	Bibliografía y referencias.....	58

Tabla de ilustraciones

Ilustración 1 Recreo de un colegio	9
Ilustración 2 Campana de Gauss.....	15
Ilustración 3 Función máximo de puntos	19
Ilustración 4 Resultados simulación	20
Ilustración 5 Resultados simulación (2)	20
Ilustración 6 Diagrama relacional	23
Ilustración 7 Tabla playground	24
Ilustración 8 Tabla playground (2)	24
Ilustración 9 Tabla pursuers	25
Ilustración 10 Tabla pursued	25
Ilustración 11 Tabla isolated	26
Ilustración 12 Persecución.....	27
Ilustración 13 Trayectorias	29
Ilustración 14 Método del triángulo	29
Ilustración 15 Método del triángulo (2).....	30
Ilustración 16 Método del triángulo (3).....	31
Ilustración 17 Resultado tabla pursuers.....	34
Ilustración 18 Resultado tabla pursued.....	34
Ilustración 19 Curva distribución normal.....	36
Ilustración 20 Curva distribución normal (2)	37
Ilustración 21 Tabla de distribución normal.....	38
Ilustración 22 Tabla de distribución normal (2)	38
Ilustración 23 Algoritmo final perseguidores	39
Ilustración 24 Radio de aislamiento.....	41
Ilustración 25 Algoritmo detector aislamiento.....	42
Ilustración 26 Menú principal.....	43
Ilustración 27 Inicio simulación.....	44
Ilustración 28 Alerta.....	44
Ilustración 29 Alerta (2)	45

Ilustración 30 Trayectorias	45
Ilustración 31 Resultados	46
Ilustración 32 Información por niño	46
Ilustración 33 Información por niño (2)	47
Ilustración 34 Resultados (2).....	48
Ilustración 35 Información por día	49
Ilustración 36 Resultados (3).....	49
Ilustración 37 Información últimos días	50
Ilustración 38 Resultados (4).....	50
Ilustración 39 Lista final de perseguidores/perseguidos	51
Ilustración 40 Resultados aislamiento	52
Ilustración 41 Resultados aislamiento (2)	52
Ilustración 42 Resultados aislamiento (3)	53
Ilustración 43 Obtención de datos de la BD.....	54
Ilustración 44 Obtención de datos de la BD (2).....	54
Ilustración 45 Tests detector débil.....	55
Ilustración 46 Tests detector débil (2).....	55

1 Introducción

1.1 El acoso escolar o bullying

España no es un país en excepción en el área del acoso escolar. Uno de cada 3 alumnos afirma que en su clase se sufren situaciones de acoso escolar, y uno de cada 10 admite que lo ha sufrido [1][2]. Los niños que son acosados pueden experimentar problemas a nivel físico, escolar y de salud mental; tales como depresión, ansiedad, cambios en los patrones alimentarios y de sueño, disminución de los logros académicos, etc. [3].

La influencia del bullying en el rendimiento escolar requiere de atención también. En España, los centros en que el 10% de su alumnado sufre este tipo de hostigamiento obtienen un resultado 21 puntos inferior en Ciencias que aquellos centros educativos donde las víctimas de acoso se reducen al 5% [4].



Ilustración 1 Recreo de un colegio

1.2 El acoso escolar en etapas tempranas

Durante el último año de la etapa de educación infantil, en edades de entre 5 y 6 años, surgen las actitudes, conductas y comportamientos que serán luego dominantes en la etapa escolar.

Los niños y niñas de familias funcionales no son "inmunes" a transformarse en acosadores, ya que también influye cómo se desenvuelve el grupo escolar, la cohesión del grupo, la calidad y el buen desempeño de los maestros, entre otras variables. Es por

ello por lo que la educación de padres en la etapa escolar asume un papel significativo en la continuidad de la escuela.

Es "imprescindible" que los maestros, al igual que realizan sistemáticamente la valoración del desarrollo de todos los alumnos de su grupo, realicen el análisis del sistema de interrelaciones que se da entre los niños y niñas. [5]

1.3 La detección del acoso escolar

El principal mecanismo para combatir el acoso escolar de hoy en día, por desgracia trata de confiar en la observación, ya sea por parte de padres, tutores u otros profesionales. Esto no es una herramienta muy eficaz, hay muchas cosas de las que el ojo humano puede no ser consciente - no obstante, ya existen tecnologías para combatir el *bullying* que pretenden ayudar a resolver este problema.

Uno de los primeros ejemplos de tecnologías contra el acoso escolar surgió en el año 2007 en Finlandia: el método *KiVa*. Este programa, que busca prevenir y tratar estos casos en las escuelas, actualmente se aplica en el 90% de los colegios finlandeses y se ha exportado a otros países europeos. *KiVa* persigue cambiar las normas de grupo, instruir a los niños para que asuman su papel de apoyo a las víctimas y no fomenten esta práctica. Para ello, los alumnos cuentan con un videojuego y un entorno virtual multiplataforma de aprendizaje contra el bullying. [6]

La gamificación también se ha utilizado en España para combatir este problema. La empresa Nesplora ha desarrollado *Monité*, un videojuego incluido en un amplio proyecto para concienciar desde edades muy tempranas y que va dirigido a las víctimas, los agresores y la comunidad educativa. [7]

Pero éstas están orientadas a la **educación** de los niños para prevenir el bullying. No son una forma de **detección**. Existen otras herramientas como *Appvise*, una plataforma de comunicación escolar para dispositivos móviles que cuenta con mecanismos que ayudan a la identificación de casos de bullying o *Zeroacoso*, una app que mide el clima escolar de cada centro para saber si existe riesgo de bullying y, además, conecta a las víctimas con profesionales, de forma anónima, para que reciban la ayuda que necesiten. [8] [9]

Aun así, estas herramientas están basadas en la **comunicación** con la víctima, suponiendo que la víctima está dispuesta a hacerlo, lo cual, ocurre con rareza.

Por ahora, no cabe constancia de ningún sistema de detección sin involucrar ni al posible acosador, ni al acosado y usando solamente información recolectada por dispositivos o sensores externos (tal como las posiciones de cada uno). [10]

1.4 Contexto de la detección en el proyecto

Existen muchas formas de bullying, tales como el bullying físico, psicológico, verbal, sexual, social y ciberbullying.

El **bullying físico** es el tipo de acoso más común, especialmente entre chicos. Incluye golpes, empujones e incluso palizas entre uno o varios agresores contra una sola víctima, En ocasiones, se produce también el robo o daño intencionado de las pertenencias de las víctimas.

En el caso del **bullying psicológico** existe una persecución, intimidación, tiranía, chantaje, manipulación y amenazas al otro. Son acciones que dañan la autoestima de la víctima y fomentan su sensación de temor, con el problema añadido que son las más difíciles de detectar por parte de profesores o padres porque son formas de acoso o exclusión que se llevan a cabo a espaldas de cualquier persona que pueda advertir la situación. Frecuentemente, los agresores utilizan esta forma de acoso con el fin de subrayar, reforzar o resaltar acciones llevadas a cabo con anterioridad, manteniendo así latente la amenaza

El **bullying verbal** son acciones no corporales con la finalidad de discriminar, difundir chismes o rumores, realizar acciones de exclusión o bromas insultantes y repetidas del tipo poner apodos, insultar, amenazar, burlarse, reírse de los otros, generar rumores de carácter racista o sexual, etc.

En el **bullying sexual** se presenta un asedio, inducción o abuso sexual o referencias malintencionadas a partes íntimas del cuerpo de la víctima. Incluye el bullying homóforo, que es cuando el maltrato hace referencia a la orientación sexual de la víctima por motivos de homosexualidad real o imaginaria.

En cuanto al **bullying social**, se pretende aislar al niño o joven del resto del grupo, ignorándolo, aislándolo y excluyéndolo del resto. Puede ser directo: excluir, no dejar participar a la víctima en actividades, sacarlos del grupo o indirecto: ignorar, tratar como un objeto, como si no existiera o hacer ver que no está ahí.

Finalmente, tenemos el llamado **ciberbullying** que cada vez es más frecuente. Es un tipo de acoso muy grave y preocupante por la gran visibilidad y alcance que se logra de los actos de humillación contra la víctima y el anonimato en que pueden permanecer los acosadores. Los canales son muy variados: mensajes de texto en móviles, tabletas y ordenadores, páginas web y blogs, juegos online, correos electrónicos, chats, encuestas online de mal gusto, redes sociales, suplantación de identidad para poner mensajes, etc. [11]

El proyecto por tanto estaría centrado en casos específicos del bullying psicológico – la persecución, y el bullying social – el aislamiento ya que éstos son los que pueden detectarse mediante las posiciones de los niños en un recreo.

1.4.1 Limitaciones del sistema de detección

El sistema está diseñado para detectar posibles situaciones de acoso que se dan en el patio de un colegio sobre los intervalos de salida al recreo exclusivamente. Para ello, los niños deben portar un sistema de geolocalización que permita a un dispositivo central, recoger sus posiciones cada segundo y con una precisión de un metro como mínimo.

La intención inicial era usar dispositivos reales, pero debido al elevado coste por los requerimientos de precisión, la idea fue abandonada y sustituida por el uso de simulaciones de recreos automatizadas.

El proyecto trata de ocuparse del acoso que se pueda detectar mediante el posicionamiento en un eje cartesiano que representa el patio del colegio con unas dimensiones estándar. Sabiendo las posiciones de cada alumno en cada momento podemos saber cómo se comportan los alumnos en relación a sus compañeros. Podemos saber si un niño suele estar alrededor de otros o no, de cuáles y cuando. Además de si un niño en concreto suele ‘ir detrás’ de otro u otros o no. Por lo que podríamos tener una idea de si el niño persigue o es perseguido.

Las situaciones de persecución en un patio de colegio son constantes y forman parte de los juegos que habitualmente usan los niños. Por ello, a un profesor o tutor le es difícil determinar si estas persecuciones son siempre llevadas a cabo por el mismo grupo de niños y hacia el mismo niño en particular y nunca al revés, lo cual sí podría constituir una situación potencial de acoso. Es por esta razón que el sistema debe ser entendido como una herramienta de apoyo al docente y nunca como un elemento de diagnóstico. Únicamente el profesional será capaz de discernir si se trata de un acoso real o no. El sistema lo que sí facilitará es un seguimiento exhaustivo de las persecuciones y aislamientos de manera que pueda generar alertas al profesor en cuanto dichas persecuciones puedan considerarse como un riesgo potencial. Estas alertas no se llevarán a cabo observando un solo día de patio, sino una secuencia temporal suficiente de días para poder evaluar con mayor objetividad la situación real de cada niño.

2 Marco teórico-tecnológico

Las tecnologías utilizadas para la realización del proyecto están muy extendidas en la actualidad, lo cual ha facilitado el desarrollo del proyecto y sobre todo su depuración.

2.1 Java

Toda la programación base del proyecto está hecha haciendo uso del lenguaje Java.

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo. [12]

Esta característica de independencia de dispositivo/sistema operativo fue una de las características que resultaron atractivas del lenguaje para el proyecto, ya que nos permitiría extender la plataforma final a móviles, tabletas y otros.

A día de hoy, es el lenguaje más popular entre los programadores. Ha permitido programar tanto la lógica del proyecto, como la parte visual. Concretamente, para la parte visual se ha hecho un extensivo uso de la librería de *Swing*. [13]

2.2 MySQL

Para el almacenamiento de los datos de cada patio o recreo, se necesita una base de datos. Por tanto, a la hora de la gestión de la información se ha utilizado MySQL.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. [14]

2.3 La función Gaussiana

En cuanto a la lógica del proyecto, es decir, lo que es el sistema de detección, se ha desarrollado a partir de funciones estadísticas tales como la **función gaussiana** o también llamada distribución de Gauss.

En estadística y probabilidad se llama **distribución normal, distribución de Gauss, distribución gaussiana** o **distribución de Laplace-Gauss**, a una de

las distribuciones de probabilidad de variable continua que con más frecuencia aparece en estadística y en la teoría de probabilidades.

La gráfica de su función de densidad tiene una forma acampanada y es simétrica respecto de un determinado parámetro estadístico. Esta curva se conoce como campana de Gauss y es el gráfico de una función gaussiana.

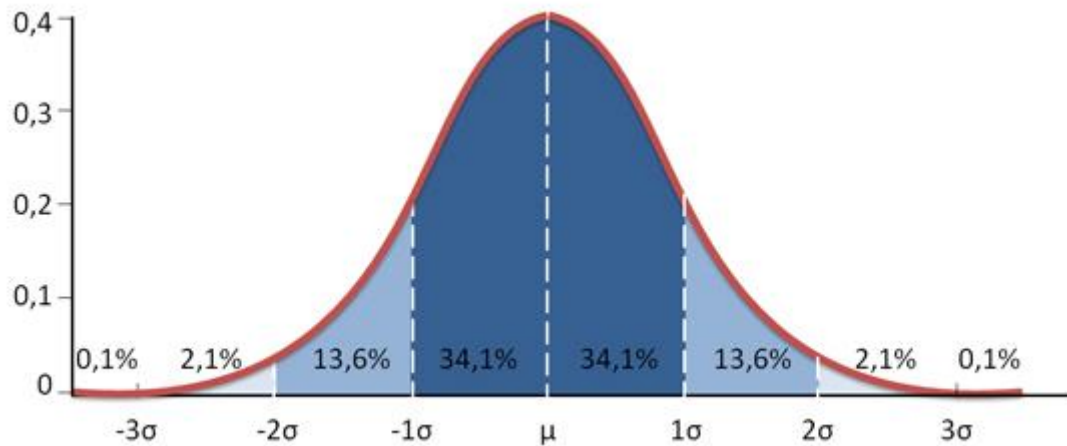


Ilustración 2 Campana de Gauss

La importancia de esta distribución radica en que permite modelar numerosos fenómenos naturales, sociales y psicológicos. Mientras que los mecanismos que subyacen a gran parte de este tipo de fenómenos son desconocidos, por la enorme cantidad de variables incontrolables que en ellos intervienen, el uso del modelo normal puede justificarse asumiendo que cada observación se obtiene como la suma de unas pocas causas independientes.

La distribución normal también aparece en muchas áreas de la propia estadística. Por ejemplo, la distribución muestral de las medias muestrales es aproximadamente normal, cuando la distribución de la población de la cual se extrae la muestra no es normal. Además, la distribución normal maximiza la entropía entre todas las distribuciones con media y varianza conocidas, lo cual la convierte en la elección natural de la distribución subyacente a una lista de datos resumidos en términos de media muestral y varianza. La distribución normal es la más extendida en estadística y muchos tests estadísticos están basados en una “normalidad” más o menos justificada de la variable aleatoria bajo estudio. [15]

La distribución normal no puede representarse sin hacer uso de la desviación típica que representa cómo se alejan los datos de la media.

La desviación típica o desviación estándar (denotada con el símbolo σ o s , dependiendo de la procedencia del conjunto de datos) es una medida de dispersión para variables de razón (variables cuantitativas o cantidades racionales) y de intervalo. Se define como la raíz cuadrada de la varianza de la variable.

Para conocer con detalle un conjunto de datos, no solo basta con conocer las medidas de tendencia central, sino que necesitamos conocer también la desviación que presentan los datos en su distribución respecto de la media aritmética de dicha distribución, con objeto de tener una visión de los mismos más acorde con la realidad al momento de describirlos e interpretarlos para la toma de decisiones. [16]

La campana de Gauss nos ha permitido asociar una “importancia” a cada detección de acoso y “repartir” la población de los niños. Es decir, los niños que son perseguidores estarán en los extremos de la campana, ya que, serán pocos, mientras que los que están alrededor de la media serán la mayoría.

La decisión principal de usar esta función en vez de otras soluciones como la IA es la simplicidad. Una técnica de redes neuronales artificiales añadiría una complejidad elevada al proyecto y a mayor coste de eficiencia, algo que no es del todo necesario y que se puede vitar. Usando la matemática, podemos conseguir justo lo que estamos buscando. Este método se ajusta a nuestro caso la estructura en forma de campana. En los extremos tendremos los casos menos probables (ningún seguimiento y elevada cantidad de seguimientos) y el valor medio sería el límite, donde se rompe el cambio de lo que está por debajo a por encima. En caso de haber podido encontrar una solución matemática al problema, posiblemente se tendría que haber recurrido al uso de redes neuronales. [17] [18]

3 Objetivos

3.1 Objetivos generales

- Detectar situaciones de persecución y de aislamiento entre los niños de un recreo de educación infantil a partir de las posiciones de cada uno en un eje de coordenadas. Teniendo en cuenta también los tiempos en los que se encuentran cada uno en cada posición.
- Analizar las situaciones detectadas de persecución y aislamiento durante un periodo específico (semanas, meses...) y generar alertas al profesor sobre riesgos potenciales de acoso.

3.2 Objetivos específicos

- Proporcionar al usuario – que en este caso sería el profesor responsable de los niños, una interfaz intuitiva de usar que permita acceder en cualquier momento los datos obtenidos después de cada recreo o cualquier otro recreo en el que el sistema ha estado puesto en marcha.
- A fin de probar la bondad del sistema, se debe realizar un simulador que genere posibles situaciones de persecución y aislamiento.
- Visualizar las persecuciones/aislamientos a petición del profesor. Toda detección que ocurra tiene que poder ser justificada gráficamente para el usuario. Por ejemplo, si en un día en concreto se ha detectado una persecución, el usuario tiene que poder ver cuáles son estas trayectorias y en qué momento ha ocurrido la detección para poder comprobarlo o juzgar la precisión del sistema.
- Almacenar en una base de datos toda la información generada en cada recreo para poder analizarla posteriormente. Toda información recolectada después de un recreo tiene que ser guardada en una base de datos que será accedida por el programa.

4 Diseño e Implementación

4.1 El simulador

Para obtener los datos con los que trabajar, el proyecto se limita a utilizar datos de simulaciones. Es decir, todas las posiciones y tiempos de los niños de un recreo se crean automáticamente por el sistema.

Esta decisión ha sido tomada principalmente por no haber podido utilizar dispositivos reales de geolocalización. Por tanto, el programa será responsable tanto de la detección, como la simulación. Sin embargo, es muy importante resaltar que estas dos funcionalidades del sistema son independientes y que por tanto el detector de acoso **funciona de la misma manera** tanto con datos reales obtenidos a partir de coordenadas GPS [19], como con los datos de simulación.

El simulador es capaz de crear la cantidad de días de recreos simulados que le indiquemos. En otras palabras, es necesario de proporcionarle cuántos días de patio van a utilizarse. Cada recreo va a ser diferente al resto.

La simulación de un recreo precisa de varios parámetros para ser creada. Estos son:

- El **número total** de niños.
- El **tiempo** máximo y mínimo que un niño necesita para pasar de una posición a otra.
- La máxima y mínima **cantidad de posiciones** por las que puede pasar un niño en un patio.

El número total se va a establecer a un valor apropiado para una clase de infantil, que en este caso es **30**.

El tiempo máximo que un niño puede tardar para pasar de una posición a otra va a ser de 180 segundos, es decir, 3 minutos. El tiempo mínimo es el tiempo mínimo es de 4.5 segundos. El tiempo a escoger será un número aleatorio entre estos dos valores. Por tanto, para que un niño pase de una posición A a una posición B, tendrán que pasar entre **4.5 y 180** segundos.

La cantidad de posiciones no es tan intuitiva de elegir. Es difícil de elegir un número de media, pero supongamos que un niño pasa por como máximo 15 posiciones.

Esto nos daría buenos resultados a la hora de lanzar el detector. Habría casos tanto de persecución como de aislamiento, sin embargo, todo ello considerando que tenemos 30 niños en el patio. Si el número decrementase a 15 niños, sería más difícil obtener unos datos lo suficientemente “buenos” para el detector, es decir, el detector podría no detectar nada aún tras varias ejecuciones del programa. Por ello, el máximo intervalo de trayectoria se ajusta en relación a la cantidad de niños; como menos niños, de media, pasarán por más posiciones, y como más niños, menos posiciones habrá por cada uno.

La función encargada de esta tarea tiene la siguiente apariencia:

```
private static int maxPoints () // en función de los días y de los niños establece el máximo intervalo de trayectoria
{
    int days = BullyingApp.NUMBER_DAYS, kids = BullyingApp.NUMBER_KIDS, maxPoints = 0;

    if (days < 11)
    {
        if (kids < 11) { maxPoints = 40; } else if (kids < 21) { maxPoints = 25; } else if (kids < 31) { maxPoints = 25; }
        else if (kids < 41) { maxPoints = 40; } else if (kids < 51) { maxPoints = 40; } else if (kids < 61) { maxPoints = 50; }
    }
    else if (days < 21)
    {
        if (kids < 11) { maxPoints = 30; } else if (kids < 21) { maxPoints = 20; } else if (kids < 31) { maxPoints = 20; }
        else if (kids < 41) { maxPoints = 20; } else if (kids < 51) { maxPoints = 20; } else if (kids < 61) { maxPoints = 20; }
    }
    else if (days < 31)
    {
        if (kids < 11) { maxPoints = 25; } else if (kids < 21) { maxPoints = 20; } else if (kids < 31) { maxPoints = 17; }
        else if (kids < 41) { maxPoints = 17; } else if (kids < 51) { maxPoints = 15; } else if (kids < 61) { maxPoints = 15; }
    }
    else if (days < 41)
    {
        if (kids < 11) { maxPoints = 20; } else if (kids < 21) { maxPoints = 20; } else if (kids < 31) { maxPoints = 20; }
        else if (kids < 41) { maxPoints = 17; } else if (kids < 51) { maxPoints = 17; } else if (kids < 61) { maxPoints = 15; }
    }
}
```

Ilustración 3 Función máximo de puntos

Todo ello se realiza con el fin de obtener unos resultados más convenientes, pero se podría establecer un valor general, digamos el 15; el inconveniente sería que tendríamos que probar con muchas ejecuciones aleatorias para obtener unos datos más plausibles.

También es importante destacar que hay que tener en cuenta la cantidad de días de simulación. Como más días pasen, mayor probabilidad habrá de que se detecte un caso de bullying y viceversa.

En resumen, la cantidad de máxima de trayectorias se equilibra con la cantidad de días de simulación y con el número de niños. Si hay pocos niños y pocos días, tenemos que aumentar el número de trayectorias, y por el contrario si hay muchos días de simulación y muchos niños, la cantidad de detecciones sería abrumadora por lo que se reduce el número de transiciones de posición.

Nunca va a haber una detección el primer día de recreo analizado. Como más días pasen, mayor probabilidad habrá de que haya una, siendo alrededor unos 10 días la cantidad mínima de tiempo requerido.

Finalmente, introduciendo como parámetros a la simulación **30 niños** y **30 días** el resultado del primer día sería el siguiente:

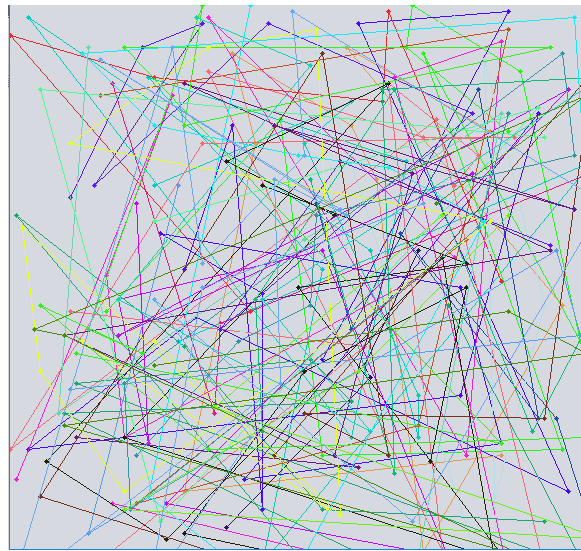


Ilustración 4 Resultados simulación

Cada color representa a un niño y cada punto del mismo color son sus coordenadas (x , y).

Obviamente el siguiente día de recreo simulado sería completamente diferente:



Ilustración 5 Resultados simulación (2)

Y así sucesivamente se obtendrían los resultados de todos los días de recreo con la cantidad de niños elegida.

4.1.2 La aleatoriedad

Es importante destacar que las simulaciones no son completamente aleatorias, si lo fueran, la probabilidad de obtener alguna detección de persecución durante varios días, por ejemplo, sería demasiado reducida, al menos que ejecutáramos una elevada cantidad de estas simulaciones.

Por tanto, las simulaciones se intentan mantener lo más realistas posibles con tal de proporcionar unos datos que puedan demostrar la efectividad del detector a una vista amplia.

Uno de los requisitos al simulador es que los niños creados, ya sean el acosador o el acosado vuelva llevar a cabo la práctica del acosamiento o vuelva a ser víctima respectivamente; ya que no es razonable que cada día obtuviéramos acosadores y acosados diferentes.

Para llevar a cabo esta función, se propone la siguiente solución:

El primer día de cada simulación siempre será aleatorio, con los parámetros que hemos establecido anteriormente, por ejemplo. En el caso de que se detecte alguna persecución este día, almacenaremos la información sobre quien es el perseguidor y quien fue el perseguido.

Al día siguiente, antes de crear la trayectoria de cada niño, miramos la información del pasado. Prestamos atención si su peso de perseguidor es mayor a 0. Si lo es, de entre sus niños perseguidos miramos cuál de ellos tienen sus trayectorias ya creadas.

Seguidamente, a partir de la tabla de probabilidades anteriormente descrita, obtenemos la probabilidad de perseguido de cada uno. Hacemos un sorteo de un número aleatorio del 1 al 100. Si la probabilidad del niño es menor que el número sorteado, elegimos este niño como el perseguido. La lista de perseguidos siempre está ordenada de mayor a menor peso, por lo que habrá mayor probabilidad de que un niño de mayor peso sea perseguido otra vez.

Teniendo en cuenta la longitud de la trayectoria del niño perseguido obtenemos la longitud de la trayectoria semi-aleatoria de persecución. Del mismo modo, el punto

inicial y final de la persecución. Todas las posiciones del perseguidor antes del punto inicial y final de persecución serán aleatoriamente creadas.

Además, para indicar cuantos eventos de acoso queremos, vamos a utilizar el límite de bullying. En vez de tenerlo establecido a un valor constante se cambiará a obtenerse mediante la raíz cuadrada al número total de niños que haya, ya que también, la curva de la raíz cuadrada representa la idea que se persigue en cuanto que haya subida de casos con mayor brusquedad al principio, pero que luego lo haga con mayor suavidad, para evitar que haya una proporcionalidad constante, ya que esto no resulta demasiado realista, resulta más realista que para 30 niños haya 3 niños acosados o acosadores, a que si hay 1000 haya 100. Con la raíz cuadrada, para 1000 presentaría en cambio unos 33 niños, que sí representa algo más fidedigno, por ejemplo.

Los puntos del perseguidor en el intervalo de persecución se situarán del mismo modo en que se hacen las detecciones débiles – en una posición aleatoria dentro del triángulo formado por los tres puntos anteriormente descritos. Y con un tiempo también aleatorio, pero no superando el tiempo de la posición final del niño perseguido. [20]

4.2 La base de datos

Cuando las simulaciones están terminadas, tendremos toda la información que necesitamos proporcionar al sistema de detección.

Por cada niño en cada día de simulación tendremos la siguiente información:

- Todas sus posiciones (x,y) .
- El tiempo en segundos en el que se encuentra en cada posición.

Utilizando los mismos parámetros (30 niños y 30 días), esto serian alrededor de 300 posiciones en total para tan solo el primer día.

Uno de los objetivos del proyecto es que todos los datos recogidos sean accesibles en cualquier momento, incluso cuando el programa no está en marcha. Es por ello que se ha de utilizar una base de datos.

Tal y como se describía anteriormente, el sistema de gestión de bases de datos escogido fue MySQL, concretamente la versión 8.0.11. Vamos a utilizar las siguientes cuatro tablas, a continuación, serán detalladas:

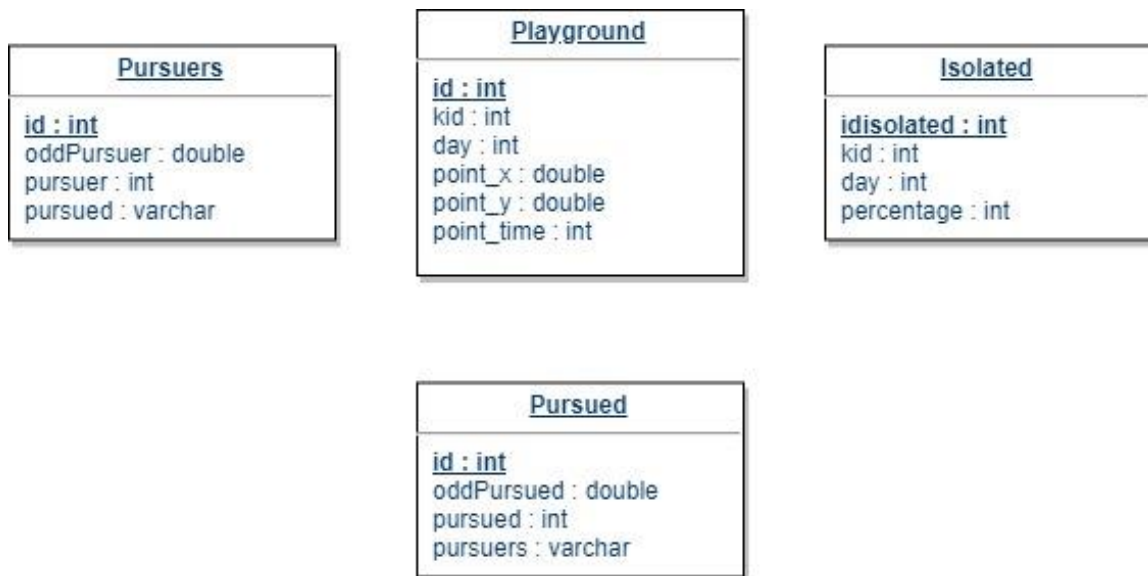


Ilustración 6 Diagrama relacional

4.2.1 Playground

La principal tabla que se utiliza es la tabla *playground*. Aquí guardamos todo lo que tenemos después de una simulación: las posiciones de cada niño junto con el tiempo en el que se encuentra en la posición en cada uno de los días.

Los campos que se utilizan son estos:

#	Field	Schema	Table	Type
1	id	bullying	playground	INT
2	kid	bullying	playground	INT
3	day	bullying	playground	INT
4	point_x	bullying	playground	DOUBLE
5	point_y	bullying	playground	DOUBLE
6	point_time	bullying	playground	INT

Ilustración 7 Tabla *playground*

Donde:

- El campo *id* es el identificador de cada fila.
- *Kid* es el identificador numérico del niño.
- *Day* es el número de día.
- *Point_x* y *point_y* son las coordenadas (x,y).
- *Point_time* es el tiempo en el que se encuentra el niño.

Y estos son algunos de los resultados:

id	kid	day	point_x	point_y	point_time
233337	0	0	0.8183333333333334	0.75	0
233338	0	0	0.29	0.81	66
233339	0	0	0.4083333333333333	0.23	45
233340	0	0	0.37	0.08	11
233341	0	0	0.5683333333333334	0.2	26
233342	0	0	0.61	0.29	122
233343	0	0	0.6883333333333334	0.91	68
233344	0	0	0.92	0.5	109
233345	0	0	0.56	0.07	122
233346	0	0	0.8	0.21	166
233347	0	1	0.77	0.17	0
233348	0	1	0.97	0.52	47
233349	0	1	0.16	0.75	50
233350	0	1	0.97	0.55	124
233351	0	1	0.2	0.89	12
233352	0	1	0.59	0.08	66
233353	0	1	0.59	0.98	105
233354	0	2	0.43	0.93	0
233355	0	2	0.98	0.96	35
233356	0	2	0.01	0.75	18

Ilustración 8 Tabla *playground* (2)

Las posiciones x e y son decimales porque en realidad son porcentajes de la anchura y altura de la ventana en la que se van a mostrar.

4.2.2 Pursuers

Esta tabla sirve para tener una constancia de los acosadores.

Field Types					
#	Field	Schema	Field	Table	Type
1	id	bullying		pursuers	INT
2	oddPursuer	bullying		pursuers	DOUBLE
3	pursuer	bullying		pursuers	INT
4	pursued	bullying		pursuers	VARCHAR

Ilustración 9 Tabla *pursuers*

4.2.3 Pursued

Del mismo modo, esta tabla sirve para almacenar la información relevante a los perseguidos (las víctimas).

Field Types					
#	Field	Schema	Field	Table	Type
1	id	bullying		pursued	INT
2	oddPursued	bullying		pursued	DOUBLE
3	pursued	bullying		pursued	INT
4	pursuers	bullying		pursued	VARCHAR

Ilustración 10 Tabla *pursued*

Como se puede observar, son muy similares. Empezando por la primera tabla tenemos los que es el *id* de la fila, después, *oddPursuer* que es la probabilidad de que este niño sea un perseguidor, *pursuer* es el id del niño y *pursued* contiene una cadena de caracteres donde se indican los niños que han sido perseguidos por éste.

La tabla *pursued* sigue el mismo principio, pero con los papeles invertidos. *oddPursued* es su probabilidad de perseguido, *pursued* es el identificador numérico del niño, y *pursuers* son los perseguidores registrados de éste.

4.2.4 Isolated

Para almacenar los eventos de aislamiento, se usa la tabla *isolated*. Sus campos son *kid* (el identificador del niño), *day* (día del aislamiento) y *percentage* (porcentaje de aislamiento sobre la trayectoria):

Field Types				
#	Field	Schema	Table	Type
1	idisolated	bullying	isolated	INT
2	kid	bullying	isolated	INT
3	day	bullying	isolated	INT
4	percentage	bullying	isolated	INT

Ilustración 11 Tabla *isolated*

Realmente, el programa puede funcionar sin estas últimas tres tablas, pero se han incluido por una posible futura extensión del sistema. Existe, sin embargo, un inconveniente en usar la BD y es que el tiempo de escritura en la misma es en ocasiones demasiado elevado. Se necesitan alrededor de unos 5 segundos para guardar toda la información recolectada por cada niño. Esta es una de las cuestiones que no se han conseguido resolver.

4.3 La detección de la persecución

Como se ha dicho anteriormente, muchas situaciones de acoso tienen lugar cuando se producen situaciones de persecución de un niño o varios niños hacia el mismo sujeto (y no al revés) de forma reiterada. Por tanto, resulta esencial detectar cada situación de persecución.

Empezando por el caso más simple de acoso que se pueda tener lugar y que se pueda detectar, es el de la persecución. Ocurre cuando un individuo sigue a otro durante una cierta cantidad tiempo con la intención de alcanzarlo.

En realidad, con tan solo un conjunto de posiciones (x,y) o trayectorias de dos individuos no podemos realmente decir que se trata de una persecución con intención de causar algún tipo de daño de uno a otro, por lo que lo llamaremos como un simple **seguimiento**; los dos niños pueden estar simplemente jugando el uno con el otro por lo que etiquetarlo con una palabra así es inadecuado por ahora.

4.3.1 Requisitos del seguimiento

Vamos a crearnos dos simples requisitos:

- El vector director del perseguidor apunta hacia la posición del perseguido.
- Las distancias entre los dos, de media, suele decrementarse y estar relativamente reducida en comparación con el resto.

Es mucho más simple replicarlo a nivel visual, por lo que una situación de seguimiento podría ser la siguiente (el rojo es el perseguidor y el azul el perseguido):

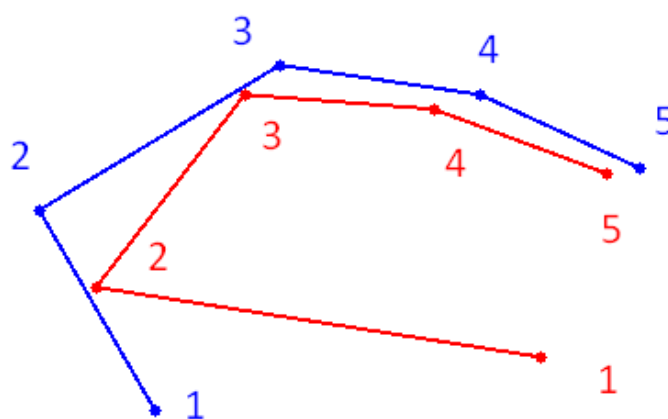


Ilustración 12 Persecución

4.3.2 Fases de detección

Para separar en fases la detección de acoso que hay en el sistema, vamos a utilizar dos conceptos: la detección “débil” será la detección de persecución que se realiza entre cada par de niños; mientras que la detección “fuerte” será la compuesta por varias detecciones débiles y que indicará si se está teniendo lugar un acoso o no.

4.3.2.1 La detección débil

Ahora que tenemos una situación de seguimiento ¿Cómo podemos detectarla?

A lo largo del desarrollo del proyecto, se han probado con varias formas. Viendo si las distancias entre los dos decrecen, viendo a donde apuntan los vectores de cada uno. Pero cada uno de estos casos trae consigo varias preguntas: ¿y si el perseguidor se acerca y luego se aleja? ¿y si la distancia no se reduce, pero sigue siendo pequeña? ¿y si la distancia aumenta? ¿y si el vector del perseguidor no apunta al perseguido? La casuística que es tan compleja que controlarlas y tener en cuenta todas no es tarea fácil.

Por tanto, la siguiente decisión a tomar fue si realmente es imprescindible poner tanto esfuerzo en completar un sistema de detección tan complejo.

Hay que recordar que, aunque en un caso hipotético tengamos un sistema de detección de seguimiento perfecto que tiene en cuenta todas estas variables, la verdadera detección de acoso no se realiza en este paso. Es decir, este sistema de detección de seguimiento “perfecto” no influenciaría el resultado final. Lo que verdaderamente nos interesa es que los eventos de persecución se detecten a lo largo de varios días, con los mismos pares de niños. Esto es lo que nos dice con mayor certeza que hay una probabilidad de acoso.

De este modo, en vez de aumentar tanto la complejidad de detecciones de seguimiento individuales para al final obtener unos resultados casi idénticos, es más apropiado simplificar lo máximo posible este paso de la detección e invertir el esfuerzo en un sistema de detección a vista más amplia.

Una forma simple, eficiente y efectiva de considerar que hay un seguimiento es lo que vamos a llamar el *método de la triangulación*.

Supongamos que tenemos dos tramos de trayectorias, una de un perseguidor (color rojo) y otra de un perseguido (color negro) como en la siguiente imagen. El punto inicial y final de ambas es A y B respectivamente según la ilustración 10.

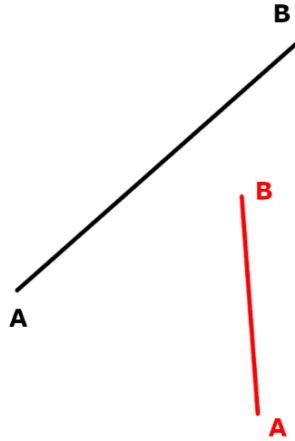


Ilustración 13 Trayectorias

Imaginémonos un triángulo formado por A, **A** y **B**.

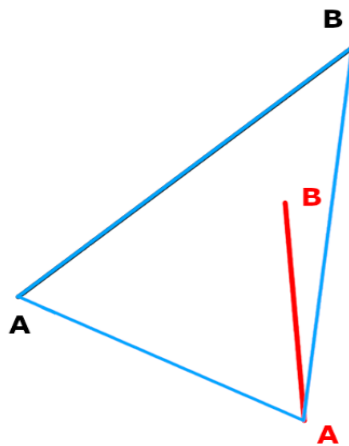


Ilustración 14 Método del triángulo

Obtenemos el triángulo de color azul. Fijémonos en el punto **B**. Éste cae dentro del área de lo que es el triángulo.

Esto nos proporciona justo lo que queríamos, pero de manera mucho más simplificada. El vector del perseguidor no apunta directamente al perseguido, sino en su dirección general, y la distancia también técnicamente se decrementa.

Como demostración, si el hipotético perseguidor no intentara alcanzar al perseguido este punto no acabaría dentro del área formada por el triángulo:

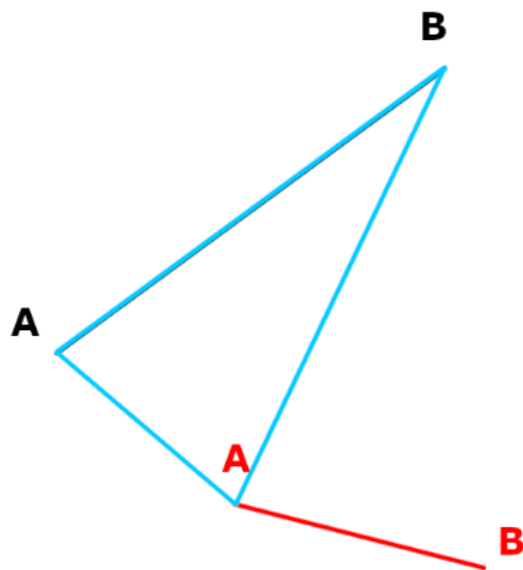


Ilustración 15 Método del triángulo (2)

Por tanto, lo único que tenemos que comprobar es si la posición final del seguidor acaba dentro de este triángulo. Si se cumple esta condición, podemos asumir que existe algún tipo de seguimiento. Eso sí, nunca está asegurado. Podemos llamar esta fase de detección como una *detección "débil"*. La verdadera detección de acoso ocurrirá cuando analicemos varios patios de diferentes días fijándonos en varios niños en concreto.

La intención es aplicar esta fórmula a cada par de tramos de trayectoria entre cada par de niños. Evidentemente se tendrán en cuenta varios factores como por ejemplo el del tiempo: tenemos que asegurarnos que las trayectorias de ambos ocurren en el mismo rango de tiempo. Lo cual, con tan solo comprobar si el tiempo en el que el perseguido alcanza su posición final es mayor o igual al tiempo en el que el perseguidor termina en su propia posición final, se consigue.

Dejemos claro los datos que necesitaremos para hacer funcionar el método.

Supongamos que el niño 1 es N1 y el niño 2 el N2. Necesitaremos saber:

- La posición inicial del tramo del primer niño: $(x_1, y_1)_{N1}$.
- La posición final del tramo del primer niño: $(x_2, y_2)_{N1}$.
- La posición inicial del tramo del segundo niño: $(x_1, y_1)_{N2}$.

- La posición final del tramo del segundo niño: $(x_2, y_2)_{N2}$.
- El tiempo en el que se encuentra en la posición final del tramo el primer niño: T_{N1} .
- El tiempo en el que se encuentra en la posición final del tramo el segundo niño: T_{N2} .

4.3.2.1.1 Formalización

Supongamos que los vértices del triángulo son V_1, V_2 y V_3 . El punto que nos interesa es P .

Teniendo en cuenta que la fórmula del área de un triángulo es $A = \frac{1}{2}bh$ podemos formar tres áreas conformadas por los vértices:

- $A_1 = (P, V_2, V_3)$
- $A_2 = (P, V_1, V_3)$
- $A_3 = (P, V_1, V_2)$

Siendo A , el área formada por V_1, V_2 y V_3 .

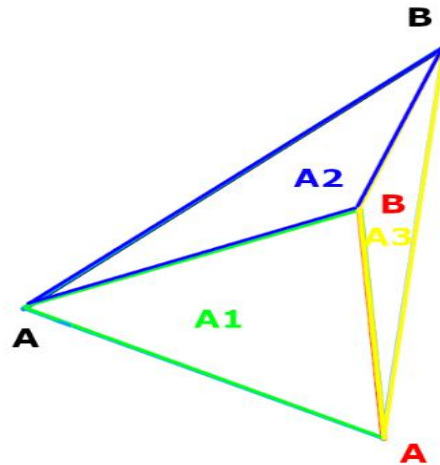


Ilustración 16 Método del triángulo (3)

Sabremos que el punto P se encuentra dentro del triángulo si $A_1 + A_2 + A_3 = A$. [21]

Si esta condición se cumple y si $T_2 \leq T_1$, tenemos un seguimiento entre estos dos intervalos.

Aun así, tenemos que elaborar algo más lo que consideraremos una persecución. Tal y como se ha visto anteriormente, podemos analizar cada par de trayectorias entre cada par de niños y decidir lo que ocurre en cada tramo.

El método definido anteriormente es aplicado a cada tramo de las trayectorias de ambos niños. Una trayectoria completa, son todas las posiciones por las que ha pasado un niño. Cogiendo las posiciones de dos en dos, generamos los tramos.

Sin embargo, ¿Y si el perseguidor solo le sigue durante una sola transición? ¿Y si el método nos devuelve resultados intercalados? Necesitamos establecer una cantidad mínima de vectores contiguos con seguimiento detectado: elijámoslo como el **40%** de las trayectorias que estamos comparando.

En otras palabras, si en un caso hipotético, se ha registrado que un niño ha estado en 10 posiciones diferentes en un patio, y en 4 posiciones contiguas ha sido seguido, se va catalogar este evento como persecución.

4.3.2.2 La detección fuerte

Tenemos un método de detección capaz de detectar situaciones de seguimiento. Pero ello no es suficiente. El principal objetivo es saber si existe una situación de riesgo de acoso y para ello se debe analizar la componente de reiteración a lo largo de sucesivos recreos.

Supongamos que es el primer día en el que el sistema se lanza en un colegio. Termina el recreo y por cada niño se han recolectado todas las posiciones en las que ha estado junto con el tiempo asociado a cada una. Estos datos están guardados en la base de datos (BD) y son accesibles en cualquier momento por el usuario.

Como nos podríamos imaginar, que el detector indique que durante varias trayectorias contiguas hubiera un seguimiento entre dos pares de niños, no significa que tuviéramos una situación de acoso, estos dos niños podrían haber estado jugando al pilla-pilla; incluso podría ocurrir coincidentemente que se cumplieran las condiciones para el detector, pero ninguno de los dos estuviera siguiendo intencionadamente al otro.

Lo que nos proporcionaría mayor certeza de que se está teniendo lugar un acosamiento es si ocurriera no solo durante un día, sino que es algo que tuviera tendencia a repetirse, posiblemente entre los mismos pares de niños.

Es muy importante destacar que a pesar de que podamos detectar posible acosamiento de este modo, nunca se tendrá certeza de ello, la labor de discernir si la respuesta del sistema está bien encaminada o no, está en manos del profesor responsable de los niños.

El programa pretende ser una herramienta de ayuda.

Ahora volvamos a nuestra suposición inicial. En el día 1, se ha dado el caso de que ha ocurrido una detección débil entre el niño 9 y el niño 8. El niño 9 ha “perseguido” al niño 8. Etiquetarlo como un acoso es demasiado precipitado en estos momentos. En cambio, lo que podemos hacer es registrar esta ocurrencia, pero no darle demasiada importancia por ahora.

Se generarían dos filas de interés en la BD. Una en la tabla *Pursuers* con el niño de identificador 9. *oddPursuer* muestra su probabilidad de perseguidor que sería del 69% y en la columna *pursued* tenemos primero el identificador del niño (o niños) que ha perseguido junto con la probabilidad asociada a esa persecución. La almohadilla (#) es el delimitador entre cada par de niño perseguido junto con la probabilidad.

id	oddPursuer	pursuer	pursued
354	0.69146	9	8-0.69146#
355	0.93319	10	1-0.93319#
356	0.84134	17	16-0.84134#
357	0.99379	23	0-0.99379#22-0.84134#
NULL	NULL	NULL	NULL

Ilustración 17 Resultado tabla *pursuers*

La otra fila creada está en la tabla *pursued*. *oddPursued* contendrá la probabilidad del niño 8 de ser un perseguido y en el campo *pursuers* está registrada la cadena de texto con el identificador de su perseguidor y probabilidad de persecución.

id	oddPursued	pursued	pursuers
1086	0.99379	0	23-0.99379#
1087	0.93319	1	10-0.93319#
1088	0.69146	8	9-0.69146#
1089	0.84134	16	17-0.84134#
1090	0.84134	22	23-0.84134#
NULL	NULL	NULL	NULL

Ilustración 18 Resultado tabla *pursued*

Como no podemos decir que el niño es un perseguidor aún, podemos asociar una probabilidad a que lo sea, un valor numérico que sea un indicio, al cual, vamos a llamar “peso”.

Cada niño tiene dos pesos: un **peso de perseguido** P_{ido} , y un **peso de perseguidor** P_{or} . De esta forma, el P_{or} del niño 9 se va a incrementar en una unidad. Al igual que el P_{ido} del niño 8. Estos datos NO se guardan en la BD, ya que es una funcionalidad propia del programa y es más apropiado que sea independiente a una base de datos relacional por lo que se guardarán en la memoria del programa, siendo calculados cada vez que se ejecute.

Pues bien, cada niño tiene asociados sus dos pares de pesos. La siguiente cuestión es ¿a partir de cuándo tendríamos que considerar de importancia los valores de los pesos? ¿dónde realizamos la delimitación?

En el caso de los perseguidores, cada unidad de peso indica que ha perseguido una vez. De nuevo, no tenemos otra opción que elegir un valor que el profesor del colegio considere como apropiado. Por ahora, se va a utilizar el **valor constante 5**. Llamaremos a este valor como el *Límite de bullying*. Más adelante, este límite nos servirá para cumplir otras funciones relacionadas a la creación de las simulaciones por lo que ya no será siempre el mismo valor para cualquier ejecución del sistema.

Como un perseguidor puede estar detrás de más de un niño, lo apropiado es utilizar una lista de pesos en vez de un solo peso. Si el niño 9 ha perseguido además al niño 11 y en dos ocasiones, el peso del acoso entre estos sería de un valor diferente, en este caso 2.

Al final, lo que se tiene es una lista de “pesos de niños perseguidos” y una lista de “pesos de niños perseguidores de”. Estas listas irán actualizándose tras la finalización de cada patio. Para saber si un niño es acosador o no, realmente sólo nos interesa el mayor peso y es éste el que se tendrá en cuenta a la hora de tomar la decisión.

En resumen, si un niño persigue a otro, aumentamos el peso del niño perseguidor y perseguido, al igual que si persigue a cualquier otro niño. Después, comprobamos si cualquiera de estos pesos supera el límite de bullying; si es así, podemos tener una “certeza” de que se trata de un riesgo de acoso y que hay que alertar al profesor.

Por tanto, el límite de bullying señala un valor sobre el cual, todo peso que esté por encima de dicho valor, será considerado para el programa que hay certeza suficiente para prestar atención de lo que esté sucediendo entre esos dos niños.

Ahora, ¿Cómo podemos discernir la importancia de cada detección? El profesor del colegio no tiene porqué saber lo que es el “peso” de un acosador o como funciona internamente el programa. Con tal de simplificar la lectura de los resultados, los pesos van a ser traducidos a probabilidades.

Es más, los pesos finales de cada niño van a ser sustituidos por una probabilidad adecuada. Es decir, cada niño tendrá una probabilidad de acosador y una probabilidad de acosado. Éstas se obtienen a partir del mayor peso de ambos casos.

Entonces ¿Cómo convertimos un peso numérico en una probabilidad? He aquí donde entra en función la estadística. Vamos a hacer uso de la función gaussiana o la

distribución normal. La curva de la distribución normal tiene la siguiente apariencia:

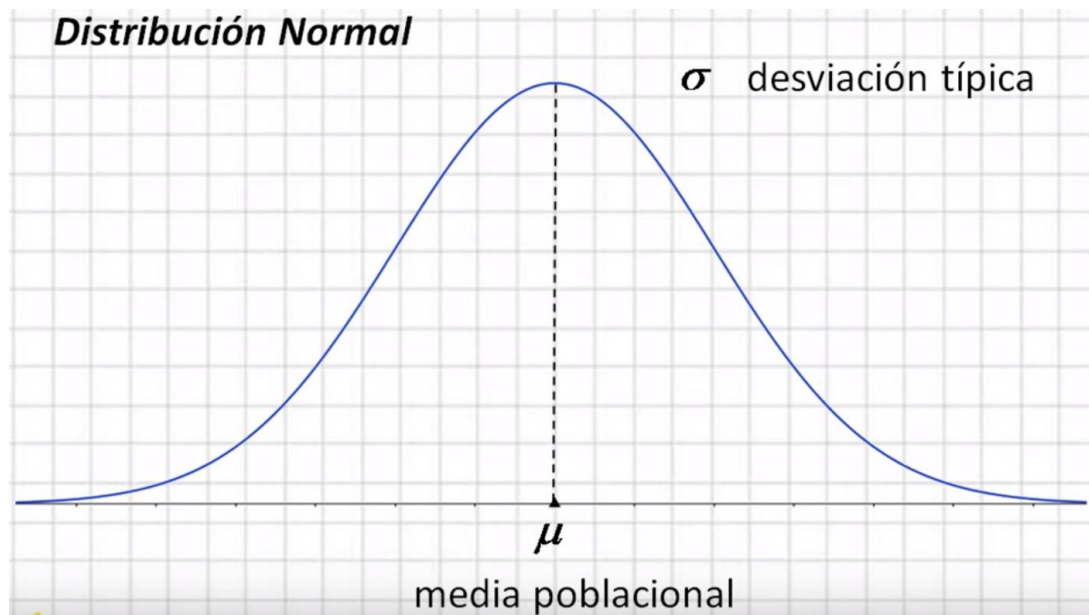


Ilustración 19 Curva distribución normal

Esto se utiliza y se puede ver muy a menudo en el estudio de poblaciones. Por ejemplo, la altura de la población, el peso, el sueldo, etc.

Gauss encontró una función que pudiera representar esta curva:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Donde μ es la media y σ es la desviación típica. La media está situada en el centro, que en nuestro caso será el límite de bullying con el valor constante que hemos definido antes. Teniendo la media establecida, lo que faltaría es fijar un valor para la desviación típica. La desviación típica nos dice en cuánto nos alejamos de la media, en nuestro caso será en el valor de una unidad.

El área encerrada entre la función y el eje horizontal es igual a 1. Es decir, haciendo la integral para obtener el área encerrada, obtendríamos 1 como el valor resultado:

$$\int_{-\infty}^{\infty} f(x) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = 1$$

Esto quiere decir que es el 100% de la población:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \int_{-\infty}^{\infty} f(x) = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = 1$$

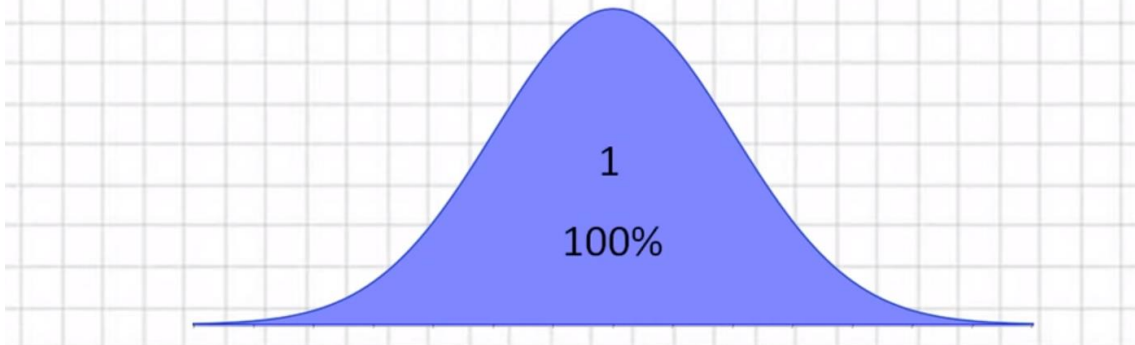


Ilustración 20 Curva distribución normal (2)

Haciendo uso de la integral anteriormente vista, podemos averiguar a qué porcentaje de la población pertenece el valor del peso.

Ahora necesitamos crear un *rango de bullying*. Si a partir del valor 5, consideramos que ha habido una posible persecución, el valor 0 significará que no se ha detectado ningún seguimiento, y, con tal de tener una simetría, vamos a definir el otro extremo del rango como el doble del límite – es decir, el valor 10.

Imaginémonos una recta; al principio de ésta tenemos el valor de peso 0, en el medio el 5 y al final el 10. Esto serían los valores del eje X de la curva; en el límite izquierdo tendríamos el valor 0 y en el límite derecho el valor 10. Lo que se pretende ahora, es asociar estos pesos a probabilidades de la curva.

Si nos fijamos en la gráfica, los valores de peso cercanos a la media serán obviamente los más comunes, mientras que, si nos alejamos, la probabilidad de encontrarse con un peso en los extremos, disminuye. Esto se adapta al modelo. Será poco probable que un niño nunca, en ningún momento fuera seguido por otro (peso 0) y, por otra parte, poco probable que sea seguido tan a menudo (peso 10).

Por tanto, hemos de decidir cuantas desviaciones típicas vamos a tener en nuestra distribución, esto se consigue dividiendo la longitud de los valores de la tabla (40) entre nuestro límite de bullying (5), dándonos como resultado el valor 8. Es decir, la curva de

distribución normal será dividida en **8 partes** (16 si consideramos la división para la parte izquierda, pero esta no se va a utilizar para la detección fuerte).

Necesitamos saber qué porcentaje de población asociar a cada peso. Con tal de evitar realizar el cálculo de la integral, vamos a utilizar la tabla de distribución normal ya creada [22]:

STANDARD NORMAL DISTRIBUTION: Table Values Represent AREA to the LEFT of the Z score.

Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.50000	.50399	.50798	.51197	.51595	.51994	.52392	.52790	.53188	.53586
0.1	.53983	.54380	.54776	.55172	.55567	.55962	.56356	.56749	.57142	.57535
0.2	.57926	.58317	.58706	.59095	.59483	.59871	.60257	.60642	.61026	.61409
0.3	.61791	.62172	.62552	.62930	.63307	.63683	.64058	.64431	.64803	.65173
0.4	.65542	.65910	.66276	.66640	.67003	.67364	.67724	.68082	.68439	.68793
0.5	.69146	.69497	.69847	.70194	.70540	.70884	.71226	.71566	.71904	.72240
0.6	.72575	.72907	.73237	.73565	.73891	.74215	.74537	.74857	.75175	.75490
0.7	.75804	.76115	.76424	.76730	.77035	.77337	.77637	.77935	.78230	.78524
0.8	.78814	.79103	.79389	.79673	.79955	.80234	.80511	.80785	.81057	.81327
0.9	.81594	.81859	.82121	.82381	.82639	.82894	.83147	.83398	.83646	.83891
1.0	.84134	.84375	.84614	.84849	.85083	.85314	.85543	.85769	.85993	.86214
1.1	.86433	.86650	.86864	.87076	.87286	.87493	.87698	.87900	.88100	.88298
1.2	.88493	.88686	.88877	.89065	.89251	.89435	.89617	.89796	.89973	.90147
1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91309	.91466	.91621	.91774
1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
2.0	.97725	.97778	.97831	.97882	.97932	.97982	.98030	.98077	.98124	.98169
2.1	.98214	.98257	.98300	.98341	.98382	.98422	.98461	.98500	.98537	.98574
2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
2.3	.98928	.98956	.98983	.99010	.99036	.99061	.99086	.99111	.99134	.99158
2.4	.99180	.99202	.99224	.99245	.99266	.99286	.99305	.99324	.99343	.99361
2.5	.99379	.99396	.99413	.99430	.99446	.99461	.99477	.99492	.99506	.99520
2.6	.99534	.99547	.99560	.99573	.99585	.99598	.99609	.99621	.99632	.99643
2.7	.99653	.99664	.99674	.99683	.99693	.99702	.99711	.99720	.99728	.99736
2.8	.99744	.99752	.99760	.99767	.99774	.99781	.99788	.99795	.99801	.99807
2.9	.99813	.99819	.99825	.99831	.99836	.99841	.99846	.99851	.99856	.99861
3.0	.99865	.99869	.99874	.99878	.99882	.99886	.99889	.99893	.99896	.99900
3.1	.99903	.99906	.99910	.99913	.99916	.99918	.99921	.99924	.99926	.99929

Ilustración 21 Tabla de distribución normal

Y para convertir los datos a código:

```
return new double[]
{
    0.50000, 0.53983, 0.57926, 0.61791, 0.65542, 0.69146, 0.72575, 0.75804, 0.78814, 0.81594,
    0.84134, 0.86433, 0.88493, 0.90320, 0.91924, 0.93319, 0.94520, 0.95543, 0.96407, 0.97128,
    0.97725, 0.98214, 0.98610, 0.98928, 0.99180, 0.99379, 0.99534, 0.99653, 0.99744, 0.99813,
    0.99865, 0.99903, 0.99931, 0.99952, 0.99966, 0.99977, 0.99984, 0.99989, 0.99993, 0.99995
};
```

Ilustración 22 Tabla de distribución normal (2)

Dependiendo de la cantidad de desviaciones típicas que tengamos, escogeremos cuántos “saltos” vamos a realizar en la tabla hasta elegir un valor de probabilidad.

Sigamos, por tanto, por el hilo de ejecución. Después de haber “filtrado” los datos por el detector débil y haber gestionado los pesos de los niños, tenemos que asociarles la probabilidad.

Por cada niño, tendremos una lista final tanto de perseguidos, como de sus perseguidores. He aquí el funcionamiento del algoritmo para obtener la lista final de perseguidos para un niño:

```
for (int i = 0; i < weightsPursued.size(); i++)
{
    if (LIMIT_BULLYING < weightsPursued.get(i))
    {
        int deviation = weightsPursued.get(i) - LIMIT_BULLYING,
            valueTable = (TABLE_ODDS.length <= (valueTable = (TABLE_ODDS.length / NUMBER_TD) * deviation))
                ? TABLE_ODDS.length - 1 : valueTable;

        resultsPursued.add(theirPursued.get(i));
        resultsOddsPursued.add(TABLE_ODDS[valueTable]);
    }
}
if (0 < resultsOddsPursued.size()) { oddPursuer = Collections.max(resultsOddsPursued); }
```

Ilustración 23 Algoritmo final perseguidores

- A. Por cada niño seguido, miramos si el peso de persecución entre ellos dos supera el límite de bullying.
 - a. Si es superado, obtenemos la desviación típica en la que se sitúa el valor; ésta la resta del peso de persecución entre el par de niños y el límite de bullying.
 - b. Obtenemos la posición de la tabla que nos interesa. Esto se consigue dividiendo la longitud de la tabla entre el rango de desviaciones típicas, y multiplicando el resultado por la desviación típica calculada anteriormente.
 - c. Si la posición obtenida supera la longitud de la tabla, utilizamos la última posición de la tabla.

Si el peso de persecución era 7 por ejemplo, la desviación típica sería 2 y la posición de interés de la tabla de distribución normal sería 10, que corresponde con la probabilidad

0.84134. La probabilidad de que este niño haya acosado al niño con el que hemos comparado sería de 84% debido al peso elevado.

Recordemos que también tenemos una probabilidad final de perseguidor o de perseguido. Para asignarla, miramos todos los niños perseguidos por éste y escogemos la mayor probabilidad. Si en nuestro caso, este niño sólo hubiera perseguido a este en concreto, su probabilidad de acosador también sería del 84%.

Finalmente, cabe destacar que, si se detecta un seguimiento entre dos niños durante un día, antes de incrementar los pesos de cada uno comprobaremos si ya ha ocurrido esto previamente el mismo día, pero con los papeles de cada niño invertidos; en cuyo caso, no aumentaremos los pesos si no que los decrementaremos debido a la previa detección y eliminaremos de sus listas de perseguidos/perseguidores el uno al otro.

El motivo es que un niño perseguido nunca va a perseguir a su perseguidor y un niño perseguidor no será perseguido por su perseguido y es por tanto que debemos de descartar esta situación de la lista de persecuciones.

4.4 La detección de aislamiento

Hasta ahora, hemos tratado con persecuciones. Detectar persecuciones es una forma eficaz de encontrar una forma de acoso. Pero no siempre estaremos en este caso. Hay muchos tipos de acoso, y otro que podamos descubrir utilizando las tecnologías que disponemos es lo que vamos a llamar el aislamiento.

¿Qué es el aislamiento? Que un niño sea acosado por sus compañeros no siempre quiere decir que le persiguen, es más, puede ocurrir todo lo contrario; en vez de ser perseguido puede estar completamente sólo en un recreo, puede estar “aislado” del resto, sin nadie acercándose a él o ella, es una consecuencia física que se deriva de un acoso de tipo no físico o psicológico.

Refinamos lo que queremos decir con “aislado”. Un niño en una determinada posición va a estar aislado del resto cuando en un radio preestablecido alrededor suyo, no se encuentra ningún otro niño. Veámoslo gráficamente:

Como ejemplo, el radio va a ser de 3 metros. Aunque en cuanto al programa al estar trabajando con píxeles se va a utilizar un porcentaje de la combinación del ancho y alto de la ventana de la interfaz de usuario.

El niño aislado es el punto en azul claro, el resto de los niños no entran en su radio y por tanto lo consideramos como aislado.

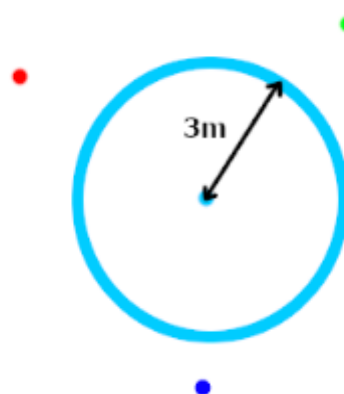


Ilustración 24 Radio de aislamiento

Al igual que con las persecuciones, se utilizará el mismo principio de detección fuerte y débil. Este paso lo consideramos como una detección de aislamiento débil.

Sin embargo, para considerar que un niño ha sido aislado durante un día vamos a construir otros dos requisitos:

- Tiene que haber estado aislado durante el 90% de su recorrido del día.
- Tiene que haber estado aislado de al menos el 80% de los niños de todo el patio.

El significado del último requisito es que tenemos que permitir que al menos alguna vez algún niño entre dentro de su radio por motivos de aleatoriedad.

Para registrar las ocurrencias de aislamiento se hará uso de la técnica de los pesos también. Cada vez que se cumplan los dos requisitos anteriores, un peso de aislamiento del niño va a ser internamente incrementado en una unidad.

Por último, del mismo modo que con las persecuciones, vamos a tener un límite de aislamiento (la detección “fuerte”); cuando el peso de aislamiento supere lo que es el 90% de los días en el que ha estado funcionando el programa, el programa va a dar un aviso al profesor.

Este es el código que representa dicho algoritmo:

```
for (int i = 0; i < NUMBER_KIDS; i++) // 1º for para coger a cada niño por orden
{
    for (int j = 0; j < NUMBER_KIDS; j++) // 2º for para comparar con el resto de los niños
    {
        contIsolationKids = 0;

        if (i != j) // si no coinciden comprobamos
        {
            int size1 = kids.get(i).getTrajectoryByDay(day).size(), size2 = kids.get(j).getTrajectoryByDay(day).size(),
            sizeMin = Math.min(size1, size2),
            limit_isolation = (int) (sizeMin * ((double)PERCENTAGE_ISOLATION_TRAJECTORY/(double)100)),
            contIsolationTrajectory = 0;

            for (int x = 0; x < sizeMin; x++)
            {
                Coord p1 = kids.get(i).getTrajectoryByDay(day).get(x),
                p2 = kids.get(j).getTrajectoryByDay(day).get(x);
                double distance = Math.sqrt(Math.pow((p1.getX() - p2.getX()), 2) + Math.pow(p1.getY() - p2.getY(), 2));

                if (distance > DISTANCE_ISOLATION) // si pasa de la distancia límite
                {
                    contIsolationTrajectory++;
                    contIsolationKids++;

                    if(contIsolationTrajectory >= limit_isolation)
                    {
                        if (contIsolationKids > LIMIT_ISOLATION_KIDS) // si pasa el límite de aislamiento sobre los demás niño
                        {
                            double percentageIsolation = ((double)contIsolationTrajectory/(double)size1)*100;
                            kids.get(i).incWeightIsolation((int)Math.round(percentageIsolation));
                        }
                    }
                }
            }
        }
    }
}
```

Ilustración 25 Algoritmo detector aislamiento

5 Prueba del sistema

Veamos cómo funciona el sistema para un supuesto usuario. Primero lanzamos el programa:

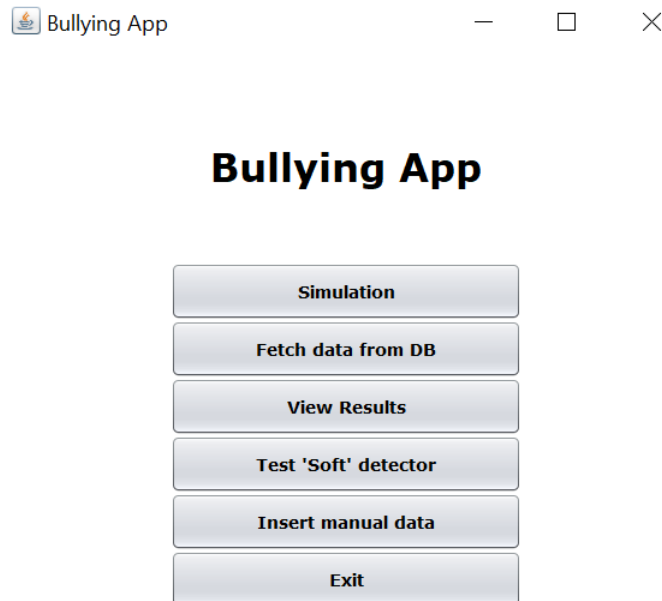


Ilustración 26 Menú principal

Como se observa, tenemos 5 principales opciones:

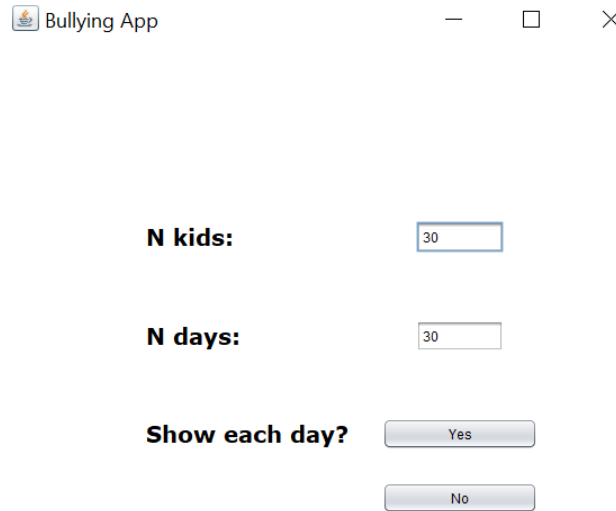
1. Iniciar una simulación.
2. Obtener los datos de la BD.
3. Ver los resultados del detector.
4. Probar el detector débil.
5. Insertar datos manuales.

Algunas de estas opciones no serían algo que el profesor del colegio tendría que ver, pero se han incluido para demostrar el funcionamiento. El profesor vería tan solo las opciones 1, 2 y 3. La opción 1 sería renombrada a *Run detector*.

Incluso se podría omitir esta última opción, el detector podría ejecutarse automáticamente en cuanto se obtengan los datos de la BD, y, es más, así es el funcionamiento actual. El detector es lanzado automáticamente también al seleccionar la opción de *Simulation* en cuanto terminen de simularse todos los días.

5.1 La simulación y detección

Al seleccionar la opción 1, nos aparecerá la siguiente pantalla, preguntándonos por los parámetros de la simulación:

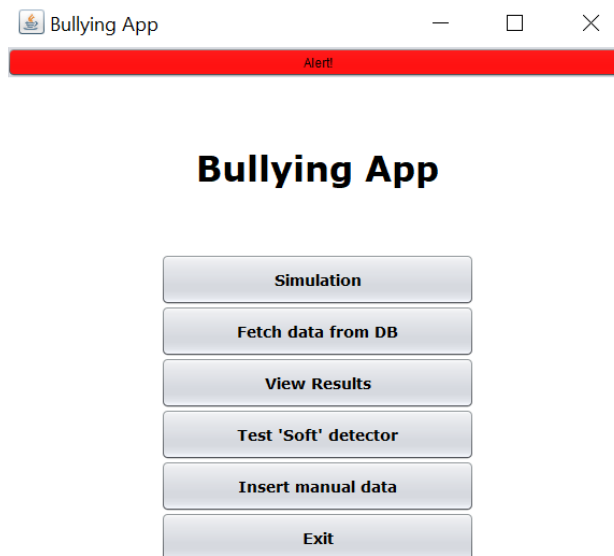


The screenshot shows a window titled "Bullying App" with standard window controls (minimize, maximize, close). The window contains the following configuration options:

- N kids:** A text input field containing the number "30".
- N days:** A text input field containing the number "30".
- Show each day?:** Two radio buttons, "Yes" and "No". The "No" button is selected.

Ilustración 27 Inicio simulación

Por defecto, el número de niños y de días de la simulación son 30, pero podemos cambiarlos. Finalmente, seleccionamos si queremos ver o no los resultados de cada patio – esta vez, vamos a decir que no.



The screenshot shows the "Bullying App" main menu. At the top, there is a red alert bar with the text "Alert!". Below the alert bar, the title "Bullying App" is displayed in a large, bold font. Underneath the title, there is a vertical stack of seven buttons:

- Simulation
- Fetch data from DB
- View Results
- Test 'Soft' detector
- Insert manual data
- Exit

Ilustración 28 Alerta

El programa nos devuelve al menú principal, pero esta vez aparece una alerta en color rojo en la parte superior, seleccionémosla:

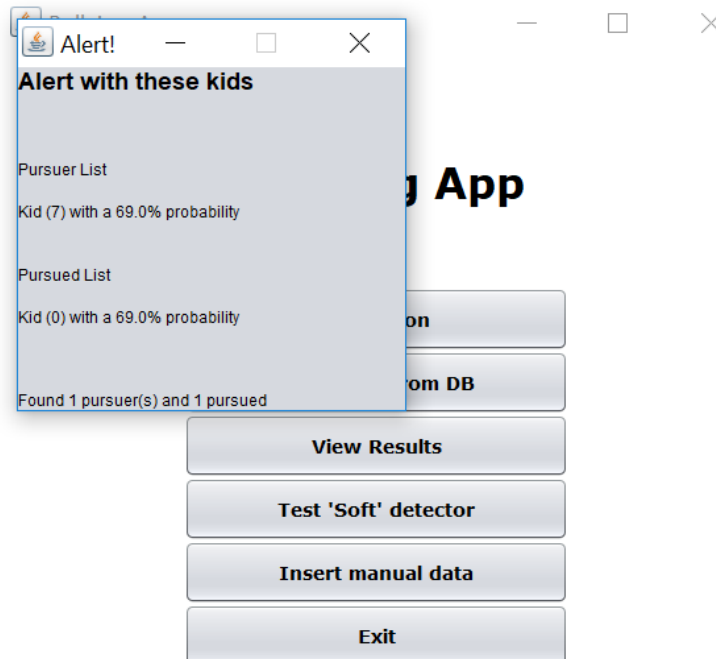


Ilustración 29 Alerta (2)

Aparece una nueva ventana diciéndonos que se ha detectado un perseguidor - el niño con identificador número 7 y un niño perseguido – el niño con id 0. En ambos casos, hay una certeza del 69%. En caso de que hubiéramos seleccionado ver los resultados de cada simulación, aparecerían todas las trayectorias de cada niño día por día:

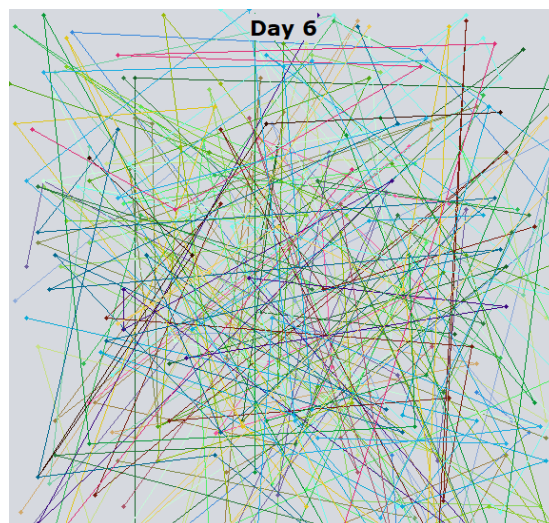


Ilustración 30 Trayectorias

5.2 Los resultados

Hemos visto que ha habido una detección, queremos ver lo que ha ocurrido, vayamos a verlo. Al seleccionar la opción, vemos la siguiente ventana:

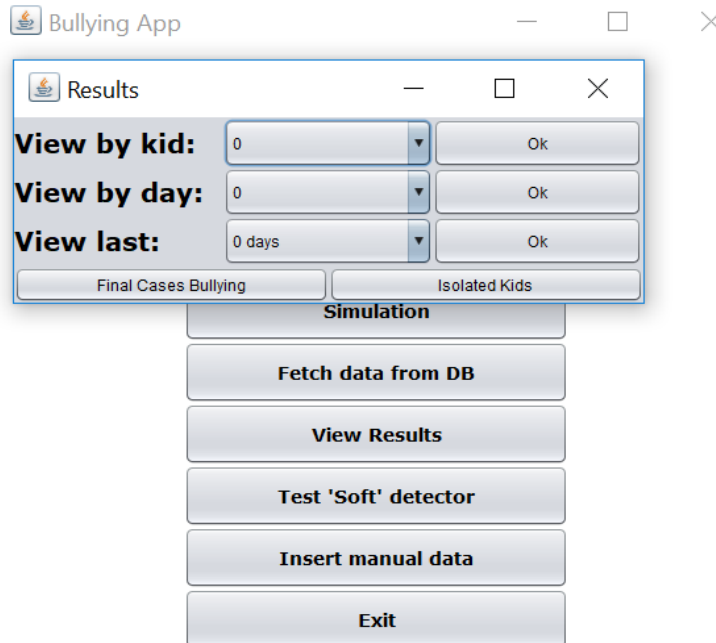


Ilustración 31 Resultados

En esta ventana, también tenemos varias opciones. Si nos interesa ver lo que ha ocurrido a un niño en concreto, seleccionamos el id del niño y veremos una pantalla similar:

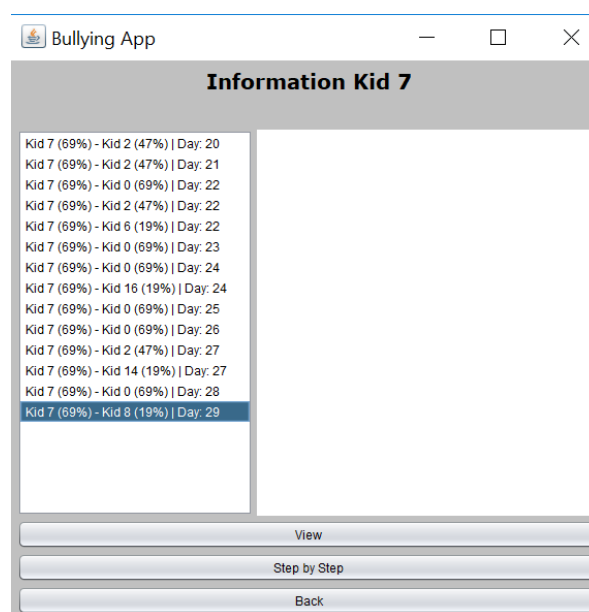


Ilustración 32 Información por niño

Aquí podemos toda la información relevante a las persecuciones entre los niños con este niño involucrado. Por ejemplo, la primera fila nos indica que el niño 7 ha perseguido al niño 2 (el niño perseguidor siempre aparece como primero) el día 20. Sin embargo, esto no aparecía en la alerta, el niño perseguido era el niño 0, no el 2.

Esto es debido a que el peso de perseguido del niño 2 no ha superado el límite de bullying, en otras palabras, no es lo suficientemente importante para aparecer en la lista final de persecuciones.

Fijémonos que el niño 0 sí que aparece en la lista y además varias veces. El día 22, 23, 24, 25, 26 y 28 fue perseguido por el niño 7. El ultimo día, se superó el límite de bullying y es por tanto que ha aparecido en la alerta. Podemos ver lo que ocurrió el último día seleccionando este evento y apretando el botón *View*:

Bullying App — □ ×

Information Kid 7

Kid 7 (69%) - Kid 2 (47%) Day: 20
Kid 7 (69%) - Kid 2 (47%) Day: 21
Kid 7 (69%) - Kid 0 (69%) Day: 22
Kid 7 (69%) - Kid 2 (47%) Day: 22
Kid 7 (69%) - Kid 6 (19%) Day: 22
Kid 7 (69%) - Kid 0 (69%) Day: 23
Kid 7 (69%) - Kid 5 (19%) Day: 23
Kid 7 (69%) - Kid 0 (69%) Day: 24
Kid 7 (69%) - Kid 16 (19%) Day: 24
Kid 7 (69%) - Kid 0 (69%) Day: 25
Kid 7 (69%) - Kid 0 (69%) Day: 26
Kid 7 (69%) - Kid 2 (47%) Day: 27
Kid 7 (69%) - Kid 14 (19%) Day: 27
Kid 7 (69%) - Kid 0 (69%) Day: 28
Kid 7 (69%) - Kid 8 (19%) Day: 29

View

Step by Step

Back

Ilustración 33 Información por niño (2)

En el panel de la derecha, nos aparecerán las trayectorias de los dos niños, junto con los triángulos formados por el método de detección débil, indicándonos con un punto negro, cuando la posición final del perseguidor ha acabado dentro del área. Si queremos ver estos datos paso por paso, tenemos la opción *Step by Step* que nos mostrará lo mismo pero par por par de trayectorias.

De la lista de selección de resultados, filtrémoslo por día. El día de este evento era el 28:

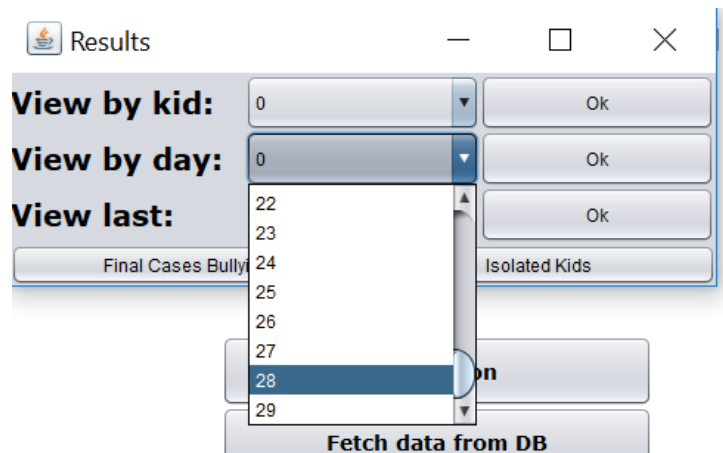


Ilustración 34 Resultados (2)

De los eventos del día, seleccionamos la persecución del niño 7 al niño 0 y vemos que el mapa creado es exactamente el mismo:

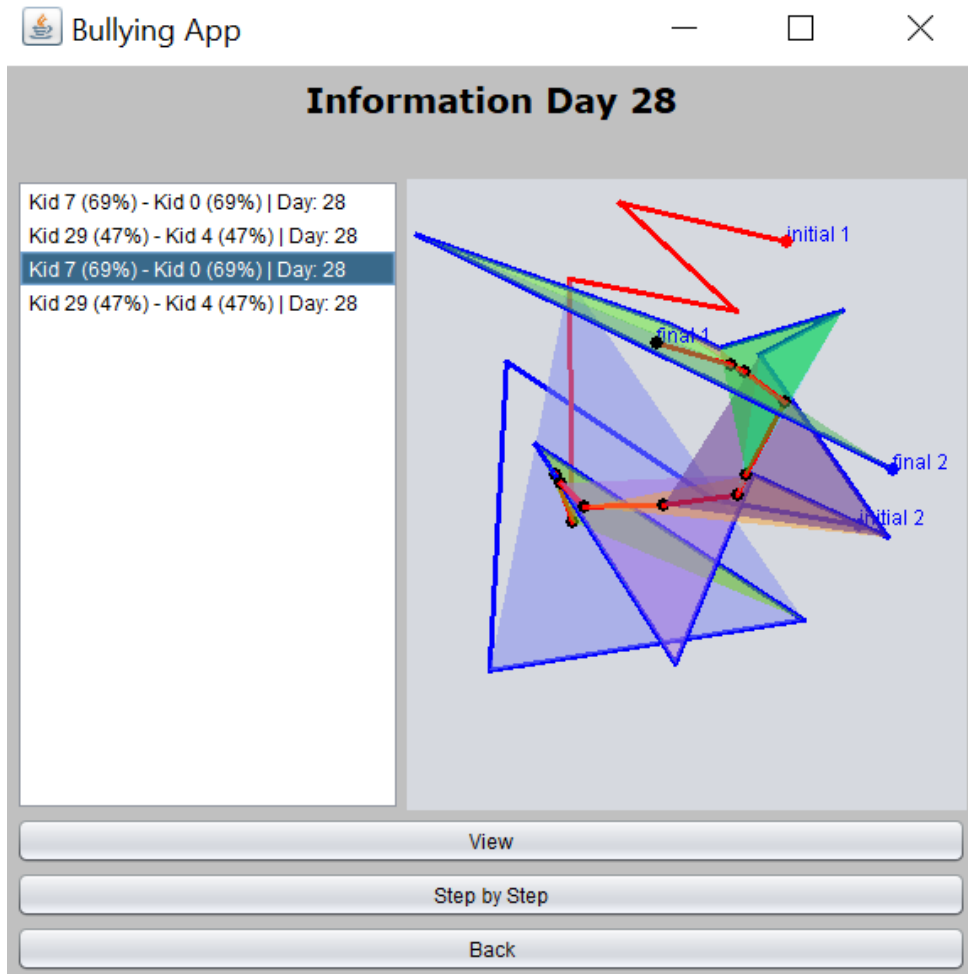


Ilustración 35 Información por día

Por último, si el usuario lo desea, puede ver lo que ha ocurrido los último X días:

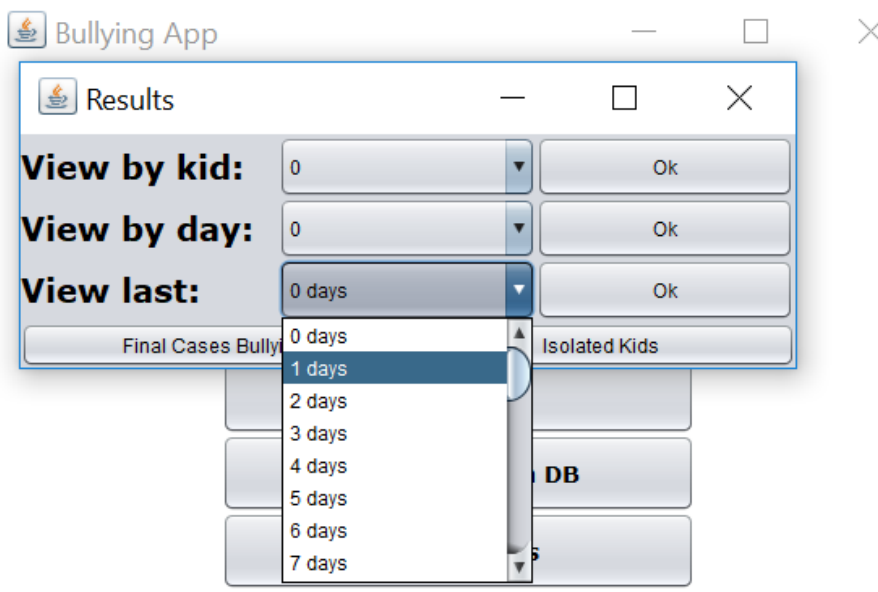


Ilustración 36 Resultados (3)

Veríamos el mismo evento:

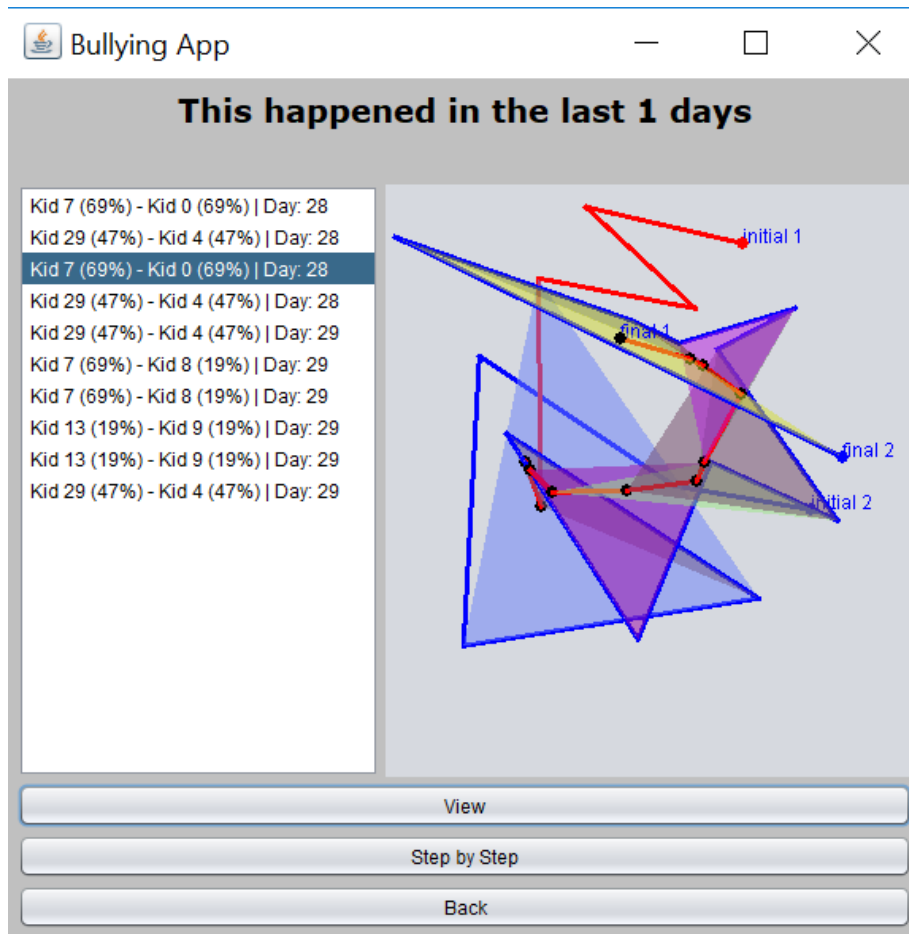


Ilustración 37 Información últimos días

Las últimas dos opciones de las que disponemos son *Final Cases Bullying* y *Isolated Kids*.

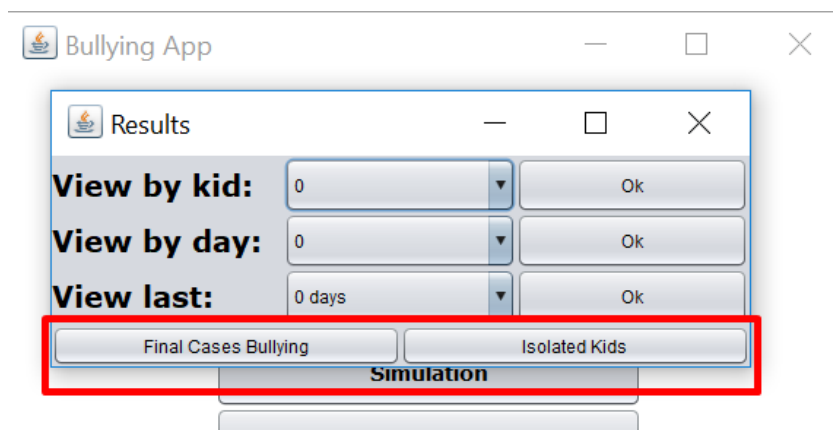


Ilustración 38 Resultados (4)

La primera de ellas es justo lo que indica, tiene una lista final de casos de acoso. La idea es que ésta sea la función principal del sistema y lo que más sea usado por el profesor de colegio.

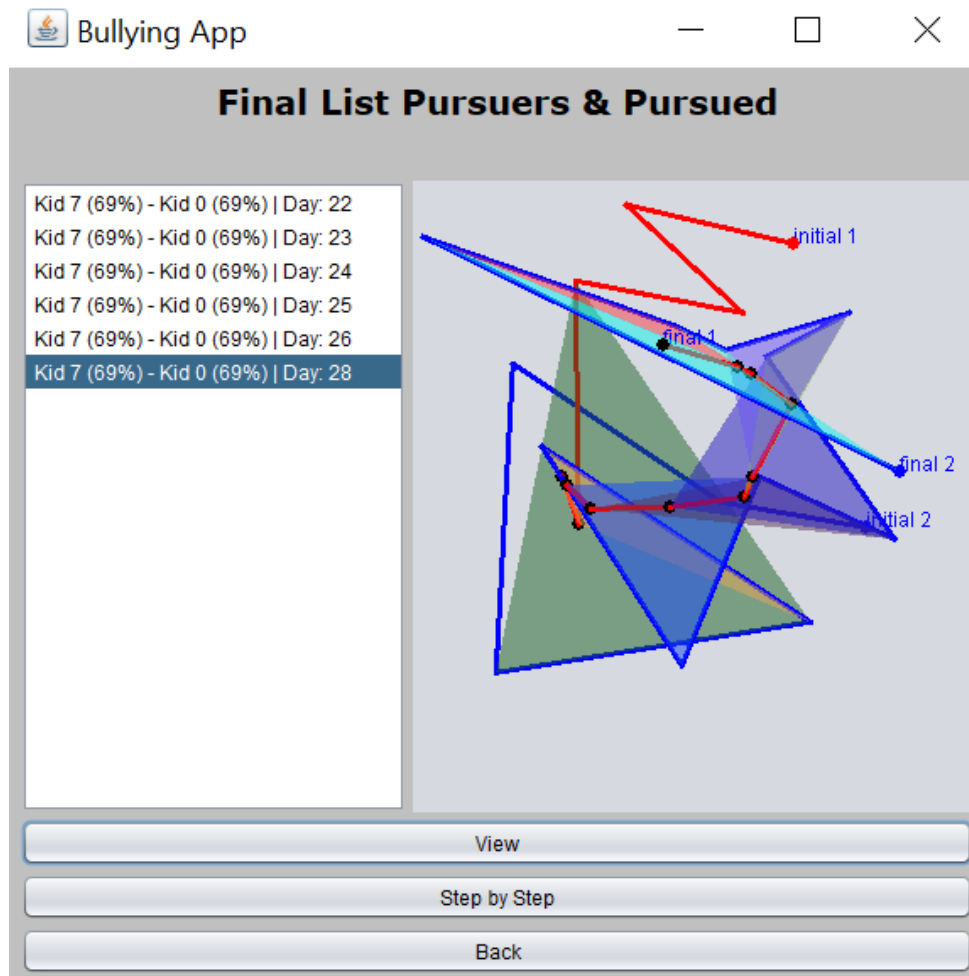


Ilustración 39 Lista final de perseguidores/perseguidos

La opción de *Isolated Kids* nos permite examinar las ocurrencias de aislamiento que ha habido. Al apretar el botón, veremos una estructura similar:

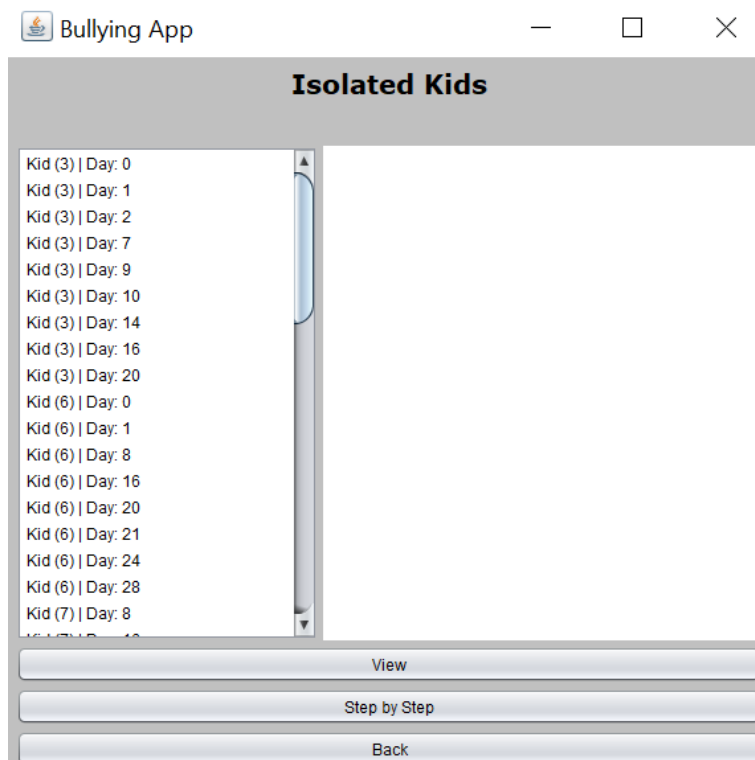


Ilustración 40 Resultados aislamiento

Seleccionando algún evento de aislamiento y apretando *View* podemos ver lo que ha ocurrido ese día.

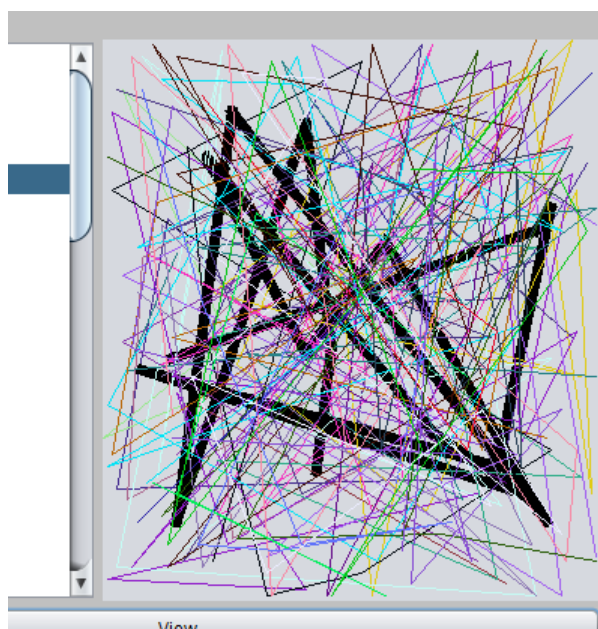


Ilustración 41 Resultados aislamiento (2)

No obstante, esto no es de mucha utilidad para el usuario, y es por ello que se ha incluido el botón *Step by Step* para poder ver lo que ha pasado en cada una de las posiciones en las que ha estado el niño:

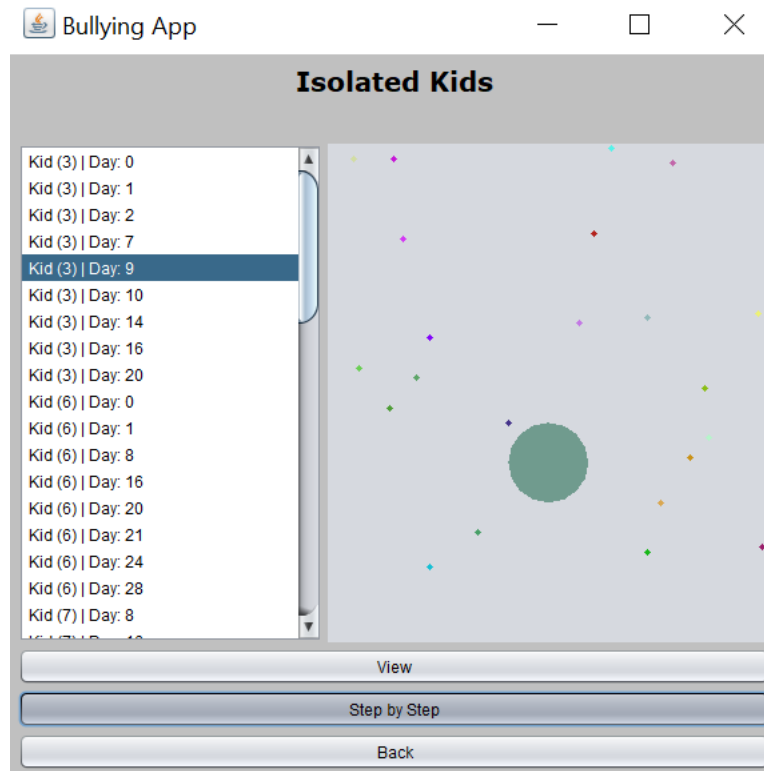


Ilustración 42 Resultados aislamiento (3)

El círculo grande representa el radio de aislamiento. Cada “punto” en el mapa son posiciones de otros niños. Aun así, es posible que algunos de los puntos caigan dentro del área círculo, pero esto es debido a las constricciones que se establecieron anteriormente: no es necesario que el niño esté aislado del 100% de los niños, sino es suficiente en que lo sea del 60%. Asimismo, tampoco tiene que estar aislado en todas sus posiciones, sino en al menos el 90% de ellas.

Recordemos que esto se ha realizado en gran parte para poder tener unos resultados tangibles a partir del simulador. Aumentando estas restricciones, sería quizás más realista, pero a la hora de la detección obtendríamos muy poco resultado ya que, al fin y al cabo, estamos tratando con simulaciones.

5.3 Base de datos

El programa tal y como nos habíamos planteado como meta, permite almacenar toda la información de las simulaciones y las detecciones para volver a obtenerla y representarla. Esto es justo lo que hace *Fetch data from DB*, si apretamos el botón, veremos la misma alerta con los mismos eventos aun después de una ejecución completamente diferente. Poder realizar esto, fue uno de los principales desafíos del proyecto.

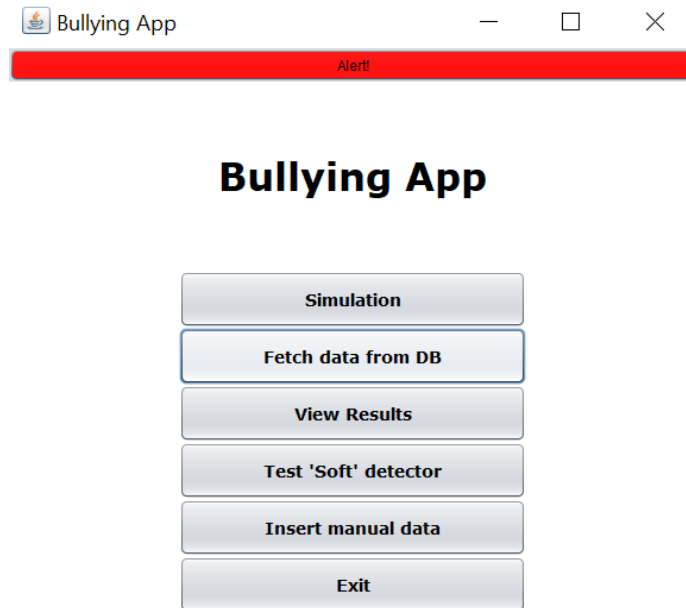


Ilustración 43 Obtención de datos de la BD

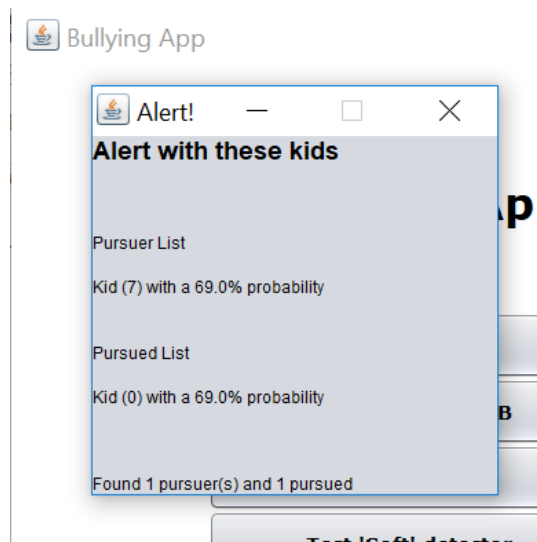


Ilustración 44 Obtención de datos de la BD (2)

5.4 Tests del detector débil

El programa permite incluso dibujar mediante clics de ratón en el panel, trayectorias de niños con tal de probar el detector. Su fin es meramente demostrativo y para sintonizar los parámetros del detector:

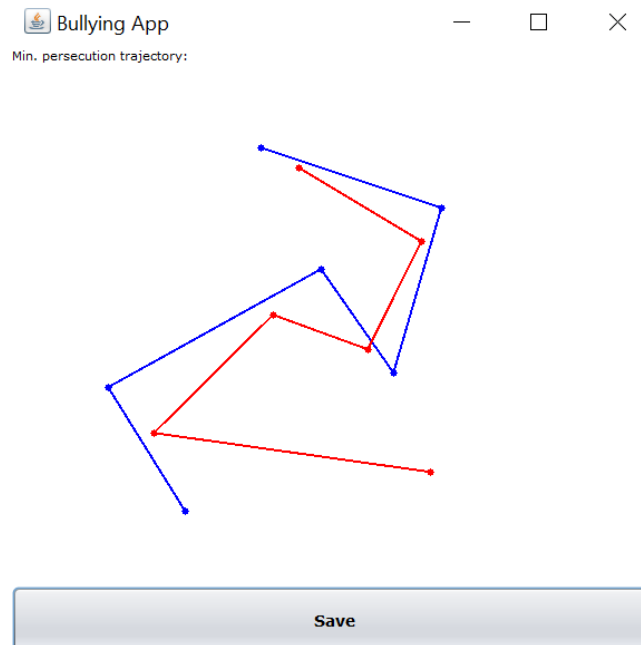


Ilustración 45 Tests detector débil

Nos dice también si ha habido una detección o no:

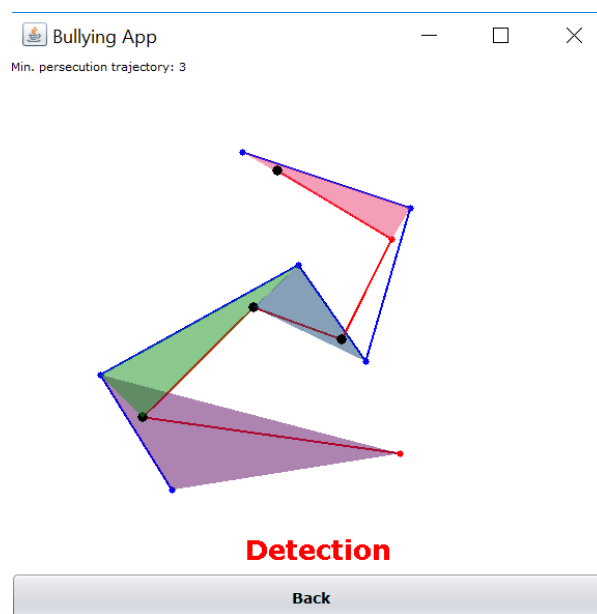


Ilustración 46 Tests detector débil (2)

6 Conclusiones

El programa cumple con los objetivos; detecta posibles situaciones de acoso, tanto de persecución como de aislamiento. Todas las detecciones son justificadas de forma visual y el profesor es capaz de comprobarlo en cualquier momento mediante la interfaz gráfica.

Aun así, a lo largo del proyecto se han encontrado varios problemas y múltiples escollos que han dificultado su realización.

Uno de los principales problemas a la hora de realizar el proyecto, es la inexistencia de datos reales. Esto se ha intentado resolver con el uso de las simulaciones. Uno puede pensar que este enfoque es más eficiente en cuanto al tiempo que se pueda dedicar al proyecto, pero realmente acabó siendo todo lo contrario.

Las simulaciones, a pesar de los intentos de hacerlas más reales, siguen siendo en su mayor parte aleatorias. Esto dificulta drásticamente lo que es la prueba del detector, sobre todo en cuanto al aislamiento, ya que poder tener un niño que se encuentra aislado en la mayoría de sus posiciones de casi todos los niños y durante muchos días, es difícil de replicar con un sistema aleatorio.

Por lo que una gran cantidad de tiempo ha sido dedicada a la creación del simulador, el cual, ha ayudado para probar el detector, pero hubiera sido muchísimo más apropiado trabajar con datos reales obtenidos por ejemplo de un GPS.

6.1 Trabajo futuro

En estos momentos, el programa funciona sobre la plataforma de Java, haciendo uso MySQL. Para emplear el sistema, un usuario necesita de un PC con una versión de JRE versión 8 o mayor y una BD con toda la información [23].

Uno de los objetivos inicialmente era proporcionar esta funcionalidad en forma de aplicación móvil. Dicho de otra manera, permitir al profesor del colegio interactuar de la misma manera con el programa, pero desde su móvil o Tablet por motivos de comodidad.

La aplicación accedería por medio de Internet a un servidor con toda esta información, y sería procesada por el detector. Obviamente no habría ninguna opción de simulación.

El profesor accedería mediante un nombre de usuario y contraseña preestablecidos y guardados en otra tabla de la BD y podría ver si ha habido detecciones o no y ver que probabilidades hay en cada caso. Esto se descartó por falta de tiempo.

También hay muchas posibilidades de cambio en cuanto a la lógica detrás del sistema de detección. Como, por ejemplo, el vínculo entre las persecuciones y el aislamiento. Actualmente, estas funciones funcionan por separado; si un niño suele ser perseguido, su peso de perseguido aumenta, pero también puede estar siendo aislado, en cuyo caso, su peso aumenta, pero no están relacionados. Lo ideal, es que hubiera un sólo peso de acosado (y acosador) y este peso fuera incrementado por eventos de persecución y aislamiento. Lo cual permite ampliar en cuanto a otros posibles tipos de detección, como por ejemplo si el niño es rodeado por otros, etc.

Asimismo, los pesos de cada niño son incrementados, pero no decrementados. Esto quiere decir que un niño puede haber estado siguiendo a otro durante los primeros días de recreo, a una unidad escasa de alcanzar el límite de bullying, pero no se ha vuelto a registrar este evento hasta dentro de 21 días. Entonces, el peso de perseguidor de este niño aumentaría y habría una detección. Sería más razonable que su peso de perseguidor decrementara en una cierta cantidad al no haber ningún registro de la misma situación durante un largo periodo de tiempo.

Otra posibilidad hubiera sido combinar los porcentajes de perseguidor y perseguido. En vez de mostrar dos porcentajes al profesor, es más intuitivo mostrar uno solo que fuera la combinación de los dos.

Obviamente, el alcance del proyecto sería utilizar dispositivos reales que ofrecerían toda la información de posicionamiento de manera encriptada en vez de simuladores. Para ello sería indispensable disponer del apoyo de un colegio que estuviera de acuerdo en que una de las clases de infantil pudiera llevar pulseras con geolocalizador durante los recreos. El profesor responsable de la clase o del recreo, sería el único que sabría a qué niño pertenece cada identificador numérico. Aunque de esto surge una cuestión que es la disposición de los padres a permitir la monitorización de sus hijos, sobre todo si sospechan que su hijo pueda un acosador.

7 Bibliografía y referencias

- [1] Acoso escolar en España. El País:
https://elpais.com/politica/2018/04/30/actualidad/1525119884_318076.html
- [2] Cifras del acoso escolar en España. El País:
https://elpais.com/elpais/2016/02/18/media/1455822566_899475.html
- [3] Efectos del bullying. Stopbullying.gov (EEUU):
<https://espanol.stopbullying.gov/en-riesgo/efectos/rs7/%C3%ADndice.html>
- [4] Acoso escolar en España. RTVE:
<http://www.rtve.es/noticias/20170419/espana-debajo-media-ocde-acoso-escolar-entre-alumnos-15-anos/1525901.shtml>
- [5] El bullying en la etapa de educación infantil. Noticias de Navarra:
<http://noticiasdenavarra.com/2018/06/13/sociedad/estado/el-bullying-se-gesta-durante-la-etapa-de-educacion-infantil-segun-un-estudio>
- [6] Programa KiVa. Accesible desde:
<http://www.kivaprogram.net/>
- [7] Método Monité. Accesible desde:
<https://monite.org/>
- [8] Aplicación Appvise. Accesible desde:
<https://www.myappvise.com/en/>
- [9] Plataforma ZeroAcoso. Accesible desde:
<https://www.zeroacoso.org/>
- [10] Tecnologías contra el bullying. El Mundo:
<http://www.impulsodigital.elmundo.es/seguridad-tecnologica/asi-son-las-tecnologias-contra-el-bullying>

[11] Formas de bullying. Universidad Internacional de Valencia:
<https://www.universidadviu.com/las-diversas-formas-de-bullying-fisico-psicologico-verbal-sexual-social-y-ciberbullying/>

[12] Leguaje de programación Java:
[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

[13] Java Swing. Oracle. Accesible desde:
<https://docs.oracle.com/javase/tutorial/uiswing/>

[14] MySQL. Oracle. Accesible desde:
<https://www.mysql.com/>

[15] Distribución normal:
https://es.wikipedia.org/wiki/Distribuci%C3%B3n_normal

[16] Desviación típica:
https://es.wikipedia.org/wiki/Desviaci%C3%B3n_t%C3%ADpica

[17] Inteligencia artificial:
https://es.wikipedia.org/wiki/Inteligencia_artificial

[18] Redes Neuronales Artificiales – Rodrigo Salas. Universidad de Valparaíso:
https://s3.amazonaws.com/academia.edu.documents/36957207/Redes_neuronales.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1536151479&Signature=pBoWgfeDCQwyaAHuUOIRL9Q8Wjw%3D&response-content-disposition=inline%3B%20filename%3DRedes_Neuronales_Artificiales.pdf

[19] GPS:
<https://es.wikipedia.org/wiki/GPS>

[20] Posicionamiento de un punto dentro del área de un triángulo. Universidad de Princeton, EEUU:
<http://www.cs.princeton.edu/~funk/tog02.pdf>

[21] Método del triángulo:

<https://www.geeksforgeeks.org/check-whether-a-given-point-lies-inside-a-triangle-or-not/>

[22] Tabla de distribución normal. Universidad de Arizona, EEUU:

<http://math.arizona.edu/~rsims/ma464/standardnormaltable.pdf>

[23] Java SE Runtime. Oracle. Accesible desde:

<http://www.oracle.com/technetwork/java/javase/jre8-downloads-2133155.html>