Título: Distribución de un simulador de incendios forestales a través de SDL

Volumen: 1

Alumno: Fco. Javier Bercero Antiller

Director/Ponente: Pau Fonseca i Casas

Departamento: EIO (Estadística e Investigación Operativa)

Data:

DATOS DEL PRYECTO

Título del Proyecto: Distribución de un simulador de incendios forestales a través de SDL

Nombre del estudiante: Fco. Javier Bercero Antiller

Titulación: Ingeniería Informática

Créditos: 37,5

Director/Ponente: Pau Fonseca i Casas

Departamento: Estadística e Investigación Operativa

MIEMBROS DEL TRIBUNAL *(nombre y asignatura)*

Presidente: José Casanovas García

Estadística e Investigación Operativa

Vocal: José Antonio Lubary Martínez

Matemática Aplicada II

Secretario: Pau Fonseca i Casas

Estadística e Investigación Operativa

CALIFICACIÓN

Calificación numérica:

Calificación descriptiva:

Fecha:

Índice

1. INTRODUCCION 8
1.1 Presentación 8
1.2 Motivación 8
1.3 Problemática 9
1.4 Objetivos 9
2 MARCO TEORICO 11
2.1 Simuladores 11
2.2 Características del fuego y de los incendios forestales 11
2.3 Simuladores del comportamiento del fuego 12
2.4 Simuladores actuales 16
2.5 Lenguaje SDL 21
2.6 Sistemas de información geográfica 24
2.7 Idridi32 26
2.8 SDLPS 27
2.9 Firelib 28
2.10 Autómatas celulares 30
3 MODELADO y DISEÑO 33
3.1 Particularidades del modelo 33
3.1.1 Extensiones 34
3.2 Especificación con el lenguaje SDL 35
3.3 Especificación del modelo 35
3.3.1 Diagrama del Sistema 36
3.3.2 Diagrama del MNCA_Fuego 38

3.3.3 El procedimiento Vecinity 41
3.3.4 El procedimiento Nucleus 45
3.3.5 Diagrama de Bloques 48
3.3.6 Diagrama de Estados 49
3.3.7 Diagrama de Procesos 51
3.4 Generación XML 65
4 IMPLEMENTACION 66
4.1 Elección lenguaje programación 66
4.2 SDK SDLPS 66
4.3 Diagrama de clases 68
4.4 Simulación distribuida 80
4.4.1 Ley de Amdahl 81
5 EXPERIMENTACION 84
5.1 Estructura del simulador 84
5.2 Verificación, validación y experimentación del modelo 84
6 PLANIFICACIÓN Y COSTES 88
6.1 Fase documentación 88
6.2 Fase Modelado y Diseño 89
6.3 Fase Implementación 89
6.4 Fase Experimentación 90
6.5 Fase de la memoria 90
7 CONCLUSIONES 94
8 TRABAJO FUTURO 96
9 ANEXOS 97
9.1 Problemas 97

9.2 Formato idrisi32 97
9.3 XML generado 98
9.4 Fichero XSD 115
9.5 Resuldados simulación Behave 127
9.6 Resultados simulación PFC no distribuido 139
9.7 Guía de estilo de programación 151
9.8 Sandrila 152
9.9 Referencia técnica y manual Firelib 153
10 GLOSARIO 154
11 ÍNDICE DE FIGURAS 156
12 BIBLIOGRAFIA 159

1. INTRODUCCION

1.1 Presentación

El proyecto que se presenta es un trabajo en el marco de la simulación y de los autómatas celulares, propuesta de Pau Fonseca, director del proyecto y profesor del departamento de Estadística e Investigación Operativa (EIO), con el objetivo de modelar el comportamiento de los incendios forestales en lenguaje SDL (basado en el modelo ya existente de Behave) y en base a esto elaborar la infraestructura para la creación de un simulador de incendios forestales para que se pueda ejecutar de manera distribuida.

1.2 Motivación

Existe un gran esfuerzo por parte del hombre en el estudio y el control de los incendios forestales ya que tienen gran importancia ecológica y económica. En las últimas décadas como consecuencia del desarrollo y la evolución de la sociedad los incendios implican pérdidas muy valiosas e importantes.

La pérdida de bosques y áreas forestales provoca la disminución de reservas naturales, cambios climáticos, falta de espacios que generen oxígeno, extinción de animales, entre otras… e incluso la más importante que no debemos de olvidar, la pérdida de vidas humanas.

Todo esto afecta a la vida en el planeta y disminuye la calidad de vida, humana, animal y vegetal. Como existe este riesgo es necesario conocer el fenómeno, identificarlo y entender los elementos que intervienen en los incendios forestales.

Este conocimiento contribuirá a la lucha contra los incendios principalmente de tres maneras. La primera, se podrán tomar medidas preventivas, de esta manera se podrá tomar medidas antes de que suceda, con la consecuencia de reducir el impacto. La segunda, se mejorara la efectividad en la lucha del frente del fuego durante un incendio y la tercera contribución será que se podrán obtener mapas de riesgos de zonas determinadas, gracias a la información que ya poseemos de condiciones meteorológicas, información del terreno o incluso factores humanos.

Lo que hace este proyecto más interesante es que es un proyecto real y en un futuro cercano se podrá utilizar en simulaciones de casos reales, incluso llegar a un mercado abierto para cualquiera, tanto el mundo universitario como laboral, ya que no existe ningún simulador como este.

1.3 Problemática

Uno de los problemas es que el comportamiento de un incendio forestal es una tarea muy compleja, ya que depende de muchas variables y elementos, por lo que se decidió utilizar el modelo del comportamiento del fuego ya existente de Behave que está bien definido.

A pesar que hay algunos simuladores de incendios, no hay ninguno que utilice el lenguaje SDL, por lo que para un investigador sin conocimientos informáticos, le es difícil de entender el modelo que está utilizando, en cambio con el lenguaje SDL, al ser más gráfico, es mucho más directo y fácil de entender, por ese motivo se eligió este lenguaje para realizar el modelo.

Otro problema que nos encontramos en este proyecto es que los simuladores para realizar una simulación en condiciones reales necesita mucha información, lo que nos provoca que al manejar toda esta información se tenga que hacer muchos cálculos por lo que se hace imposible ejecutar una simulación real en un pc de sobremesa, por lo que se optó por realizar la ejecución de manera distribuida para que se pueda ejecutar por ejemplo en los superordenadores que hay repartidos por todo el mundo, como ejemplo el que tenemos en Barcelona el Mare Nostrum.

El último problema que nos encontramos es que al realizar este proyecto se utilizarán herramientas desarrolladas, incluso aún en desarrollo de otros proyectos de la universidad (dos proyectos de la UPC y uno de otra universidad) y no sabemos qué problemas nos podrán llegar a provocar, los cuales se irán arreglando sobe la marcha.

1.4 Objetivos

El objetivo del proyecto se podría dividir en varias tareas a conseguir.

La primera tarea sería la de buscar y analizar toda la información necesaria sobre el comportamiento del fuego, pero no de un modelo cualquiera, ni un modelo realizado desde cero, sino que utilizaremos un modelo sobre el comportamiento del fuego que ya existe, y es muy usado en las simulaciones de incendios forestales más comunes, más en concreto el modelo elegido será el modelo de Behave.

El segundo objetivo sería coger este modelo de Behave y transformarlo al lenguaje SDL y a su vez, comprobar que el comportamiento tanto en el modelo Behave como en nuestro modelo en lenguaje SDL es el mismo.

Para finalizar, el último objetivo que pretendemos alcanzar, será el de implementar este modelo en lenguaje SDL, estructurándolo de tal manera que se pueda ejecutar una simulación de manera distribuida, en varias maquinas a la vez.

Validar el modelo no es parte de este proyecto aunque los resultados obtenidos se tendrán que aproximar a valores que darían en la realidad y que ya están validados en el modelo de Behave.

2 MARCO TEORICO

2.1 Simuladores

Los simuladores son herramientas que nos permiten la emulación de un sistema, es decir, reproducir el comportamiento de un sistema, reproducen sensaciones que en realidad no están sucediendo. Por lo general se basan en que tienen unas ciertas entradas (normalmente recogidas de datos reales) que describen el sistema y sus características, donde un proceso va desarrollando su comportamiento paso a paso y al final obtiene una salida con un resultado.

Las ventajas de los simuladores son varias, nos permiten observar diversos fenómenos que no resultan fáciles de ver ya sea porque no se sabe cuando y donde van a ocurrir u otros factores. Los simuladores nos permiten predecir los fenómenos, si tenemos hecha una simulación podemos entender su proceso, tomar medidas y actuar en consecuencia. El estudio de estos fenómenos, no entraña efectos negativos al reproducirlos ya que son simulaciones y podemos reproducirlo tantas veces como queramos y bajo los efectos que se deseen, sin ningún tipo de límite.

En los incendios forestales es común usar simuladores para predecir el comportamiento del fuego y de esta manera poder tomar medidas para disminuir el impacto del incendio e incluso tomar medidas preventivas más efectivas. No siempre todo es tan bonito en los simuladores de este tipo porque no siempre las simulaciones son validas ya que muchas veces la salida de la simulación difiere de la real ya que pueden intervenir factores que alteren la simulación con la realidad.

2.2 Características del fuego y de los incendios forestales

El fuego es una reacción química producida por la ignición y la combustión materiales, para que esto ocurra son necesarios tres elementos, el combustible a quemar, el aire para obtener oxigeno y una fuente de calor que sea capaz de llevar la fuente de calor a la ignición. Reduciendo uno de estos tres elementos es posible controlar y extinguir un incendio.

Los incendios suelen iniciarse en un foco y el fuego se propaga y se extiende debido a diversos factores como la convección, radiación y conducción del calor que provocan

las llamas. La principal fuente de incendios son: los rayos, las negligencias, causas fortuitas, intencionadas o reproducciones de incendios anteriores (el 90% de los incendios provocados por los humanos).

Hay tres tipos de incendios: Los incendios de superficie, donde el incendio se propaga por la superficie del terreno, se quema combustible que está en el suelo (hojas secas, ramas caídas, etc.), en este tipo de incendio el fuego suele propagarse rápido. Los incendios de copas, son los que se propagan a través de las copas de los arboles, en estos las llamas alcanzan grandes alturas y se propagan de la copa de un árbol a otro mediante la conducción. Por último estaría los incendios de subsuelo, el cual se propaga por debajo de la superficie del terreno, el fuego quema y se propaga a través de las raíces y materia orgánica seca.

Para cada tipo de incendio tenemos características tanto de propagación como comportamiento de fuego diferente por lo que nosotros nos basaremos en las de superficie que son las cuales se basa Behave.

2.3 Simuladores del comportamiento del fuego

Normalmente los simuladores utilizados en este campo intentan emular como avanza el fuego en un incendio forestal, tomando características del ambiente donde se desarrolla el incendio en un determinado momento y lo que hacen es tratar de simular como se propagará en un instante de tiempo posterior.

Anteriormente explique los tres tipos de incendios que podían suceder en un área forestal, de superficie, de copas y subterráneas. Aunque los tres tipos podrían desarrollase simultáneamente (eso significa que podrían influir entre ellos) en nuestro caso solo se contemplara un tipo, los de superficie. Esto es porque el modelo definido por Behave que será el que modelemos se centra solo en este tipo de incendio sin tener en cuenta el resto. Al igual que en el momento de un incendio se hay muchos factores que determinan como se va a propagar el incendio, como puede ser la pendiente, la vegetación, el clima, la humedad, etc. Todos estos factores y muchos otros interactúan entre sí e influyen en la propagación del fuego, naturalmente un simulador completo debería de tener todas estas consideraciones y relaciones, pero el

tiempo de este proyecto no permite abarcarlas todas, por lo que se utilizará el modelo Behave que está más simplificado. Además si se quisiera obtener resultados en un tiempo razonable en un simulador completo, por su complejidad nos sería imposible por lo que por lo general se suele optar por simular solo un tipo de incendio (y asumiendo ciertas características del entorno), esto nos hace que los simuladores no sean tan precisos como realmente se desea, intentado en todo momento buscar la máxima precisión-tiempo más conveniente.

Los incendios forestales tienen unas ciertas características que hay que tener en cuenta y que son las siguientes.

El tipo de combustible, el clima y la topología del sitio donde se desarrolla el incendio determina como será su propagación. Estos factores hará que el fuego no se propague, que se extinga o por el contrario que el fuego crezca y se propague más rápidamente.

La cantidad y tipo de combustible inflamable que rodea el incendio se conoce como carga (cantidad de combustible por unidad de área), a más carga el fuego se propagará más rápido y con una mayor intensidad.

Otro factor determinante que influye en la propagación del fuego es el tamaño del material que conforma el combustible, si es más pequeño por ejemplo será más fácil de calentar y de quemar que un material grande que tardaría más en ambos procesos (esto se debe a la relación entre la superficie total del material y su volumen).

La humedad también afecta a la velocidad en que el material es quemado, a mayor humedad mayor será el tiempo en que la fuente de calor seque el combustible y hacer que alcance su temperatura de ignición (un material con alto contenido de humedad absorbe el calor del fuego haciendo más difícil su propagación). Por ejemplo las temporadas de altas precipitaciones provocarán que haya un alto contenido de humedad lo que hará que sea más difícil que se propague el fuego, por el contrario si es época de sequia amplia la posibilidad de producirse un incendio y si lo hay que se propague más rápido.

La temperatura de ambiente también influye ya que provoca que el combustible esté más o menos calientes, secos o menos secos, dependiendo si es época de temperaturas altas o no.

El viento tiene gran importancia también porque determina la forma en que avanza el fuego, a más intensidad de viento, más rápida será la propagación ya que empuja las llamas acelerando su propagación. Incluso si el viento es bastante intenso determinará la dirección de propagación. También puede arrastrar combustible encendido y crear otros focos de incendio. Este factor es bastante impredecible, rara vez sus ráfagas son constantes en tiempo e intensidad.

La pendiente del terreno afecta porque determina la velocidad y dirección de propagación del fuego. El fuego se propaga más rápidamente subiendo la pendiente, ya que el huma y el calor suben por la pendiente, precalentando el combustible y haciéndolo más fácil de quemar.

El siguiente cuadro muestran los factores principales que influyen (favorecen o no) en la propagación:

Factor	Influencia
Carga	Favorece
Tamaño	No favorece
Humedad	No favorece
Temperatura	Favorece
Viento	Favorece
Pendiente	Favorece

Los modelos de propagación utilizan características del entorno donde se desarrolla el fuego para determinar el comportamiento del fuego y los índices de riesgo. Las aéreas forestales suelen encontrarse gran variedad de especies vegetales pero es bastante común que haya un tipo de material determine como se propagaría el fuego en caso de incendio y eso es precisamente lo que hacen los modelos de combustible, intentan representar estos materiales para simplificar su uso en de los modelos matemáticos de propagación del fuego.

Actualmente se utilizan trece modelos para describir los tipos de combustible, divididos a su vez en cuatro clases: pastos, arbustos, leñosos y ramas. Cada uno de estas clases propaga el fuego de forma distinta. Estos modelos describen las características de los combustibles que son justamente las entradas de los modelos matemáticos del comportamiento del fuego.

La siguiente tabla muestra los trece modelos de combustible:

Modelo	Grupo
1. Pasto Corto	Pastos
2. Pasto y arbustos (pequeños)	Pastos
3. Pasto alto (o paja)	Pastos
4. Matorral	Arbustos
5. Arbustos pequeños	Arbustos
6. Arbustos con ramas pequeñas	Arbustos
7. Arbustos con muchas ramas	Arbustos
8. Pasto u hojas cortas (debajo arboles)	Leñosos
9. Arbustos pequeños (debajo arboles)	Leñosos
10. Arbustos secos (debajo arboles)	Leñosos
11. Algunas ramas caídas	Ramosos
12.Más ramas caídas	Ramosos
13. Muchas ramas caídas	Ramosos

Saber si realmente es posible predecir el comportamiento del fuego, depende de la precisión que se busca en la predicción, hay tantos factores a tener en cuenta y que influyen en mayor o menor medida al comportamiento del fuego que hace que se descarte la idea de predicciones absolutas.

Richard Rothermel ha definido un modelo de comportamiento del fuego asumiendo que el mismo avanza sobre superficies con combustible continuo. Este modelo evalúa la energía generada por el fuego, la transmisión de calor desde el fuego al combustible próximo al fuego y la energía absorbida por el combustible, se consideran los combustibles muertos y vivos, sus contenidos de humedad, el efecto del viento, la pendiente en la transferencia de calor, la carga, lo compacto que esta el combustible,

el tamaño de la partículas del combustible, etc. Todo esto determina la forma y velocidad en que el fuego se propaga. No se considera ni modela el fuego de copas, torbellinos de fuego y la creación de más focos de fuego.

El modelo de Rothermel es uno de los más utilizados para la predicción del comportamiento del fuego, la mayoría de los simuladores del comportamiento se basan sus cálculos en este modelo que a grandes rasgos calculan el índice de máxima propagación y la intensidad de reacción del fuego conociendo ciertas propiedades del combustible y del ambiente.

2.4 Simuladores actuales

En la actualidad existen diversos simuladores (Fire Behavior and Fire Danger Software s.f.) Para predecir el comportamiento del fuego, a continuación expondré los más utilizados.

BehavePlus: Se considera el sucesor de Behave (Andrews 1986), es un simulador que se ejecuta bajo Windows. Behave Plus es una colección de modelos que describen el comportamiento del fuego, su entorno y sus efectos. Es un programa recomendado para usuarios experimentados o al menos familiarizados con el problema, aunque tiene la opción de utilizar una serie de ventanas que permiten el ingreso de los parámetros y muestra los resultados de una forma amigable y fácil de utilizar. Muestra relaciones entre humedades, viento, pendiente e índice de propagación de forma gráfica después de la simulación. Los resultados son mediante gráficas, tablas, diagramas y puede ser utilizado por programas externos.

Inputs: SURFACE, SIZE

Description		Compare fire behavior for 4 fuel models
Fuel/Vegetation, Surface/Understory		
Fuel Model		2, 5, gr2, sh2
Fuel Moisture		
Dead Fuel Moisture	%	5
Live Fuel Moisture	%	75
Weather		
Midflame Wind Speed (upslope)	mi/h	4
Terrain		
Slope Steepness	%	0
Fire		
Elapsed Time	h	1

Figura 1: Entrada de datos en BehavePlus

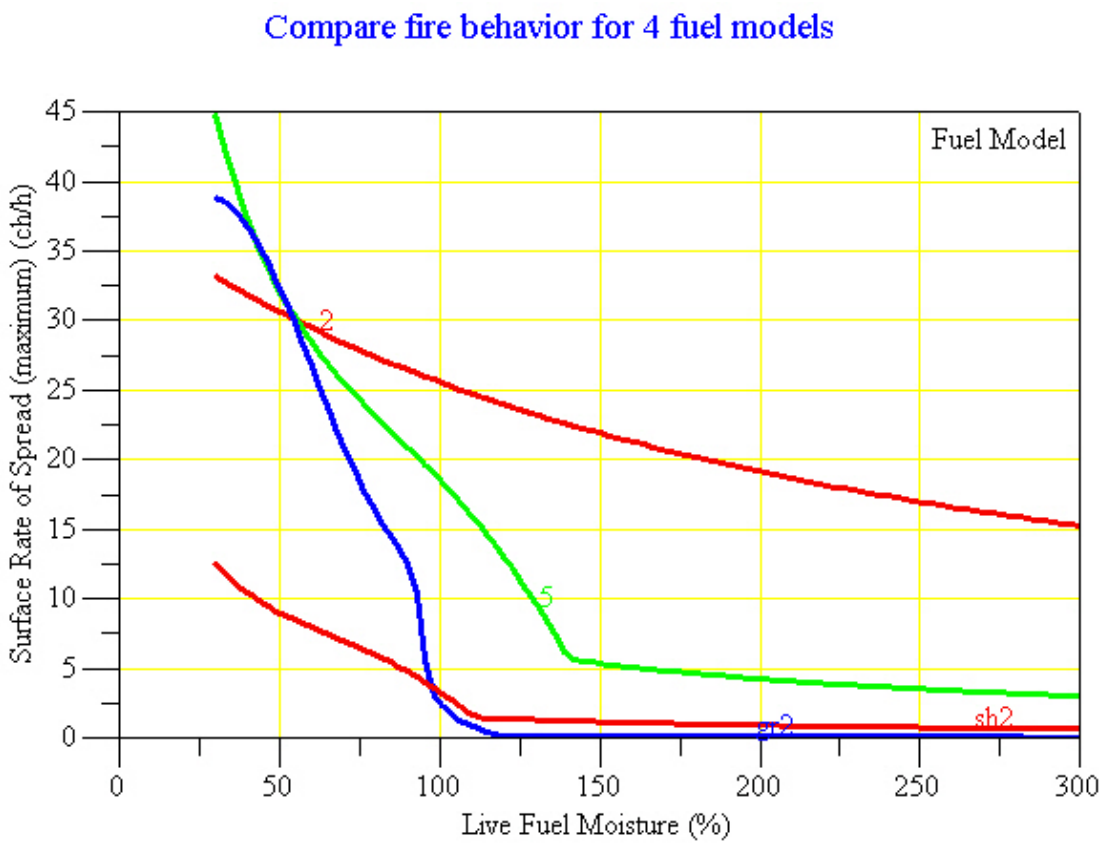

Figura 2: Salida de BehavePlus, comparativa de 4 modelos de combustible

Farsite: Es un modelo de simulación de crecimiento del fuego, es un simulador que se ejecuta bajo Windows y tiene una interfaz gráfica. Necesita como entrada información de sistemas de información geográfica. Utiliza información sobre la tipología, el clima,

el viento y el combustible, este simulador incluye modelos de propagación en superficie, en copas y spotting, realiza las simulaciones en dos dimensiones. Es un modelo para la simulación en tiempo y espacio de la propagación y comportamiento de fuegos bajo condiciones de terreno, combustible y clima heterogéneas. Está basado en un principio de propagación de ondas propuesto por Huyens. Su salida es la proyección del perímetro del frente del fuego y el comportamiento del fuego, pudiéndose exportar a otras aplicaciones o sistemas de información geográfica. Es una aplicación orientada a usuarios con conocimientos sobre combustibles, clima, topologías, situaciones de incendios, etc.

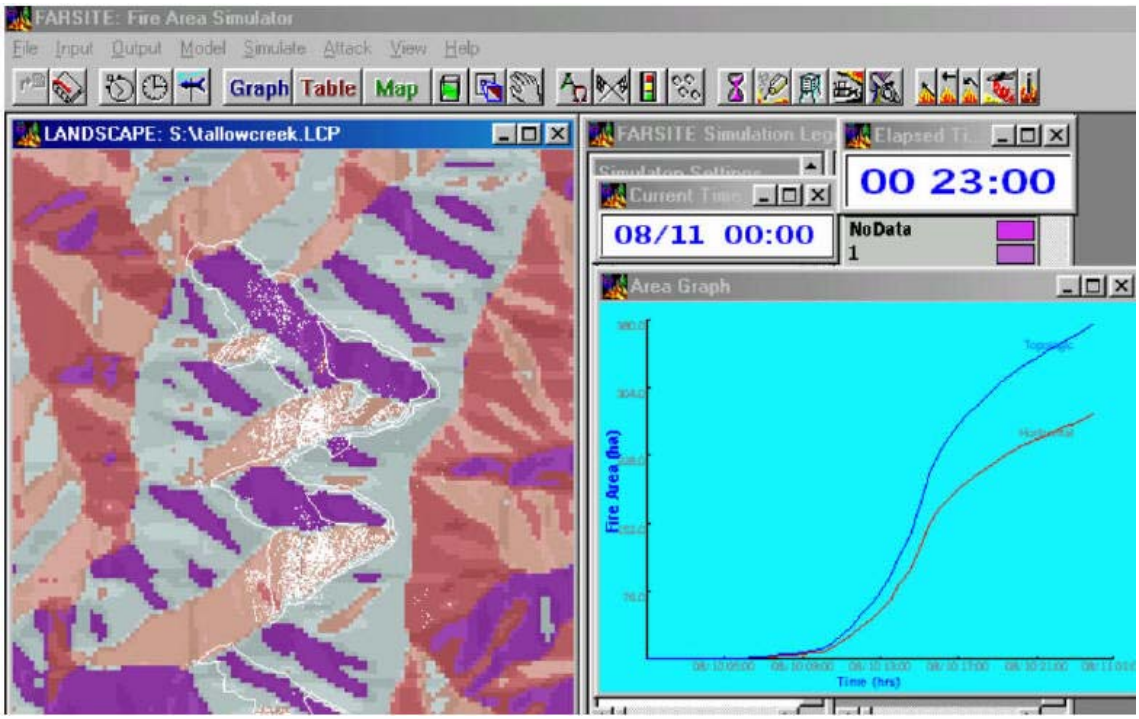

Figura 3: Pantalla de simulación en Farsite

Nexus: es una aplicación que conecta modelos de predicción de fuegos de copas y superficies, es útil para estudiar el riesgo de copas potencial y formas alternativas de tratarlos. Contiene también herramientas visuales que son útiles para entender la interacción entre los incendios de copas y superficies. Está destinado a usuarios familiarizados con Behave o BehavePlus, usuarios que ya posean un conocimiento sobre el comportamiento del fuego y sobre los modelos de fuego de copas.

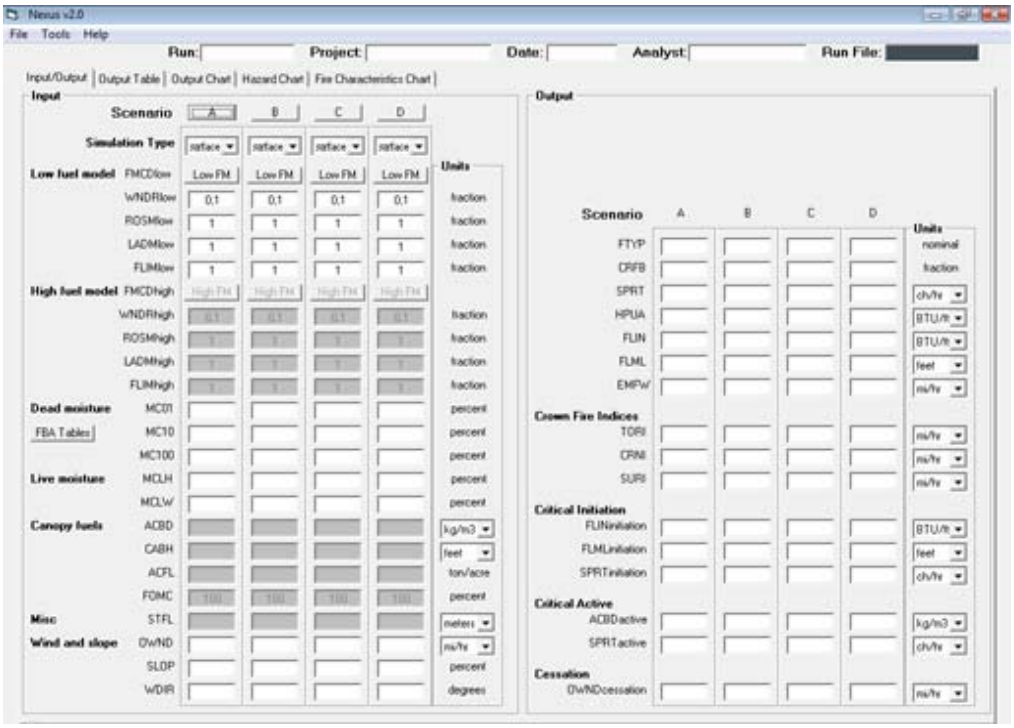

Figura 4: Entada de datos en Nexus

Fofem: es una aplicación que modela los efectos, tanto directos como indirectos, que son consecuencia de los incendios. Establece los efectos secundarios como la mortalidad de los arboles, consumición de combustible, el humo o el calor del suelo. Es útil para planificar mejor los incendios controlados.

Figura 5: Gráfico de combustible en Fofem

Después de ver unos cuantos simuladores que están actualmente en el mercado, podemos ver que todas las aplicaciones necesitan que el usuario tenga un alto grado de conocimiento sobre el comportamiento del fuego y conocer los modelos que utilizan estas aplicaciones. El resultado de las simulaciones, en general, son bastante pobres gráficamente y la simulación se tiene que realizar en una única computadora.

Por eso las ventajas que tiene el simulador de este proyecto respecto del software actual son:

- A nivel de usuario, es más fácil entender cómo funciona el modelo, ya que se presenta de forma gráfica en diagramas SDL, no es condición necesaria que el usuario sea un experto en incendios.
- El resultado de la simulación se podrá ver en VRML ó en Google Earth, más visual que las tablas y gráficos que muestran las aplicaciones actuales.
- Permite ejecutar la simulación de forma distribuida, ninguna de las ofertas actuales lo permite.

No todo son ventajas claro está, como desventaja principal tiene, que de momento solo se está desarrollando el modelo para incendios de superficie, en trabajos futuros se pueden seguir implementando los demás tipos de incendios.

2.5 Lenguaje SDL

Uno de los lenguajes que encontramos para formalizar modelos de simulación es el lenguaje SDL (Telecommunication standardization sector of ITU 1999), es un lenguaje orientado a la especificación y diseño de sistemas de telecomunicaciones que funcionan en tiempo real.

Existen dos formas sintácticas diferentes para representación de sistemas mediante SDL, la primera es SDL/GR (Graphical Representation), la cual es un leguaje gráfico que define la estructura y flujos del control del sistema. La segunda representación es la SDL/PR (Phrase Representation), donde este es un lenguaje de programación. Para la realización de este proyecto hemos usado la representación SDL/GR ya que nos permite trabajar mejor a la hora de crear el modelo de la propagación del incendio forestal.

Este lenguaje se compone de entidades en forma de jerarquía, donde la principal es el sistema, el cual está compuesto por bloques. Los bloques están conectados entre sí y con el entorno mediante canales (que transportan señales). Cada bloque está compuesto por procesos y cada procesos está definido por una maquina de estados finita extendida, con variables, parámetros, eventos y temporizadores. Los procesos se comunican también se comunican entre ellos por canales que transportan señales.

El sistema es la entidad de más alto nivel y en ella podemos encontrar todo lo que queremos modelar, está separado del entorno por la frontera del sistema, el cual se comunica mediante canales que transportan señales. El sistema estará compuesto por uno o varios bloques.

Los bloques tienen una o más definiciones de procesos del sistema. El bloque se encarga de agrupar procesos que realizan cierta función, estos bloques también se comunican mediante canales que transportan señales.

Los procesos están al nivel más bajo de la especificación y se representan como una máquina de estados finita extendida, la cual contiene unos ciertos estados finitos y existen transiciones para pasar de un estado a otro. Cada transición viene producida por una señal que viene de otro proceso o del entorno y que permiten manipular variables locales del proceso, enviar señales o incluso ejecutar procedimientos. Pueden existir varias instancias de un mismo proceso ejecutándose concurrentemente o de instancias de procesos distintos, los procesos tienen acceso al tiempo absoluto para realizar cálculos de tiempos. Hay señales de entrada y de salida, para la recepción de señales los procesos tienen una única cola de señales por proceso y se van suministrando mediante FIFO (la primera que entra es la primera en salir).

Respecto al canal que transporta señales hay que decir que esta comunicación entre bloques (o el entorno) puede ser en ambos sentidos, pero nunca puede haber un canal que empiece y termine en un mismo bloque. Las señales son un flujo de información a transportar.

La manera de representar todo esto es la siguiente:

Representación SDL/GR

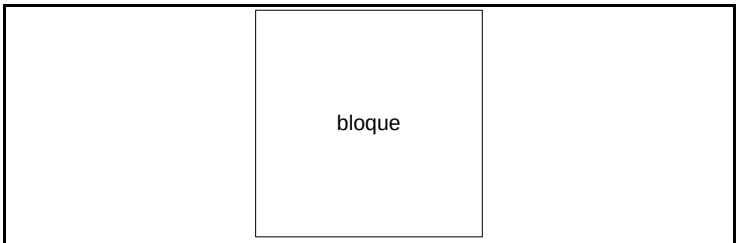

Figura 6: Bloque

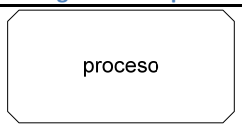

Figura 7: Proceso

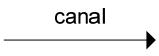

Figura 8: Canal

Figura 9: Señal

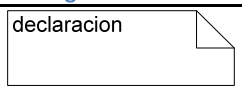

Figura 10: Declaración

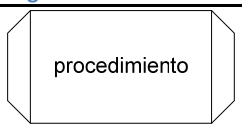

Figura 11: Procedimiento

Figura 12: Comentario

Figura 13: Estado

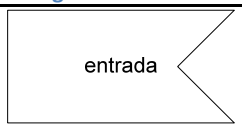

Figura 14: Evento de entrada

Figura 15: Evento de salida

Figura 16: Tarea

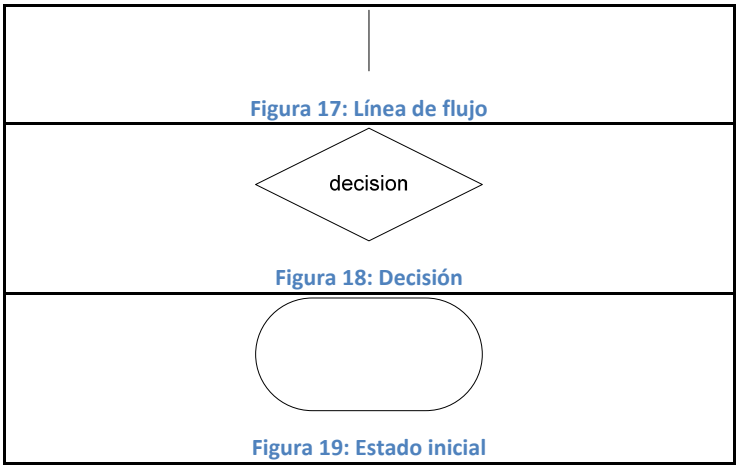

Figura 17: Línea de flujo

Figura 18: Decisión

Figura 19: Estado inicial

2.6 Sistemas de información geográfica

Un Sistema de Información Geográfica (SIG o GIS, en su acrónimo inglés Geographic Information System) es una integración organizada de *hardware*, *software* y datos geográficos diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

La tecnología de los Sistemas de Información Geográfica puede ser utilizada para investigaciones científicas, la gestión de los recursos, gestión de activos, la arqueología, la evaluación del impacto ambiental, la planificación urbana, la cartografía, la sociología, la geografía histórica, el marketing, la logística por nombrar unos pocos. Por ejemplo, un SIG podría permitir a los grupos de emergencia calcular fácilmente los tiempos de respuesta en caso de un desastre natural.

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos

gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Los datos SIG representan los objetos del mundo real (carreteras, el uso del suelo, altitudes). Los objetos del mundo real se pueden dividir en dos abstracciones: objetos discretos (una casa) y continuos (cantidad de lluvia caída, una elevación). Existen dos formas de almacenar los datos en un SIG: raster y vectorial.

Un tipo de datos raster es, en esencia, cualquier tipo de imagen digital representada en mallas. El modelo de SIG raster o de retícula se centra en las propiedades del espacio más que en la precisión de la localización. Divide el espacio en celdas regulares donde cada una de ellas representa un único valor.

En un SIG, las características geográficas se expresan con frecuencia como vectores, manteniendo las características geométricas de las figuras. En los datos vectoriales, el interés de las representaciones se centra en la precisión de localización de los elementos geográficos sobre el espacio y donde los fenómenos a representar son discretos, es decir, de límites definidos. Cada una de estas geometrías está vinculada a una fila en una base de datos que describe sus atributos. Para modelar digitalmente las entidades del mundo real se utilizan tres elementos geométricos: el punto, la línea y el polígono.

Las ventajas de utilizar el raster es que las estructura de los datos es muy simple, las operaciones de superposición sin sencillas, el formato es óptimo para variaciones altas de datos y el buen almacenamiento de imágenes digitales.

Las desventajas de los datos raster son que necesitan mayor requerimiento de memoria, las reglas topológicas son más difíciles de generar y las salidas graficas son menos vistosas y estéticas.

Las ventajas de los datos vectoriales son que la estructura de datos es más compacta, hay una codificación eficiente de la topología y las operaciones espaciales, tiene una buena salida grafica, tienen mayor compatibilidad con entornos de bases de datos relacionales y permite una mayor capacidad de análisis.

Las desventajas de los datos vectoriales son que la estructura de datos es más compleja, las operaciones de superposición son difíciles de implementar y de representar, pierde eficacia cuando la variación de datos es alta, es un formato más difícil de mantener actualizado.

Ninguno de los dos tipos de datos es mejor que el otro, simplemente tienen utilidades diferentes, en nuestro caso utilizaremos los datos raster que son los que necesitamos para la simulación.

2.7 Idridi32

Idrisi23 es un sistema de información geográfica principalmente de datos raster, aunque se puede incorporar información vectorial, pero que posiblemente, según el problema, también se acabará transformando en datos raster.

Este sistema se basa en dos tipos de ficheros, uno para dos y otro para metadatos. Los ficheros de datos contienen la tabla bidimensional coordenada – valor del atributo, mientras que los fichero de metadatos contienen la información sobre los propios datos, el nombre, su origen, la precisión, etc., viene a ser como la documentación del fichero de datos. Cada uno de estos ficheros es almacenado por separado y con extensiones diferentes.

Los ficheros de datos se representan mediante una lista secuencial de valores, donde el primer valor será el que represente la celda superior izquierda de la tabla y el último la celda inferior derecha.

El fichero de metadatos es el que se encarga guardar la información de los datos, de decir cuántas filas y columnas tiene la tabla y de cómo se han de representar la tabla de manera bidimensional.

Respecto a la manera que tiene idrisi32 de representar los datos de modo vectorial, decir que como tiene que reconocer, puntos, líneas y polígonos, todos estos han de estar en ficheros diferentes, no detallaré como están compuestos porque no se utilizarán para este proyecto.

2.8 SDLPS

SDLPS es un simulador distribuido que permite la definición y ejecución de modelos utilizando el lenguaje SDL, con una extensión para permitir la definición completa de tiempo en los modelos de simulación. Esta definición completa del comportamiento del modelo permite su simulación sin la necesidad de la aplicación, simplificando los procesos de validación y de verificación.

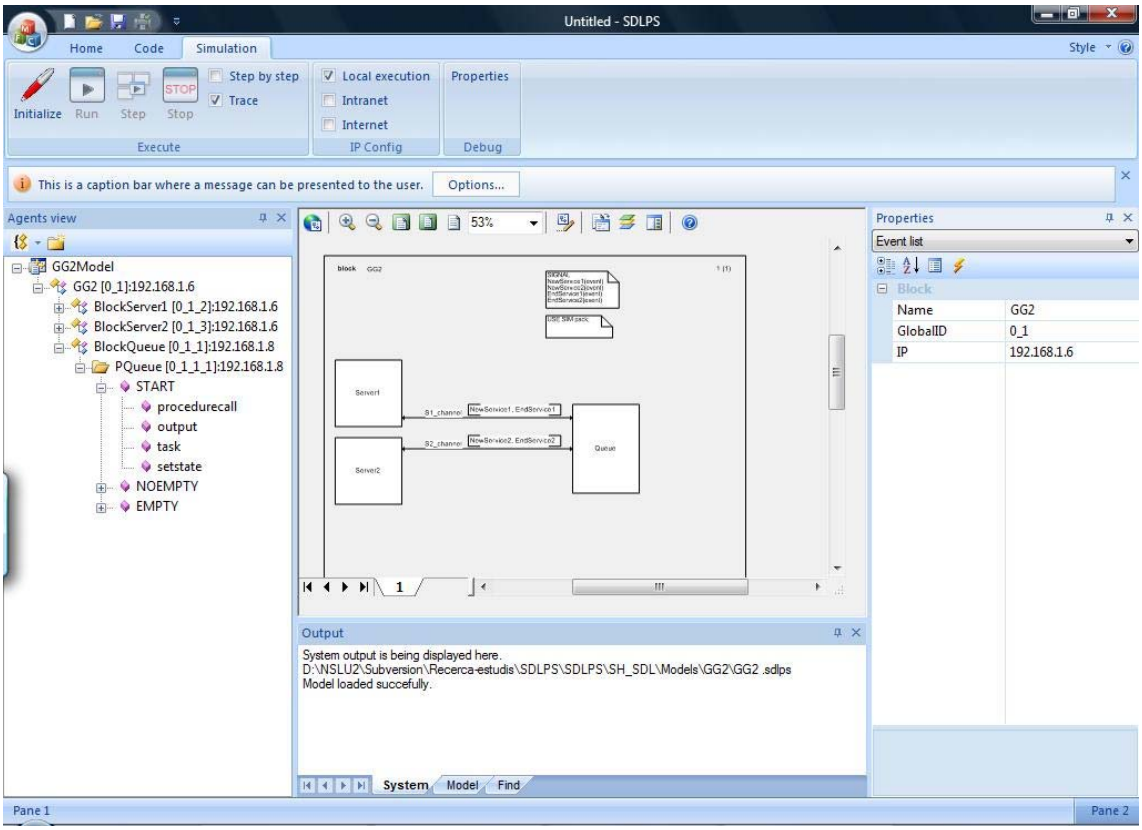

Figura 20: Software SDLPS

2.9 Firelib

Firelib (Bevins s.f.) es una librería desarrollada en lenguaje c para la simulación de incendios forestales basados en algoritmos de Behave para predecir la propagación del fuego en dos dimensiones. Esta librería a grandes rasgos calcula el índice de propagación, la intensidad, la longitud de la llama y la altura de la brasa para incendios de superficie.

Contiene trece funciones, pero con cuatro de ellas se tiene suficiente para crear un simulador simple pero a la vez eficiente y funcional, tal y como necesitamos para este proyecto.

Este simulador recibe como entradas, el mapa con el frente del fuego inicial y los parámetros de entrada necesarios que aportan toda la información necesaria para la predicción y da como resultado de salida un mapa del incendio simulado para un cierto instante posterior al inicial.

Los parámetros de entrada necesarios para la predicción son los siguientes:

- El numero del modelo de combustible (uno de los trece modelos definidos por la librería)
- El contenido de humedad del combustible muerto (para una hora, diez horas o cien horas)
- El contenido de humedad del combustible herbáceo vivo
- La velocidad y dirección del viento
- La dirección e inclinación del terreno

Los mapas representan el terreno donde se desarrolla el incendio. Estos mismos están divididos en celdas y cada celda contiene el momento en el que el fuego alcanza el centro de la misma (un valor de cero significa que nunca fue alcanzada).

Si el tipo de combustible encontrado en el terreno no coincide con ninguno de los trece modelos propuestos por la librería, permite crear un tipo de combustible nuevo y definir todas sus características para que sea más representativo del combustible real.

Para hacer esto se debe de especificar un catalogo de combustible nuevo, el modelo de combustible y las particularidades que conforman el combustible.

En el momento de la simulación, el simulador itera sobre cada una de las celdas del mapa propagando el fuego de una celda a otra hasta llegar a dos casos, llegar a un punto en que la intensidad del fuego no es suficiente como para seguir propagándose o llegar al borde del terreno.

En definitiva el proceso de simulación se podría resumir en cuatro pasos (uno por cada función principal de la librería). Todos estos pasos se realizan por cada celda del mapa que puede llegar a propagar fuego hacia una celda vecina.

Los pasos de la simulación son los siguientes:

Número Paso	Entradas	Salidas
Combustible	Características del combustible	Características generales del combustible: carga, densidad, altura, etc.
Humedad	Humedad del combustible	Índice de propagación sin viento, ni pendiente. Intensidad de reacción, humedad de extinción del combustible vivo, etc.
Viento y pendiente	Dirección y velocidad del viento, dirección e inclinación del terreno	Índice de máxima propagación y su dirección
Dirección	Dirección de máxima propagación	Intensidad de propagación, longitud de la llama, altura de las brasas por cada una de las 8 direcciones principales.

En el primer paso, se establecen y se asignan los valores que tienen relación con el tipo de combustible. Aunque en la naturaleza se pueden tener distintos tipos de combustible en el mismo área, en este simulador el tipo de combustible se considera uniforme tanto espacial como temporalmente.

En el segundo paso, se tiene como salida el índice de propagación sin considerar todavía la influencia del viento y de la pendiente. Lo que determina en este proceso la intensidad de propagación es el tipo de combustible y el contenido de humedad del combustible (las características se describen en el paso uno y la información del contenido de humedad es la entrada de este proceso).

El tercer paso obtiene el índice de la máxima propagación y la dirección en la que ocurre esta máxima propagación. Para obtener estos dos datos es necesario considerar la intensidad y la dirección del viento, así como la inclinación y dirección de la pendiente. Aunque la pendiente se considera también uniforme en todo el terreno y toda la simulación, se utiliza porque el modelo de propagación de Rothermel modela el efecto producido por la pendiente de la misma forma que el efecto producido por el viento, combinándolos y tratándolos en conjunto para poder determinar el ángulo en que el fuego alcanza su mayor velocidad de propagación.

En el último paso, el cuarto, se obtiene la intensidad y el índice de propagación para cada una de las ocho direcciones principales (norte, noroeste, oeste, sudoeste, sur, sudeste, este y noreste). En este paso se obtiene en qué momento el fuego llega a cada una de las celdas vecinas, en el caso de llegar claro, dependiendo de la propagación, de la dirección de máxima propagación, del crecimiento elíptico del modelo, la dirección y la distancia en que se encuentra el fuego en relación a cada una de las celdas vecinas.

2.10 Autómatas celulares

Los autómatas celulares (Wolfram's s.f.) es una metodología que permite el modelado de sistemas dinámicos complejos, de variables y tiempo discretos, es decir que tanto el tiempo, como el espacio y los estados son discretos, creado originalmente por Von Newman y S. Ulam.

Un autómata celular es un conjunto infinito n-dimensional de celdas ubicadas geométricamente en una parrilla y cada celda tiene un estado elegido de un alfabeto que es finito. Cada celda contiene el mismo mecanismo de cálculo que las otras y se pueden conectar entre sí. El estado de las celdas se actualiza simultáneamente y de

forma independiente de las demás en pasos de tiempo discretos, para ello se define una vecindad de una celda, que es un conjunto de celdas cercanas, esta vecindad es homogénea para todas las celdas.

Los estados de las celdas se actualizan de acuerdo a una regla local, esto quiere decir que el estado de una celda en un momento dado sólo depende de su propio estado en el instante previo y de los estados de sus vecinos en ese instante previo. Todas las celdas se actualizan de manera sincronizada, pero el estado de toda la celda avanza en pasos de tiempo discretos. Las celdas actualizan sus valores usando una función de transición, que toma como entrada su estado actual y un conjunto de celdas cercanas (la vecindad).

Un ejemplo que muestra la vecindad de Moore de una celda en la fila i, columna j.

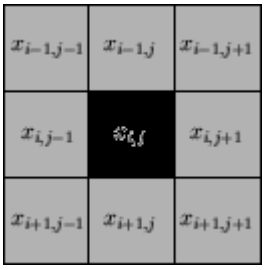

Figura 21: Vecindad de Moore

Por lo tanto un autómata celular se podría definir especificando tres parámetros:

- Un conjunto de estados
- La vecindad de Moore
- La función de transición

A pesar de que los autómatas celulares son una herramienta poderosa, también tienen unas restricciones importantes. Por una parte, se asume que en el espacio de celdas, las celdas están activas simultáneamente, como en un sistema paralelo, pero hay una restricción importante en la actividad simultánea: todas cambian de estado simultáneamente. Otro problema es que el espacio de celdas es infinito, para que pueda ser simulado, el modelo debe de acotarse a un numero finito de celdas en cada paso y la forma más fácil de hacerlo es limitar el modelo a un área finita, lo que hace que se pierda homogeneidad, esto se ve claro en el caso de los bordes de la parrilla,

¿Cómo se deberían de tratar?, para solucionar esto se puede hacer mediante aproximaciones, aunque se pierde precisión. El gran numero de celdas hace que en general se estén haciendo cálculos innecesarios, por ejemplo una celda ya no puede cambiar de estado si su vecindad no ha cambiado de estado.

3 MODELADO y DISEÑO

Como en la mayoría de casos, en el desarrollo de software, la clave es hacer un buen diseño y en este Proyecto Final de Carrera concretamente es la parte más importante, por lo que es la parte donde hay que tener más cuidado y realizarla de lo más minuciosamente posible. Una vez tenemos definido como se comportan los incendios forestales en la naturaleza, ahora tenemos toda esa información para crear el diseño técnico que simule un incendio forestal real.

3.1 Particularidades del modelo

Como el campo de la simulación de incendios forestales se ha podido ver en el apartado de marco teórico, es muy extenso y muy complejo hay que tener en cuenta unas características y restricciones que debemos de tener en cuenta para realizar este proyecto.

Para modelar el comportamiento del fuego este proyecto se basará en el modelo Behave, que utiliza un modelo matemático elaborado por Richard Rothermel, los cuales ya están validados.

Este modelo tiene varias particularidades, una de ellas es que se centra únicamente en los incendios de superficie, no se tiene en cuenta los incendios por copas por ejemplo.

Otro aspecto a tener en cuenta es que se toma como entrada que el combustible de la zona afectada es continuo, es decir, que al realizar la simulación toma un único tipo de combustible para toda la zona.

La pendiente del terreno se considera uniforme a lo largo del terreno y a lo largo de toda la simulación.

Un parámetro dinámico como es el viento, que seguro que varía y sigue un comportamiento difícil de determinar a lo largo de un incendio forestal, en nuestro modelo de simulación también lo contabilizaremos como estático.

Todas estas restricciones y que a veces los datos tampoco pueden ser medidos directamente (hay que tomar estimaciones), hacen que haya imprecisiones en los

datos de entrada, aunque para el caso de los incendios forestales seguro que la simulación nos da un resultado aproximado a la realidad.

Para conseguir el objetivo de que el modelo se ejecute de manera distribuida, al modelo realizado en lenguaje SDL estándar se le han añadido extensiones especiales, que no son propias del SDL y que se explicarán en el próximo apartado.

3.1.1 Extensiones

Gracias a dos extensiones extras al lenguaje SDL conseguiremos definir nuestro modelo de manera que pueda ejecutarse de manera distribuida y estas dos extensiones son:

- La extensión Cell-DEVS
- La extensión m:n CA^k cellular automata

El formalismo Cell-DEVS (Javier Ameghino s.f.) resuelve los problemas que tiene el autómata celular cuando se usa para simular sistemas complejos, ya que el uso del tiempo discreto trae acarreada restricciones en la precisión del modelo. En el paradigma Cell-DEVS cada celda se define como un modelo atómico con un conjunto de estados para los valores de su vecindad, un comportamiento propio y una demora de actualización. Cada celda tiene asociada una demora de tiempo, esto permite diferir la ejecución, de la función de cálculo y los resultados de salida de cada celda no son enviados instantáneamente sino hasta que se consuma el tiempo de demora.

Hay que destacar que esta extensión es del lenguaje DEVS, no es lenguaje SDL y está actualmente pendiente de estandarización. La manera de representarlo en nuestros diagramas en lenguaje SDL será mediante el gráfico de extensión y una variable con las demora de tiempo.

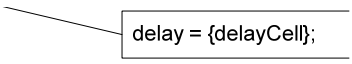

Figura 22: Extensión Cell-Devs

En cuanto a la extensión m:n CA^k cellular automata lo que nos permite es, poder enviar señales a una o varias celdas a la vez, para que comiencen a simularse. De esta manera

permitimos que una celda pueda propagar el fuego a sus celdas más próximas, su vecindad, que en nuestro caso puede llegar a un máximo de ocho celdas vecinas.

En este caso la extensión forma parte del lenguaje SDL, aunque todavía no se ha cursado los procedimientos legales para que acepte su estandarización, pero próximamente se podrá ver en la *Summer Simulation Conference* (International s.f.) a través de un poster de la simulación de la sucesión de Fibonacci. Nuestros diagramas en leguaje SDL representaran esta extensión mediante el gráfico de extensión y dentro una array con los números de las celdas que se han de propagar.

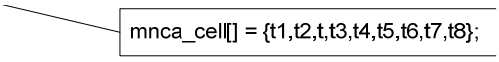

Figura 23: Extensión M:N CA^k

3.2 Especificación con el lenguaje SDL

Se ha escogido el lenguaje SDL para realizar esta especificación, porque es un lenguaje utilizado para diseñar sistemas distribuidos y que se utiliza y se aplica mucho en el campo de la simulación.

Una de las principales características que dan ventaja al lenguaje SDL respecto a otros lenguajes como puede ser el UML, es que este es más versátil, manejable y su especificación en SDL/GR se realiza de manera gráfica, lo que hace que sea más legible y factible para cualquier persona que cualquier otro lenguaje. La especificación se realiza de una manera más natural, lo que facilita que personas ajenas a la informática también puedan entender y aportar información a los modelos diseñados con el lenguaje SDL.

3.3 Especificación del modelo

Mediante el lenguaje SDL se especificará el modelo de simulación de los incendios forestales, el comportamiento de sus celdas y de todos los componentes que forman parte de este sistema.

Para el modelado se ha dividido la especificación en cinco secciones, utilizando el enfoque de arriba hacia abajo, clásico en las ingenierías, empezaremos desde el nivel más alto hasta llegar al nivel más bajo. Las secciones son las siguientes:

- El diagrama del sistema
- El diagrama del MNCA
- El procedimiento Vecinity
- El procedimiento Nucleus
- El diagrama de bloques
- El diagrama de estados
- Los diagramas de procesos

Hay que tener en cuenta que para llegar al modelo final, primero se diseñó un modelo que lo que pretendía era una simulación global, básicamente que funcionara como un único conjunto de celdas donde estaba todo el terreno a incendiar y de aquí obtener una salida correcta.

Después de ver que los resultados del primer modelo eran satisfactorios se pasó a modificarlo para hacer que en lugar de trabajar como un conjunto se trabajara a nivel de celda, permitiendo así un modelo distribuido que era el objetivo de este proyecto. Para conseguir que el modelo final se pudiese ejecutar de manera distribuida al lenguaje se le tuvo que añadir unas extensiones que no son propias del lenguaje SDL, estas extensiones se añadieron tanto en los diagramas del modelo como en el XML.

De esta manera en la especificación se detallará el primer modelo y el modelo final, con sus respectivas diferencias.

3.3.1 Diagrama del Sistema

El sistema inicial estaba compuesto por una instancia de BlockCelda, el motor, el reloj y el entorno. La instancia de BlockCelda es la diferencia respecto al modelo final. BlockCelda se corresponde con toda la zona de terreno a simular ó incendiar, tratándola globalmente como una unidad de terreno y en este primer modelo sin distinguir celdas ni parcelas de terreno. El resto de instancias, el motor, el reloj y el entorno de este primer modelo funcionan igual que en el modelo final, así que, se detallarán más adelante en la versión definitiva. En los dos casos no hay señales que viajen hacia, ni desde el exterior ya no se pueden modificar los datos de entrada en

tiempo real, esto se está investigando en otros proyectos y en un futuro próximo se podrá aplicar ampliaciones de este proyecto.

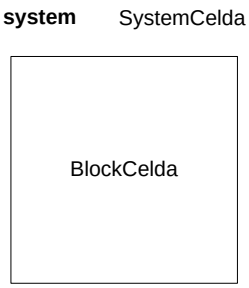

Figura 24: Diagrama Sistema versión 1

El sistema del modelo final está compuesto por diversas instancias, referentes al autómata celular, el motor, el reloj y el entorno. En este modelo final habrá N instancias que representaran los bloques del autómata celular, según el tamaño del terreno a incendiar, mientras que del resto de instancias, el motor, el reloj y el entorno, solo habrá una única instancia por cada una de ellas.

El entorno es el que gestiona la interacción del usuario y el simulador, se encarga de enviar al motor la configuración inicial, inicializar la simulación, indicar cuándo se ha de empezar, realizar la simulación paso a paso y también es el responsable de parar la simulación, mostrando de una manera u otra la salida ó resultado de la simulación al usuario.

El motor es el que guía al proceso de la simulación desde el estado inicial al estado final y el reloj es el que marcará al motor cuando ha de actuar.

El autómata celular, que es una extensión del lenguaje SDL, está compuesto por una ó N instancias de este bloque llamado MNCA_Fuego, una instancia de este bloque representa a una única celda ó parcela del terreno a simular (en la primera versión se cogía todo el terreno) y contiene toda la información necesaria (distribuida por capas), para ser simulada por sí sola, aunque la capa principal será el estado, que será la que determine su comportamiento a lo largo de la simulación. Esta es la gran diferencia

respecto al modelo anterior, en este caso los datos de las capas que necesita cada celda, se cargan en cada instancia, cada una realiza los cálculos necesarios con la información que ya posee y los resultados obtenidos se guardan por separado, esto lo que nos permite es distribuir el modelo del sistema.

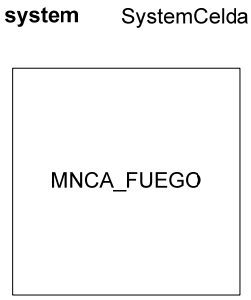

Figura 25: Diagrama del sistema versión final

3.3.2 Diagrama del MNCA_Fuego

Este diagrama no existe en la primera versión, porque no se utilizaba ninguna instancia del MNCA_Fuego, de manera que en nuestro modelo final quedó como voy a explicar a continuación.

Como ya he comentado anteriormente, el MNCA_Fuego representa una o N instancias, donde cada una se corresponde con una parcela del terreno a simular. A este nivel, tenemos que definir las dimensiones del autómata celular, es decir, las dimensiones del terreno a incendiar y lo expresamos de la siguiente forma:

Existe un valor entero que hace referencia a las dimensiones del terreno y es el siguiente:

int mnca_DIM = 2;

En este caso como es una simulación en dos dimensiones lo represento con un 2.

Hay otros dos valores definen el tamaño del terreno, es decir, el ancho y alto del terreno forestal.

int mnca_D1 = 101;

int mnca_D2 = 101;

Este caso por ejemplo sería para un terreno de 101 celdas x 101 celdas.

Cada instancia de celda ha de contener ya, toda la información necesaria para poder realizar todos los cálculos necesarios en su simulación, por lo que aquí es donde tenemos que declarar todas las capas con la información relativa cada celda, que son la siguiente:

Los valores que vienen a continuación definen el contenido de humedad del combustible muerto, pasada 1 hora, 10 horas y 100 horas respectivamente.

double mnca_M1[];

double mnca_M10[];

double mnca_M100[];

El siguiente valor es el mapa de la simulación y nos marcará el instante de tiempo en que el fuego llega al centro de la celda, como en un principio no hay ninguna celda incendiada todos los valores estarán a cero, queriendo decir que no fue alcanzada por el fuego.

double mnca_BlockCelda[];

La dirección del viento la obtendremos en la siguiente variable.

double mnca_MDirVents[];

El contenido de humedad de la madera y de la hierba los definimos en las siguientes variables.

double mnca_MFusta[];

double mnca_MHerba[];

Y la información que corresponde a la orientación y pendiente del terreno las definimos así.

double mnca_MOrientacio[];

double mnca_MPendent[];

La velocidad del viento se mantiene en esta variable.

double mnca_MVelVents[];

El tipo de combustible que contiene cada celda.

double mnca_Model[];

En este diagrama también se definen dos procedimientos como son Vecinity y Nucleus, en este punto solo comentaré por encima lo que hacen, más adelante se entrará más en detalle de cómo funcionan.

El procedimiento Vecinity lo que hace es calcular la vecindad de Moore de una celda, recibe como parámetro la posición de la celda y nos devolverá su vecindad (celdas vecinas) respecto de la dada.

En cambio el procedimiento Nucleus calcula cual es la celda actual y salvará todos los valores importantes para poder sacar conclusiones de la simulación.

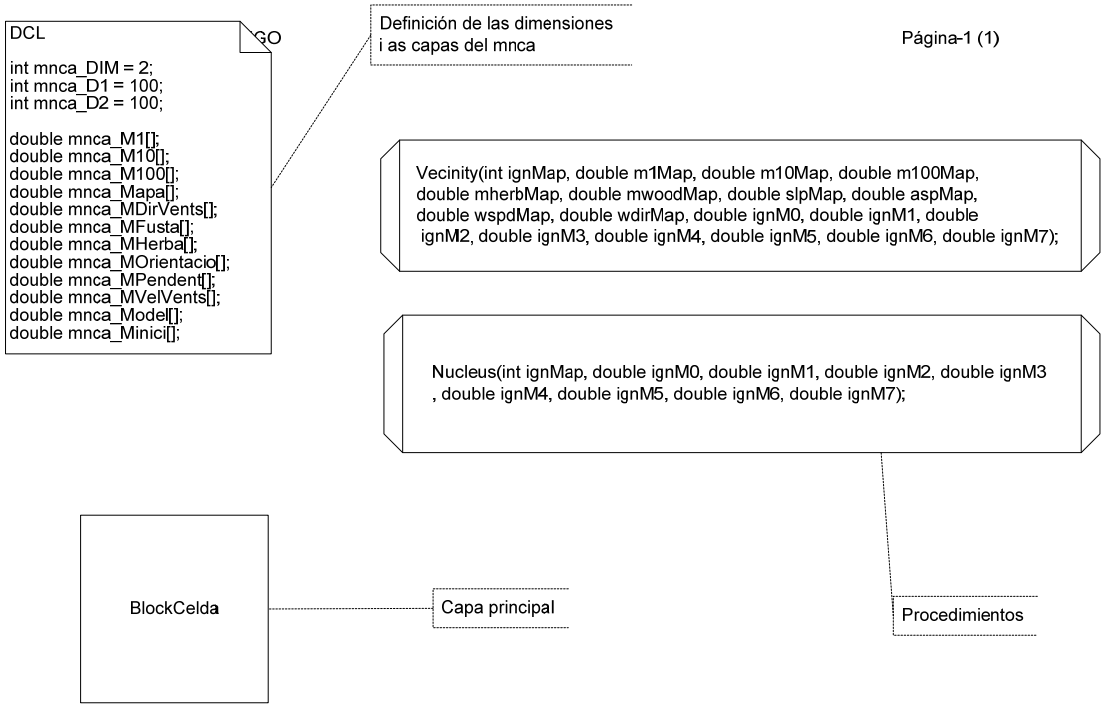

Figura 26: Diagrama del MNCA_Fuego

3.3.3 El procedimiento Vecinity

Como estamos utilizando un autómata celular necesitamos una función que calcule la vecindad de Moore, en nuestro caso en dos dimensiones, en concreto se encargará de encontrar las ocho celdas que rodean a una celda a tratar, justamente la celda que se estará simulando de manera distribuida.

El procedimiento Vecinity y sus parámetros son los siguientes:

Vecinity(int ignMap, double m1Map, double m10Map, double m100Map, double mherbMap, double mwoodMap, double slpMap, double aspMap, double wspdMap, double wdirMap, double ignM0, double ignM1, double ignM2, double ignM3, double ignM4, double ignM5, double ignM6, double ignM7);

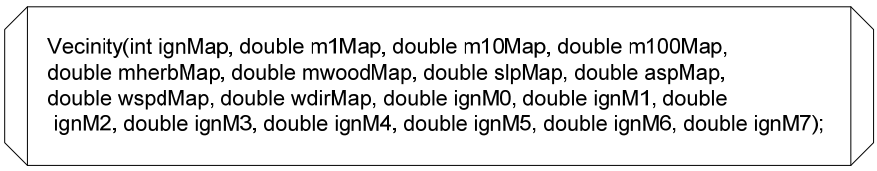

Figura 27: Procedimiento Vecinity

Al llamar a este procedimiento nos encontramos en una celda en concreto que se encontrará dentro de una matriz en la fila i, columna j. Lo que pretendemos es obtener toda la información necesaria para poder ejecutar la simulación en esta celda, por lo que se le pasan todos sus parámetros por referencia.

Guardaremos la información relativa a la celda actual en los parámetros siguientes:

Contiene el tiempo de ignición en el parámetro

int ignMap.

Los tres parámetros siguientes hacen referencia a la humedad de combustible muerto pasadas 1, 10 y 100 horas

dobule m1Map

double m10Map

double m100Map,

El contenido de humedad de la hierba y de madera estará en los parámetros

double mherbMap

double mwoodMap

La información relativa a la pendiente del terreno

double slpMap

La información de la orientación del terreno

double aspMap

También necesitamos la información de la velocidad y dirección del viento

double wspdMap

double wdirMap

Aparte de toda la información de la celda central se necesita saber la información relativa al tiempo de ignición de las celdas vecinas, estas de dejarán en los parámetros

double ignM0

double ignM1

double ignM2

double ignM3

double ignM4

double ignM5

double ignM6

double ignM7

Ahora procederé a explicar cómo funciona a más bajo nivel el procedimiento.

Lo primero que se hace es obtener la posición de la celda actual y guardarla en una variable temporal llamada tmpCell, gracias a las funciones que nos aporta el SDK del SDLPS.

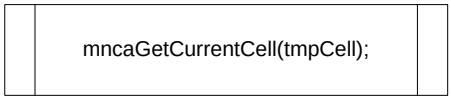

Figura 28: Obtención celda actual

A continuación se obtiene toda la información de las capas (con funciones del SDK) necesaria para la simulación de esta celda actual y se guarda en sus respectivas variables.

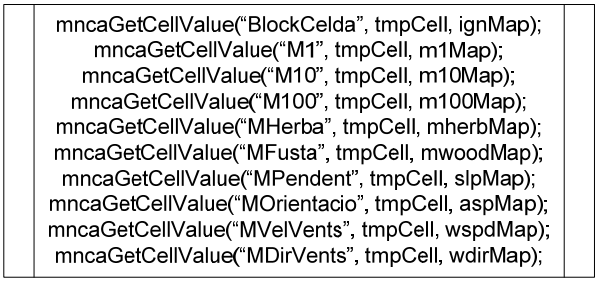

Figura 29: Obtención información celda actual

Ahora nos falta obtener la información de las celdas vecinas, por lo que a partir de ahora lo que se hace es, primero calcular la posición de una celda vecina, obtener la información (con la ayuda de las funciones de SDK) y guardarla en su variable, así hasta completar las ocho celdas vecinas. Como la parrilla es una matriz (bidimensional) y en el SDLS solo se pueden utilizar arrays, para calcular las celdas vecinas hay que saber la fila y columna de las celdas, esto se hace con las funciones de getPosicion, se modifica la fila y columna de la celda hasta llevarla a una celda vecina (según la vecindad de Moore) y se calcula la posición final de la celda vecina, de esta manera se obtiene la posición dentro del array y al final se obtiene el valor dentro de la capa.

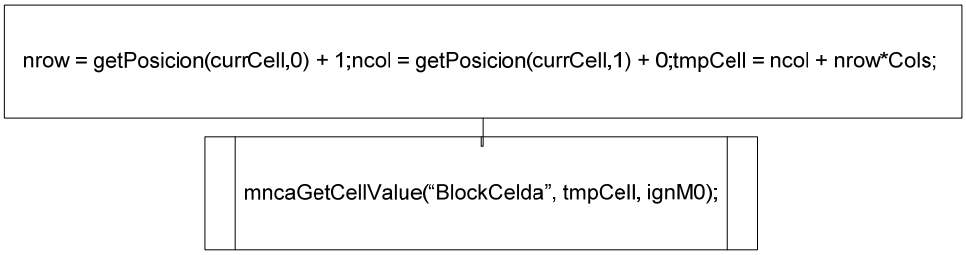

Figura 30: Obtención información celda vecina

Un ejemplo de esto sería podría ser el caso siguiente. Estamos en una celda que está en la posición 2575 de la parrilla de celdas (de 101 x 101 celdas), de este modo el procedimiento funcionaría así:

Obtengo la posición con la función mncaGetCurrentCell, por lo tanto en la variable tmpCell tengo la posición 2575.

Como ya tengo la posición ahora obtengo la información de esta celda con la función que proporciona el SDK mncaGetValue y la guardo en sus respectivas variables

Ahora obtengo la posición de las ocho celdas vecinas a la de la posición 2575, que son estas 2676, 2677, 2576, 2475, 2474, 2473, 2574, 2675. Como ya tengo sus posiciones con la función mncaGetValue puedo obtener su tiempo de ignición, de esta manera ya estamos preparados para ejecutar la función de transición de este autómata.

Por último se puede ver cómo resultó el procedimiento completo en Lenguaje SDL.

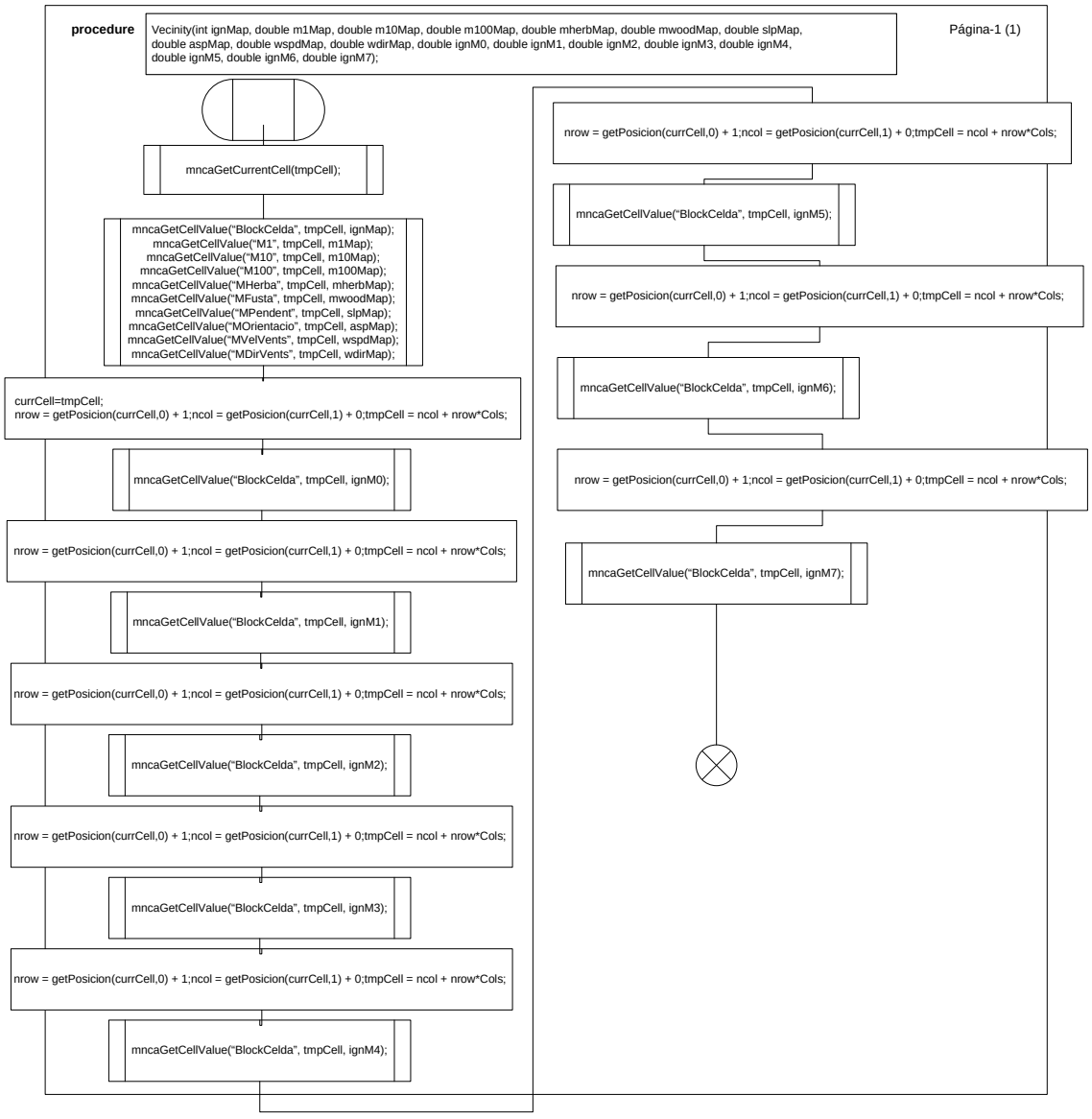

Figura 31: Procedimiento Vecinity

3.3.4 El procedimiento Nucleus

En el procedimiento Nucleus lo que hacemos es básicamente, salvar a las capas del modelo los resultados obtenidos por la función de transición del autómata celular.

El procedimiento Nucleus y sus parámetros son los siguientes:

Nucleus(int ignMap, double ignM0, double ignM1, double ignM2, double ignM3, double ignM4, double ignM5, double ignM6, double ignM7);

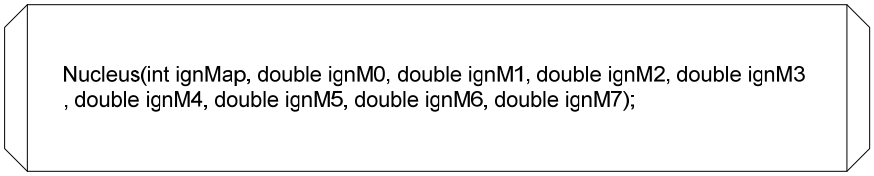

Figura 32: Procedimiento Nucleus

La variable que contiene el tiempo de ignición de la celda actual es:

Int ignMap

Los siguientes ocho parámetros se corresponden con los tiempos de ignición de las celdas vecinas

double ignM0

double ignM1

double ignM2

double ignM3

double ignM4

double ignM5

double ignM6

double ignM7

El funcionamiento del procedimiento a más bajo nivel sería el siguiente.

Igual que el procedimiento anterior lo que se hace es obtener en que celda nos encontramos y la guardamos en la variable tmpCell.

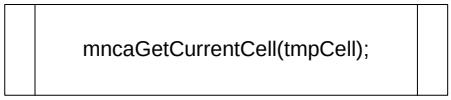

Figura 33: Obtención información celda actual

A continuación se guarda el valor del tiempo de ignición de la celda en su correspondiente capa, utilizando la función del SDK del SDLPS mncaSetCellValue.

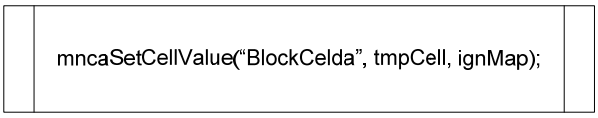

Figura 34: Salvar información celda actual

Ahora como en el procedimiento Vecinity se pasaría a calcular la posición de las ocho celdas vecinas y salvar su valor

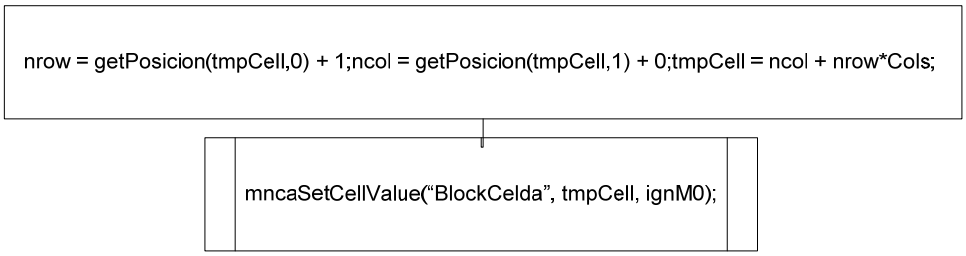

Figura 35: Salvar información celda vecina

A modo de ejemplo, siguiendo con el mismo ejemplo que el apartado anterior, nos encontramos con la celda en la posición 2575 y al entrar en este procedimiento funcionaria así:

Obtengo el valor de la posición de la celda actual con mncaGetCurrentCell y lo guardo en tmpCell que será igual a 2575.

A continuación guardo el tiempo de ignición de esta celda con la función mncaSetCellValue.

Después de esto paso a calcular las posiciones vecinas a la 2575, que en este caso son, 2676, 2677, 2576, 2475, 2474, 2473, 2574 y 2675.

Por último con guardo sus tiempos de ignición que los tengo por los parámetros en las capas correspondientes.

En SDL el procedimiento completo quedaría como a continuación.

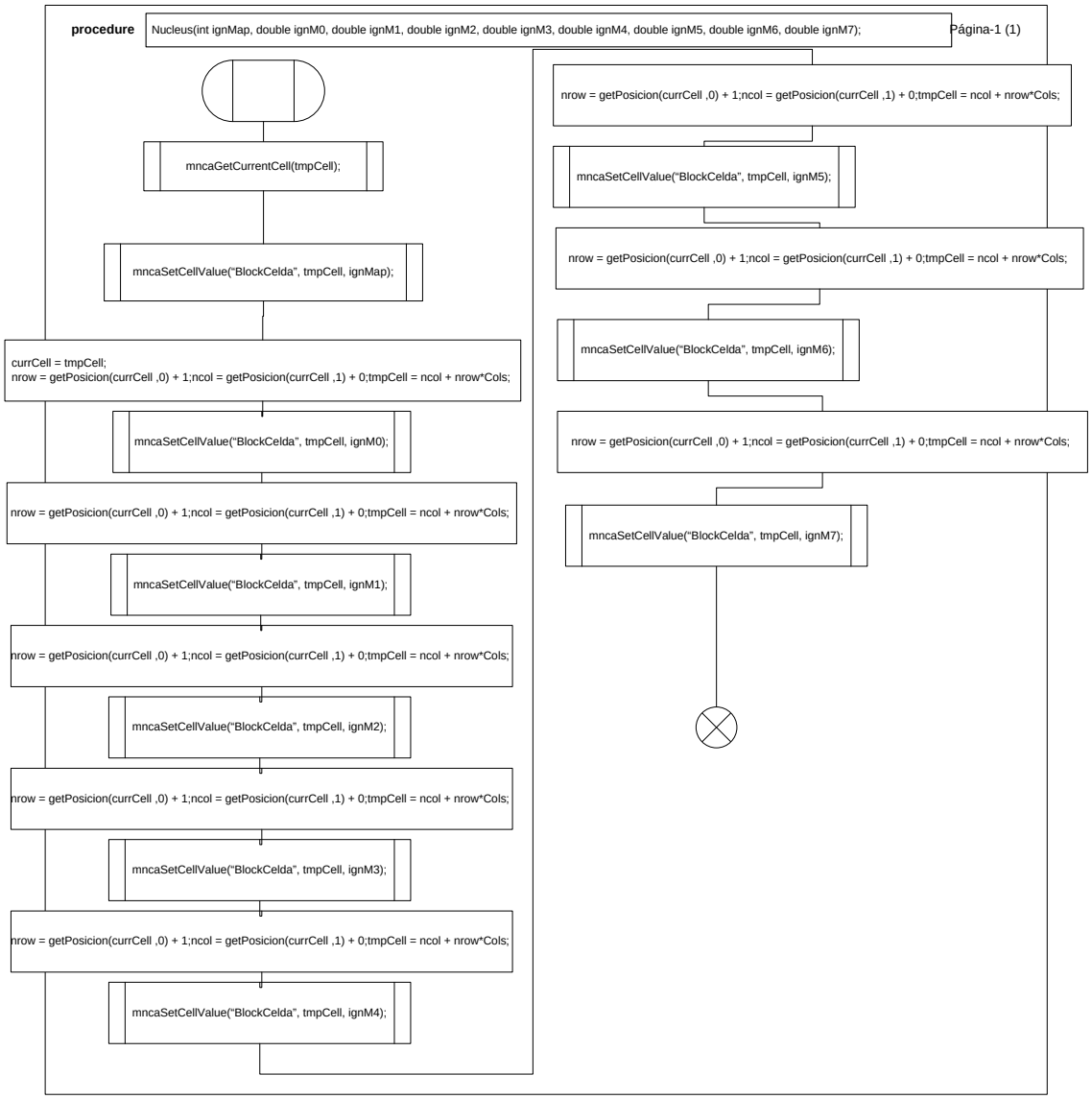

Figura 36: Procedimiento Nucleus

3.3.5 Diagrama de Bloques

Seguimos bajando por el modelado y nos encontramos con el diagrama de bloques, que nos representa en este punto un único proceso, en la primera versión representaba a toda el área de terreno en cambio en la versión definitiva solo es representativo de una única celda, parcela de terreno.

En la versión inicial el bloque como representaba todo el terreno, poseía toda la información de todo el terreno y realizaba todas las acciones en base a todo el terreno teniendo en cuenta en todo momento todas las celdas de la zona forestal.

En la versión final es diferente porque en este momento el bloque nos representa una celda en concreto y solo conoce la información relativa a ella misma, es la única que es capaz de realizar sus modificaciones y sus cálculos necesarios para avanzar en la simulación, por eso no necesita conocer y de hecho no conoce, la información relativa al resto de las otras celdas.

En nuestro caso, los bloques se ejecutan de manera paralela, ya que cada bloque podría ejecutarse en una computadora diferente, es decir, los bloques se ejecutarían a la vez en maquinas distintas. Para el caso de los procesos que hay dentro del bloque es diferente porque estos se ejecutarán de manera concurrente, es decir, una misma máquina alternará la ejecución de varios procesos en posiciones fijas de tiempo, aunque en nuestro caso solo hay un único proceso, en el caso de haber más es lo que sucedería.

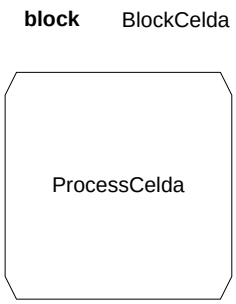

Figura 37: Diagrama de bloques

3.3.6 Diagrama de Estados

Lo primero que hay que aclarar que este diagrama no es del lenguaje SDL, pero creo que es muy importante explicarlo para que se comprenda el modelo. El diagrama de estados no varía del primer modelo al modelo final, lo único que hay que tener en cuenta que para el primer modelo el estado representaba a todo el terreno y en la

versión final representa a cada celda de terreno, con lo que explicaré este diagrama para versión definitiva y para el primer modelo es igual, solo hay que cambiar lo que comento a nivel de celda por el nivel de terreno.

Cada celda contiene la información relativa a sus capas y puede modificar su información a lo largo de la simulación. El estado de una celda es el que determina el comportamiento que esta tendrá dentro de la simulación y un cambio en este estado se podrá producir si recibe una señal producida por algún evento de alguna otra celda y esta también podrá producir eventos hacia otras celdas lo que producirán cambios de estados en otras celdas.

Los posibles estados de una celda son:

- **No Quemado**, representa que la celda no está incendiada, es decir, en su estado forestal natural.
- **Quemado**, representa que la celda ya ha sido devastada por el fuego.
- **Ardiendo**, representa que la celda está ardiendo y el fuego se puede propagar hacia alguna de las otras celdas.

A continuación se explicará los eventos (marcados en cursiva) que reciben cada estado (marcados en negrita) y los posibles cambios de estados que estos puedan producir.

Primero de todo hay que decir que todas las celdas empiezan con un estado inicial de **No Quemado**, como es natural todas las parcelas de terreno están en su estado original y cuando estas reciben un evento de *Propagar* es cuando pasan a modificar su estado, a un estado de **Ardiendo**, exceptuando la primera celda que se incendiará que recibirá el evento de *Arder*, que solo se producirá al inicio de la simulación.

Cuando nos encontramos en el estado de **Ardiendo** puede haber varios sucesos, una es que reciba un evento de *Extinguir*, lo que haría que modificase el estado de la celda a un estado de **Quemado,** acabando aquí la simulación de esta celda. Otra opción es que la celda reciba el evento de *Arder* (significará que la celda está ardiendo), *ActualizarDatos* (esto quiere decir que se ha modificado información de la celda en tiempo real y por lo tanto que volver a realizar sus cálculos) ó *Propagar* (significa que

el fuego se propaga a celdas vecinas) pero como su estado ya era de **Ardiendo**, no le ocasionará ninguna modificación al estado.

El evento de *Propagar* producirá que otras celdas modifiquen su estado, en cambio los eventos *Arder* y *ActualizarDatos* son propios de la propia celda y como mucho acabarán produciendo que la misma celda reciba el evento de **Extinguir**.

El estado de **Quemado** se llegará por que han enviado un evento de *Extinguir*, una vez llegado a este estado dará igual lo que reciba, porque la celda ya no podrá modificar su estado en ningún otro momento de la simulación, este estado es final porque la parcela de terreno ya esta devastada, por ejemplo si recibe un evento de *Arder* como ya no hay más combustible por arder, no le afectará.

El diagrama de estados completo es el que se muestra a continuación.

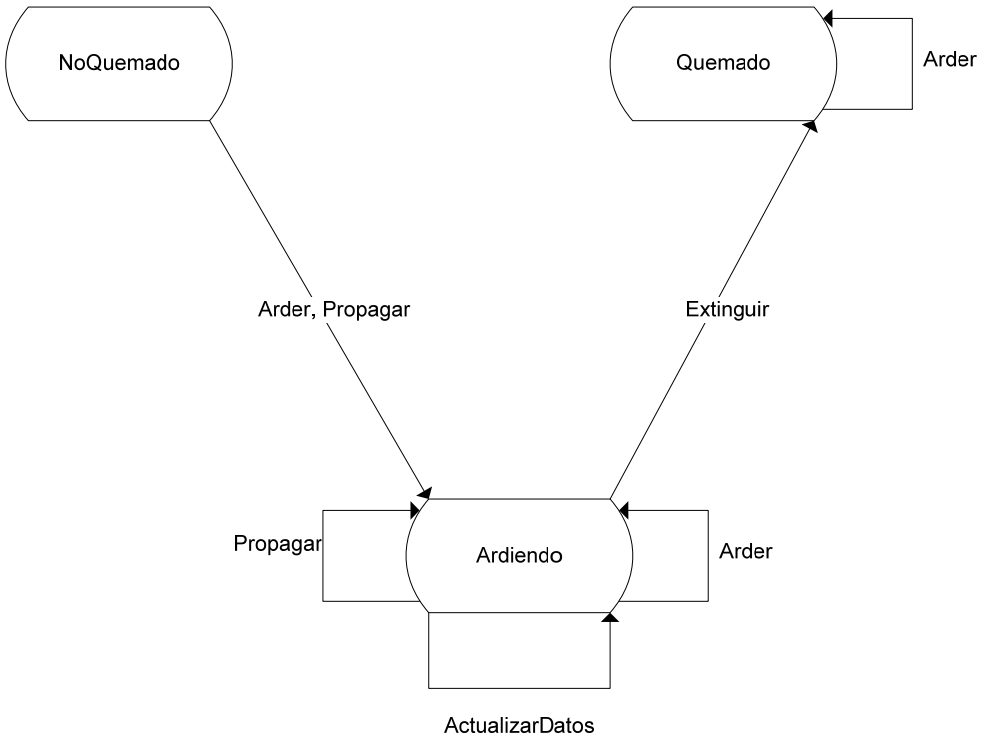

Figura 38: Diagrama de estados

3.3.7 Diagrama de Procesos

Este es el nivel más bajo de la especificación, el proceso ProcessCelda se compone de tres estados (marcados en negrita) explicados anteriormente **No Quemado**, **Quemado** y **Ardiendo**. Al igual que diagrama anterior el modelo inicial y final son iguales con el

cambio de tratar a nivel de celda o de terreno, por lo que solo explicaré la versión definitiva.

Lo primero de todo en el proceso hay declarados unos valores que son necesarios para la simulación y representan lo siguiente.

El entero que viene a continuación nos marcará si la celda propaga el fuego a otras o no, un valor de cero representa que no se propaga y un valor de uno quiere decir que se propaga a una o varias celdas vecinas.

Int PROPAGA;

Hay otro entero que nos dirá en todo momento si el fuego se ha extinguido de la celda actual, un valor de cero representa que no está extinguido y un uno en el caso contrario.

Int EXTINGUIDO;

Utilizaremos una variable para mantener el tiempo actual de simulación y será el que se muestra a continuación.

double ProcessCelda_t;

Para guardar en todo momento la información relativa a la celda a simular, utilizaremos estas variables.

el tiempo de ignición en la siguiente variable

int ignMap.

Las tres variables siguientes hacen referencia a la humedad de combustible muerto pasadas 1, 10 y 100 horas

dobule m1Map

double m10Map

double m100Map,

El contenido de humedad de la hierba y de madera estará en las variables

double mherbMap

double mwoodMap

La información relativa a la pendiente del terreno

double slpMap

La información de la orientación del terreno

double aspMap

También necesitamos la información de la velocidad y dirección del viento

double wspdMap

double wdirMap

Aparte de toda la información de la celda a simular se necesita saber la información relativa al tiempo de ignición de las celdas vecinas, estas de dejarán en las variables.

double ignM0

double ignM1

double ignM2

double ignM3

double ignM4

double ignM5

double ignM6

double ignM7

Ahora se mostrarán los procesos de los tres estados por separado para una mayor claridad y entendimiento.

Comenzaremos la simulación por el estado inicial y lo primero que hacemos es buscar cual es la primera celda que se ha de incendiar, inicializaremos las características del fuego y enviaremos el evento *Arder* para empezar con la simulación.

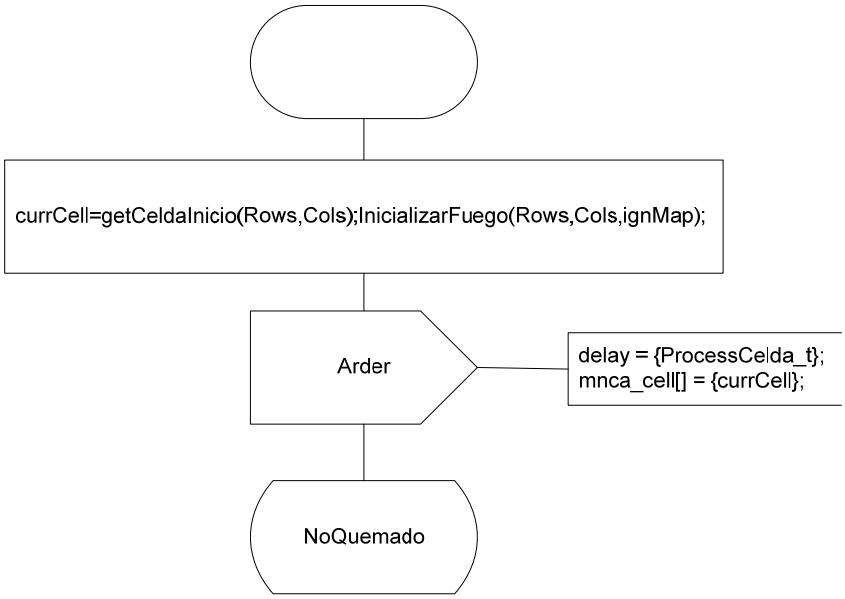

Figura 39: Enviando la señal arder.

A partir de aquí, toda celda empieza en un estado inicial de **No Quemado**. Por ser la primera celda esta habrá recibido el evento *Arder*, por lo que calculará la vecindad de Moore con el procedimiento **Vecinity**.

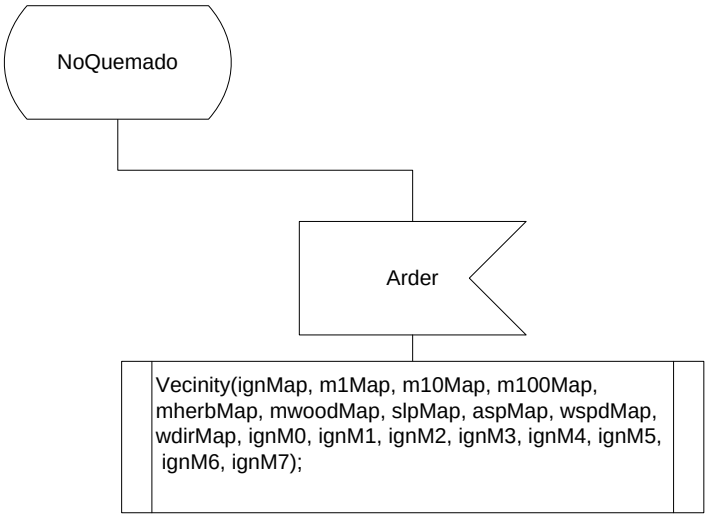

Figura 40: Recibiendo la señal arder

A continuación ejecutará una tarea que contiene la función de **PropagarFuego** que calculará cuales son las celdas vecinas a las que se propaga el fuego, después

obtendremos el valor del tiempo de ignición de estas celdas vecinas mediante la función **ObtenerCeldasIncendiadas** (estas funciones se explicarán con detalle más adelante).

```
PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Map,m10Map,
m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wdirMap,
ignMap,flMap,currCell,
ignM0,ignM1,ignM2,ignM3,ignM4,ignM5,ignM6,ignM7);
propCell1=ObtenerCeldasIncendiadas(0);
propCell2=ObtenerCeldasIncendiadas(1);
propCell3=ObtenerCeldasIncendiadas(2);
propCell4=ObtenerCeldasIncendiadas(3);
propCell5=ObtenerCeldasIncendiadas(4);
propCell6=ObtenerCeldasIncendiadas(5);
propCell7=ObtenerCeldasIncendiadas(6);
propCell8=ObtenerCeldasIncendiadas(7);
```

Figura 41: Obteniendo la propagación

El siguiente paso a realizar es guardar en las capas correspondientes los tiempos de ignición de las nuevas celdas en las que se ha de propagar el fuego, esto se hace con el procedimiento **Nucleus**.

```
Nucleus(ignMap, ignM0, ignM1, ignM2,
ignM3, ignM4, ignM5, ignM6, ignM7);
```

Figura 42: Salvando los resultados

Ya solo nos queda enviar el evento *Propagar* a las nuevas celdas vecinas que hay que simular y esto lo hacemos con las extensiones extras que hemos creado, mnca_cell[] donde le indicaremos las celdas que deseamos propagar el fuego y en delay el retraso con el que quieres que se ejecuten.

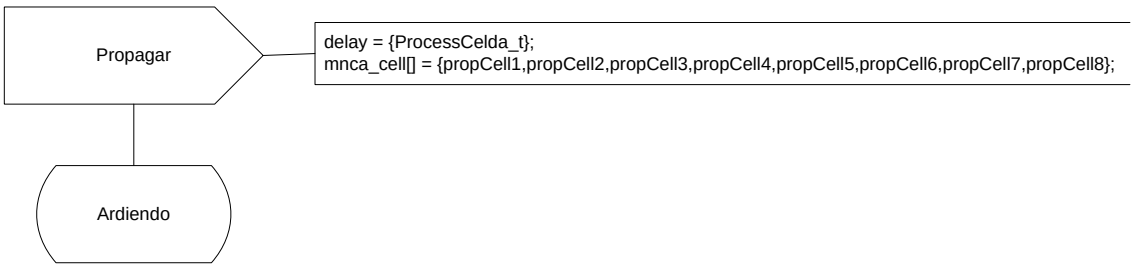

Figura 43: Enviando la señal propagar

Esto era para el caso de la celda inicial, pero para el resto de celdas, recibirán el evento *Propagar* (en lugar de uno de *Arder* como era el otro caso), lo que quiere decir que se

han de incendiar y por lo tanto han de cambiar de estado a un estado de **Ardiendo** donde continuarán su simulación, en SDL está representado de la siguiente forma.

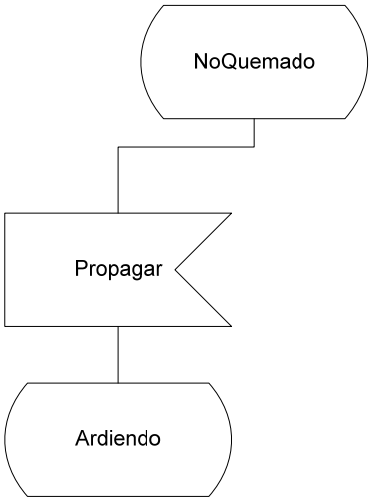

Figura 44: Recibiendo la señal propagar

El diagrama completo sería el que viene a continuación:

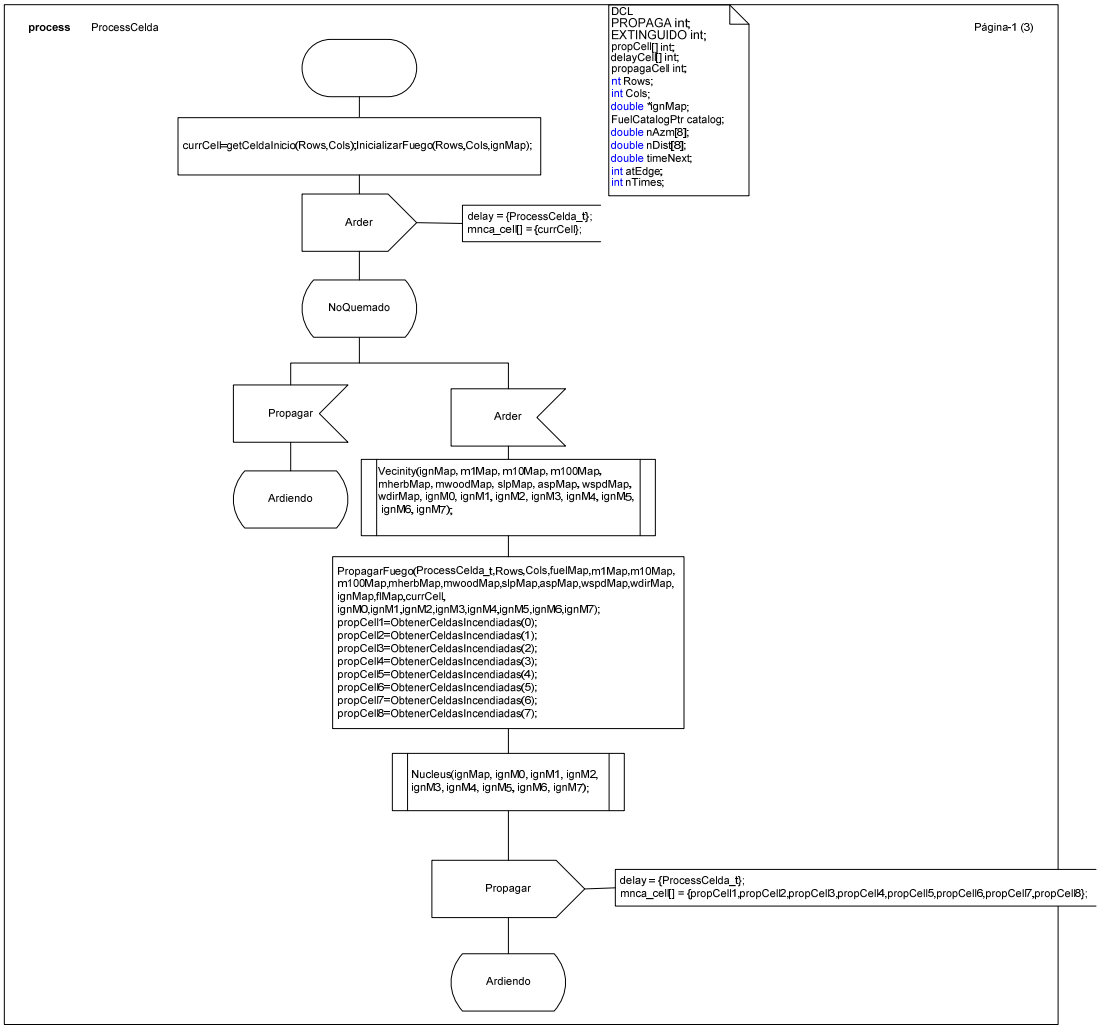

Figura 45: Diagrama de procesos, parte 1

Ahora detallaré el segundo estado, el estado **Ardiendo** que puede recibir cuatro eventos.

Si recibe el evento *Arder* significa que la celda se está quemando porque todavía tiene combustible y mientras tenga combustible continuará en este mismo estado.

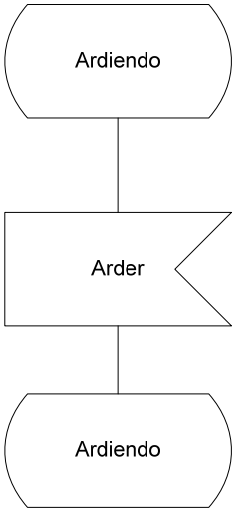

Figura 46: Recibiendo señal arder

Si recibe el evento de *Extinguir* significa que el fuego de esta celda ya se ha apagado porque el combustible se ha acabado y no se puede seguir ardiendo, ni propagándose a ninguna otra celda, por lo tanto la celda ha de cambiar de estado a uno de **Quemado**.

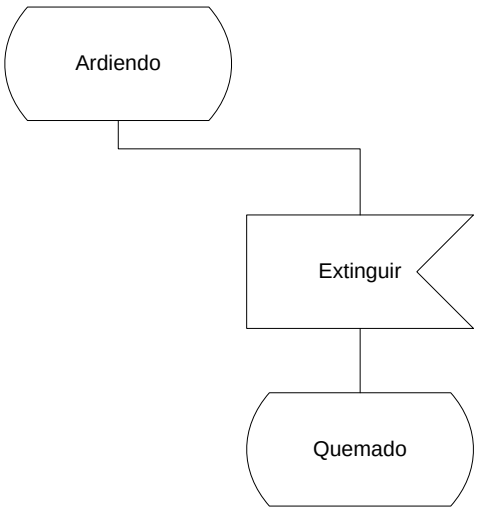

Figura 47: Recibiendo la señal de extinguir

También puede recibir el evento de *Propagar* esto significa que una celda del terreno vecina le ha propagado el fuego y por eso le ha enviado este evento, esto quiere decir que la celda empieza a arder y podría propagarse a otra celda por lo que a continuación se calcula la vecindad de Moore con el procedimiento **Vecinity**.

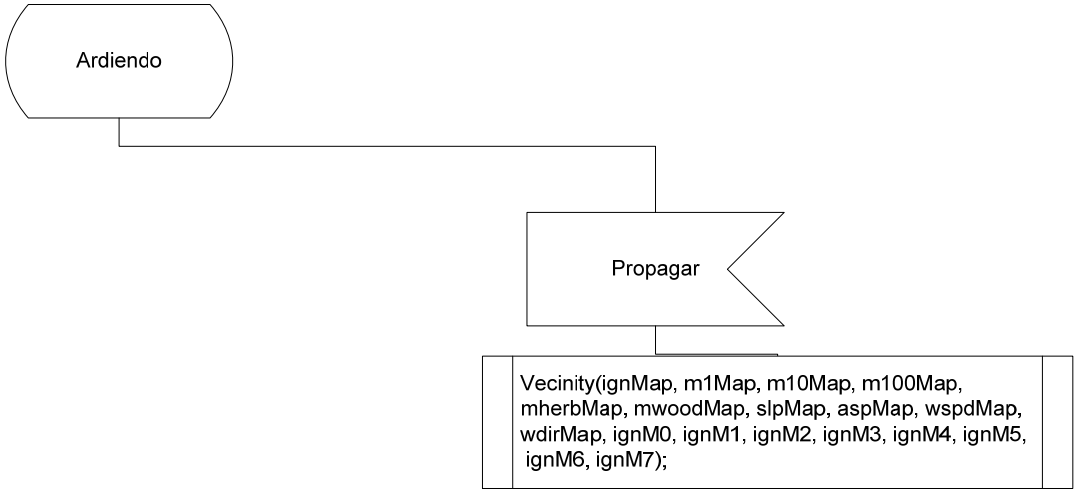

Figura 48: Recibiendo la señal de propagar

Una vez conocemos toda la información relativa a la celda y a sus vecinas pasará a ejecutarse una tarea donde la primera función **PropagarFuego** nos calculará que celdas vecinas empiezan a incendiarse y en qué tiempo y después con la función **ObtenerCeldasIncendiadas** sabremos las celdas vecinas que se les ha propagado el fuego y por ultimo con las funciones **gerPtopagar** y **getExtinguir** podremos saber si la celda propaga el fuego a otra celda y si se extingue el fuego de esta celda, guardando en sus respectivas variables PROPAGA y EXTINGUIDO su valor para poder tomar las decisiones convenientes en los próximos pasos.

```
PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Map,m10Map,
m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wdirMap,
ignMap,flMap,currCell,
ignM0,ignM1,ignM2,ignM3,ignM4,ignM5,ignM6,ignM7);
propCell1=ObtenerCeldasIncendiadas(0);
propCell2=ObtenerCeldasIncendiadas(1);
propCell3=ObtenerCeldasIncendiadas(2);
propCell4=ObtenerCeldasIncendiadas(3);
propCell5=ObtenerCeldasIncendiadas(4);
propCell6=ObtenerCeldasIncendiadas(5);
propCell7=ObtenerCeldasIncendiadas(6);
propCell8=ObtenerCeldasIncendiadas(7);
PROPAGA=getPropagar();
EXTINGUIDO=getExtinguir();
```

Figura 49: Calculando la propagación

Lo siguiente será guardar esta información en las capas con el procedimiento **Nucleus**.

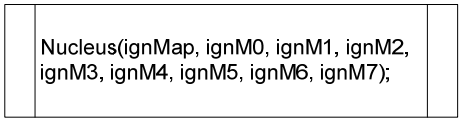

Figura 50: Salvando los resultados obtenidos

Ahora llegamos a un punto en el que pueden pasar varias cosas, si la celda se propaga hacia otras celdas, significará que esta celda sigue todavía ardiendo y por lo tanto enviará un evento de *Arder*, en caso contrario no hará nada.

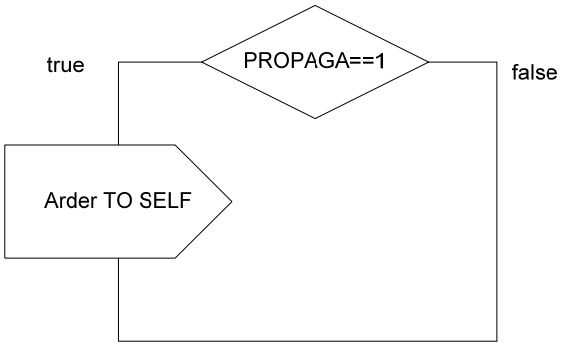

Figura 51: Tomando la decisión si se propaga el fuego

Después llegamos a otro punto decisional, dependiendo si la celda se ha extinguido o no, si la celda ya ha agotado su combustible, significará que ya no puede arder y por lo tanto tenemos que cambiar de estado a esta celda enviándole un evento de *Extinguir* y llevándola a un estado de **Quemado**.

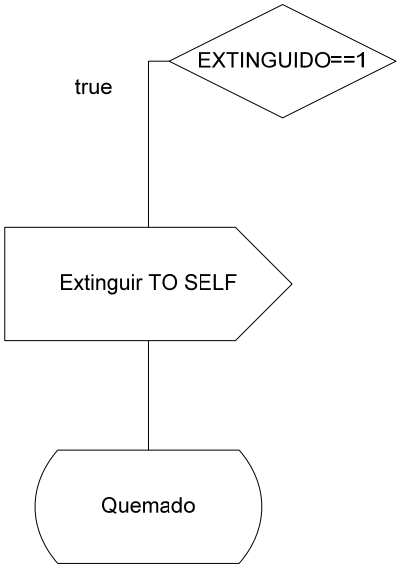

Figura 52: Enviando señal de extinguir

Si por el contrario, no se ha extinguido, significa que el fuego podría propagarse a alguna de sus celdas vecinas, por eso se le ha de enviar el evento *Propagar*, con la ayuda de las nuevas extensiones mnca_cell[] y delay, hacia las nuevas celdas propagadas.

Figura 53: Enviando la señal de propagar

Por último, también puede recibir el evento de *ActualizarDatos*, este evento se enviará cuando se modifiquen los datos de las celdas en tiempo real, por ejemplo cuando se realice una simulación con la ayuda de GPS que permitan enviar y modificar datos al sistema en tiempo real. En este caso lo primero que tenemos que hacer es obtener la información modificada con el procedimiento **Vecinity**.

Figura 54: Recibiendo la señal de actualizardatos

Después procederemos a ejecutar la tarea que calcule si con los nuevos datos el fuego de la celda actual hace que el fuego se propague a otras celdas vecinas y saber cuáles son, esto lo obtendremos con las funciones **PropagarFuego** y **ObtenerCeldasIncendiadas**.

```
PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Map,m10Map,
m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wdirMap,
ignMap,flMap,currCell,
ignM0,ignM1,ignM2,ignM3,ignM4,ignM5,ignM6,ignM7);
propCell1=ObtenerCeldasIncendiadas(0);
propCell2=ObtenerCeldasIncendiadas(1);
propCell3=ObtenerCeldasIncendiadas(2);
propCell4=ObtenerCeldasIncendiadas(3);
propCell5=ObtenerCeldasIncendiadas(4);
propCell6=ObtenerCeldasIncendiadas(5);
propCell7=ObtenerCeldasIncendiadas(6);
propCell8=ObtenerCeldasIncendiadas(7);
```

Figura 55: Calculando la propagación

Guardaremos los resultados obtenidos en las capas de información con el procedimiento **Nucleus** y enviaremos el evento *Propagar* a las nuevas celdas incendiadas con la ayuda de las extensiones extra.

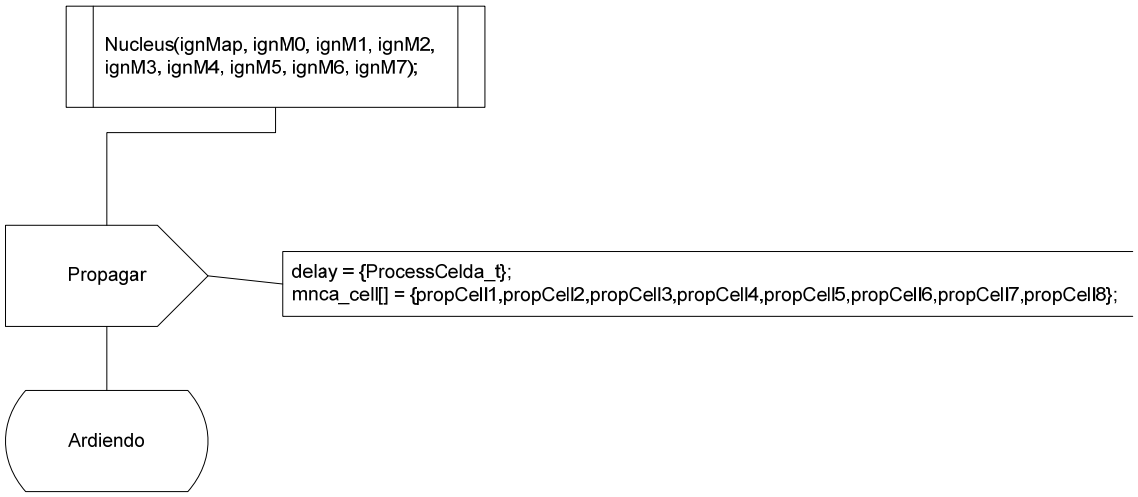

Figura 56: Enviando la señal de propagar

El diagrama completo quedaría como se ve a continuación.

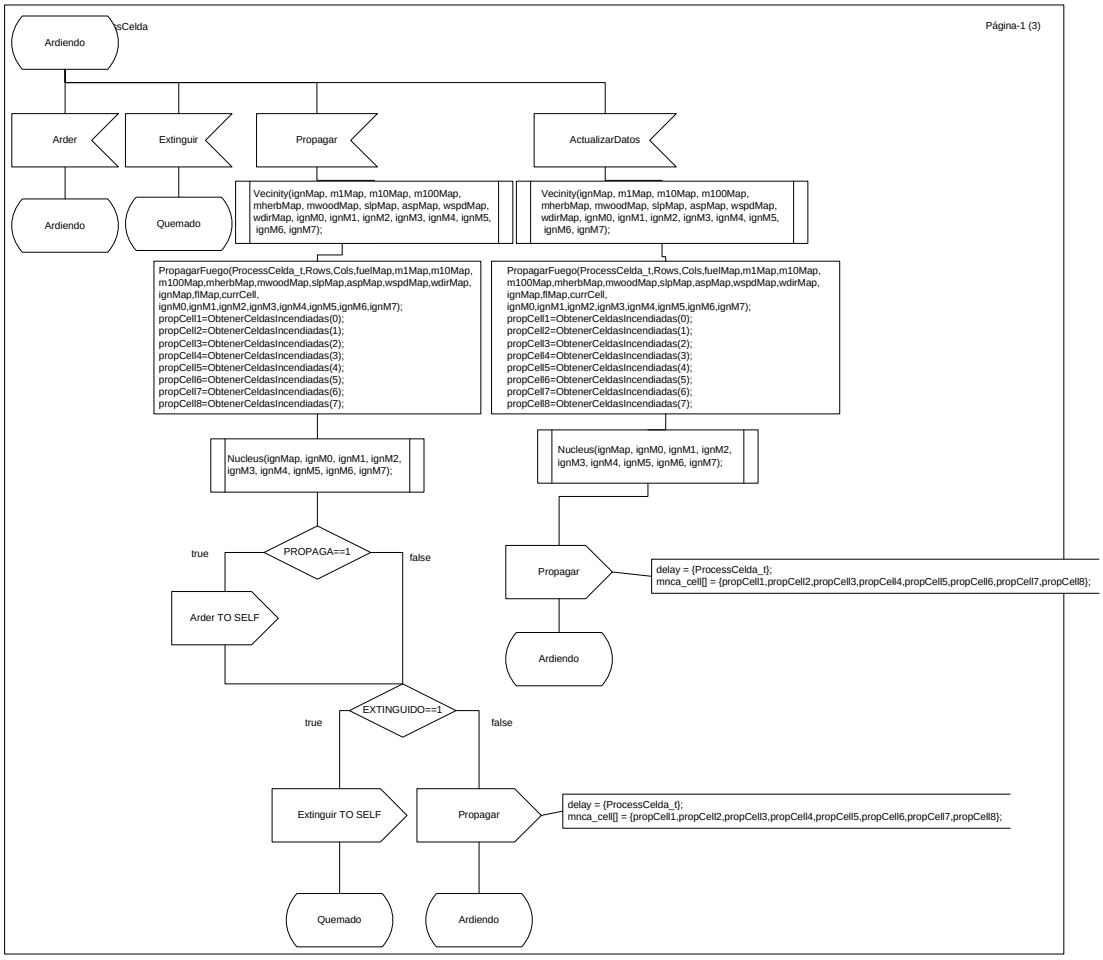

Figura 57: Diagrama de procesos, parte 2

Para finalizar el último estado, es el de **Quemado**, si se llega a este estado es porque se ha recibido un evento de *Extinguir*, lo que significa que ya no hay combustible en esta celda y por lo tanto acabará la simulación para esta celda.

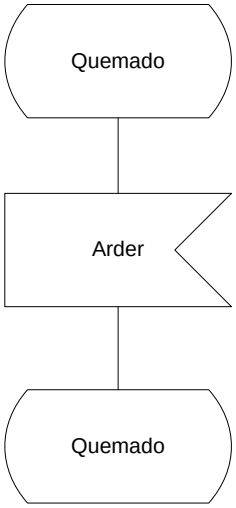

Figura 58: Diagrama de procesos, parte 3

A lo largo de la simulación hay varias funciones que se detallarán más adelante y aquí solo explicaré un poco por encima. Una de ellas es la de **PropagarFuego** que calcula si se propaga a las celdas vecinas o no y el tiempo que empieza la ignición de estas. Básicamente los parámetros que posee son relativos a la información de la celda actual y los tiempos de ignición de sus celdas vecinas.

void PropagarFuego(double timeNext, int Rows, int Cols, int fuelMap, double m1Map, double m10Map, double m100Map, double mherbMap, double mwoodMap, double slpMap, double aspMap, double wspdMap, double wdirMap, double ignMap, double flMap, int cell, double i0,double i1,double i2,double i3,double i4,double i5,double i6,double i7);

Otra función es la de **ObtenerCeldasIncendiadas**, que nos dice la posición de una de las ocho celdas vecinas que se ha de propagar el fuego.

int ObtenerCeldasIncendiadas(int i);

Con la función **getPropagar** podemos saber si la celda propaga el fuego a alguna de sus ocho celdas vecinas o no.

int getPropagar();

Y la función **getExtinguir** no permitirá saber si el combustible de la celda se ha agotado y por lo tanto no puede seguir ardiendo la celda.

int getExtinguir();

Todas estas funciones están explicadas en su totalidad y en detalle, junto con el diagrama de clases en su apartado correspondiente.

3.4 Generación XML

Una vez realizada la especificación, se procedió a traducir el modelo obtenido para pasarlo a un formato xml que el software SDLPS pudiese entender para probar la simulación. La relación entre los diagramas del modelo y el xml es directa por lo que no entraré en mucho detalle sobre el formato de este documento, si se desea se completo se puede ver en el anexo.

Como todo archivo xml, funciona mediante etiquetas:

Hay una etiqueta principal llamada “system” que representa todo nuestro sistema, y aquí es donde se especifica lo explicado en el diagrama del Sistema.

Dentro de esta etiqueta nos encontramos con una etiqueta que se llama “mnca” y nos representa lo expresado en el diagrama de MNA_Fuego, con todas las declaraciones y procesos explicados anteriormente en su correspondiente diagrama.

No seguimos adentrando por el xml y dentro de la etiqueta anterior nos encontramos la etiqueta de “block” que representa el diagrama de Bloques.

Inmediatamente después se encuentra la etiqueta “process” que nos permite especificar el diagrama de Procesos de nuestro modelo, detallando en todo momento al igual que los diagramas, las declaraciones, las funciones, los estados y las extensiones que hemos añadido al lenguaje SDL.

Como se veía en los diagramas, desde el nivel más alto vas entrando al siguiente nivel inferior, y cada nivel a su inferior respectivamente, hasta llegar al nivel más bajo que sería en nuestro caso el nivel de procesos y en el xml se hace igual, cada etiqueta va encerrando a su nivel inferior de esta manera se sigue la estructura de un xml correcto y de los diagramas especificados anteriormente.

4 IMPLEMENTACION

4.1 Elección lenguaje programación

Uno de lo que había que decidir era que lenguaje de programación elegir para implementar el modelo, había varios lenguajes, C, C++, .NET, Java… llegando a la conclusión que el lenguaje más adecuado era el lenguaje C, por la razones que explicaré a continuación.

Una de las grandes ventajas que tiene este lenguaje C respecto a otro como Java por ejemplo es la gran eficiencia que tiene a la hora de tratar mucha información y a la hora realizar los cálculos de esta información.

Se tenía que utilizar parte de código de la librería externa firelib, la cual está realizada en C, cosa que nos facilitaría el trabajo, aunque pasarla a C++ también hubiera sido posible.

También se han tenido que dejar de lado otros aspectos como la portabilidad, puede que en este aspecto Java sea mucho más portable que cualquiera de los otros lenguajes, pero en el caso de este proyecto mi proyecto viene condicionado con la utilización del programa SDLPS, que está desarrollado en C++, por lo que quedaba descartado el lenguaje Java. Finalmente nos quedaban dos lenguajes C++ y C, lo lógico hubiera sido elegir C++, pero por cuestiones del programa SDLPS era más fácil de mantener realizarlo en C que en C++.

De esta manera quedaba claro que los objetos importantes del modelo se compilaría en C++ (no permitiendo cambio del modelo alguno) mediante el programa SDLPS y los procesos externos, menos importantes como por ejemplo el cálculo de la propagación de Richard Rothermel (puede que alguien le interese utilizar otro cálculo de la propagación), permitiría la modificación por usuarios externos, se realizaría en C.

4.2 SDK SDLPS

El SDLPS nos proporciona un Kit de desarrollo de software, es decir, un conjunto de herramientas de desarrollo que nos permite que sea más fácil a la hora de programar.

Más concretamente nos proporciona las siguientes funciones:

- mncaGetCurrCell
- mncaGetCellValue
- mncaSetCellValue

La primera función **mncaGetCurrCell** nos permite saber en qué celda está la simulación y tiene como parámetro un entero que será donde nos guarda el valor de la posición de la celda actual.

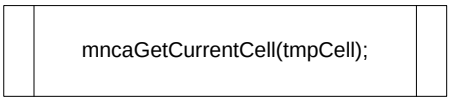

Figura 59: Función mncaGetCurrCell

La función **mncaGetGetValue** nos permite saber la información que contiene una capa de nuestro modelo, indiferentemente de la capa que se precise consultar, por ejemplo puede contener el tiempo de ignición de las celdas, la dirección del viento ó la pendiente del terreno, entre muchas otras capas. Recibe como parámetros un string con el nombre de la capa, un entero con la posición de la celda que se quiere consultar y un tercer parámetro donde guardar el resultado.

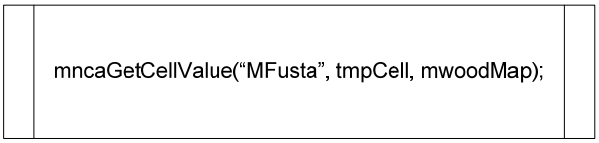

Figura 60: Función mncaGetGetValue

La tercera función **mncaSetValue** nos permite guardar la información, que hemos podido manipular ó volver calcular, y salvarla de nuevo a la capa que le corresponda. Recibe como parámetros un string con el nombre de la capa, un entero con la posición de la celda que se está tratando y un último parámetro que será el que dejará en la capa.

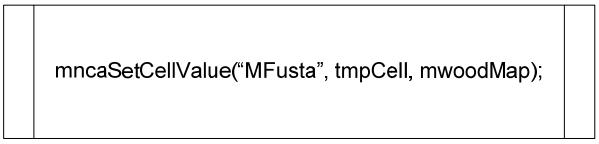

Figura 61: Función mncaSetCellValue

Con estas tres funciones tenemos más que suficiente para poder llevar a cabo la simulación.

4.3 Diagrama de clases

El diagrama de clases que se expone a continuación solo es de referente al código externo que utiliza el simulador, ya que el simulador SDLPS crea automáticamente el código del modelo y posteriormente una librería que no podemos modificar.

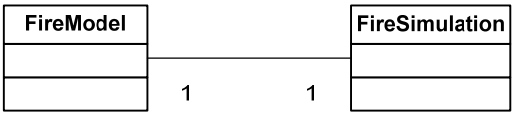

Figura 62: Diagrama de clases

En este diagrama nos encontramos con la clase FireModel que representa el modelo del fuego, en nuestro caso extraído directamente de la librería firelib y que se define toda la clase en el anexo.

La clase FireSimulation es la que nos define las funciones relativas a la propagación del fuego y las únicas que el usuario podría modificar si fuera el caso que lo necesitase, porque prefiere usar otras funciones de propagación.

Los métodos de la clase FireSimulation son los siguientes:

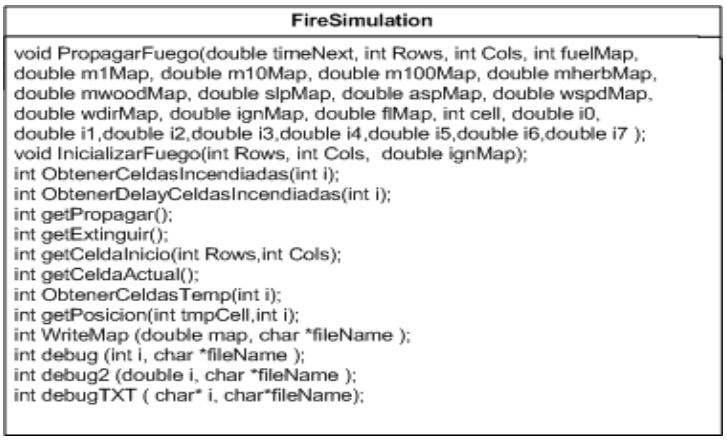

Figura 63: Clase FireSimulation

PropagarFuego:

Tipo de retorno: void

Descripción: Ejecuta todos los calculos necesarios para saber si una celda incendiada propaga el fuego a una de sus ocho celdas vecinas.

Parámetros:

timeNext

Expresión del tipo: double

Tipo: entrada

Descripción: Valor con el tiempo actual de la simulación.

Rows

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de filas de la parrilla de celdas

Cols

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de columnas de la parrilla de celdas

fuelMap

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el tipo de combustible de la celda

m1Map

Expresión del tipo: double

Tipo: entrada

Descripción: Contenido de humedad del combustible muerto pasada 1 hora

m10Map

Expresión del tipo: double

Tipo: entrada

Descripción: Contenido de humedad del combustible muerto pasada 10 horas

m100Map

Expresión del tipo: double

Tipo: entrada

Descripción: Contenido de humedad del combustible muerto pasada 100 horas

mherbMap

Expresión del tipo: double

Tipo: entrada

Descripción: Contenido de humedad de la hierba

mwoodMap

Expresión del tipo: double

Tipo: entrada

Descripción: Contenido de humedad de la madera

slpMap

Expresión del tipo: double

Tipo: entrada

Descripción: Contiene la pendiente del terreno

aspMap

Expresión del tipo: double

Tipo: entrada

Descripción: Valor de con la orientación del terreno

wspdMap

Expresión del tipo: double

Tipo: entrada

Descripción: Valor con la velocidad del viento

wdirMap

Expresión del tipo: double

Tipo: entrada

Descripción: Valor con la dirección del viento

ignMap

Expresión del tipo: double

Tipo: entrada

Descripción: El tiempo de ignicion del la celda

flMap

Expresión del tipo: double

Tipo: entrada

Descripción: Valor con la longitud de la llama

cell

Expresión del tipo: int

Tipo: entrada

Descripción: Valor de la posición de la celda a tratar

i0

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la primera celda vecina

i1

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la segunda celda vecina

i2

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la tercera celda vecina

i3

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la cuarta celda vecina

i4

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la quinta celda vecina

i5

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la sexta celda vecina

i6

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la septima celda vecina

i7

Expresión del tipo: double

Tipo: entrada

Descripción: Tiempo de ignición de la octava celda vecina

InicializarFuego:

Tipo de retorno: void

Descripción: Ejecuta todos los calculos necesarios para iniciar la simulación.

Parámetros:

Rows

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de filas de la parrilla de celdas

Cols

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de columnas de la parrilla de celdas

IgnMap

Expresión del tipo: double

Tipo: entrada

Descripción: El tiempo de ignición de la celda

ObtenerCeldasIncendiadas:

Tipo de retorno: int

Descripción: Obtienes la posición de una celda vecina.

Parámetros:

i

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de la celda vecina que quieres obtener

ObtenerDelayCeldasIncendiadas

Tipo de retorno: int

Descripción: Obtienes el retardo de una celda vecina.

Parámetros:

i

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de la celda vecina que quieres obtener

getPropagar:

Tipo de retorno: int

Descripción: Obtienes si la celda propaga el fuego o no.

getExtinguir:

Tipo de retorno: int

Descripción: Obtienes si el fuego de la celda se ha extinguido.

getCeldaInicio:

Tipo de retorno: int

Descripción: Obtienes la posición de la celda de inicio.

Parámetros:

Rows

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de filas de la parrilla de celdas

Cols

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de columnas de la parrilla de celdas

getCeldaActual:

Tipo de retorno: int

Descripción: Obtienes la posición de la celda en curso.

ObtenerCeldasTemp:

Tipo de retorno: int

Descripción: Obtienes el tiempo de ignición de la celda pedida (método de debug)

Parámetros:

i

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de la celda que quieres obtener

getPosicion:

Tipo de retorno: int

Descripción: Permite obtener la posición exacta en fila ó columna de una determinada celda

Parámetros:

tmpCell

Expresión del tipo: int

Tipo: entrada

Descripción: Valor con el número de la celda que quieres obtener

i

Expresión del tipo: int

Tipo: entrada

Descripción: Valor que permite devolver la fila (0) ó la columna (1)

WriteMap:

Tipo de retorno: int

Descripción: Permite escribir en un fichero el resultado de un mapa de terreno (método de debug)

Parámetros:

map

Expresión del tipo: double

Tipo: entrada

Descripción: El mapa que quiere observar

fileName

Expresión del tipo: char *

Tipo: entrada

Descripción: El path completo del fichero a crear

debug:

Tipo de retorno: int

Descripción: Escribe en un fichero la variable dada (método de debug)

Parámetros:

i

Expresión del tipo: int

Tipo: entrada

Descripción: La variable interesada

fileName

Expresión del tipo: char *

Tipo: entrada

Descripción: El path completo a crear

int debug2:

Tipo de retorno: int

Descripción: Escribe en un fichero la variable interesada (método de debug)

Parámetros:

i

Expresión del tipo: double

Tipo: entrada

Descripción: La variable interesada

fileName

Expresión del tipo: char *

Tipo: entrada

Descripción: El path completo del fichero a crear

debugTXT:

Tipo de retorno: int

Descripción: Escribe en un fichero el texto pasado por parámetro (método de debug)

Parámetros:

i

Expresión del tipo: char *

Tipo: entrada

Descripción: La cadena de caracteres interesada

fileName

Expresión del tipo: char *

Tipo: entrada

Descripción: El path completo del fichero a crear

4.4 Simulación distribuida

Los sistemas de eventos discretos son de gran utilidad para hacer simulaciones de todo tipo, pero en muchos casos la ejecución de estas simulaciones es poco eficiente. El principal problema de es la velocidad, ya que tiene que manejar gran cantidad de información, además de la complejidad de los modelos y cálculos. Para que sea significativa una simulación debe de generar un número suficiente de evoluciones del sistema.

Por eso tenemos que obtener una simulación más rápida y a primera vista eso lo podemos hacer dedicando más recursos, se puede ver claro que podemos acelerar la simulación utilizando un sistema de multiprocesador en lugar de un solo procesador, pero claro para una simulación, en la que son sistemas que tienen muchos componentes operando en paralelo, lo más razonable sería explotar esto y dar un gran salto de calidad para solventar este problema, lo mejor sería hacer el modelo distribuido, es decir una simulación distribuida.

La simulación distribuida sería la ejecución de la simulación en diversas computadoras, y esto lo podemos hacer porque hay muchos componentes operando en paralelo, de manera que mejoraríamos la velocidad de la simulación.

Además de la velocidad, a veces el sistema puede que no tenga suficiente memoria para manejar toda la simulación, de esta manera distribuida, haciendo que la memoria no sea compartida, también arreglamos este problema.

Llegados a este punto, hace falta hacerse una pregunta ¿Cómo hacer la simulación distribuida? Para realiza estoy hay varias formas:

Hacer replicaciones independientes de una simulación existente en computadoras por separado y luego recoger los resultados. Aunque parezca que sea muy eficiente porque cada computadora ejecuta una simulación diferente requiere que cada una tenga memoria suficiente para contener toda la simulación, cosa que no siempre es posible.

Hacer las funciones distribuidas, asignando tareas diferentes a cada computadora. Lo malo es que no explota el paralelismo ya que cada tarea se tiene que hacer una detrás de otra.

Utilizar eventos distribuidos aprovechando que solo se cambia de estado en puntos discretos de tiempo, es decir cuando se produce un evento.

Descomponer el modelo en varias partes, cada parte requerirá menos información y en consecuencia menos memoria, uno de los problemas comentados anteriormente. El único problema es que hay que tener cuidado con la sincronización.

La última forma es la que más beneficios nos aportará, por eso se decidió descomponer el modelo, además que el software utilizado, el SDLPS, nos permite este tipo de simulación, lo más lógico sería que en este tipo de simulaciones se utilizasen supercomputadoras porque estas poseen capacidades de cálculo muy superiores a las comunes, permitiendo incluso lanzar muchas simulaciones a la vez con los datos diferentes para estudiar y comparar sus resultados, cosa que con una computadora común se hace inviable.

Gracias la simulación distribuida podemos organizar los diferentes componentes de una simulación en diferentes máquinas, por ejemplo, en una maquina el mapa de vientos, en otra el mapa de la lluvia, etc. De esta manera no solo sirve cada máquina para una única simulación, sino una maquina que contiene un componente como puede ser el mapa de lluvias, puede ser aprovechada por otras simulaciones que también necesiten datos de este mapa.

4.4.1 Ley de Amdahl

Con la ley de Amdahl (J.M. Llavería s.f.) podemos medir el rendimiento de un computador, ya que podemos conocer la ganancia que nos aporta cuando ejecutamos la simulación de manera distribuida respecto de la original.

La Ley de Amdalh establece que “la ganancia por añadir la una mejora en un diseño está limitada por la fracción del tiempo original en que se utiliza la mejora”.

La fórmula original de la ley de Amdahl es la siguiente:

$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$

Figura 64: Ley de Amdahl

A es la ganancia conseguida debido a la mejora

F_m es la fracción de tiempo que se utiliza la mejora

A_m es la ganancia cuando se utiliza la mejora

Si aplicamos la ley de Amdahl a nuestro caso concreto para ver la ganancia que se tiene al hacer la simulación distribuida obtenemos los resultados siguientes:

F_m (% del tiempo)	G_m	Ganancia (%)
25	2	14
50	2	33
75	2	59
100	2	100
25	3	20
50	3	49
75	3	100
100	3	203

F_m representa el % del tiempo que se ha utilizado la mejora, en nuestro caso es el tiempo que ha estado la simulación distribuida entre varias computadoras.

G_m representa la ganancia cuando se utiliza dicha mejora, en nuestro caso la ganancia es igual al número de computadoras utilizadas (suponiendo que tienen un procesador cada una).

Según los resultados obtenidos se puede ver que cuantos más computadoras tengamos disponibles para la simulación distribuida, mejor será el tiempo de ejecución de la simulación (más ganancia), pero no solo eso, sino que, el aspecto más relevante es que cuanto más distribuido tengamos el modelo, mayor fracción de tiempo podremos utilizar esta mejora y como consecuencia directa, se puede ver en la tabla, que la ganancia aumenta bastante más con la variación de esta fracción de tiempo, por

lo tanto se puede concluir que la realizar una distribución del modelo, tiene más peso y obtiene más ganancia que elevar el número de computadoras.

5 EXPERIMENTACION

5.1 Estructura del simulador

La estructura del simulador está compuesta por la carpeta raíz y unas subcarpetas y ficheros con diferente información, su estructura es la siguiente:

El archivo principal del simulador es el modelo en SDL, que está especificado en la carpeta raíz del proyecto y se llama SystemCelda.xml

En la carpeta raíz también nos aparecen los archivos que genera automáticamente el SDLPS, SDLCode.c, SDLCOde.o y SDLCode.dll, que contienen el código fuente, código objeto y la librería del simulador. También nos podremos encontrar dependiendo de la simulación un fichero llamado trace.csv que contiene todos los pasos que se han realizado en la simulación y que sirve para debugar.

Dentro de la capeta raíz tenemos cuatro subcarpetas:

La carpeta data contiene la información de las capas relativas a las características del terreno a simular, hay dos tipos de archivos, unos son los ".doc" que describen como están compuestos los datos y otros archivos ".img" que contienen los datos raster.

Otra carpeta doc, contiene los diagramas del modelo realizados con el Microsoft Visio y Sandrila, exactamente tal y como están descritos en el archivo xml principal.

En la carpeta Memoria está la documentación realizada para este proyecto.

Y por ultimo una última carpeta SDLCodeExternal que contiene las cabeceras, archivos ".h" y el código fuente, archivos ".c" de los métodos externos que necesitamos para llevar a cabo la simulación.

5.2 Verificación, validación y experimentación del modelo

En las simulaciones hay unas fases que tienen gran importancia, una de ellas es la verificación, debemos de saber si hemos construido correctamente el modelo, es un paso importante para probar el modelo, pero aunque resulte positiva, no nos asegura que el modelo sea correcto.

Otra fase es la validación, que consiste en saber si el modelo que hemos conseguido a lo largo de este proyecto, es el correcto para el sistema real del comportamiento del fuego en los incendios forestales o no.

Nuestro modelo obtendrá como resultados unas salidas que nos servirán para obtener conclusiones sobre nuestro sistema y lo comparemos con el sistema real, por lo tanto con esta comparativa nos aseguraremos que nuestro modelo sea válido.

En nuestro caso se ha utilizado el modelo de Behave, en particular para la propagación del fuego se ha utilizado la librería firelib, la cual ya está validada y por ese motivo no se entrará en muchos detalles.

Partiendo de este modelo ya validado lo que haremos será comparar los resultados del modelo de Behave que ya es correcto y compararlo con los resultados obtenidos por nuestro modelo, de esta manera sabremos si los resultados que estamos logrando con nuestra simulación son correctos o no.

Empezaremos comprobando que las todos los datos relevantes que tiene el sistema real también están en nuestro sistema.

Para empezar los datos de entrada tanto del sistema real como de nuestro sistema son idénticos, ya que cogeremos la información sobre combustible, viento, pendiente… de un ejemplo de las librerías firelib.

En una primera fase del proyecto se ha ejecutado la simulación en nuestro modelo no distribuido y el resultado han sido salidas exactas para nuestro modelo y el modelo de Behave, por lo tanto se puede decir que nuestro modelo es fiable, que ha pasado la verificación y la validación.

El resultado gráfico de la simulación del modelo Behave y nuestro modelo es el siguiente:

Figura 65: Resultado Behave

Figura 66: Resultado simulador no distribuido

La zona azul marca las celdas ó parcela de terreno forestal que no ha sido incendiado, por lo tanto no ha sido afectada por el fuego y la zona blanca marca la zona de terreno donde se ha propagado el fuego. El resultado que hemos obtenido por ambas simulaciones es una matriz con los tiempos en que se inicia el fuego en la celda (se puede encontrar en los anexos) aunque para un entendimiento más claro, como el SDLPS también muestra el resultado gráfico, se decidió pasar el resultado obtenido por el modelo de Behave a gráfico también.

En la segunda fase, la ejecución de la simulación en un simulador distribuido, por razones de tiempo y porque el SDLPS aún está en pleno desarrollo, no hemos podido obtener los resultados para compararlos con el modelo de Behave. Aunque no debería haber problemas para obtener unos resultados satisfactorios, ya que el modelo y el

código externo es prácticamente el mismo, añadiendo las extensiones al SDL ya explicadas anteriormente.

Este simulador te permitiría crear varios inicios de fuegos en el mismo terreno, lo único que no podemos validar los resultados ya que no disponemos de otros resultados válidos para poder compararlos.

6 PLANIFICACIÓN Y COSTES

6.1 Fase documentación

La primera fase, que va desde el día 1 de Septiembre de 2009 hasta el día 15 de Octubre de 2009, se centró en documentar y recopilar información sobre los siguientes temas:

- Estudio de los simuladores.
- Estudio de las características del fuego y de los incendios forestales
- Estudio del modelo de Behave y librería firelib
- Estudio de la herramienta SDLPS
- Estudio de los sistemas de información geográfica
- Estudio de los autómatas celulares distribuidos

Esta fase se realizó con éxito, según la planificación y se dispuso de los conocimientos necesarios para realizar el proyecto con garantías y para que sea representativo de un comportamiento de un incendio forestal real.

El reparto del tiempo en esta fase quedó de la manera siguiente:

Tema	Tiempo (días)
Simuladores	2
Características del fuego e incendios forestales	5
Simulación del comportamiento del fuego	10
Lenguaje SDL	6
Sistemas de información geográfica	2
Idrisi32	2
SDLPS	5
Firelib	10
MNCA	9

Después de realizar está fase ha dado como resultado un total de 33 días de trabajo, si se suma la columna de días el resultado no dará 33 esto es porque había tareas que se podían realizar simultáneamente, esto se puede ver en el diagrama de Gantt.

6.2 Fase Modelado y Diseño

La fase de modelado y diseño, comprende entre los meses Noviembre y Diciembre, esta es una de las fases más importantes y en la cual había que tratar lo siguiente:

- La especificación del Modelo, donde había que diseñar el modelo en lenguaje SDL, especificando claramente los diagramas de sistema, mnca, bloque, estado y de procesos.
- Generación del XML del modelo creado

La fase de diseño se completó con éxito en el periodo planificado, aunque como era de esperar, durante la implementación ha habido puntos conflictivos que no se habían tenido en cuenta, producidos por la inexperiencia en este terreno, y eso ha hecho modificar el diseño durante la fase de implementación, aunque han sido pequeños cambios que no han tenido gran repercusión en los diagramas.

El tiempo destinado a esta fase queda de la siguiente manera:

Tema	Tiempo (días)
Diagrama de sistema	2
Diagrama de MNCA	13
Diagrama proceso Vecinity	2
Diagrama proceso Nucleus	2
Diagrama de bloque	2
Diagrama de estados	1
Diagrama de procesos	16
Generación XML	19

El total de días destinados para esta fase es de 58 días, esta era la fase más importante del proyecto y por eso ninguna tarea se hace simultáneamente, hasta no tener una clara no se pasó a la siguiente.

6.3 Fase Implementación

La fase de implementación comprende entre los meses de enero, febrero, marzo y abril, los temas a tratar son los siguientes:

- Elección del lenguaje de programación
- Diagrama de clases
- Adaptación firelib
- Paralelización

El reparto de tiempo para estas tareas quedo de la siguiente manera:

Tema	Tiempo (días)
Elección de lenguaje	2
Diagrama de clases	7
Adaptación firelib	45
Paralelización	17

El total de días es de 70 días de trabajo

6.4 Fase Experimentación

La fase de experimentación comprende los meses de marzo, abril, mayo y se trataban los temas que nombro a continuación:

Tema	Tiempo (días)
Experimentación	16

El total de días es de 16

6.5 Fase de la memoria

La fase de memoria comprende los meses que van desde enero hasta la finalización del proyecto en mayo.

Tema	Tiempo (días)
Creación memoria	99

Las fechas exactas y diagrama de Gantt se muestran a continuación:

Diagrama de Gantt

	Nombre de tarea	Duración	Comienzo	Fin
4	Simulacion del comportamiento del fuego	10 días	vie 04/09/09	jue 17/09/09
5	Lenguaje SDL	6 días	vie 11/09/09	vie 18/09/09
6	Sistemas de Información Geográfica	2 días	mar 15/09/09	mié 16/09/09
7	Idrisi32	2 días	mar 15/09/09	mié 16/09/09
8	SDLPS	5 días	mar 15/09/09	lun 21/09/09
9	Firelib	10 días	lun 21/09/09	vie 02/10/09
10	MNCA	9 días	lun 05/10/09	jue 15/10/09
11	**Modelado**	**58 días**	**jue 15/10/09**	**lun 04/01/10**
12	**Especificacion del modelo**	**39 días**	**jue 15/10/09**	**mar 08/12/09**
13	Diagrama de Sistema	2 días	jue 15/10/09	vie 16/10/09
14	Diagrama de Bloques	2 días	lun 19/10/09	mar 20/10/09
15	Diagrama estados	1 día	mié 21/10/09	mié 21/10/09
16	Diagrama de Procesos	16 días	jue 22/10/09	jue 12/11/09
17	Diagrama MNCA	13 días	vie 13/11/09	mar 01/12/09
18	Diagrama proceso Vecinity	2 días	jue 03/12/09	vie 04/12/09
19	Diagrama proceso Nucleus	2 días	lun 07/12/09	mar 08/12/09
20	Generación XML	19 días	mié 09/12/09	lun 04/01/10
21	**Implementación**	**70 días**	**vie 15/01/10**	**lun 19/04/10**
22	Elección lenguaje programación	2 días	vie 15/01/10	lun 18/01/10
23	Diagrama de clases	7 días	mar 19/01/10	mié 27/01/10
24	Adaptación Firelib	45 días	jue 28/01/10	vie 26/03/10
25	Paralelización	17 días	vie 26/03/10	lun 19/04/10
26	Experimentación	16 días	mar 27/04/10	mar 18/05/10
27	Documentación	99 días?	vie 15/01/10	vie 28/05/10

Figura 67: Fechas planificación

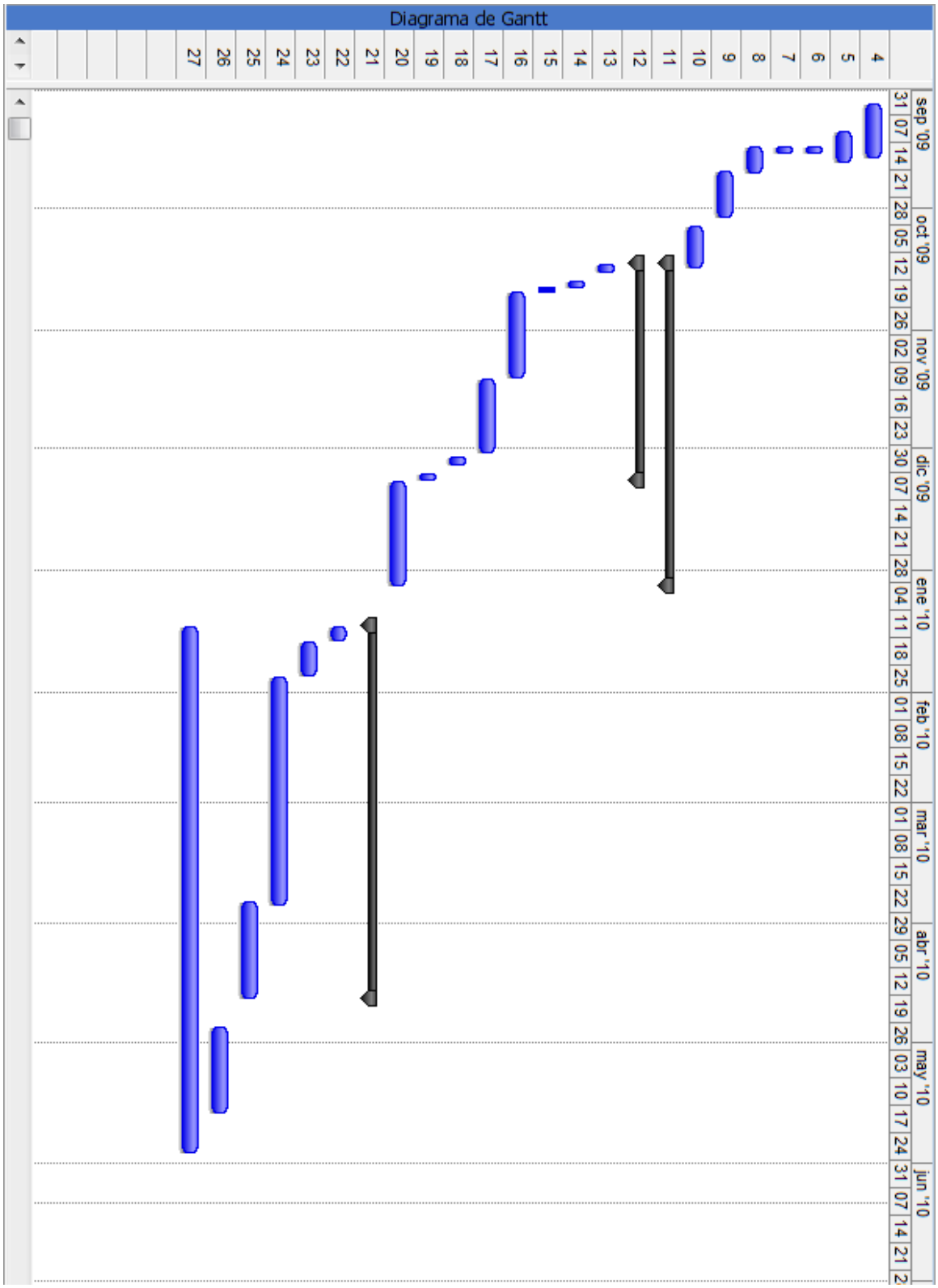

Figura 68: Diagrama de Gantt

Los costes de este proyecto están divididos en dos partes, por un lado los costes referentes a materiales, básicamente software y por otro lado el coste por días de trabajo.

Los costes imputados al material son los siguientes:

Software	Coste (€)
Windows Vista	210
Microsoft Office	208
SDLPS	0
Sandrila	23
TOTAL	441

El total de este coste es de 441€.

Los costes de personal son los siguientes:

Fase	Días	Coste por día (€)	Coste fase (€)
Documentación	33	480	15840
Diseño	58	480	27840
Implementación	70	320	22400
Experimentación	16	320	5120
TOTAL			71200

El total del coste de personal sería de 71200 €.

El coste total del proyecto sería la suma del coste imputados al material más el coste imputado al personal, en definitiva de unos 71641 €.

7 CONCLUSIONES

Hay que tener en cuenta que con este proyecto no se ha tratado toda la problemática del fuego, más concretamente solo se trató la parte de la propagación del fuego en un incendio forestal, en futuros proyectos se podrá ampliar este mismo proyecto para que siga tratando otros problemas de interés que aparecen en los incendios forestales como sería la extinción del fuego, por poner un ejemplo claro.

Con este proyecto también se ha podido ver las posibilidades que ofrecen las nuevas tecnologías y las diferentes herramientas como el SDLPS, que permiten crear simuladores complejos a partir de diagramas especificados en lenguaje SDL desvinculando a la persona de sus conocimientos o no de los lenguajes de programación existentes, simplemente conociendo el lenguaje SDL que es mucho más visual y fácil de aprender que cualquier lenguaje de programación, cada uno se podrá crear su modelo y ejecutar una simulación sin tener que tocar código.

A todo esto hay que decir que como se pueden ejecutar simulaciones distribuidas hace que se puedan ejecutar más simulaciones a la vez y poder recoger más resultados diferentes para poder tomar las decisiones adecuadas.

Las simulaciones son bastante próximas a la realidad ya que permiten recoger los datos de entrada de sistemas GIS, información del terreno real, que están en constante actualización y mientras más precisión de los datos de entrada más cerca estaremos de un incendio real.

También hay que destacar que hasta ahora las herramientas que había para realizar simulaciones de incendios forestales, los resultados se mostraban mediante tablas ó gráficos simples al final de la simulación, en cambio, ahora nos permitirá mostrar los resultados en tiempo real a la vez que se ejecuta la simulación y mostrar los resultados en tres dimensiones, de una manera más amigable y entendible.

Resumiendo se podría concluir subrayando los siguientes puntos:

- Los sistemas GIS nos permiten representar las condiciones muy próximas a las condiciones reales de un incendio.

- Cualquiera podría construir un simulador complejo, partiendo de un buen modelo (en mi caso fue Behave), sin necesidad de tocar código, solo conociendo el lenguaje SDL.

- La simulación distribuida nos proporciona más velocidad de ejecución, la posibilidad de realizar más ejecuciones a la vez y por lo tanto de obtener más resultados.

- Podemos ver en tres dimensiones y en tiempo real como transcurre un incendio forestal, lo que hace más cercano el incendio virtual, al incendio real.

- Lo último a remarcar es que a este proyecto, se puede seguir ampliándolo para tratar las demás problemáticas no tratadas de los incendios forestales y llegar hasta un punto que la simulación sea lo más próxima posible al incendio real.

8 TRABAJO FUTURO

En este proyecto se podrían mejorar varios aspectos e implementar otros ya que por la duración del proyecto ha sido imposible.

Los aspectos a mejorar serían los siguientes:

- Se podrían añadir los otros modelos de incendios, ya que hemos visto que en este proyecto solo trata los incendios de superficie, por ejemplo tratar el incendio de copas, incluso como interactuarían entre ellos.
- Respecto a la salida de la simulación se podría mejorar haciéndola de una manera gráfica, más vistosa, agradable como puede ser mediante VRML (para representaciones 3D) ó también mostrar cómo transcurre el incendio en la zona afecta real en Google Earth.

Los puntos que se podrían añadir serían los que vienen a continuación:

- Integrar la simulación en tiempo real con los sistemas GPS actuales, en la universidad están los GPS rino que te permitirían saber la posición exacta en el terreno forestal y la comunicación vía radio, de manera que te permitiría modificar los datos de entrada de la simulación sobre el terreno y realizar unas simulaciones más precisas.
- Añadir la problemática de la extinción del fuego sería una opción muy interesante que podría acabar de complementar a este simulador.

9 ANEXOS

9.1 Problemas

Muchos de los problemas que me he encontrado al realizar este proyecto, vienen sobretodo porque es un proyecto real y derivan de la colaboración con otros proyectos que actualmente se están desarrollando (uno con dos estudiantes de la UPC y otro con un estudiante de otra universidad), por lo que he tenido que ir sorteando estos problemas y solucionándolos como he podido.

El paso de los diagramas a xml se podía hacer mediante un plugin del Microsoft Visio realizado por otros proyectistas, pero justamente para mi caso, no lo hacía correctamente, ya que en mis diagramas hay un caso que hay un bucle y esto no lo trataba bien, por lo que decidí finalmente coger los DTD y realizarlo a mano.

9.2 Formato idrisi32

El fichero de datos será de tipo raster.

Raster: divide el espacio en celdas regulares, donde cada una de ellas representa un único valor. La estructura lógica es una malla ó matriz, pero se guarda físicamente en un fichero con una columna.

Un ejemplo de datos raster

1	2	3
4	5	6
7	8	9

En disco lo guardaremos así:

1

2

3

4

5

6

7

8

9

Los valores raster pueden ser enteros o números con coma flotante y los archivos serán de tipo ASCII, donde cada información raster estará compuesta por un documento ".doc" con la información relativa a los datos raster y los datos raster estarán en ficheros con la extensión ".img". Los dos archivos tendrán el mismo nombre.

9.3 XML generado

El archivo que especifica el modelo y lo ejecuta en el programa SDLPS es el siguiente:

```
<?xml version="1.0"?>
<system id="0" name="SystemCelda" implementation=""
IP="192.168.0.137" portRead="8695" version="1.0">
     <channels>
     </channels>
    <mnca id="1" name="MNCA_FUEGO" implementation=""
IP="192.168.0.137" portRead="8689">
     <DCLS>
          <DCL name="mnca_DIM" type="int" value="2"/>
          <DCL name="mnca_D1" type="int" value="101"/>
          <DCL name="mnca_D2" type="int" value="101"/>
    <DCL name="mnca_M1[]" type="double" value=""/>
    <DCL name="mnca_M10[]" type="double" value=""/>
    <DCL name="mnca_M100[]" type="double" value=""/>
    <DCL name="mnca_Mapa[]" type="double" value=""/>
    <DCL name="mnca_MDirVents[]" type="double" value=""/>
    <DCL name="mnca_MFusta[]" type="double" value=""/>
    <DCL name="mnca_MHerba[]" type="double" value=""/>
    <DCL name="mnca_MOrientacio[]" type="double" value=""/>
    <DCL name="mnca_MPendent[]" type="double" value=""/>
    <DCL name="mnca_MVelVents[]" type="double" value=""/>
    <DCL name="mnca_Model[]" type="double" value=""/>
    <DCL name="mnca_MInici[]" type="double" value=""/>
     </DCLS>
     <channels>
     </channels>
     <block id="1" name="BlockCelda" implementation=""
IP="192.168.0.137" portRead="8689">
```

```
          <channels>
          </channels>
          <process id="1" name="ProcessCelda"
implementation="" IP="192.168.0.137" portRead="8690">
               <DCLS>
                    <DCL name="PROPAGA" type="int"
value="0"></DCL>
                    <DCL name="EXTINGUIDO" type="int"
value="0"></DCL>
                    <DCL name="ProcessCelda_t"
type="double" value="0"></DCL>
        <DCL name="currCell" type="int" value="0"></DCL>
        <DCL name="tmpCell" type="int" value=""></DCL>
        <DCL name="sum" type="double" value="0"></DCL>
        <DCL name="tmp1" type="double" value="0"></DCL>
        <DCL name="tmp2" type="double" value="0"></DCL>
        <DCL name="nrow" type="int" value="0"></DCL>
        <DCL name="ncol" type="int" value="0"></DCL>
        <DCL name="Rows" type="int" value="101"></DCL>
        <DCL name="Cols" type="int" value="101"></DCL>
        <DCL name="ignMap" type="int" value="1"></DCL>
        <DCL name="timeNext" type="double" value="0"></DCL>
        <DCL name="nTimes" type="int" value="0"></DCL>
        <DCL name="atEdge" type="int" value="0"></DCL>
        <DCL name="fuelMap" type="int" value="1"></DCL>
        <DCL name="m1Map" type="double" value=".10"></DCL>
        <DCL name="m10Map" type="double" value=".10"></DCL>
        <DCL name="m100Map" type="double"
value=".10"></DCL>
        <DCL name="mherbMap" type="double"
value="1."></DCL>
        <DCL name="mwoodMap" type="double"
value="1.50"></DCL>
        <DCL name="slpMap" type="double" value="0"></DCL>
        <DCL name="aspMap" type="double" value="0"></DCL>
        <DCL name="wspdMap" type="double" value="4."></DCL>
        <DCL name="wdirMap" type="double" value="0"></DCL>
        <DCL name="flMap" type="double" value="0"></DCL>
        <DCL name="ignM0" type="double" value="9999"></DCL>
        <DCL name="ignM1" type="double" value="9999"></DCL>
        <DCL name="ignM2" type="double" value="9999"></DCL>
        <DCL name="ignM3" type="double" value="9999"></DCL>
        <DCL name="ignM4" type="double" value="9999"></DCL>
        <DCL name="ignM5" type="double" value="9999"></DCL>
        <DCL name="ignM6" type="double" value="9999"></DCL>
        <DCL name="ignM7" type="double" value="9999"></DCL>
        <DCL name="propCell1" type="int" value="-1"></DCL>
        <DCL name="propCell2" type="int" value="-1"></DCL>
        <DCL name="propCell3" type="int" value="-1"></DCL>
        <DCL name="propCell4" type="int" value="-1"></DCL>
```

```
        <DCL name="propCell5" type="int" value="-1"></DCL>
        <DCL name="propCell6" type="int" value="-1"></DCL>
        <DCL name="propCell7" type="int" value="-1"></DCL>
        <DCL name="propCell8" type="int" value="-1"></DCL>
        <DCL name="delayCell" type="int" value="-1"></DCL>
      </DCLS>
              <procedures>
                   <procedure id="1" name="ArrivalTime"
implementation="">
                        <params>
                             <param name="ArrivalTime_t"
type="double" defvalue="" ref="yes"></param>
                        </params>
                        <body>
                             <task id="1"
name="">ArrivalTime_t=1;</task>
                        </body>
                   </procedure>
                   <procedure id="2" name="PropagarFuego"
implementation="">
                        <body>
                        </body>
                   </procedure>
        <procedure id="3" name="InicializarFuego"
implementation="">
          <body>
          </body>
        </procedure>
        <procedure id="4" name="mncaGetCurrentCell"
implementation="CSDLOperationProcedureCallmncaGetCurrentCel
l">
          <params>
            <param name="MNCA_CURR_CELL" type="int"
defvalue="" ref="yes"></param>
          </params>
        </procedure>
        <procedure id="5" name="mncaGetCellValue"
implementation="CSDLOperationProcedureCallmncaGetCellValue"
>
          <params>
            <param name="MNCA_GET_LAYER" type="char*"
defvalue="" ref="yes"></param>
            <param name="MNCA_GET_CELL" type="int"
defvalue="" ref="yes"></param>
            <param name="MNCA_GET_VALUE" type="double"
defvalue="" ref="yes"></param>
          </params>
        </procedure>
```

```
        <procedure id="6" name="mncaSetCellValue"
implementation="CSDLOperationProcedureCallmncaSetCellValue"
>
          <params>
            <param name="MNCA_SET_LAYER" type="char*"
defvalue="" ref="yes"></param>
            <param name="MNCA_SET_CELL" type="int"
defvalue="" ref="yes"></param>
            <param name="MNCA_SET_VALUE" type="double"
defvalue="" ref="yes"></param>
          </params>
        </procedure>
        <procedure id="6" name="FindDestination"
implementation="CSDLOperationProcedureCallFindDestination">
          <params>
            <param name="nextCell" type="int" defvalue=""
ref="yes"></param>
          </params>
        </procedure>
        <procedure id="7" name="CalcProperties"
implementation="CSDLOperationProcedureCallCalcProperties">
          <params>
            <param name="Strength_1" type="double"
defvalue="" ref="yes"></param>
            <param name="Thickness_1" type="double"
defvalue="" ref="yes"></param>
            <param name="Speed_1" type="double" defvalue=""
ref="yes"></param>
            <param name="Time_1" type="double" defvalue=""
ref="yes"></param>
            <param name="Strength_2" type="double"
defvalue="" ref="yes"></param>
            <param name="Thickness_2" type="double"
defvalue="" ref="yes"></param>
            <param name="Speed_2" type="double" defvalue=""
ref="yes"></param>
            <param name="Time_2" type="double" defvalue=""
ref="yes"></param>
          </params>
        </procedure>
        <procedure id="8" name="CalculaDestiFrac"
implementation="CSDLOperationProcedureCallCalculaDestiFrac"
>
          <params>
            <param name="nextCell" type="int" defvalue=""
ref="yes"></param>
            <param name="alcMin" type="double" defvalue=""
ref="yes"></param>
          </params>
        </procedure>
```

```
        <procedure id="4" name="Vicinity"
implementation="">
          <params>
            <param name="ignMap" type="int" defvalue=""
ref="yes"></param>
            <param name="m1Map" type="double" defvalue=""
ref="yes"></param>
            <param name="m10Map" type="double" defvalue=""
ref="yes"></param>
            <param name="m100Map" type="double" defvalue=""
ref="yes"></param>
            <param name="mherbMap" type="double"
defvalue="" ref="yes"></param>
            <param name="mwoodMap" type="double"
defvalue="" ref="yes"></param>
            <param name="slpMap" type="double" defvalue=""
ref="yes"></param>
            <param name="aspMap" type="double" defvalue=""
ref="yes"></param>
            <param name="wspdMap" type="double" defvalue=""
ref="yes"></param>
            <param name="wdirMap" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM0" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM1" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM2" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM3" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM4" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM5" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM6" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM7" type="double" defvalue=""
ref="yes"></param>
          </params>
          <body>
            <procedurecall id="1"
name="mncaGetCurrentCell">
              <param name="MNCA_CURR_CELL"
value="tmpCell"></param>
            </procedurecall>
            <procedurecall id="2" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
```

```
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignMap"></param>
            </procedurecall>
            <procedurecall id="3"
name="mncaGetCellValue"><!-- NO FUNCIONA CON TODAS LAS
CAPAS !!!-->
              <param name="MNCA_SET_LAYER"
value="'M1'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="m1Map"></param>
            </procedurecall>
            <procedurecall id="4" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'M10'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="m10Map"></param>
            </procedurecall>
            <procedurecall id="5" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'M100'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="m100Map"></param>
            </procedurecall>
            <procedurecall id="6" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MHerba'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="mherbMap"></param>
            </procedurecall>
            <procedurecall id="7" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MFusta'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="mwoodMap"></param>
            </procedurecall>
            <procedurecall id="8" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MPendent'"></param>
```

```
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="slpMap"></param>
            </procedurecall>
            <procedurecall id="9" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MOrientacio'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="aspMap"></param>
            </procedurecall>
            <procedurecall id="10" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MVelVents'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="wspdMap"></param>
            </procedurecall>
            <procedurecall id="11" name="mncaGetCellValue">
              <param name="MNCA_SET_LAYER"
value="'MDirVents'"></param>
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="wdirMap"></param>
            </procedurecall>
            <task id="12" name="">currCell=tmoCell;nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) +
0;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="13" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM0"></param>
            </procedurecall>
            <task id="14" name="">nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="15" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM1"></param>
```

```
            </procedurecall>
            <task id="16" name="">nrow =
getPosicion(tmpCell,0) + 0;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="17" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM2"></param>
            </procedurecall>
            <task id="18" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="19" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM3"></param>
            </procedurecall>
            <task id="20" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) +
0;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="21" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM4"></param>
            </procedurecall>
            <task id="22" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) 1
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="23" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM5"></param>
            </procedurecall>
            <task id="24" name="">nrow =
getPosicion(tmpCell,0) + 0;ncol = getPosicion(tmpCell,1) -
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="25" name="mncaGetCellValue">
```

```
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM6"></param>
            </procedurecall>
            <task id="26" name="">nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) -
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="27" name="mncaGetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="ignM7"></param>
            </procedurecall>
          </body>
        </procedure>
        <procedure id="5" name="Nucleus" implementation="">
          <params>
            <param name="ignMap" type="int" defvalue=""
ref="yes"></param>
            <param name="ignM0" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM1" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM2" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM3" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM4" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM5" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM6" type="double" defvalue=""
ref="yes"></param>
            <param name="ignM7" type="double" defvalue=""
ref="yes"></param>
          </params>
          <body>
            <procedurecall id="1"
name="mncaGetCurrentCell">
              <param name="MNCA_CURR_CELL"
value="tmpCell"></param>
            </procedurecall>
            <procedurecall id="2" name="mncaSetCellValue">
              <param name="MNCA_SET_LAYER"
value="'BlockCelda'"></param>
```

```
              <param name="MNCA_SET_CELL"
value="tmpCell"></param>
              <param name="MNCA_SET_VALUE"
value="ignMap"></param>
            </procedurecall>
            <task id="3" name="">nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) +
0;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="4" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell0"></param>
            </procedurecall>
            <task id="5" name="">nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="6" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell1"></param>
            </procedurecall>
            <task id="7" name="">nrow =
getPosicion(tmpCell,0) + 0;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="8" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell2"></param>
            </procedurecall>
            <task id="9" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) +
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="10" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell3"></param>
            </procedurecall>
```

```
            <task id="11" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) +
0;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="12" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell4"></param>
            </procedurecall>
            <task id="13" name="">nrow =
getPosicion(tmpCell,0) - 1;ncol = getPosicion(tmpCell,1) 1
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="14" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell5"></param>
            </procedurecall>
            <task id="15" name="">nrow =
getPosicion(tmpCell,0) + 0;ncol = getPosicion(tmpCell,1) -
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="16" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell6"></param>
            </procedurecall>
            <task id="17" name="">nrow =
getPosicion(tmpCell,0) + 1;ncol = getPosicion(tmpCell,1) -
1;tmpCell = ncol + nrow*Cols;</task>
            <procedurecall id="18" name="mncaSetCellValue">
              <param name="MNCA_GET_LAYER"
value="'BlockCelda'"></param>
              <param name="MNCA_GET_CELL"
value="tmpCell"></param>
              <param name="MNCA_GET_VALUE"
value="propCell7"></param>
            </procedurecall>
          </body>
        </procedure>
      </procedures>
              <start>
        <procedurecall id="1" name="ArrivalTime">
```

```
          <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
        </procedurecall>
        <task id="2"
name="">currCell=getCeldaInicio(Rows,Cols);InicializarFuego
(Rows,Cols,ignMap);</task>
        <output id="3" name="Arder" self="yes" to=""
via="">
            <param name="creationtime" value="0"></param>
            <param name="executiontime"
value="ProcessCelda_t"></param>
            <param name="delay"
value="ProcessCelda_t"></param>
            <param name="priority" value="0"></param>
            <param name="type" value="Arder"></param>
            <param name="mnca_cell[]"
value="{currCell}"></param>
            <userparam name="currCell"
value="currCell"></userparam>
        </output>
        <setstate id="4" name="NoQuemado"></setstate>
              </start>
              <state name="NoQuemado">
                   <input id="1" name="Arder"></input>
        <procedurecall id="2" name="Vicinity">
          <param name="ignMap" value="ignMap"></param>
          <param name="m1Map" value="m1Map"></param>
          <param name="m10Map" value="m10Map"></param>
          <param name="m100Map" value="m100Map"></param>
          <param name="mherbMap" value="mherbMap"></param>
          <param name="mwoodMap" value="mwoodMap"></param>
          <param name="slpMap" value="slpMap"></param>
          <param name="aspMap" value="aspMap"></param>
          <param name="wspdMap" value="wspdMap"></param>
          <param name="wdirMap" value="wdirMap"></param>
          <param name="ignM0" value="ignM0"></param>
          <param name="ignM1" value="ignM1"></param>
          <param name="ignM2" value="ignM2"></param>
          <param name="ignM3" value="ignM3"></param>
          <param name="ignM4" value="ignM4"></param>
          <param name="ignM5" value="ignM5"></param>
          <param name="ignM6" value="ignM6"></param>
          <param name="ignM7" value="ignM7"></param>
        </procedurecall>
        <task id="3"
name="">PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Ma
p,m10Map,m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wd
irMap,ignMap,flMap,currCell,ignM0,ignM1,ignM2,ignM3,ignM4,i
gnM5,ignM6,ignM7);propCell1=ObtenerCeldasIncendiadas(0);pro
pCell2=ObtenerCeldasIncendiadas(1);propCell3=ObtenerCeldasI
```

```
ncendiadas(2);propCell4=ObtenerCeldasIncendiadas(3);propCel
l5=ObtenerCeldasIncendiadas(4);propCell6=ObtenerCeldasIncen
diadas(5);propCell7=ObtenerCeldasIncendiadas(6);propCell8=O
btenerCeldasIncendiadas(7);</task>
        <procedurecall id="4" name="Nucleus"><!-- NO SALVAR
CELDAS -1 -->
          <param name="CellValue" value="ignMap"></param>
          <param name="ignM0" value="ignM0"></param>
          <param name="ignM1" value="ignM1"></param>
          <param name="ignM2" value="ignM2"></param>
          <param name="ignM3" value="ignM3"></param>
          <param name="ignM4" value="ignM4"></param>
          <param name="ignM5" value="ignM5"></param>
          <param name="ignM6" value="ignM6"></param>
          <param name="ignM7" value="ignM7"></param>
        </procedurecall>
                    <procedurecall id="5"
name="ArrivalTime">
                        <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                    </procedurecall>
                    <task id="6" name=""></task>
                    <output id="7" name="Propagar"
self="yes" to="" via=""><!-- NO ENVIAR A LAS CELDAS -1 -->
          <param name="delay"
value="ProcessCelda_t"></param>
                        <param name="priority"
value="0"></param>
          <param name="mnca_cell[]"
value="{propCell1,propCell2,propCell3,propCell4,propCell5,p
ropCell6,propCell7,propCell8}"></param>
          <userparam name="delayCell"
value="ProcessCelda_t"></userparam>
          <userparam name="propCell1"
value="propCell1"></userparam>
                    </output>
                    <setstate id="8"
name="Ardiendo"></setstate>
        <input id="9" name="Propagar"></input>
        <procedurecall id="10" name="ArrivalTime">
          <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
        </procedurecall>
        <output id="11" name="Propagar" self="yes" to=""
via="">
          <param name="delay"
value="ProcessCelda_t"></param>
          <param name="priority" value="0"></param>
        </output>
        <setstate id="12" name="Ardiendo"></setstate>
```

```
                </state>
                <state name="Quemado">
                     <input id="1" name="Arder"></input>
                     <procedurecall id="2"
name="ArrivalTime">
                          <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                     </procedurecall>
                     <output id="2" name="" self="yes" to=""
via="">
                          <param name="delay"
value="ProcessCelda_t"></param>
                          <param name="priority"
value="0"></param>
                     </output>
                     <setstate id="3"
name="Quemado"></setstate>
                </state>
                <state name="Ardiendo">
                     <input id="1" name="Arder"></input>
                     <procedurecall id="2"
name="ArrivalTime">
                          <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                     </procedurecall>
                     <output id="3" name="" self="yes" to=""
via="">
                          <param name="delay"
value="ProcessCelda_t"></param>
                          <param name="priority"
value="0"></param>
                     </output>
                     <setstate id="4"
name="Ardiendo"></setstate>
                     <input id="5" name="Extinguir"></input>
                     <procedurecall id="6"
name="ArrivalTime">
                          <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                     </procedurecall>
                     <output id="6" name="" self="yes" to=""
via="">
                          <param name="delay"
value="ProcessCelda_t"></param>
                          <param name="priority"
value="0"></param>
                     </output>
                     <setstate id="7"
name="Quemado"></setstate>
                     <input id="8" name="Propagar"></input>
```

```
                    <procedurecall id="9"
name="ArrivalTime">
                        <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                    </procedurecall>
        <procedurecall id="10" name="Vicinity">
          <param name="ignMap" value="ignMap"></param>
          <param name="m1Map" value="m1Map"></param>
          <param name="m10Map" value="m10Map"></param>
          <param name="m100Map" value="m100Map"></param>
          <param name="mherbMap" value="mherbMap"></param>
          <param name="mwoodMap" value="mwoodMap"></param>
          <param name="slpMap" value="slpMap"></param>
          <param name="aspMap" value="aspMap"></param>
          <param name="wspdMap" value="wspdMap"></param>
          <param name="wdirMap" value="wdirMap"></param>
          <param name="ignM0" value="ignM0"></param>
          <param name="ignM1" value="ignM1"></param>
          <param name="ignM2" value="ignM2"></param>
          <param name="ignM3" value="ignM3"></param>
          <param name="ignM4" value="ignM4"></param>
          <param name="ignM5" value="ignM5"></param>
          <param name="ignM6" value="ignM6"></param>
          <param name="ignM7" value="ignM7"></param>
        </procedurecall>
        <task id="11"
name="">PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Ma
p,m10Map,m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wd
irMap,ignMap,flMap,currCell,ignM0,ignM1,ignM2,ignM3,ignM4,i
gnM5,ignM6,ignM7);propCell1=ObtenerCeldasIncendiadas(0);pro
pCell2=ObtenerCeldasIncendiadas(1);propCell3=ObtenerCeldasI
ncendiadas(2);propCell4=ObtenerCeldasIncendiadas(3);propCel
l5=ObtenerCeldasIncendiadas(4);propCell6=ObtenerCeldasIncen
diadas(5);propCell7=ObtenerCeldasIncendiadas(6);propCell8=O
btenerCeldasIncendiadas(7);</task>
        <procedurecall id="12" name="Nucleus">
          <!-- NO SALVAR CELDAS -1 -->
          <param name="CellValue" value="ignMap"></param>
          <param name="ignM0" value="ignM0"></param>
          <param name="ignM1" value="ignM1"></param>
          <param name="ignM2" value="ignM2"></param>
          <param name="ignM3" value="ignM3"></param>
          <param name="ignM4" value="ignM4"></param>
          <param name="ignM5" value="ignM5"></param>
          <param name="ignM6" value="ignM6"></param>
          <param name="ignM7" value="ignM7"></param>
        </procedurecall>
        <decision id="13" name="PROPAGA" iftrue="19"
iffalse="14">PROPAGA==1</decision>
```

```
                    <decision id="14" name="EXTINGUIDO"
iftrue="23" iffalse="15">EXTINGUIDO==1</decision>
        <output id="15" name="Propagar" self="yes" to=""
via="">
          <param name="delay"
value="ProcessCelda_t"></param>
          <param name="priority" value="0"></param>
          <param name="mnca_cell[]"
value="{propCell1,propCell2,propCell3,propCell4,propCell5,p
ropCell6,propCell7,propCell8}"></param>
          <userparam name="delayCell"
value="ProcessCelda_t"></userparam>
          <userparam name="propCell1"
value="propCell1"></userparam>
                    </output>
                    <setstate id="16"
name="Ardiendo"></setstate>
                    <output id="17" name="Extinguir"
self="yes" to="" via="">
                         <param name="delay"
value="ProcessCelda_t"></param>
                         <param name="priority"
value="0"></param>
                    </output>
                    <setstate id="18"
name="Quemado"></setstate>
                    <output id="19" name="Arder" self="yes"
to="" via="">
                         <param name="delay"
value="ProcessCelda_t"></param>
                         <param name="priority"
value="0"></param>
                    </output>
                    <decision id="20" iftrue="23"
name="EXTINGUIDO" iffalse="21">EXTINGUIDO==1</decision>
                    <output id="21" name="Propagar"
self="yes" to="" via="">
                         <param name="delay"
value="ProcessCelda_t"></param>
                         <param name="priority"
value="0"></param>
          <param name="mnca_cell[]"
value="{propCell1,propCell2,propCell3,propCell4,propCell5,p
ropCell6,propCell7,propCell8}"></param>
          <userparam name="delayCell"
value="ProcessCelda_t"></userparam>
          <userparam name="propCell1"
value="propCell1"></userparam>
        </output>
```

```
                    <setstate id="22"
name="Ardiendo"></setstate>
                    <output id="23" name="Extinguir"
self="yes" to="" via="">
                        <param name="delay"
value="ProcessCelda_t"></param>
                        <param name="priority"
value="0"></param>
                    </output>
                    <setstate id="24"
name="Quemado"></setstate>
                    <input id="25"
name="ActualizarDatos"></input>
                    <procedurecall id="26"
name="ArrivalTime">
                        <param name="ArrivalTime_t"
value="ProcessCelda_t"></param>
                    </procedurecall>
        <procedurecall id="27" name="Vicinity">
          <param name="ignMap" value="ignMap"></param>
          <param name="m1Map" value="m1Map"></param>
          <param name="m10Map" value="m10Map"></param>
          <param name="m100Map" value="m100Map"></param>
          <param name="mherbMap" value="mherbMap"></param>
          <param name="mwoodMap" value="mwoodMap"></param>
          <param name="slpMap" value="slpMap"></param>
          <param name="aspMap" value="aspMap"></param>
          <param name="wspdMap" value="wspdMap"></param>
          <param name="wdirMap" value="wdirMap"></param>
          <param name="ignM0" value="ignM0"></param>
          <param name="ignM1" value="ignM1"></param>
          <param name="ignM2" value="ignM2"></param>
          <param name="ignM3" value="ignM3"></param>
          <param name="ignM4" value="ignM4"></param>
          <param name="ignM5" value="ignM5"></param>
          <param name="ignM6" value="ignM6"></param>
          <param name="ignM7" value="ignM7"></param>
        </procedurecall>
        <task id="28"
name="">PropagarFuego(ProcessCelda_t,Rows,Cols,fuelMap,m1Ma
p,m10Map,m100Map,mherbMap,mwoodMap,slpMap,aspMap,wspdMap,wd
irMap,ignMap,flMap,currCell,ignM0,ignM1,ignM2,ignM3,ignM4,i
gnM5,ignM6,ignM7);propCell1=ObtenerCeldasIncendiadas(0);pro
pCell2=ObtenerCeldasIncendiadas(1);propCell3=ObtenerCeldasI
ncendiadas(2);propCell4=ObtenerCeldasIncendiadas(3);propCel
l5=ObtenerCeldasIncendiadas(4);propCell6=ObtenerCeldasIncen
diadas(5);propCell7=ObtenerCeldasIncendiadas(6);propCell8=O
btenerCeldasIncendiadas(7);</task>
        <procedurecall id="29" name="Nucleus">
          <!-- NO SALVAR CELDAS -1 -->
```

```
            <param name="CellValue" value="ignMap"></param>
            <param name="ignM0" value="ignM0"></param>
            <param name="ignM1" value="ignM1"></param>
            <param name="ignM2" value="ignM2"></param>
            <param name="ignM3" value="ignM3"></param>
            <param name="ignM4" value="ignM4"></param>
            <param name="ignM5" value="ignM5"></param>
            <param name="ignM6" value="ignM6"></param>
            <param name="ignM7" value="ignM7"></param>
          </procedurecall>
          <output id="30" name="Propagar" self="yes" to=""
via="">
            <param name="delay"
value="ProcessCelda_t"></param>
            <param name="priority" value="0"></param>
            <param name="mnca_cell[]"
value="{propCell1,propCell2,propCell3,propCell4,propCell5,p
ropCell6,propCell7,propCell8}"></param>
            <userparam name="delayCell"
value="ProcessCelda_t"></userparam>
            <userparam name="propCell1"
value="propCell1"></userparam>
                    </output>
                    <setstate id="31"
name="Ardiendo"></setstate>
                </state>
            </process>
      </block>
      </mnca>
</system>
```

9.4 Fichero XSD

El fichero xsd que define la estructura del fichero xml generado para el simulador con el software SDLPS es el siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<!--Created with Liquid XML Studio - 30 Day Trial Edition
7.0.5.906 (http://www.liquid-technologies.com)-->
<xs:schema xmlns:xsd="undefined"
attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="system">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="blockType">
          <xs:attribute name="version" type="xs:decimal"
use="required" />
```

```
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="processType">
    <xs:sequence>
      <xs:element minOccurs="0" name="DCLS">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="DCL">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="type" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="procedures">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="procedure">
              <xs:complexType>
                <xs:complexContent mixed="false">
                  <xs:extension base="procedureType" />
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="start"
type="startType" />
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="state">
        <xs:complexType>
          <xs:complexContent mixed="false">
            <xs:extension base="stateType">
              <xs:attribute name="name" type="xs:string"
use="optional" />
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
```

```
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"
use="required" />
    <xs:attribute name="name" type="xs:string"
use="required" />
    <xs:attribute name="implementation" type="xs:string"
use="required" />
    <xs:attribute name="IP" type="xs:string" use="required"
/>
    <xs:attribute name="portRead" type="xs:unsignedShort"
use="required" />
  </xs:complexType>
  <xs:complexType name="blockType">
    <xs:sequence>
      <xs:element minOccurs="0" name="representations">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0"
name="representation">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="mesh"
type="xs:string" />
                  <xs:element minOccurs="0" name="position"
/>
                </xs:sequence>
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="channels">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="channel">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0"
maxOccurs="unbounded" name="event">
                    <xs:complexType>
                      <xs:attribute name="name"
type="xs:string" use="optional" />
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="name" type="xs:string"
use="optional" />
```

```
                <xs:attribute name="start" type="xs:string"
use="optional" />
                <xs:attribute name="end" type="xs:string"
use="optional" />
                <xs:attribute name="dual" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded"
name="mnca" type="mncaType" />
          <xs:element minOccurs="0" maxOccurs="unbounded"
name="block" type="blockType" />
        </xs:sequence>
        <xs:element maxOccurs="unbounded" name="process"
type="processType" />
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"
use="required" />
    <xs:attribute name="name" type="xs:string"
use="required" />
    <xs:attribute name="implementation" type="xs:string"
use="required" />
    <xs:attribute name="IP" type="xs:string" use="required"
/>
    <xs:attribute name="portRead" type="xs:unsignedShort"
use="required" />
  </xs:complexType>
  <xs:complexType name="mncaType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="dim">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="size" type="xs:unsignedByte" />
          </xs:sequence>
          <xs:attribute name="size" type="xs:unsignedByte"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="channels">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="channel">
              <xs:complexType>
```

```
                <xs:sequence>
                  <xs:element minOccurs="0"
maxOccurs="unbounded" name="event">
                    <xs:complexType>
                      <xs:attribute name="name"
type="xs:string" use="optional" />
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="start" type="xs:string"
use="optional" />
                <xs:attribute name="end" type="xs:string"
use="optional" />
                <xs:attribute name="dual" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="block"
type="blockType" />
      <xs:element minOccurs="0" name="representations">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0"
name="representation">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="mesh"
type="xs:string" />
                  <xs:element minOccurs="0" name="position"
/>
                </xs:sequence>
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"
use="required" />
    <xs:attribute name="name" type="xs:string"
use="required" />
    <xs:attribute name="implementation" type="xs:string"
use="required" />
```

```
    <xs:attribute name="IP" type="xs:string" use="required"
/>
    <xs:attribute name="portRead" type="xs:string"
use="required" />
  </xs:complexType>
  <xs:complexType name="procedureType">
    <xs:sequence>
      <xs:element minOccurs="0" name="params">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="type" type="xs:string"
use="optional" />
                <xs:attribute name="defvalue"
type="xs:string" use="optional" />
                <xs:attribute name="ref" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" name="body" type="bodyType"
/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
    <xs:attribute name="name" type="xs:string"
use="optional" />
    <xs:attribute name="implementation" type="xs:string"
use="optional" />
  </xs:complexType>
  <xs:complexType name="stateType">
    <xs:choice maxOccurs="unbounded">
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="decision">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
              <xs:attribute name="iftrue"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="iffalse"
type="xs:unsignedByte" use="optional" />
```

```
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="create">
        <xs:complexType>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="target" type="xs:string"
use="optional" />
          <xs:attribute name="qtt" type="xs:unsignedByte"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="output">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="userparam">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="self" type="xs:string"
use="optional" />
          <xs:attribute name="to" type="xs:string"
use="optional" />
          <xs:attribute name="via" type="xs:string"
use="optional" />
```

```
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="procedurecall">
        <xs:complexType mixed="true">
          <xs:sequence minOccurs="0">
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="task">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="setstate">
        <xs:complexType>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="input">
        <xs:complexType>
          <xs:sequence minOccurs="0">
            <xs:element minOccurs="0" name="param">
              <xs:complexType>
```

```
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="bodyType">
    <xs:choice maxOccurs="unbounded">
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="decision">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
              <xs:attribute name="iftrue"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="iffalse"
type="xs:unsignedByte" use="optional" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="create">
        <xs:complexType>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="target" type="xs:string"
use="optional" />
          <xs:attribute name="qtt" type="xs:unsignedByte"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="output">
        <xs:complexType>
```

```
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="userparam">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="self" type="xs:string"
use="optional" />
          <xs:attribute name="to" type="xs:string"
use="optional" />
          <xs:attribute name="via" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="procedurecall">
        <xs:complexType mixed="true">
          <xs:sequence minOccurs="0">
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
```

```
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="task">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="startType">
    <xs:choice maxOccurs="unbounded">
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="decision">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
              <xs:attribute name="iftrue"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="iffalse"
type="xs:unsignedByte" use="optional" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="create">
        <xs:complexType>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="target" type="xs:string"
use="optional" />
          <xs:attribute name="qtt" type="xs:unsignedByte"
use="optional" />
        </xs:complexType>
      </xs:element>
```

```
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="output">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="userparam">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
          <xs:attribute name="self" type="xs:string"
use="optional" />
          <xs:attribute name="to" type="xs:string"
use="optional" />
          <xs:attribute name="via" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="procedurecall">
        <xs:complexType mixed="true">
          <xs:sequence minOccurs="0">
            <xs:element minOccurs="0" maxOccurs="unbounded"
name="param">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string"
use="optional" />
                <xs:attribute name="value" type="xs:string"
use="optional" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
```

```
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="task">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="id"
type="xs:unsignedByte" use="optional" />
              <xs:attribute name="name" type="xs:string"
use="optional" />
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="setstate">
        <xs:complexType>
          <xs:attribute name="id" type="xs:unsignedByte"
use="optional" />
          <xs:attribute name="name" type="xs:string"
use="optional" />
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

9.5 Resuldados simulación Behave

El resuldado del modelo Behave ha sido generado con un ejemplo que trae la labreria firelib y es el que viene a continuación.

north: 10100

south: 0

east: 10100

west: 0

rows: 101

cols: 101

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 206 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 203 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 201 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
206 198 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
204 195 187 195 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
201 193 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
206 198 190 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
204 196 188 179 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 190 182 174 182 190 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 174 166 174 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 164 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 185 177 169 161 169 177 185 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 158 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 196 188 180 172 164 156 164 172 180 188 196 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 169 161 153 161 169 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 159 151 159 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 180 172 164 156 148 156 164 172 180 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 153 145 153 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 191 183 175 167 159 151 143 151 159 167 175 183 191 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 164 156 148 140 148 156 164 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 154 146 137 146 154 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 175 167 159 151 143 135 143 151 159 167 175 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 165 157 148 140 132 140 148 157 165 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 186 178 170 162 154 146 138 130 138 146 154 162 170 178 186 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 176 168 159 151 143 135 127 135 143 151 159 168 176 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 165 157 149 141 132 124 132 141 149 157 165 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 170 162 154 146 138 130 122 130 138 146 154 162 170 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 176 168 160 152 143 135 127 119 127 135 143 152 160 168 176 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 189 181 173 165 157 149 141 133 125 116 125 133 141 149 157 165 173 181 189 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 171 163 154 146 138 130 122 114 122 130 138 146 154 163 171 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 200 192 184 176 168 160 152 144 136 127 119 111 119 127 136 144 152 160 168 176 184 192 200 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 165 157 149 141 133 125 117 109 117 125 133 141 149 157 165 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 171 163 155 147 138 130 122 114 106 114 122 130 138 147 155 163 171 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 184 176 168 160 152 144 136 128 120 111 103 111 120 128 136 144 152 160 168 176 184 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 166 158 149 141 133 125 117 109 101 109 117 125 133 141 149 158 166 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 195 187 179 171 163 155 147 139 131 122 114 106 98 106 114 122 131 139 147 155 163 171 179 187 195 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 160 152 144 136 128 120 112 104 95 104 112 120 128 136 144 152 160 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 166 158 150 142 133 125 117 109 101 93 101 109 117 125 133 142 150 158 166 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 179 171 163 155 147 139 131 123 115 106 98 90 98 106 115 123 131 139 147 155 163 171 179 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 161 153 144 136 128 120 112 104 96 88 96 104 112 120 128 136 144 153 161 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 190 182 174 166 158 150 142 134 126 117 109 101 93 85 93 101 109 117 126 134 142 150 158 166 174 182 190 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 163 155 147 139 131 123 115 107 99 90 82 90 99 107 115 123 131 139 147 155 163 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 161 153 145 137 128 120 112 104 96 88 80 88 96 104 112 120 128 137 145 153 161 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 174 166 158 150 142 134 126 118 110 101 93 85 77 85 93 101 110 118 126 134 142 150 158 166 174 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 164 156 148 139 131 123 115 107 99 91 83 74 83 91 99 107 115 123 131 139 148 156 164 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 185 177 169 161 153 145 137 129 121 112 104 96 88 80 72 80 88 96 104 112 121 129 137 145 153 161 169 177 185 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 158 150 142 134 126 118 110 102 94 85 77 69 77 85 94 102 110 118 126 134 142 150 158 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 196 188 180 172 164 156 148 140 132 123 115 107 99 91 83 75 67 75 83 91 99 107 115 123 132 140 148 156 164 172 180 188 196 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 169 161 153 145 137 129 121 113 105 96 88 80 72 64 72 80 88 96 105 113 121 129 137 145 153 161 169 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 159 151 143 134 126 118 110 102 94 86 78 69 61 69 78 86 94 102 110 118 126 134 143 151 159 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 180 172 164 156 148 140 132 124 116 107 99 91 83 75 67 59 67 75 83 91 99 107 116 124 132 140 148 156 164 172 180 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 153 145 137 129 121 113 105 97 89 80 72 64 56 64 72 80 89 97 105 113 121 129 137 145 153 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 191 183 175 167 159 151 143 135 127 118 110 102 94 86 78 70 62 53 62 70 78 86 94 102 110 118 127 135 143 151 159 167 175 183 191 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 164 156 148 140 132 124 116 108 100 91 83 75 67 59 51 59 67 75 83 91 100 108 116 124 132 140 148 156 164 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 214 194 186 178 170 162 154 146 137 129 121 113 105 97 89 81 73 64 56 48 56 64 73 81 89 97 105 113 121 129 137 146 154 162 170 178 186 194 214 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 239 203 184 175 167 159 151 143 135 127 119 111 102 94 86 78 70 62 54 46 54 62 70 78 86 94 102 111 119 127 135 143 151 159 167 175 184 203 239 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 212 192 173 165 157 148 140 132 124 116 108 100 92 84 75 67 59 51 43 51 59 67 75 84 92 100 108 116 124 132 140 148 157 165 173 192 212 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 237 201 182 162 154 146 138 130 122 113 105 97 89 81 73 65 57 48 40 48 57 65 73 81 89 97 105 113 122 130 138 146 154 162 182 201 237 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 210 191 171 151 143 135 127 119 111 103 95 86 78 70 62 54 46 38 46 54 62 70 78 86 95 103 111 119 127 135 143 151 171 191 210 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 235 199 180 160 141 132 124 116 108 100 92 84 76 68 59 51 43 35 43 51 59 68 76 84 92 100 108 116 124 132 141 160 180 199 235 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 189 169 149 130 122 114 106 97 89 81 73 65 57 49 41 32 41 49 57 65 73 81 89 97 106 114 122 130 149 169 189 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 217 197 178 158 139 119 111 103 95 87 79 70 62 54 46 38 30 38 46 54 62 70 79 87 95 103 111 119 139 158 178 197 217 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 242 206 187 167 148 128 108 100 92 84 76 68 60 52 43 35 27 35 43 52 60 68 76 84 92 100 108 128 148 167 187 206 242 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 215 196 176 156 137 117 98 90 81 73 65 57 49 41 33 25 33 41 49 57 65 73 81 90 98 117 137 156 176 196 215 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 240 204 185 165 146 126 106 87 79 71 63 54 46 38 30 22 30 38 46 54 63 71 79 87 106 126 146 165 185 204 240 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 213 194 174 155 135 115 96 76 68 60 52 44 36 27 19 27 36 44 52 60 68 76 96 115 135 155 174 194 213 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 238 203 183 163 144 124 105 85 65 57 49 41 33 25 17 25 33 41 49 57 65 85 105 124 144 163 183 203 238 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 211 192 172 153 133 113 94 74 55 47 38 30 22 14 22 30 38 47 55 74 94 113 133 153 172 192 211 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 236 201 181 161 142 122 103 83 64 44 36 28 20 11 20 28 36 44 64 83 103 122 142 161 181 201 236 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 210 190 170 151 131 112 92 72 53 33 25 17 9 17 25 33 53 72 92 112 131 151 170 190 210 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 235 199 179 160 140 120 101 81 62 42 22 14 6 14 22 42 62 81 101 120 140 160 179 199 235 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 188 168 149 129 110 90 71 51 31 12 4 12 31 51 71 90 110 129 149 168 188 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 216 197 177 158 138 119 99 79 60 40 21 1 21 40 60 79 99 119 138 158 177 197 216 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 242 222 202 183 163 144 124 104 85 65 46 38 46 65 85 104 124 144 163 183 202 222 242 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 227 208 188 169 149 130 110 90 82 74 82 90 110 130 149 169 188 208 227 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 233 213 194 174 155 135 127 119 111 119 127 135 155 174 194 213 233 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 238 219 199 180 172 163 155 147 155 163 172 180 199 219 238 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 224 216 208 200 192 184 192 200 208 216 224 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 237 228 220 228 237 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0

0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

9.6 Resultados simulación PFC no distribuido

El resultado generado por nuestro simulador ejecutado con el SDLPS es el siguiente:

rows: 101

cols: 101

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 206 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 203 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 201 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
206 198 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
204 195 187 195 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
201 193 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
206 198 190 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
204 196 188 179 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
201 193 185 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
207 199 190 182 174 182 190 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
204 196 188 180 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
201 193 185 177 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207
199 191 183 174 166 174 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204
196 188 180 172 164 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202
194 185 177 169 161 169 177 185 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207
199 191 183 175 167 158 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205
196 188 180 172 164 156 164 172 180 188 196 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 169 161 153 161 169 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 159 151 159 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 180 172 164 156 148 156 164 172 180 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 153 145 153 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 191 183 175 167 159 151 143 151 159 167 175 183 191 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 164 156 148 140 148 156 164 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 154 146 137 146 154 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 175 167 159 151 143 135 143 151 159 167 175 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 165 157 148 140 132 140 148 157 165 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 186 178 170 162 154 146 138 130 138 146 154 162 170 178 186 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 176 168 159 151 143 135 127 135 143 151 159 168 176 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 165 157 149 141 132 124 132 141 149 157 165 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 170 162 154 146 138 130 122 130 138 146 154 162 170 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 192 184 176 168 160 152 143 135 127 119 127 135 143 152 160 168 176 184 192 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 189 181 173 165 157 149 141 133 125 116 125 133 141 149 157 165 173 181 189 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 171 163 154 146 138 130 122 114 122 130 138 146 154 163 171 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 200 192 184 176 168 160 152 144 136 127 119 111 119 127 136 144 152 160 168 176 184 192 200 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 165 157 149 141 133 125 117 109 117 125 133 141 149 157 165 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 203 195 187 179 171 163 155 147 138 130 122 114 106 114 122 130 138 147 155 163 171 179 187 195 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 184 176 168 160 152 144 136 128 120 111 103 111 120 128 136 144 152 160 168 176 184 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 166 158 149 141 133 125 117 109 101 109 117 125 133 141 149 158 166 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 195 187 179 171 163 155 147 139 131 122 114 106 98 106 114 122 131 139 147 155 163 171 179 187 195 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 160 152 144 136 128 120 112 104 95 104 112 120 128 136 144 152 160 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 198 190 182 174 166 158 150 142 133 125 117 109 101 93 101 109 117 125 133 142 150 158 166 174 182 190 198 206 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 179 171 163 155 147 139 131 123 115 106 98 90 98 106 115 123 131 139 147 155 163 171 179 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 161 153 144 136 128 120 112 104 96 88 96 104 112 120 128 136 144 153 161 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 190 182 174 166 158 150 142 134 126 117 109 101 93 85 93 101 109 117 126 134 142 150 158 166 174 182 190 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 163 155 147 139 131 123 115 107 99 90 82 90 99 107 115 123 131 139 147 155 163 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 201 193 185 177 169 161 153 145 137 128 120 112 104 96 88 80 88 96 104 112 120 128 137 145 153 161 169 177 185 193 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 174 166 158 150 142 134 126 118 110 101 93 85 77 85 93 101 110 118 126 134 142 150 158 166 174 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 204 196 188 180 172 164 156 148 139 131 123 115 107 99 91 83 74 83 91 99 107 115 123 131 139 148 156 164 172 180 188 196 204 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 185 177 169 161 153 145 137 129 121 112 104 96 88 80 72 80 88 96 104 112 121 129 137 145 153 161 169 177 185 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 158 150 142 134 126 118 110 102 94 85 77 69 77 85 94 102 110 118 126 134 142 150 158 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 196 188 180 172 164 156 148 140 132 123 115 107 99 91 83 75 67 75 83 91 99 107 115 123 132 140 148 156 164 172 180 188 196 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 169 161 153 145 137 129 121 113 105 96 88 80 72 64 72 80 88 96 105 113 121 129 137 145 153 161 169 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 207 199 191 183 175 167 159 151 143 134 126 118 110 102 94 86 78 69 61 69 78 86 94 102 110 118 126 134 143 151 159 167 175 183 191 199 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 180 172 164 156 148 140 132 124 116 107 99 91 83 75 67 59 67 75 83 91 99 107 116 124 132 140 148 156 164 172 180 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 202 194 186 178 170 162 153 145 137 129 121 113 105 97 89 80 72 64 56 64 72 80 89 97 105 113 121 129 137 145 153 162 170 178 186 194 202 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 200 191 183 175 167 159 151 143 135 127 118 110 102 94 86 78 70 62 53 62 70 78 86 94 102 110 118 127 135 143 151 159 167 175 183 191 200 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 205 197 189 181 173 164 156 148 140 132 124 116 108 100 91 83 75 67 59 51 59 67 75 83 91 100 108 116 124 132 140 148 156 164 173 181 189 197 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 214 194 186 178 170 162 154 146 137 129 121 113 105 97 89 81 73 64 56 48 56 64 73 81 89 97 105 113 121 129 137 146 154 162 170 178 186 194 214 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 239 203 184 175 167 159 151 143 135 127 119 111 102 94 86 78 70 62 54 46 54 62 70 78 86 94 102 111 119 127 135 143 151 159 167 175 184 203 239 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 212 192 173 165 157 148 140 132 124 116 108 100 92 84 75 67 59 51 43 51 59 67 75 84 92 100 108 116 124 132 140 148 157 165 173 192 212 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 237 201 182 162 154 146 138 130 122 113 105 97 89 81 73 65 57 48 40 48 57 65 73 81 89 97 105 113 122 130 138 146 154 162 182 201 237 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 210 191 171 151 143 135 127 119 111 103 95 86 78 70 62 54 46 38 46 54 62 70 78 86 95 103 111 119 127 135 143 151 171 191 210 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 235 199 180 160 141 132 124 116 108 100 92 84 76 68 59 51 43 35 43 51 59 68 76 84 92 100 108 116 124 132 141 160 180 199 235 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 189 169 149 130 122 114 106 97 89 81 73 65 57 49 41 32 41 49 57 65 73 81 89 97 106 114 122 130 149 169 189 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 217 197 178 158 139 119 111 103 95 87 79 70 62 54 46 38 30 38 46 54 62 70 79 87 95 103 111 119 139 158 178 197 217 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 242 206 187 167 148 128 108 100 92 84 76 68 60 52 43 35 27 35 43 52 60 68 76 84 92 100 108 128 148 167 187 206 242 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 215 196 176 156 137 117 98 90 81 73 65 57 49 41 33 25 33 41 49 57 65 73 81 90 98 117 137 156 176 196 215 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 240 204 185 165 146 126 106 87 79 71 63 54 46 38 30 22 30 38 46 54 63 71 79 87 106 126 146 165 185 204 240 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 213 194 174 155 135 115 96 76 68 60 52 44 36 27 19 27 36 44 52 60 68 76 96 115 135 155 174 194 213 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 238 203 183 163 144 124 105 85 65 57 49 41 33 25 17 25 33 41 49 57 65 85 105 124 144 163 183 203 238 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 211 192 172 153 133 113 94 74 55 47 38 30 22 14 22 30 38 47 55 74 94 113 133 153 172 192 211 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 236 201 181 161 142 122 103 83 64 44 36 28 20 11 20 28 36 44 64 83 103 122 142 161 181 201 236 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 210 190 170 151 131 112 92 72 53 33 25 17 9 17 25 33 53 72 92 112 131 151 170 190 210 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 235 199 179 160 140 120 101 81 62 42 22 14 6 14 22 42 62 81 101 120 140 160 179 199 235 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 208 188 168 149 129 110 90 71 51 31 12 4 12 31 51 71 90 110 129 149 168 188 208 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 216 197 177 158 138 119 99 79 60 40 21 1 21 40 60 79 99 119 138 158 177 197 216 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 242 222 202 183 163 144 124 104 85 65 46 38 46 65 85 104 124 144 163 183 202 222 242 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 227 208 188 169 149 130 110 90 82 74 82 90 110 130 149 169 188 208 227 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 233 213 194 174 155 135 127 119 111 119 127 135 155 174 194 213 233 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 238 219 199 180 172 163 155 147 155 163 172 180 199 219 238 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 224 216 208 200 192 184 192 200 208 216 224 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 237 228 220 228 237 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0

0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0

9.7 Guía de estilo de programación

Breve resumen de los principales puntos de la guía de programación.

Variables globales: se presentan siempre con caracteres alfanuméricos, pero siempre ha de empezar con una letra minúscula y si está compuesta por dos palabras, la segunda tendrá que empezar con mayúscula.

tipo ejemploVariable;

Variables auxiliares: se representan también con caracteres alfanuméricos y siempre comenzará por minúscula, pero si está formada por dos palabras, tendrá que estas separada la primera de la segunda palabra por el símbolo "_".

tipo var_auxiliar;

Funciones: se representan con caracteres alfanuméricos y siempre comenzará por una letra en mayúscula y en caso de contener varias palabras, todas empezaran siempre por mayúscula. Los parámetros estarán escritos en minúsculas.

Tipo FuncionUno(int parametro);

Clases: las clases estarán formadas por nombres que la identifiquen y tanto si está formada por una o varias palabras, estas tendrán que empezar con una letra mayúscula.

ClaseUno

Todas las clases han de estar definidas en un fichero ".h" que contendrá las cabeceras e implementadas en u n fichero ".c".

Comentarios: los comentarios en el código estarán delimitados por los símbolos "/*" al principio del comentario y "*/" al final del mismo.

/* esto es un comentario */

9.8 Sandrila

La herramienta Sandrila (Paul Herber's Sandrila Ltd s.f.) se ha utilizado para crear los diagramas en Lenguaje SDL a través del Microsoft Visio y comprobar sintácticamente si el diagrama era correcto.

Para más información visitar la web oficial:

http://www.sdl.sandrila.co.uk/

9.9 Referencia técnica y manual Firelib

Se puede encontrar la información completa de la librería el siguiente web oficial:

http://www.fire.org/downloads/fireLib/1.0.4/firelib.pdf

10 GLOSARIO

Autómata celular: Es un modelo matemático que modela un sistema dinámico que evoluciona en pasos discretos.

Behave: Modelo matemático que calcula el comportamiento del fuego a partir de unos modelos de combustibles.

Combustible: Cualquier tipo de substancia que es capaz de liberar energía cuando se quema y cambiar ó transformar luego su estructura química.

Datos raster: Es una matriz de celdas en un área determinada.

Datos vectoriales: Representan la información mediante vectores. Puede representar puntos, líneas ó polígonos.

Fuego: Proceso exotérmico de oxidación violenta de una materia de combustible con el desprendimiento de llamas, calor y gases.

Idrisi32: Es un sistema de información geográfica.

Propagación del fuego: Comportamiento que sigue el fuego a lo largo de una superficie.

Sandrila: Software para la creación de diagramas SDL con Microsoft Office Visio.

SDK: Software Develpment Kit, es un kit de desarrollo de software proporcionado normalmente por el creador del software, que ayuda al desarrollo de aplicaciones.

SDL: Lenguaje para formalizar modelos de simulación.

SDLPS: Software para la ejecución de simulaciones distribuidas a partir de modelos formalizados en lenguaje SDL.

SIG: Sistema de información geográfica, es una colección de hardware, software y datos geográficos.

Simulación paralela: Es la ejecución de una simulación con procesos repartidos en una misma computadora, de manera que comparten memoria.

Simulación discreta: Se centra en los eventos, la simulación consiste en seguir los cambios de estado del sistema en pasos de tiempo discretos.

Simulación distribuida: Es la ejecución de una simulación con procesos repartidos en diversas computadoras, de manera que no hay compartición de datos, ni de memoria entre ellas.

Vrml: Virtual Reality Modeling Language, es un formato de archive normalizado para la representación de gráficos en tres dimensiones.

11 ÍNDICE DE FIGURAS

Figura 1: Entrada de datos en BehavePlus 17
Figura 2: Salida de BehavePlus, comparativa de 4 modelos de combustible 17
Figura 3: Pantalla de simulación en Farsite 18
Figura 4: Entada de datos en Nexus 19
Figura 5: Gráfico de combustible en Fofem 20
Figura 6: Bloque 23
Figura 7: Proceso 23
Figura 8: Canal 23
Figura 9: Señal 23
Figura 10: Declaración 23
Figura 11: Procedimiento 23
Figura 12: Comentario 23
Figura 13: Estado 23
Figura 14: Evento de entrada 23
Figura 15: Evento de salida 23
Figura 16: Tarea 23
Figura 17: Línea de flujo 24
Figura 18: Decisión 24
Figura 19: Estado inicial 24
Figura 20: Software SDLPS 27
Figura 21: Vecindad de Moore 31
Figura 22: Extensión Cell-Devs 34
Figura 23: Extensión M:N CA^k 35
Figura 24: Diagrama Sistema versión 1 37
Figura 25: Diagrama del sistema versión final 38
Figura 26: Diagrama del MNCA_Fuego 41
Figura 27: Procedimiento Vecinity 41
Figura 28: Obtención celda actual 43
Figura 29: Obtención información celda actual 44
Figura 30: Obtención información celda vecina 44

Figura 31: Procedimiento Vecinity 45
Figura 32: Procedimiento Nucleus 46
Figura 33: Obtención información celda actual 47
Figura 34: Salvar información celda actual 47
Figura 35: Salvar información celda vecina 47
Figura 36: Procedimiento Nucleus 48
Figura 37: Diagrama de bloques 49
Figura 38: Diagrama de estados 51
Figura 39: Enviando la señal arder. 54
Figura 40: Recibiendo la señal arder 54
Figura 41: Obteniendo la propagación 55
Figura 42: Salvando los resultados 55
Figura 43: Enviando la señal propagar 55
Figura 44: Recibiendo la señal propagar 56
Figura 45: Diagrama de procesos, parte 1 57
Figura 46: Recibiendo señal arder 58
Figura 47: Recibiendo la señal de extinguir 58
Figura 48: Recibiendo la señal de propagar 59
Figura 49: Calculando la propagación 59
Figura 50: Salvando los resultados obtenidos 60
Figura 51: Tomando la decisión si se propaga el fuego 60
Figura 52: Enviando señal de extinguir 60
Figura 53: Enviando la señal de propagar 61
Figura 54: Recibiendo la señal de actualizardatos 61
Figura 55: Calculando la propagación 62
Figura 56: Enviando la señal de propagar 62
Figura 57: Diagrama de procesos, parte 2 63
Figura 58: Diagrama de procesos, parte 3 64
Figura 59: Función mncaGetCurrCell 67
Figura 60: Función mncaGetGetValue 67
Figura 61: Función mncaSetCellValue 67

Figura 62: Diagrama de clases 68
Figura 63: Clase FireSimulation 68
Figura 64: Ley de Amdahl 82
Figura 65: Resultado Behave 86
Figura 66: Resultado simulador no distribuido 86
Figura 67: Fechas planificación 91
Figura 68: Diagrama de Gantt 92

12 BIBLIOGRAFIA

Bevins, Collin D. "fireLib User Manual and Technical Reference."

Bitmanagement, Interactive web3d graphics. http://www.bitmanagement.com/.

Casas, Pau Fonseca I. *Formalització de models de simulació.*

Casas, Pau Fonseca I. "SDL, a graphical language useful to describe social simulation models."

Fire Behavior and Fire Danger Software. http://firemodels.fire.org/.

IEC. *SDL Tutorial.* http://www.iec.org/online/tutorials/sdl/topic04.html (accessed May 2010).

International Telecommunication Union . http://www.itu.int/.

International, Society for Modeling & Simulation. *SCS.* http://www.scs.org/.

J.M. Llavería, E. Herrada y A. Olivé. *Arquitectura de computadors (AC) Teoria 1.*

Javier Ameghino, Gabriel Wainer. "Modelando sistemas complejos con Cell-Devs."

Pau Fonseca, Josep Casanovas y Jordi Montero. "A celular autómata and intelligent agents use to model natural disasters with discrete simulation."

Paul Herber's Sandrila Ltd. http://www.sandrila.co.uk/.

Public Domain Software for the Windland Fire Community. http://www.fire.org/.

Reed, Rick. "SDL-2000 for New Millennium Systems."

Sastre, David Esteban. *Introducción a SDL.*

Telecommunication standardization sector of ITU. "Specification and Description Language (SDL)." *Series Z: Languages and general software aspects for telecommunication systems.* International Telecommunication Union. 1999. http://www.itu.int/ITU-T/studygroups/com17/languages/index.html (accessed April 2008).

Wainer, Gabriel A. "Introducción a la simulación de eventos discretos."

Wolfram's, Stephen. *Anew kind of science.*

Z.100, UIT-T. "Técnicas de descripción formal – Lenguaje de especificación y descripción."