



Universidad
Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

Proyecto Final de Carrera

Ingeniería Informática

Curso 2010/2011

Anexos

ESTRUCTURA DE ACELERACIÓN PARA RENDER 3D

DISTRIBUIDA MEDIANTE COMUNICACIONES PEER TO PEER

3/3

Diego Ignacio Mallea Lobera

Director:

Juan Antonio Magallón Lacarta

Grupo de Informática Gráfica Avanzada

Departamento de Informática e Ingeniería de Sistemas

Centro Politécnico Superior

Universidad de Zaragoza

Junio de 2011

*A mis padres Jorge José y María Pilar Olga, y a mis hermanas Gabriela y
Eugenia,
por insistirme en que termine mis estudios.*

*A mi novia Nadia,
por no reclamar el tiempo empleado.*

*A mis amigos de toda la vida,
por secuestrarme e impedirme que me centre en acabar la carrera.*

Derechos de Autor

Los derechos de la presente obra perteneces a D. Diego Ignacio Mallea Lobera y al Dr. D. Juan Antonio Magallón Lacarta, del Departamento de Informática e Ingeniería de Sistemas del Centro Politécnico Superior de la Universidad de Zaragoza. Queda prohibida la reproducción total o parcial de esta obra, por cualquier medio, sin el permiso escrito de los autores.

Ficha Técnica

Proyecto fin de carrera

Título:	ESTRUCTURA DE ACCELERACIÓN PARA RENDER 3D DISTRIBUIDA MEDIANTE COMUNICACIONES PEER TO PEER
Autor:	D. Diego Ignacio Mallea Lobera
DNI:	72981973-Y
Promoción:	2004/2011
Especialidad:	Informática
Director:	Juan Antonio Magallón Lacarta
Departamento:	Informática e Ingeniería de Sistemas
Centro:	Centro Politécnico Superior
Universidad:	Universidad de Zaragoza
Fecha	Junio 2011

Resumen

Hoy en día, se puede afirmar que el acceso a computadoras con una gran potencia de cálculo está al alcance en nuestros hogares o puestos de trabajo.

Estas pequeñas estaciones, con las que consultamos el correo o las cuentas de nuestras redes sociales, están la mayoría del tiempo sin aprovechar al máximo su poder de cálculo. Hace tiempo que programas como el SETI, Folding o figthAids entre otros aprovechan estos tiempos de baja actividad de nuestras computadoras para procesar en horas grandes cantidades de datos, las cuales requerirían de supercomputadoras que requieren de costosas inversiones y un mantenimiento tan costoso que solo grandes gobiernos u organismos pueden hacer frente.

Por otro lado tenemos que ámbitos tan dispares como el cine, los videojuegos, en la aeronáutica y un largo etcétera, se hace necesario el computar ingentes cantidades de datos con cálculos increíblemente complejos.

Un ámbito en el cual se tiene una serie de cálculos que se pueden paralelizar y tiene una complejidad tan alta que requiere de más de un computador, es el mundo de la animación 3D.

El presente proyecto busca evaluar los fundamentos básicos tras la producción de una animación 3D, en un sistema distribuido Peer to Peer.

En primera instancia se decidió el evaluar las búsquedas en árboles *kd* mediante el empleo de dicha red, para ello se decidió el implementar una red de dichas características para poder distribuir el cálculo y los nodos de la red.

Posteriormente se implemento una serie de algoritmos comúnmente empleados para la generación de dichas escenas animadas, búsquedas en *kd-tree*, modificando estos para poder realizar los cálculos de una manera distribuida entre los componentes de la red.

Según se avanzo en el estudio se hizo patente un hambre por evaluar e investigar las posibilidades de dicho sistema para nubes de volúmenes incurriendo en unas posibilidades de investigar las posibilidades de las mismas.

Tras el desarrollo del PFC, se puede concluir que el uso de redes P2P abre un sinfín de posibilidades y mejoras. Además la ampliación de dichos algoritmos del procesado de nubes de puntos a nubes de volúmenes abre posibilidades que congenian con técnicas de colisiones.

CONTENIDOS

Anexo I. Planificación	1
I.I. Ciclo de vida.....	1
I.II. Metas iniciales.....	1
I.II.I. Metas iniciales del primer ciclo	1
I.II.II. Metas iniciales del segundo ciclo.....	1
I.II.III. Metas iniciales del tercer ciclo	1
I.III. Primera estimación de la planificación	1
I.IV. Resumen del primer ciclo	2
I.IV.I. Objetivos del primer ciclo.....	2
I.IV.II. Riesgos analizados en el primer ciclo.....	2
I.IV.III. Análisis del primer ciclo	2
I.IV.IV. Ajuste en la planificación	3
I.V. Resumen del segundo ciclo.....	4
I.V.I. Objetivos del segundo ciclo	4
I.V.II. Riesgos analizados en el segundo ciclo	4
I.V.III. Análisis del ciclo segundo ciclo.....	4
I.V.IV. Ajuste en la planificación	4

I.VI. Resumen del tercer ciclo	5
I.VI.I. Objetivos del tercer ciclo.....	5
I.VI.II. Riesgos analizados.....	5
I.VI.III. Análisis del ciclo	5
I.VI.IV. Ajuste en la planificación	6
I.VII. Resumen del cuarto ciclo	7
I.VII.I. Objetivos.....	7
I.VII.II. Riesgos analizados	7
I.VII.III. Análisis del ciclo	7
I.VII.IV. Ajuste en la planificación	7
Anexo II. Análisis.....	9
II.I. Diagramas de Estructura	9
II.I.I. Diagrama de Clases	9
II.I.II. Diagrama de Componentes.....	20
II.I.III. Diagrama de Objetos.....	21
II.I.IV. Diagrama de Despliegue	22
II.I.V. Diagrama de Paquetes	22
II.II. Diagramas de Comportamiento	23
II.II.I. Diagrama de Casos de Uso.....	23
II.II.II. Diagrama de Estados	27
II.III. Diagramas de Interacción	27
II.III.I. Diagrama de Secuencia	27
Anexo III. Simulación.....	31
III.I. Resultados de los experimentos	31
III.I.I. Cálculo del tiempo base	31
Anexo IV. Algoritmos.....	45
IV.I. Generación del kd tree de una nube de puntos	45
IV.I.I. Explicación	45
IV.I.II. Desarrollo	45
IV.I.III. Pseudocódigo de la construcción de un kdtree en una nube de puntos	48
IV.II. Generación del un kd tree con elementos con volumen	49
IV.II.I. Explicación	49

IV.II.II.	Desarrollo	49
IV.II.III.	Pseudocódigo de construcción de un árbol kdtree de una nube de volúmenes	52
IV.III.	Búsqueda de puntos contenidos en una esfera de una nube de puntos	55
IV.III.I.	Explicación	55
IV.III.II.	Desarrollo	55
IV.III.III.	Estudiémoslo con un ejemplo:.....	59
IV.III.IV.	Pseudocódigo de la búsqueda esférica en una nube de puntos	63
IV.IV.	Búsqueda de volúmenes contenidos en una esfera de una nube de volúmenes.....	64
IV.IV.I.	Explicación	64
IV.IV.II.	Desarrollo	64
IV.IV.III.	Pseudocódigo de la búsqueda esférica en una nube de volúmenes	70
IV.V.	Búsqueda del punto más cercano	73
IV.V.I.	Explicación	73
IV.V.II.	Desarrollo	73
IV.V.III.	Pseudocódigo de la búsqueda del más cercano en una nube de puntos	75
IV.VI.	Búsqueda del volumen más cercano	77
IV.VI.I.	Explicación	77
IV.VI.II.	Desarrollo	77
IV.VI.III.	Pseudocódigo de la búsqueda del más cercano en una nube de volúmenes	79
IV.VII.	Búsqueda ortogonal en una nube de puntos.....	81
IV.VII.I.	Explicación	81
IV.VII.II.	Desarrollo	81
IV.VII.III.	Pseudocódigo de la búsqueda ortogonal en una nube de puntos	83
IV.VIII.	Búsqueda ortogonal en una nube de volúmenes	84
IV.VIII.I.	Explicación	84
IV.VIII.II.	Desarrollo.....	84
IV.VIII.III.	Pseudocódigo de la búsqueda ortogonal en una nube de volúmenes	86

Anexo V. Tabla de Hash Distribuido.....	89
V.I. Preámbulo	89
V.II. Definición	89
V.III. Historia	90
V.IV. Estructura	93
V.V. Particionamiento del espacio de claves	95
V.VI. Red de overlay	97
V.VII. Algoritmos para redes de overlay	98
V.VIII. Propiedades	98
V.IX. Implementaciones de DHT	98
Anexo VI. Estimación de Costes	101
VI.I. Presupuesto.....	101
VI.II. Comunicado del desarrollador:	101
VI.III. Ofertas de formación	102
VI.III.I. Curso de desarrollo de aplicaciones RMI:	102
VI.III.II. Curso de desarrollo de un servicio de mensajería:	102
VI.III.III. Curso de desarrollo de un programa P2P:	102
Anexo VII. Pliego de Condiciones	103
VII.I. Condiciones Generales	103
VII.I.I. Documentación del proyecto	103
VII.I.II. Documentación del propietario	103
VII.II. Instrucciones de utilización	103
VII.II.I. Descripción técnica	103
VII.II.II. Plazo de garantía	104
VII.II.III. Exclusión de responsabilidad	105
VII.II.IV. Formalización del contrato.	105
VII.II.V. Rescisión de contrato.	105
VII.II.VI. Plazo de ejecución	105
VII.II.VII. Limitación de responsabilidad	105
VII.III. Normas, leyes y reglamentos	106
VII.IV. Condiciones contractuales.	107
VII.IV.I. Compatibilidad y relación entre documentos.	107
VII.IV.II. Representante del contratista.....	107

VII.IV.III.	Interpretación del proyecto.....	107
VII.IV.IV.	Modificaciones del diseño del proyecto.....	107
VII.IV.V.	Condiciones del proyecto.....	107
VII.IV.VI.	Revisión o alteración del proyecto.	107
VII.V.	Condiciones técnicas y materiales	108
VII.VI.	Exclusividad del proyecto	109
VII.VII.	Plazos de entrega	112
VII.VIII.	Derechos de autor.....	113
VII.IX.	Licencia de software.....	114
Anexo VIII.	Razones Geométricas	121
VIII.I.	Preámbulo	121
VIII.II.	Distancia de un punto a una recta	121
VIII.II.I.	Datos.....	121
VIII.II.II.	Cálculos.....	121
VIII.III.	Distancia de un punto a un segmento	122
VIII.III.I.	Datos.....	122
VIII.III.II.	Cálculos	122
VIII.IV.	Distancia de un punto a un plano	123
VIII.IV.I.	Datos.....	123
VIII.IV.II.	Cálculos	123
VIII.V.	Distancia de un punto a un triángulo	124
VIII.V.I.	Datos.....	124
VIII.V.II.	Cálculos.....	124
VIII.VI.	Distancia de un punto a un cuadrado	125
VIII.VI.I.	Datos.....	125
VIII.VI.II.	Cálculos	125
VIII.VII.	Distancia de un punto a un cubo	126
VIII.VII.I.	Datos.....	126
VIII.VII.II.	Cálculos	126
VIII.VIII.	Proyección de un punto a un plano	127
VIII.VIII.I.	Datos.....	127
VIII.VIII.II.	Cálculos	127
VIII.IX.	Proyección de un punto en un segmento	128

VIII.IX.I. Datos	128
VIII.IX.II. Cálculos	128
VIII.X. Proyección de un punto en un triángulo	129
VIII.X.I. Datos	129
VIII.X.II. Cálculos.....	130
VIII.XI. Intersección de un punto con una recta	131
VIII.XI.I. Datos	131
VIII.XI.II. Cálculos	131
VIII.XII. Intersección de un punto con un plano.....	132
VIII.XII.I. Datos.....	132
VIII.XII.II. Cálculos	132
VIII.XIII. Intersección de un punto con un segmento.....	133
VIII.XIII.I. Datos.....	133
VIII.XIII.II. Cálculos	133
VIII.XIV. Intersección de un punto con un triángulo	134
VIII.XIV.I. Datos.....	134
VIII.XIV.II. Cálculos	135
VIII.XV. Intersección de un punto con un cubo ortogonal	136
VIII.XV.I. Datos.....	136
VIII.XV.II. Cálculos	136
VIII.XVI. Intersección de una recta con una recta	137
VIII.XVI.I. Datos.....	137
VIII.XVI.II. Cálculos	137
VIII.XVII. Intersección de una segmento con un segmento.....	138
VIII.XVII.I. Datos.....	138
VIII.XVII.II. Cálculos.....	138
VIII.XVIII. Intersección de una recta con un plano	139
VIII.XVIII.I. Datos	139
VIII.XVIII.II. Cálculos.....	139
VIII.XIX. Intersección de un segmento con un plano.....	141
VIII.XIX.I. Datos.....	141
VIII.XIX.II. Cálculos	141
VIII.XX. Intersección de un segmento con un cuadrado	143

VIII.XX.I. Cálculos	143
VIII.XXI. Intersección de un cubo con un cuadrado	144
VIII.XXI.I. Cálculos	144
VIII.XXII. Intersección de un triángulo con un cubo	145
VIII.XXII.I. Cálculos	145
VIII.XXIII. Intersección de un triángulo con un segmento.....	146
VIII.XXIII.I. Cálculos.....	146
VIII.XXIV. Intersección de un segmento con un cubo	147
VIII.XXIV.I. Cálculos	147
VIII.XXV. Intersección de un cubo con un cubo	148
VIII.XXV.I. Cálculos	148

Anexo I. PLANIFICACIÓN

I.I. Ciclo de vida

El ciclo de vida elegido para llevar la planificación del proyecto es un un ciclo de vida en espiral. Se descarto el modelo de vida en cascada, al tener que tener todos los requisitos al comienzo, no estando estos totalmente definidos. El ciclo de vida en V tampoco nos era fiel a la realidad.

Después de evaluar otros tantos modelos se decanto por un modelo de desarrollo en espiral, al ser este más adaptable frente a la aparición de riesgos.

I.II. Metas iniciales

Las metas iniciales fueron las de finalizar el proyecto en los 6 meses siguientes al inicio del mismo. Para ello se plantearon tres ciclos.

I.II.I. Metas iniciales del primer ciclo

El primer ciclo se destinaría a bocetar el proyecto en general, dando lugar a una estructura de seguimiento, una plataforma de desarrollo y un primer prototipo, sin todas las funcionalidades, de la aplicación.

I.II.II. Metas iniciales del segundo ciclo

El segundo ciclo se emplearía en el desarrollo y la mejora del prototipo, además de actualizar la documentación del proyecto.

I.II.III. Metas iniciales del tercer ciclo

El tercer ciclo seria el destinado a depurar y testear la solución, además de emplearlo en recopilar datos de su funcionamiento. Por último plasmaríamos los datos recopilados en la documentación una vez se hubiera terminado su análisis.

I.III. Primera estimación de la planificación

Primera estimación de los tiempos que llevaría terminar el proyecto.

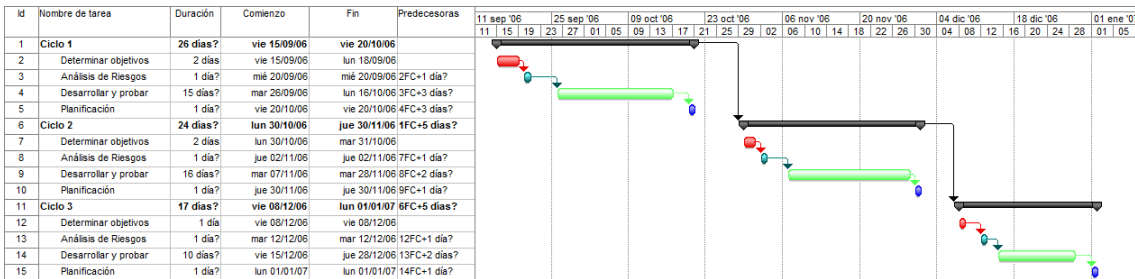


Ilustración 1 Primera estimación de la planificación

I.IV. Resumen del primer ciclo

I.IV.I. *Objetivos del primer ciclo*

Los objetivos planteados en el primer ciclo fueron.

Objetivos del primer ciclo
Bocetar los documentos de seguimiento del proyecto.
Estudiar la técnica.
Elección del entorno de desarrollo.
Implementar primer prototipo.

Tabla 1 Objetivos del primer ciclo

I.IV.II. *Riesgos analizados en el primer ciclo*

Los riesgos analizados en el primer ciclo fueron:

Riesgos del primer ciclo
Complejidad de la técnica.
Incumplimiento de los tiempos planificados.

Tabla 2 Riesgos analizados en el primer ciclo

I.IV.III. *Análisis del primer ciclo*

El tiempo estimado para el desarrollo de los bocetos de la documentación se excedió.

El tiempo empleado en el estudio de la técnica se excedió.

La elección del entorno de desarrollo fue llevada en menos tiempo del estimado.

La implementación del primer prototipo excedió el tiempo estimado, el prototipo no fue concluido, dando paso a un replanteamiento de la planificación del proyecto.

La planificación desarrollada para el primer ciclo no fue cumplida. Debido a contratiempos, dicho ciclo se alargó más de lo previsto.

Los objetivos marcados para esta primera fase no fueron cumplidos. Los principales contratiempos que causaron este desbarajuste fueron:

Contratiempos del primer ciclo
Largos tiempos de inactividad.
Decisiones de desarrollo mal tomadas.
Complejidad de los algoritmos.

Tabla 3 Contratiempos del primer ciclo

I.IV.IV. Ajuste en la planificación

La planificación fue reajustada consecuentemente con los resultados:

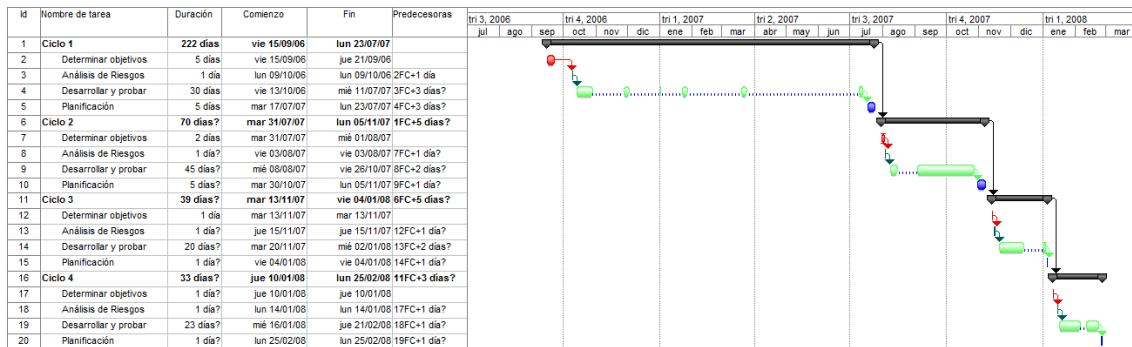


Ilustración 2 Ajuste en la planificación

Se añadió un ciclo nuevo a la vida del proyecto.

El segundo ciclo se destinaría al desarrollo de un prototipo y mantenimiento de la documentación.

El tercer ciclo a depuración y testeo de la utilidad además del mantenimiento de la documentación.

El último ciclo se destinara al análisis de los datos obtenidos y finalización de la documentación.

I.V. Resumen del segundo ciclo

I.V.I. Objetivos del segundo ciclo

Los objetivos planteados en el segundo ciclo fueron.

Objetivos del segundo ciclo

- Mantener los documentos de seguimiento del proyecto.
- Implementar el segundo prototipo.

Tabla 4 Objetivos del segundo ciclo

I.V.II. Riesgos analizados en el segundo ciclo

Los riesgos analizados en el segundo ciclo fueron:

Riesgos del segundo ciclo

- Retrasos por la complejidad de la técnica.
- Problemática del prototipo.
- No cumplimiento de los tiempos planificados.

Tabla 5 Riesgos del segundo ciclo

I.V.III. Análisis del ciclo segundo ciclo

Las fechas y el tiempo empleados en la implementación del prototipo se excedieron.

Se consiguió realizar un primer prototipo. Se ha dejado para la tercera fase la implementación y optimización de ciertas funcionalidades.

Se ha vuelto a incurrir en tiempos de inactividad. El avance en la implementación ha sido muy lento por la complejidad de los algoritmos.

I.V.IV. Ajuste en la planificación

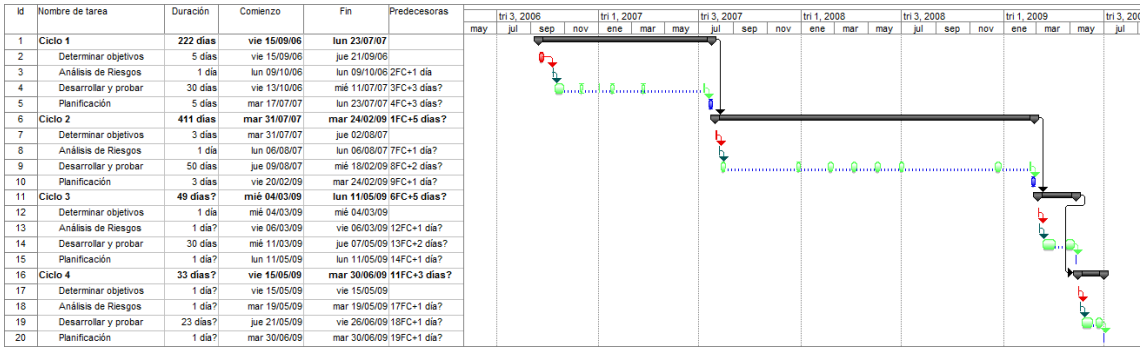


Ilustración 3 Ajuste en la planificación

Se decidió mantener la planificación de los ciclos pero ajustando la duración de los mismos con el fin de hacerla más realista.

I.VI. Resumen del tercer ciclo

I.VI.I. *Objetivos del tercer ciclo*

Los objetivos planteados en el tercer ciclo fueron.

Objetivos del tercer ciclo
Mantener los documentos de seguimiento del proyecto.
Resolver errores detectados en la fase anterior.
Ampliar el modelo para tratar con volúmenes.
Añadir automatizaciones a la aplicación.
Añadir visualizaciones gráficas de la estructura del árbol.
Añadir visualizaciones gráficas 3D de la estructura de las formas.
Completar la implantación de las funcionalidades.
Desarrollo de una batería de test.

Tabla 6 Objetivos del tercer ciclo

I.VI.II. *Riesgos analizados*

Los riesgos analizados en el tercer ciclo fueron:

Riesgos del tercer ciclo
Implementaciones mal llevadas a cabo.
No cumplimiento de los tiempos planificados.

Tabla 7 Riesgos del tercer ciclo

I.VI.III. *Análisis del ciclo*

El inicio del ciclo se vio retrasado. Las fechas y el tiempo empleados en resolver los errores se excedieron.

Durante el desarrollo se encontraron más errores de los inicialmente detectados.

Contratiempos del tercer ciclo
Se ha vuelto a incurrir en tiempos de inactividad y lapsos de tiempo sin avanzar en el desarrollo por la complejidad de los algoritmos.
Las visualizaciones gráficas excedieron el tiempo planificado.
Las automatizaciones excedieron el tiempo planificado.
Las baterías de test excedieron el tiempo planificado.
Se decidió documentar el código, objetivo que inicialmente no estaba contemplado.

Tabla 8 Contratiempos del tercer ciclo

I.VI.IV. Ajuste en la planificación

La planificación fue reajustada consecuentemente con los resultados:

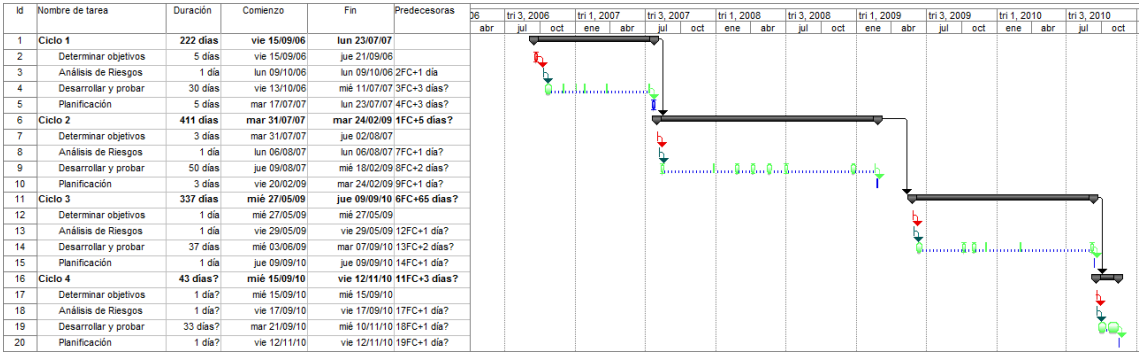


Ilustración 4 Ajuste en la planificación

La planificación del cuarto ciclo se decidió reajustar los tiempos.

I.VII. Resumen del cuarto ciclo

I.VII.I. Objetivos

Los objetivos planteados en el cuarto ciclo fueron.

Objetivos del cuarto ciclo

- Mantener los documentos de seguimiento del proyecto.
- Resolver errores detectados.
- Añadir automatizaciones a la aplicación.
- Recolección y análisis de datos.

Tabla 9 Objetivos del cuarto ciclo

I.VII.II. Riesgos analizados

Los riesgos analizados en el cuarto ciclo fueron:

Riesgos del cuarto ciclo

- No cumplimiento de los tiempos planificados.

Tabla 10 Riesgos del cuarto ciclo

I.VII.III. Análisis del ciclo

El inicio del ciclo se vio retrasado por problemas laborales.

Las fechas y el tiempo empleados en resolver los errores se excedieron.

Las fechas y el tiempo empleados en recoger datos y analizarlos se excedieron.

Los tiempos en elaborar la documentación se excedieron.

I.VII.IV. Ajuste en la planificación

La planificación fue reajustada consecuentemente con los resultados:

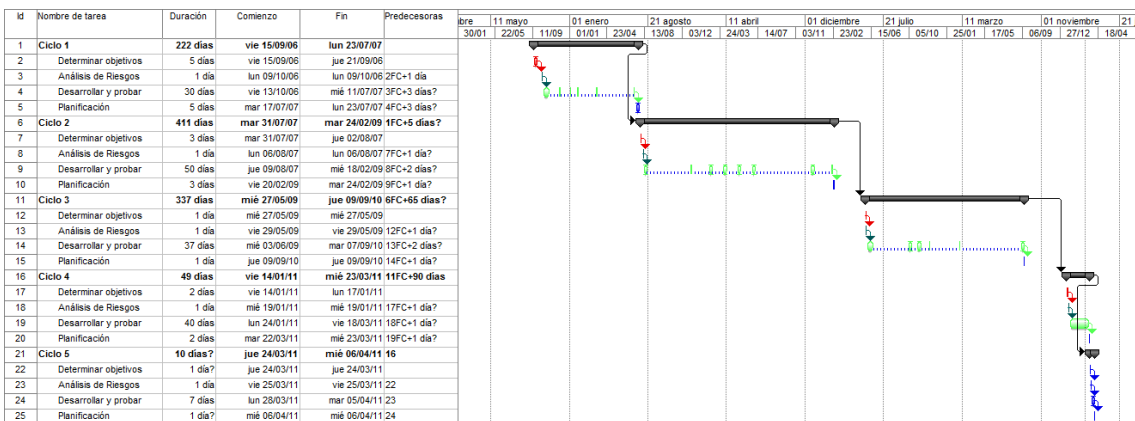


Ilustración 5 Ajuste en la planificación

Se decidió añadir un quinto ciclo, este está destinado a la elaboración de la presentación del mismo. Solo estará estimado al ser posterior al plazo de presentación del mismo.

Anexo II. ANÁLISIS

II.I. Diagramas de Estructura

II.I.I. Diagrama de Clases

Estructura del sistema:



Ilustración 6 Estructura del sistema

El sistema contará de una estructura MVC (Modelo, vista y controlador). Para poder permitir la evolución del sistema se ha creado una estructura modular prevista de clases abstractas implementadas por los componentes principales con el fin de poder ser mejorados o remplazados en posteriores versiones. Así si en un futuro se quisiera dar soporte o mejoras su compatibilidad estará garantizada.

Estructura del modelo

El modelo de datos abarca los modelos relacionados con el listado de estaciones de render, el listado de formas y el árbol kd.

Modelo	Descripción
Estaciones de render	Las estaciones de render nos sirven para saber los datos de enrutamiento con las otras estaciones de render.
Listado de formas	El listado de formas nos servirá para poder generara el árbol kd.
Árbol kd	El árbol kd cobra vida al procesar el listado de formas para su generación. Tiene la finalidad de ser la herramienta usada para la realización de las simulaciones.

Tabla 11 Estructura de los datos del modelo

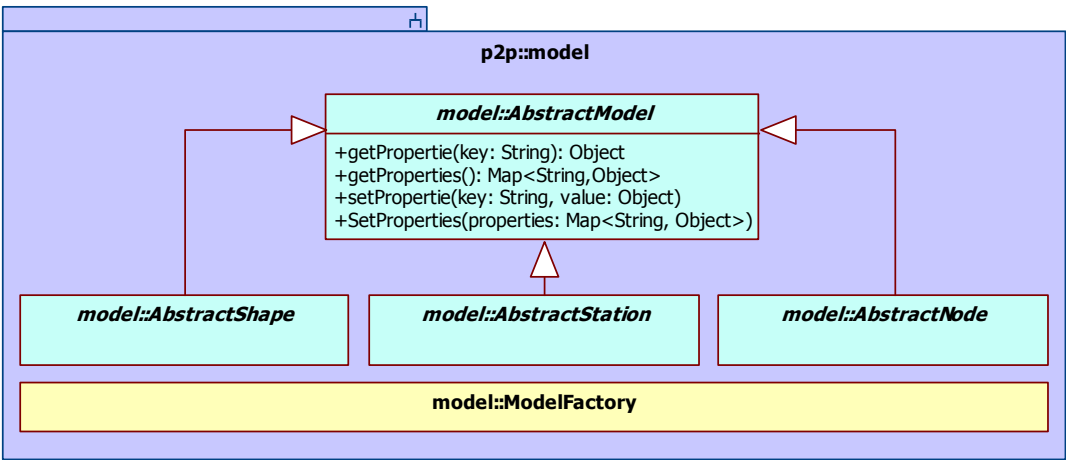


Ilustración 7 Estructura del modelo

Descripción
Todos los modelos heredan de una clase abstracta con el fin de proveer un soporte básico común para los modelos. La factoría nos permitirá poder crear distintas implementaciones de los modelos de una manera fácil y sencilla.

Tabla 12 Estructura del modelo

Modelo del listado de estaciones de render

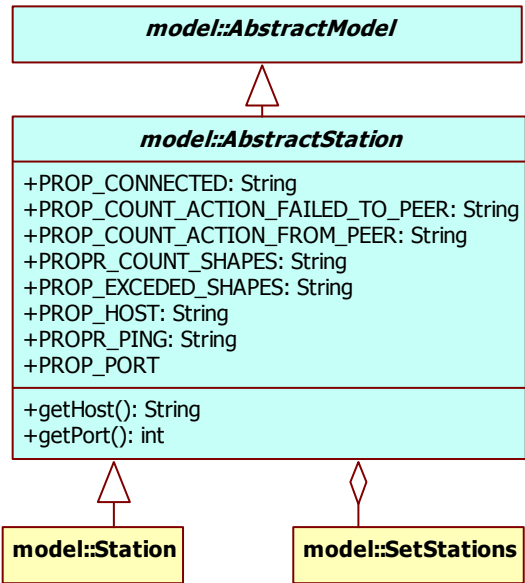


Ilustración 8 Modelo del listado de estaciones de render

Descripción
El modelo abstracto de la estación de render provee de unas operaciones y propiedades básicas para la conexión de las estaciones, la implementación realizada así como posibles implementaciones futuras heredarán esta estructura, asegurando la compatibilidad.

Tabla 13 Modelo del listado de estaciones de render

Modelo del listado de formas

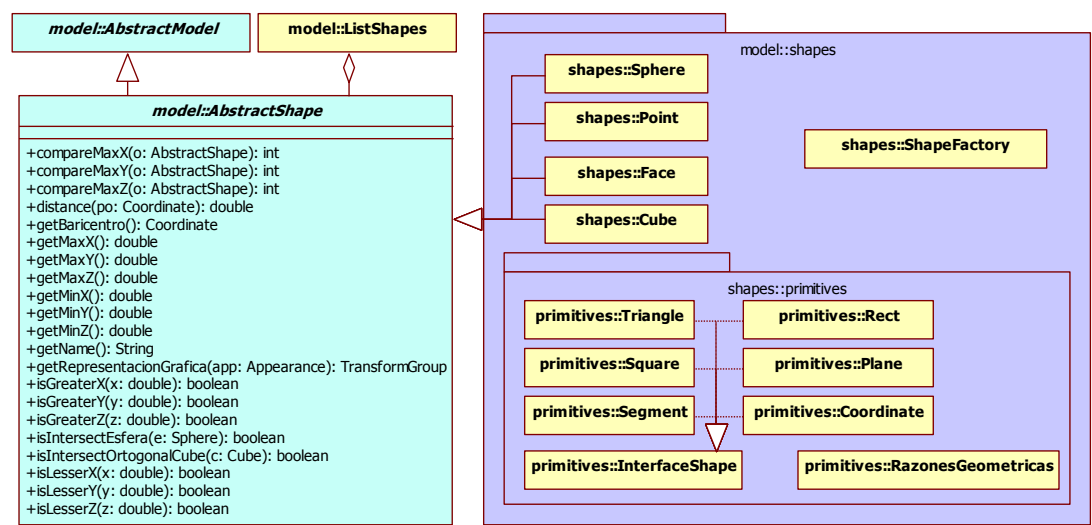


Ilustración 9 Modelo del listado de formas

Descripción

El modelo abstracto de las formas provee de unas operaciones y propiedades básicas para las operaciones entre las mismas, la implementación realizada así como posibles implementaciones futuras heredarán esta estructura, asegurando la compatibilidad. El modelo del listado de formas utilizara un conjunto de primitivas, así como unas razones geométricas para las distintas implementaciones de las formas. La factoría nos da una manera de obtener la implementación de una forma de manera sencilla.

Tabla 14 Modelo del listado de formas

Modelo del árbol kd

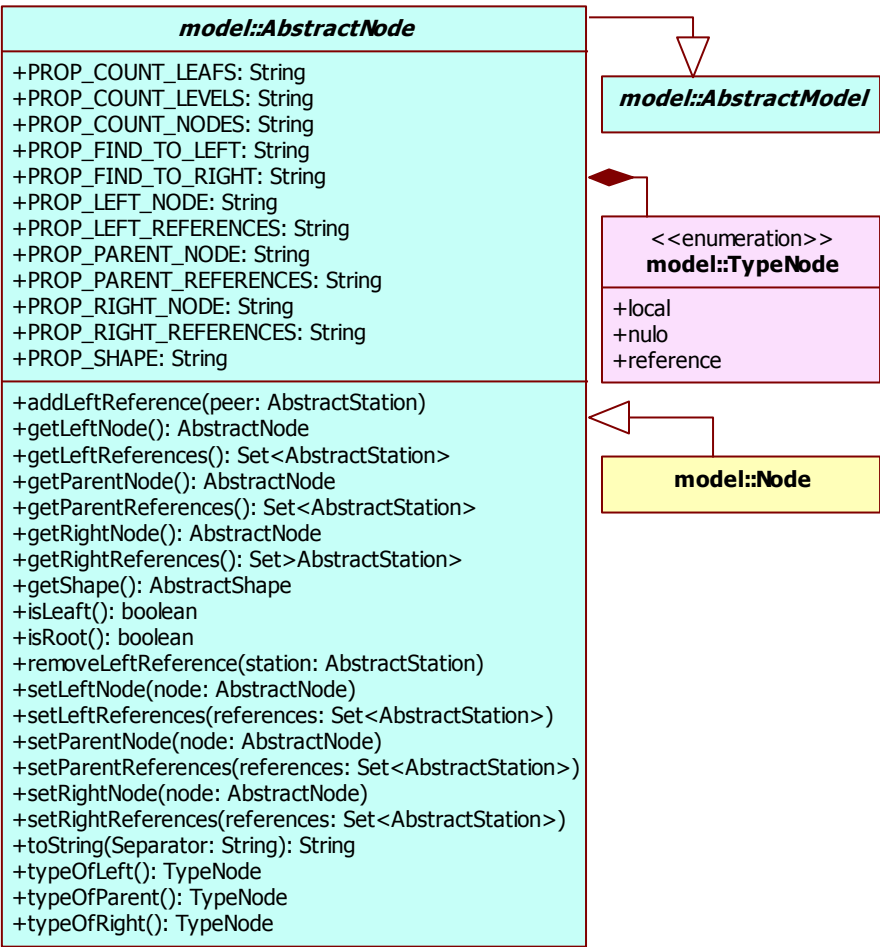


Ilustración 10 Modelo del árbol kd

Descripción

El modelo abstracto del nodo del árbol provee de unas operaciones y propiedades básicas para las operaciones con el mismo, la implementación realizada así como posibles implementaciones futuras heredarán esta estructura, asegurando la compatibilidad. El árbol hará uso de varios nodos para construir su estructura.

Tabla 15 Modelo del árbol kd

Estructura del controlador

En el controlador se han implementado una estructura modular.

Se ha creado una capa de abstracción con el fin de poder modificar la implementación de los módulos sin afectar en lo más mínimo en cuanto a funcionalidades.

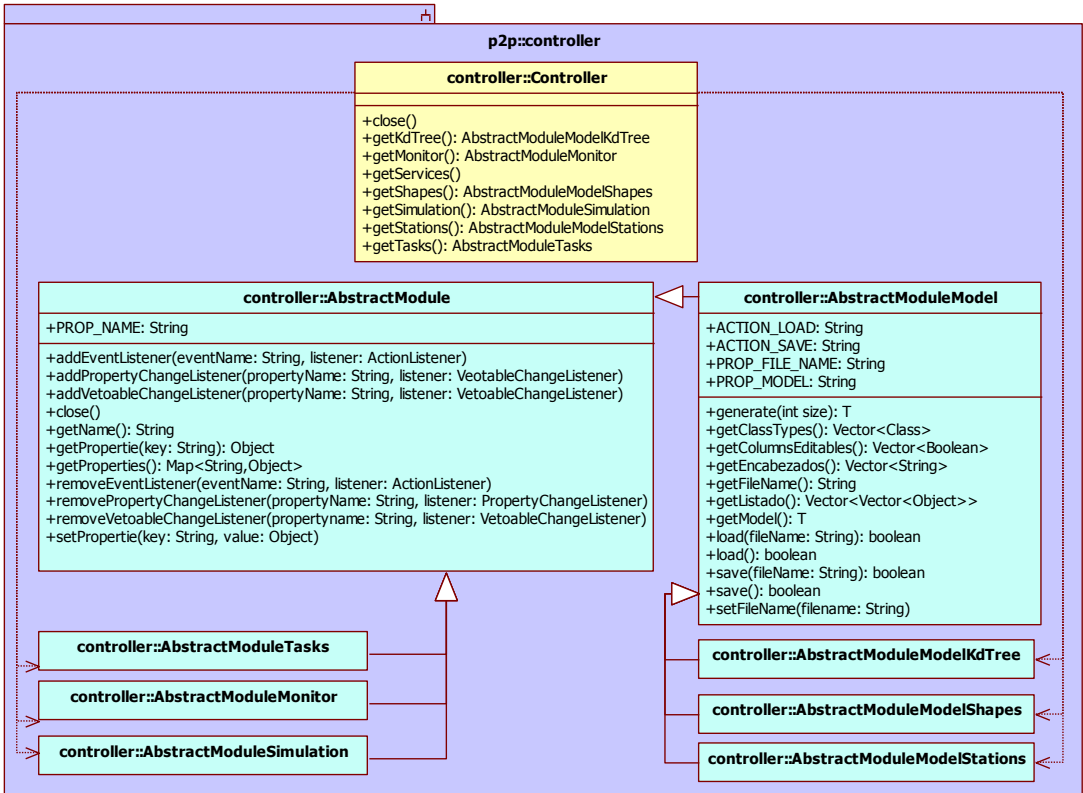


Ilustración 11 Estructura del controlador

Descripción

El controlador esta subdividido en varios módulos. Cada módulo proporciona una serie de funcionalidades básicas a la aplicación. La descripción de dichas funcionalidades básicas esta descrita por módulos abstractos, con el fin de poder ampliar dicha funcionalidad, manteniendo la compatibilidad de las distintas versiones.

Tabla 16 Estructura del controlador

Módulo de servicios del controlador

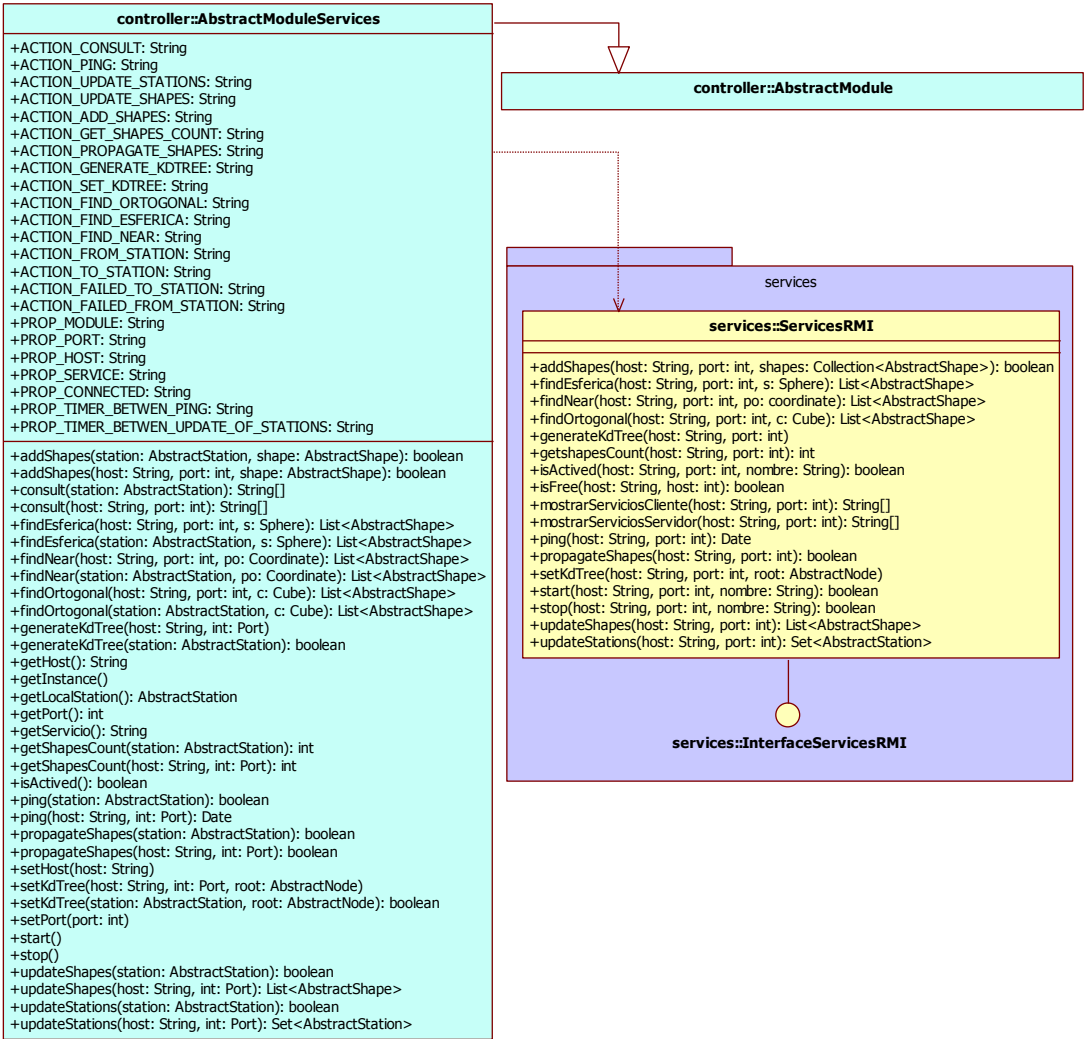


Ilustración 12 Módulo de servicios del controlador

Descripción

El módulo de servicios es el eje principal de la aplicación en el ámbito de las comunicaciones. Este módulo es el que se encarga de de dar soporte a las comunicaciones con otras estaciones de render.

Tabla 17 Módulo de servicios del controlador

Módulo de estaciones de render del controlador

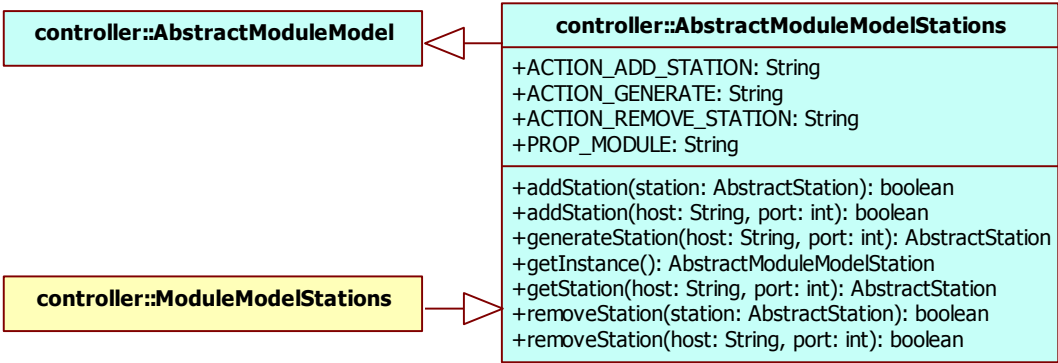


Ilustración 13 Módulo de estaciones de render del controlador

Descripción
El módulo del listado de estaciones es el encargado de gestionar localmente las estaciones de render que posteriormente serán utilizadas como cache local en las operaciones remotas.

Tabla 18 Módulo de estaciones de render del controlador

Módulo de formas del controlador

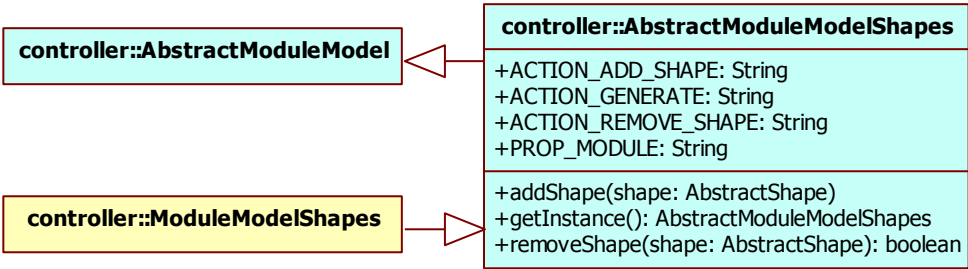


Ilustración 14 Módulo de formas del controlador

Descripción
El módulo del listado de formas es el encargado de gestionar localmente las formas que posteriormente se almacenaran en el árbol.

Tabla 19 Módulo de formas del controlador

Módulo del árbol del controlador

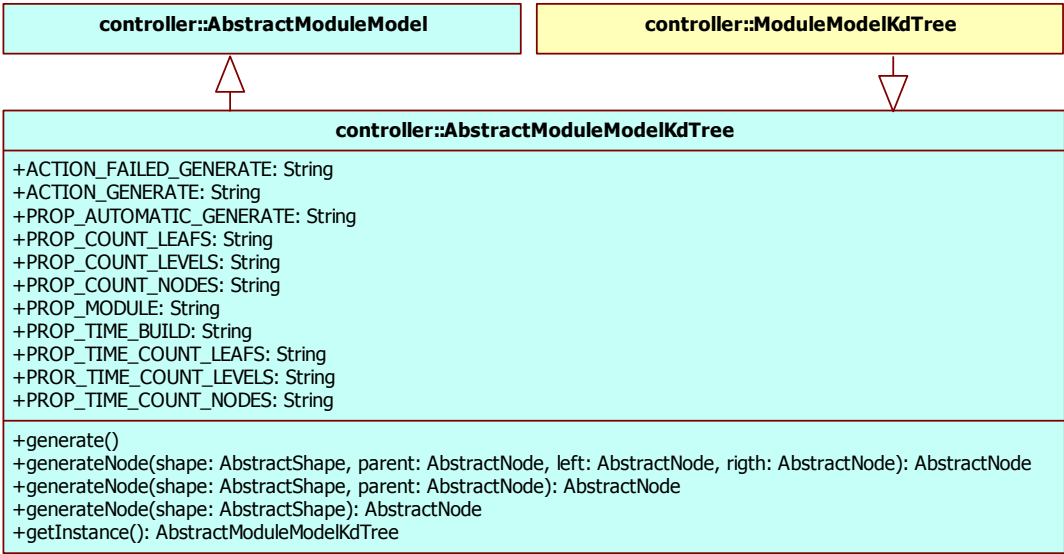


Ilustración 15 Módulo del árbol del controlador

Descripción

El módulo del árbol kd nos permite gestionar y navegar por el árbol.

Tabla 20 Módulo del árbol del controlador

Módulo de simulación del controlador

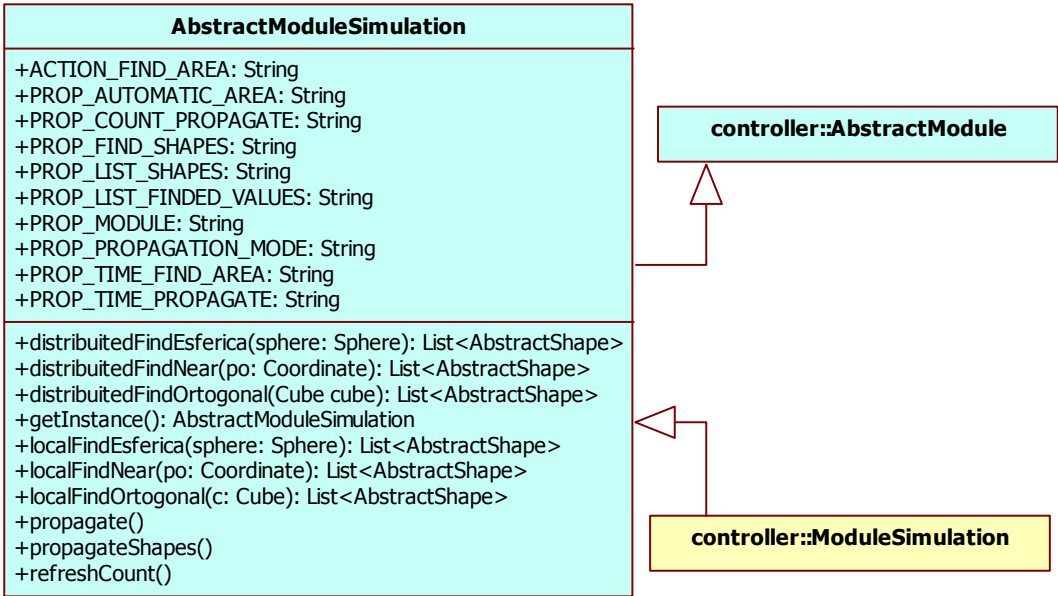


Ilustración 16 Módulo de simulación del controlador

Descripción

El módulo de simulación nos servirá para procesar los distintos algoritmos a evaluar, tanto los de ámbito local como distribuido.

Tabla 21 Módulo de simulación del controlador

Módulo del monitor del controlador

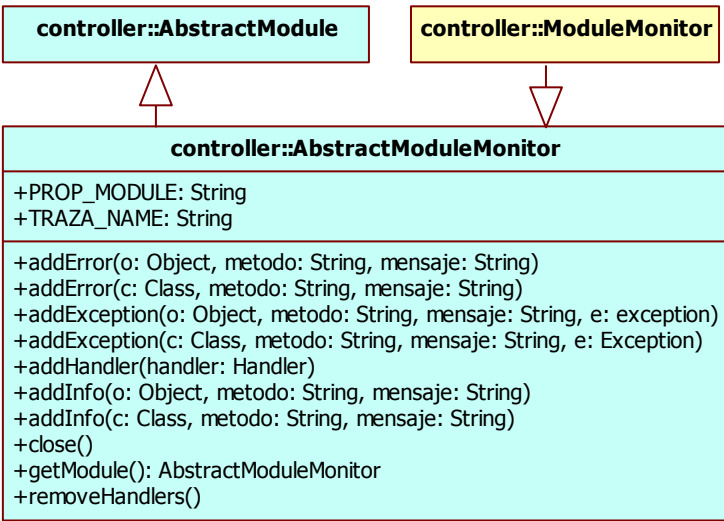


Ilustración 17 Módulo del monitor del controlador

Descripción

El módulo de monitorización nos permite hacer un seguimiento del estado de los procesos, permitiendo mostrarnos un seguimiento selectivo de la aplicación..

Tabla 22 Módulo del monitor del controlador

Módulo de tareas del controlador

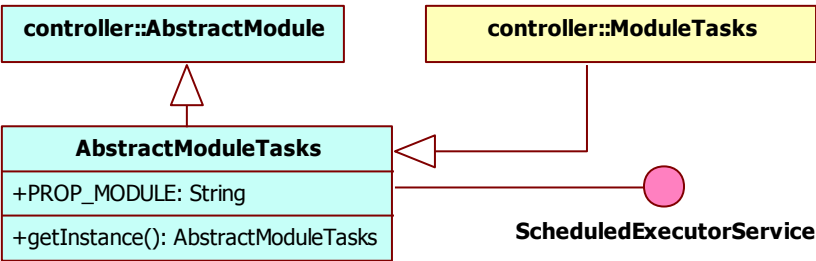


Ilustración 18 Módulo de tareas del controlador

Descripción

El módulo de tareas es una implementación de un programador de tareas. Nos permite hacer ejecución paralela. En este caso implementamos el interfaz de java **java.util.concurrent.ScheduledExecutorService** para este cometido.

Tabla 23 Módulo de tareas del controlador

Estructura del Interfaz del Usuario

En nuestra interfaz de usuario, tenemos una serie de paneles por cada sección.

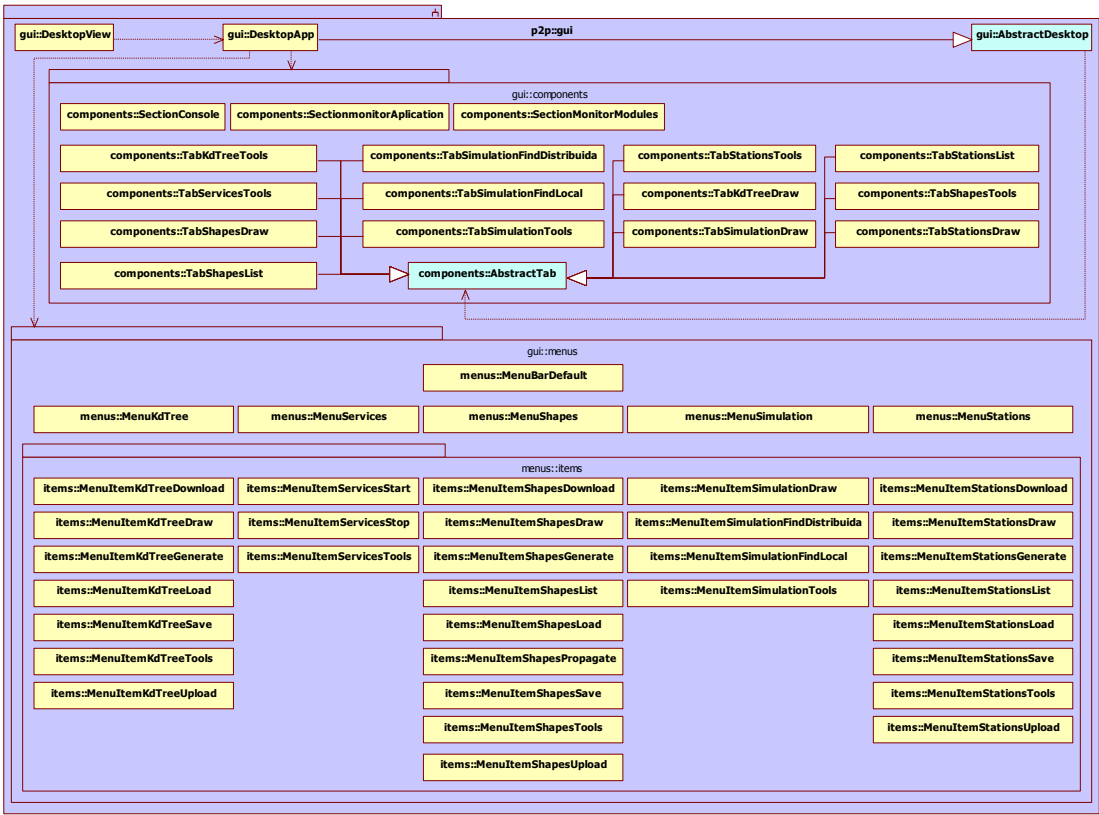


Ilustración 19 Estructura del Interfaz del Usuario

Descripción

La vista estará compuesta por una serie de componentes que dispararan distintas funcionalidades del controlador. Tendremos una clase principal que servirá para ir posicionando los distintos componentes que serán accedidos mediante la barra de menú.

Tabla 24 Estructura del Interfaz del Usuario

II.1.II. Diagrama de Componentes

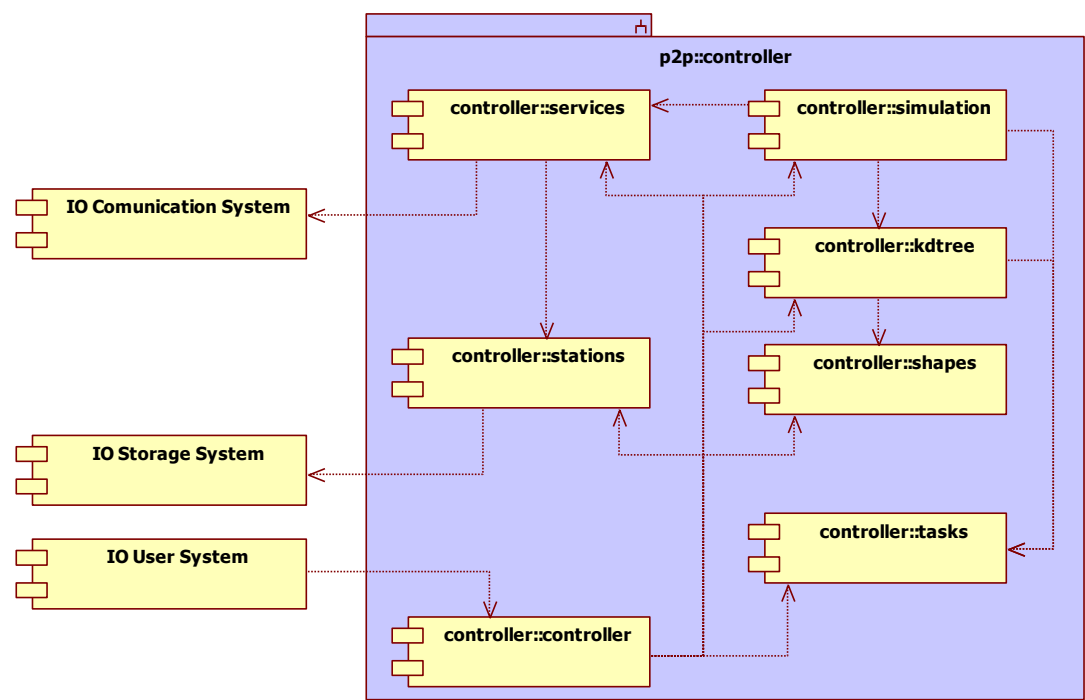


Ilustración 20 Diagrama de Componentes

Descripción
Aquí se muestran los componentes principales implicados en la aplicación así como sus interacciones.

Tabla 25 Diagrama de Componentes

II.1.III. Diagrama de Objetos

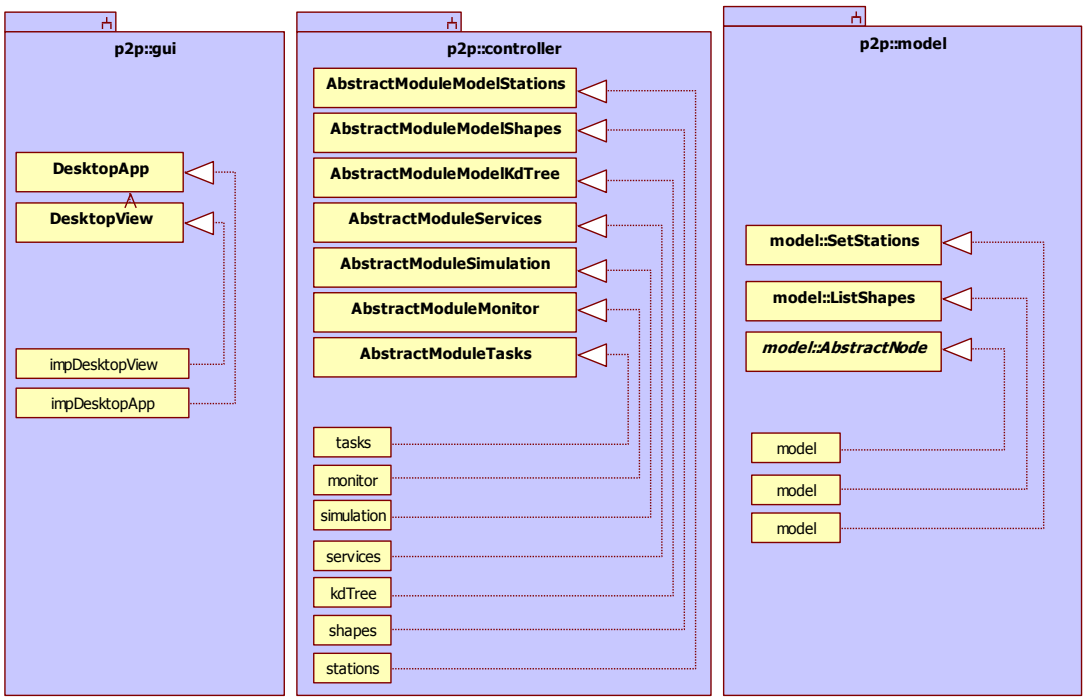


Ilustración 21 Diagrama de Objetos

Descripción

Los principales objetos que se encuentran instanciados durante toda vida de la aplicación, son los módulos de la aplicación. Los modelos básicos, y el interfaz de usuario de la aplicación (Con los componentes fijos que no figuran en este esquema).

Tabla 26 Diagrama de Objetos

II.I.IV. Diagrama de Despliegue

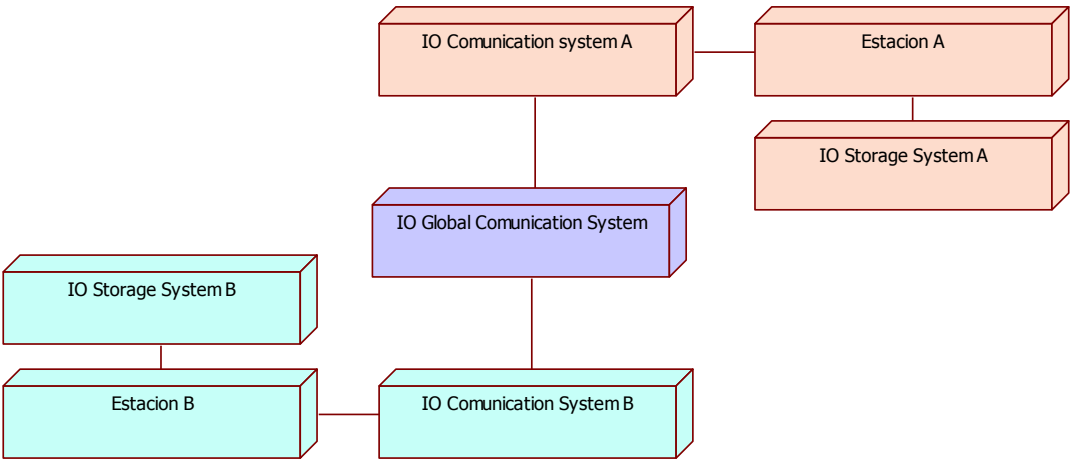


Ilustración 22 Diagrama de Despliegue

Descripción

El despliegue del sistema básico implica un sistema de almacenamiento por cada estación, un sistema de comunicación por cada estación, y un sistema global de comunicaciones.

Tabla 27 Diagrama de Despliegue

II.I.V. Diagrama de Paquetes

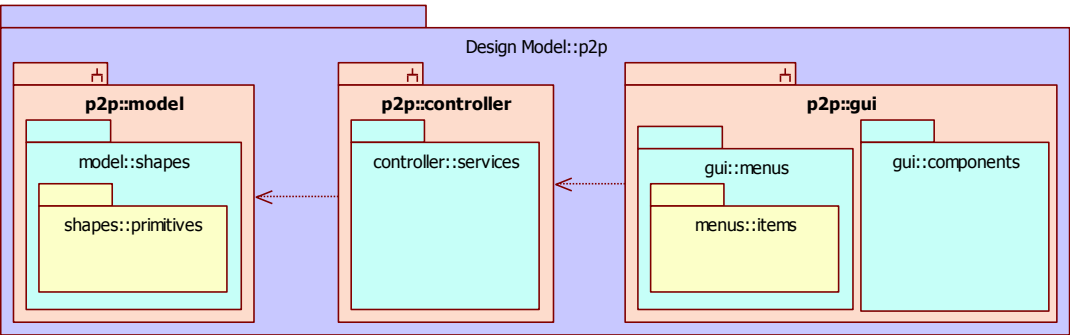


Ilustración 23 Diagrama de Paquetes

Descripción

En este esquema se puede observar que cada proyecto de la aplicación representa un subsistema el cual está representado por distintos paquetes en el código.

Tabla 28 Diagrama de Paquetes

II.II. Diagramas de Comportamiento

II.II.I. Diagrama de Casos de Uso

Casos de uso de aplicación

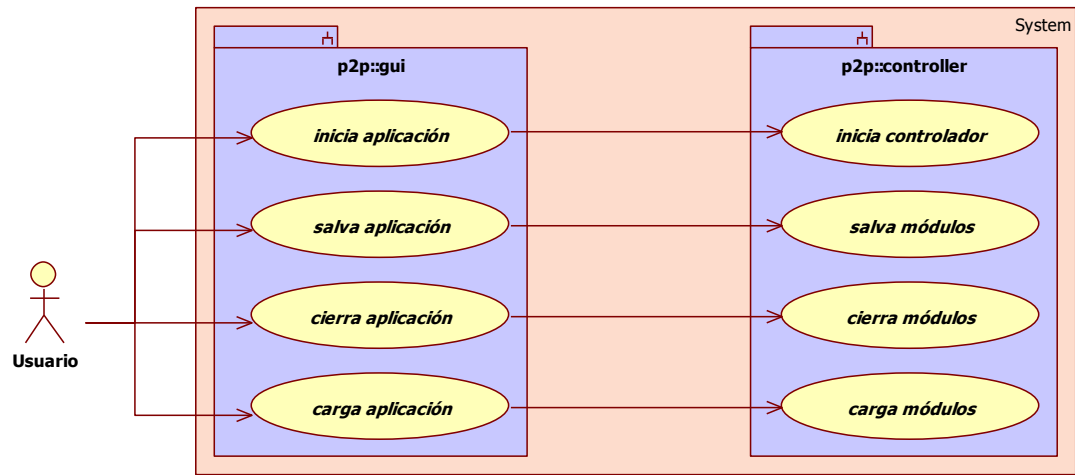


Ilustración 24 Casos de uso de aplicación

Descripción
Son los casos de uso que implican a toda la aplicación en general.
Tabla 29 Casos de uso de aplicación

Casos de uso relacionados con los servicios

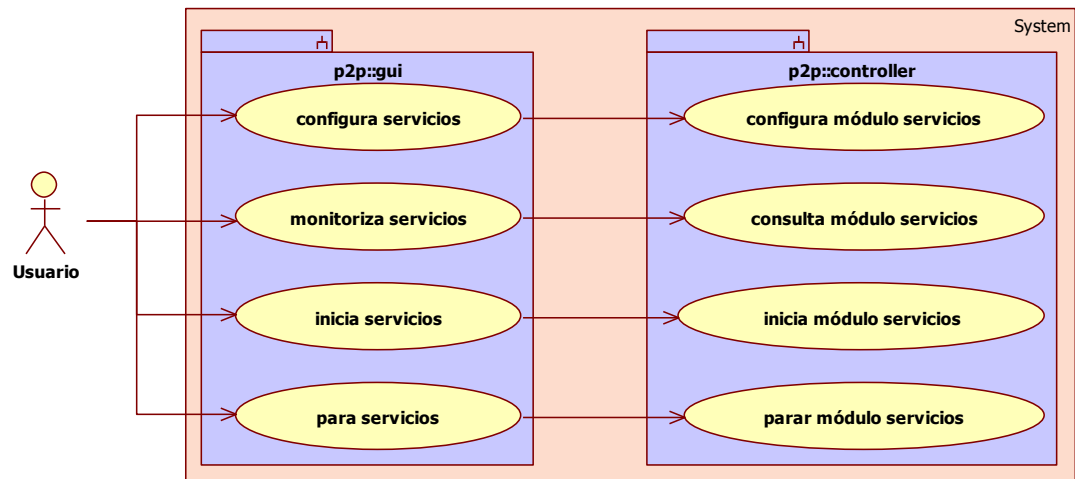


Ilustración 25 Casos de uso relacionados con los servicios

Descripción
Son los casos de uso que implican tareas relacionadas con la gestión de los servicios.
Tabla 30 Casos de uso relacionados con los servicios

Casos de uso relacionados con las estaciones de render

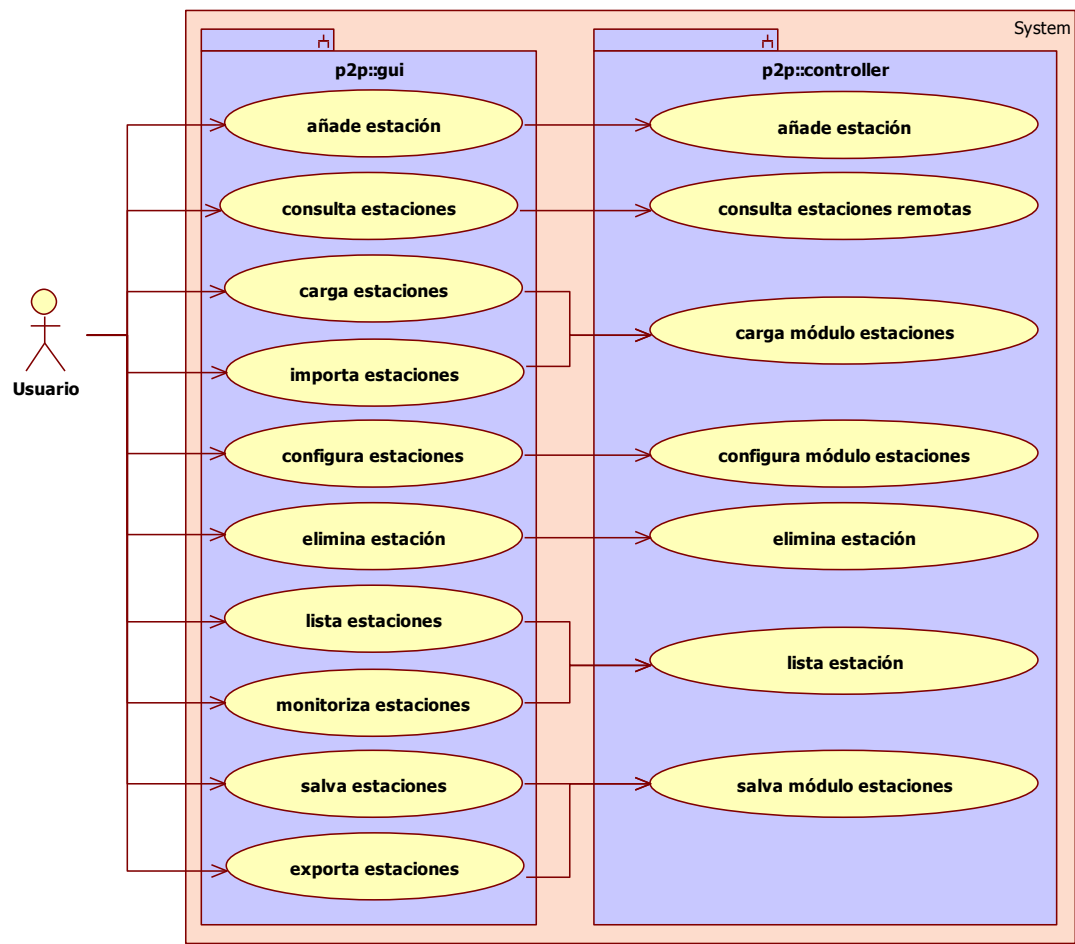


Ilustración 26 Casos de uso relacionados con las estaciones de render

Descripción
Son los casos de uso que implican el mantenimiento de las estaciones de render.

Tabla 31 Casos de uso relacionados con las estaciones de render

Casos de uso relacionados con las formas

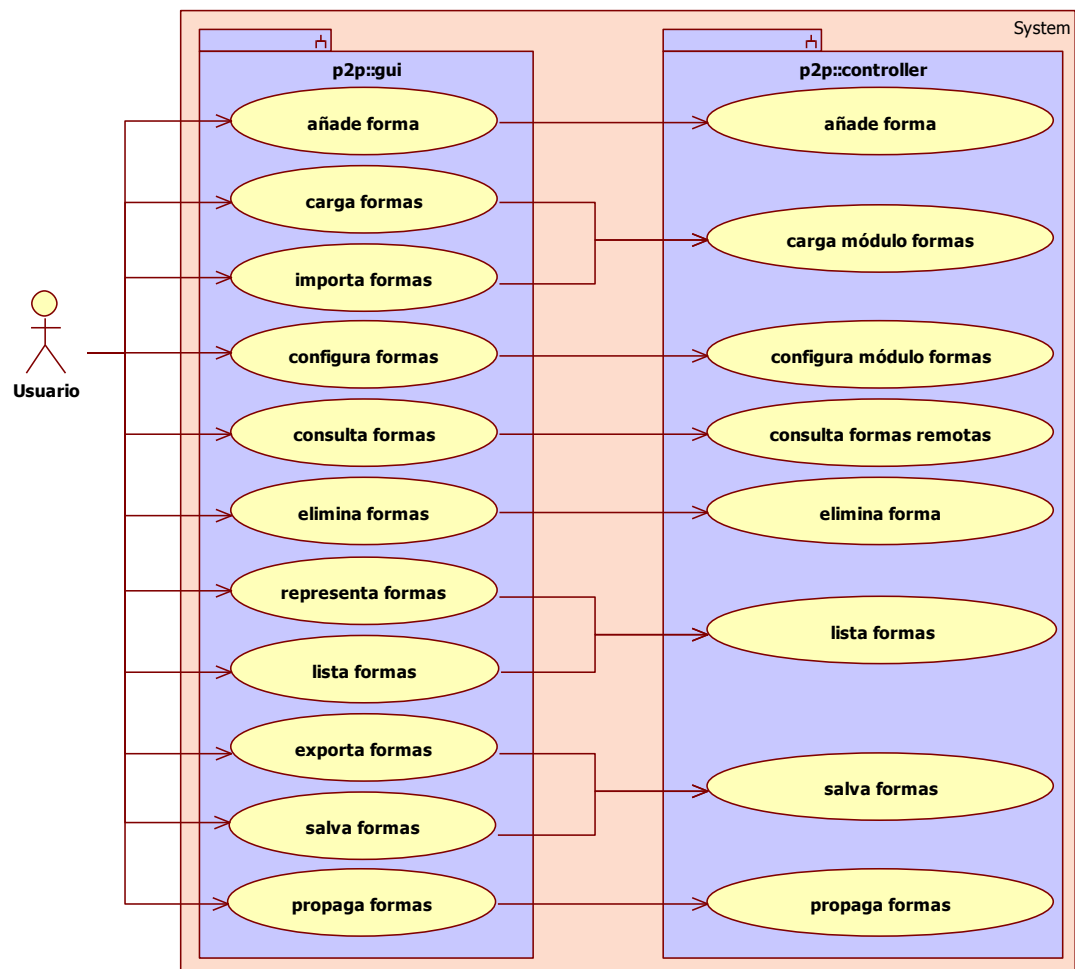


Ilustración 27 Casos de uso relacionados con las formas

Descripción
Son los casos de uso que implican el mantenimiento de las formas del espacio.
Tabla 32 Casos de uso relacionados con las formas

Casos de uso relacionados con el árbol kd

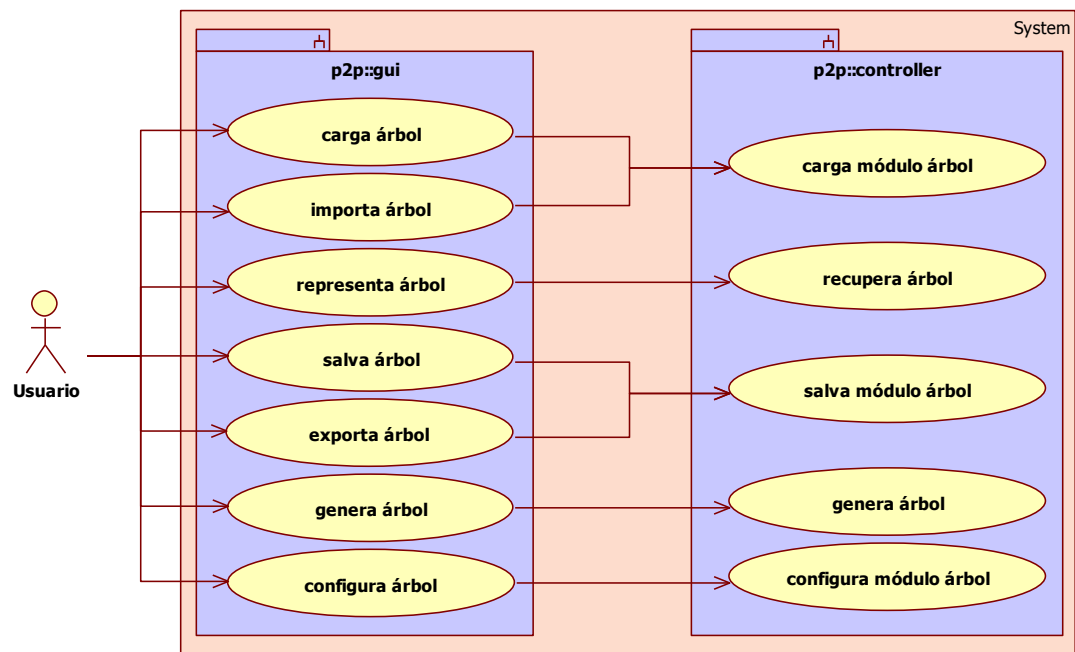


Ilustración 28 Casos de uso relacionados con el árbol kd

Descripción

Son los casos de uso que implican el mantenimiento del árbol kd.

Tabla 33 Casos de uso relacionados con el árbol kd

Casos de uso relacionados con la simulación

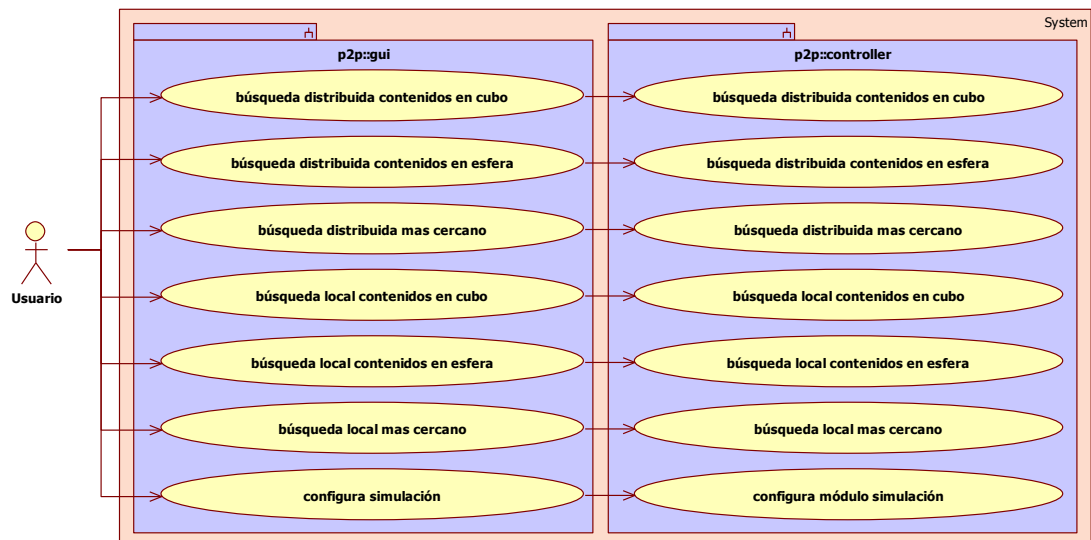


Ilustración 29 Casos de uso relacionados con la simulación

Descripción

Son los casos de uso que implican funcionalidades relacionadas con la simulación.

Tabla 34 Casos de uso relacionados con la simulación

II.II.II. Diagrama de Estados

Estados del servicio de red

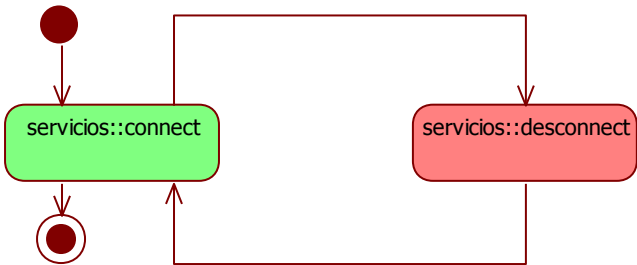


Ilustración 30 Estados del servicio de red

Descripción
Los servicios de red pueden estar conectados o desconectados.

Tabla 35 Estados del servicio de red

II.III. Diagramas de Interacción

II.III.I. Diagrama de Secuencia

Inicio de vida de un módulo cualquiera

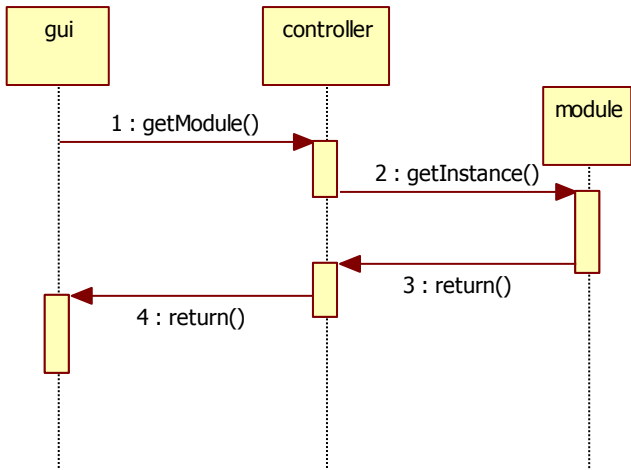


Ilustración 31 Inicio de vida de un módulo cualquiera

Descripción
El ciclo de vida de un módulo cualquiera se inicia nada más realizar una instanciación del mismo.

Tabla 36 Inicio de vida de un módulo cualquiera

Secuencia de cierre de la aplicación

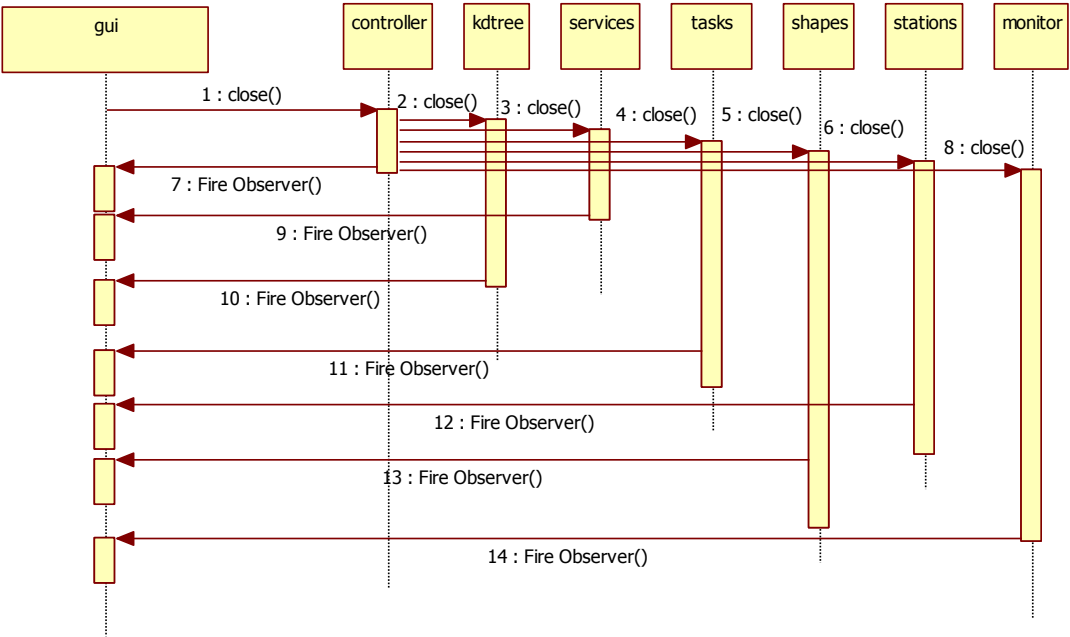


Ilustración 32 Secuencia de cierre de la aplicación

Descripción

Al cerrar la aplicación mandaremos una orden de cierre de los módulos de la aplicación. Estos a su vez harán las tareas necesarias para dar por cerrada la aplicación. Luego dispararan un evento que será capturado por la aplicación. Esta una vez a capturado el cierre de los módulos procederá a cerrar el interfaz de usuario.

Tabla 37 Secuencia de cierre de la aplicación

Ejecución de una funcionalidad cualquiera

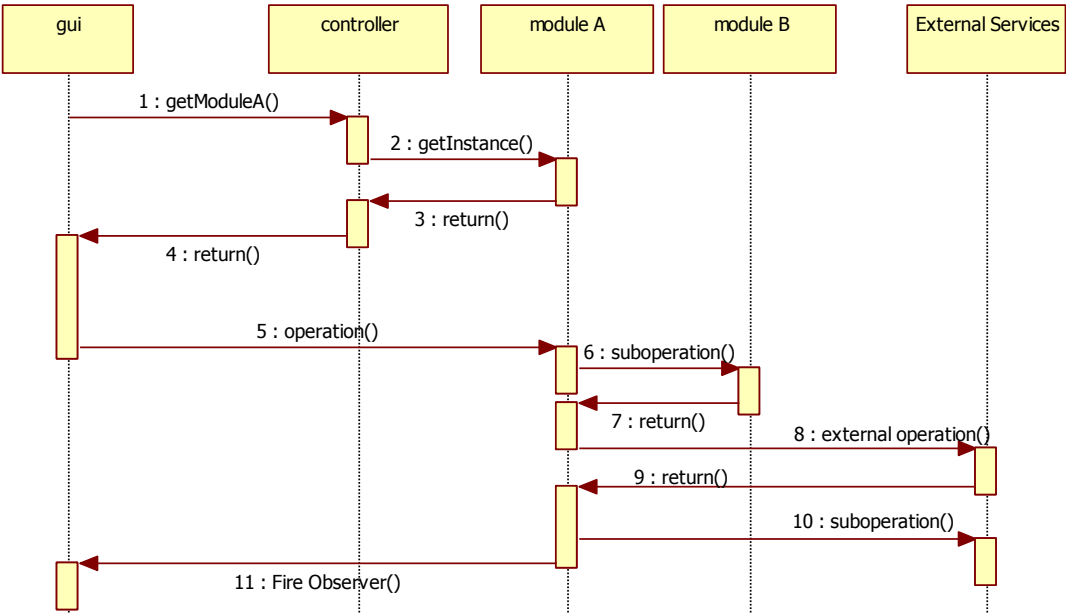


Ilustración 33 Ejecución de una funcionalidad cualquiera

Descripción

Cuando realizamos una operación cualquiera, se ven implicados uno o más módulos, además se hace uso de operaciones a servicios externos del núcleo como operaciones de disco o de red. Además las operaciones no son siempre síncronas, muchas veces la respuesta viene dada por la activación de un evento.

Tabla 38 Ejecución de una funcionalidad cualquiera

Anexo III. SIMULACIÓN

III.I. Resultados de los experimentos

III.I.I. Cálculo del tiempo base

El tiempo base, consiste en la experimentación de las distintas configuraciones con un numero de muestras nulo. Esto nos permite saber el tiempo perdido en operaciones distintas del cálculo.

Resultados del cálculo del tiempo base de la estación A sin uso de red

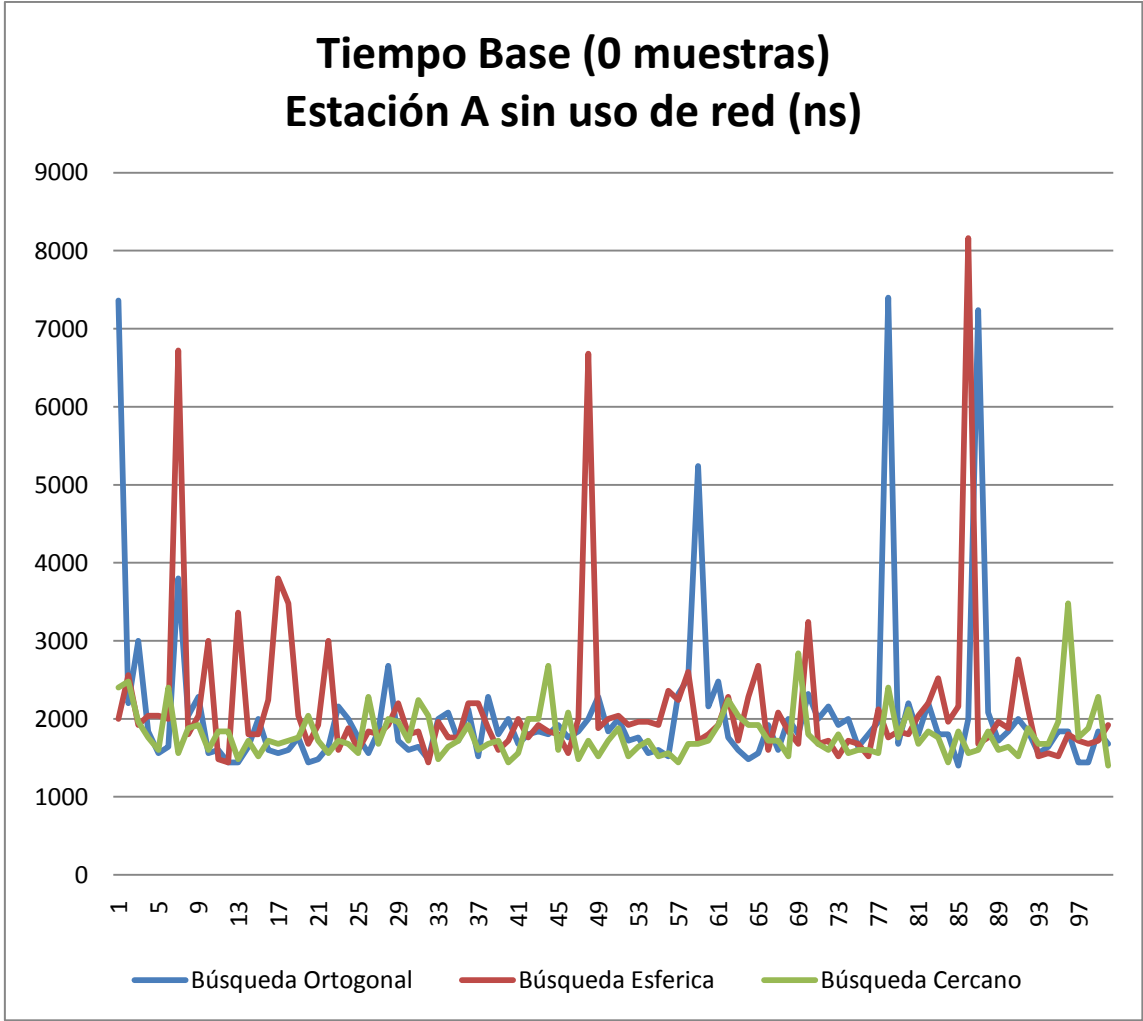


Ilustración 34 Tiempo Base Estación A sin uso de red

Descripción

Aquí se puede ver cómo serían 100 simulaciones de 0 muestras. Esto nos da una idea del tiempo perdido en tareas que no son de cálculo para la estación A (sin operaciones de red). Los picos los achacamos a momentos de ocupación de CPU empleados por otros programas ajenos a la simulación.

Tabla 39 Tiempo Base Estación A sin uso de red

Análisis del cálculo del tiempo base de la estación A sin uso de red

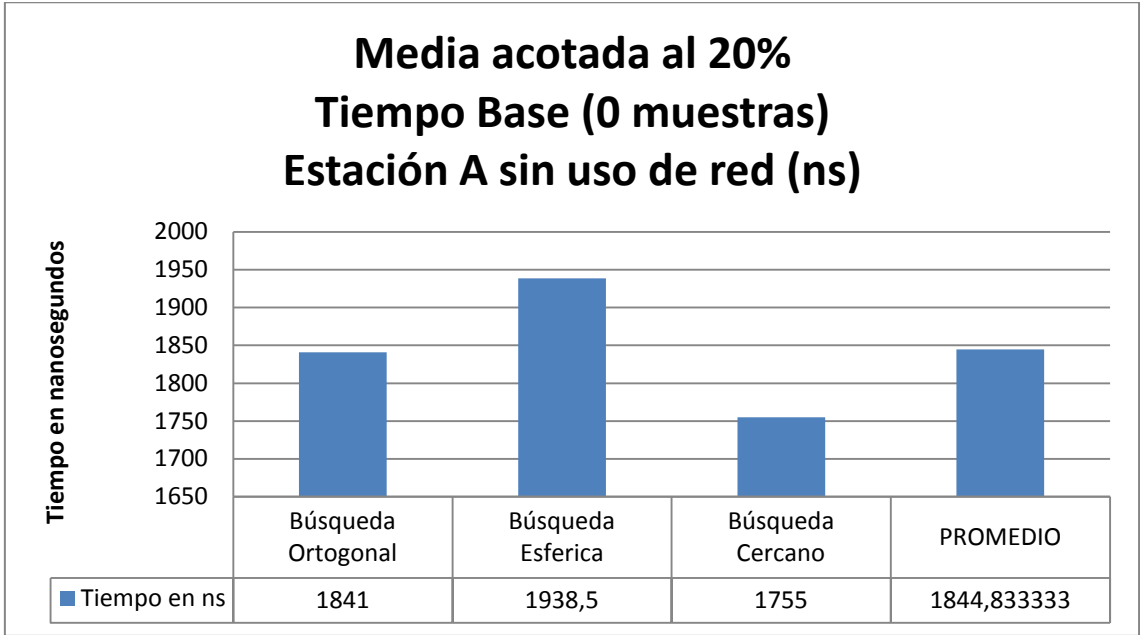


Ilustración 35 Análisis del cálculo del tiempo base de la estación A sin uso de red

Descripción

Para poder obtener el tiempo base lo que haremos es calcular la media acotada al 20%, con ello eliminamos los picos de CPU achacables a otros procesos externos a las pruebas. Como podemos observar las tres búsquedas tienen una media acotada similar, alrededor de los 1800 nanosegundos, esto es debido a que los tiempos aquí reflejados son debidos a operaciones no relacionadas con los algoritmos.

Tabla 40 Análisis del cálculo del tiempo base de la estación A sin uso de red

Resultados del cálculo del tiempo base de la estación A con uso de red en local

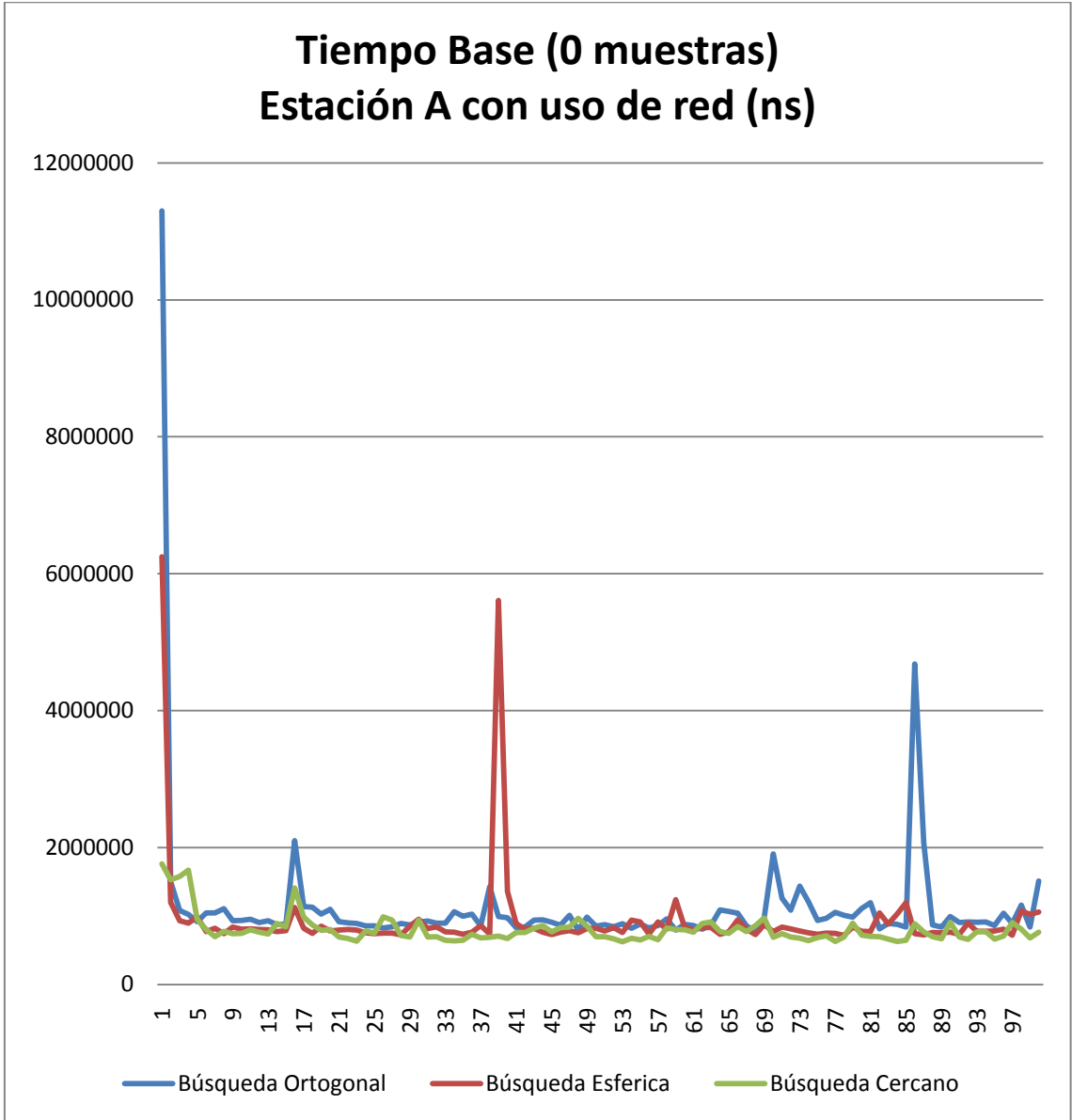


Ilustración 36 Tiempo base de la estación A con uso de red en local

Descripción

Aquí se puede ver cómo serían 100 simulaciones de 0 muestras. Esto nos da una idea del tiempo perdido en tareas que no son de cálculo para la estación A (con operaciones de red). Los picos los achacamos a momentos de ocupación de CPU empleados por otros programas ajenos a la simulación y ocupación de la red por parte de otras estaciones ajenas a la experimentación. Además hay un pico muy significativo al principio de la simulación, esto es debido a que el sistema se configura los protocolos de red, en un entorno de uso intensivo es de esperar que se llegue a un comportamiento estable de la red.

Tabla 41 Tiempo base de la estación A con uso de red en local

Análisis del cálculo del tiempo base de la estación A con uso de red en local

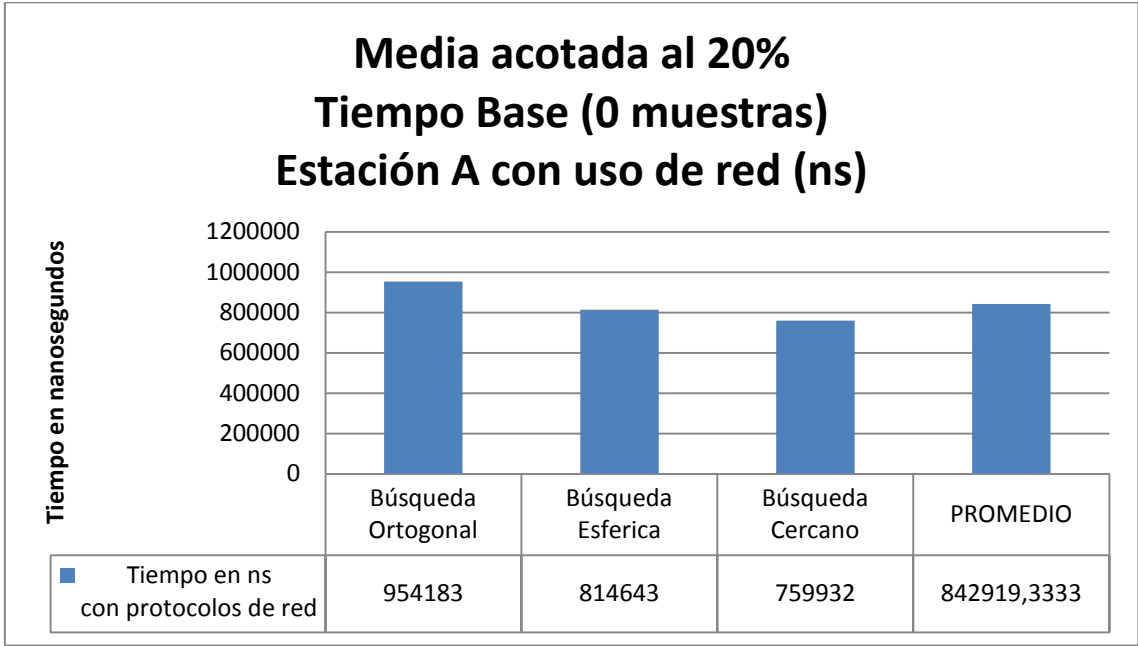


Ilustración 37 Tiempo base de la estación A con uso de red en local

Descripción

Para poder obtener el tiempo base lo que haremos es calcular la media acotada al 20%, con ello eliminamos los picos de CPU achacables a otros procesos externos a las pruebas y los picos de inicialización de los protocolos de red. Como podemos observar las tres búsquedas tienen una media acotada similar, alrededor de los 842000 nanosegundos, esto es debido a que los tiempos aquí reflejados son debidos a operaciones no relacionadas con los algoritmos.

Tabla 42 Tiempo base de la estación A con uso de red en local

Análisis del cálculo del tiempo base de la estación A con uso de red en local y sin uso de red

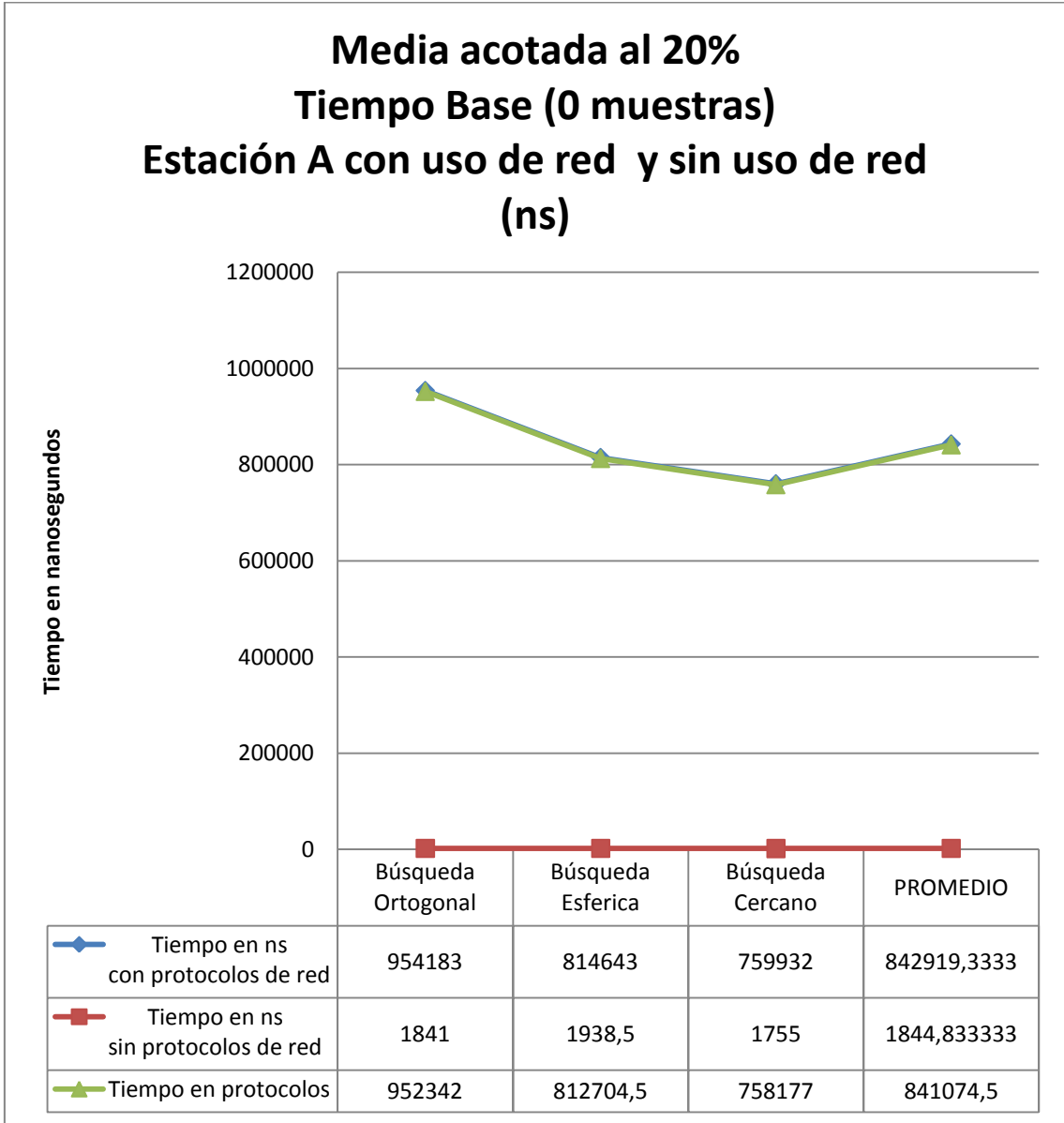


Ilustración 38 Tiempo base de la estación A con uso de red en local y sin uso de red

Descripción

Si comparamos los tiempos anteriores descubriremos que el tiempo perdido en protocolos de red es bastante elevado.

$$\begin{array}{r} 842919,3333 \text{ ns} \\ - 1844,8333 \text{ ns} \\ \hline 841074,499967 \text{ ns} \end{array}$$

Como podemos observar el tiempo perdido en los protocolos de red hace insignificante el tiempo perdido en otras operaciones, en la grafica casi se solapan al ser tan próximos.

Tabla 43 Tiempo base de la estación A con uso de red en local y sin uso de red

Resultados del cálculo del tiempo base de la estación B con uso de protocolos de red en configuración remota

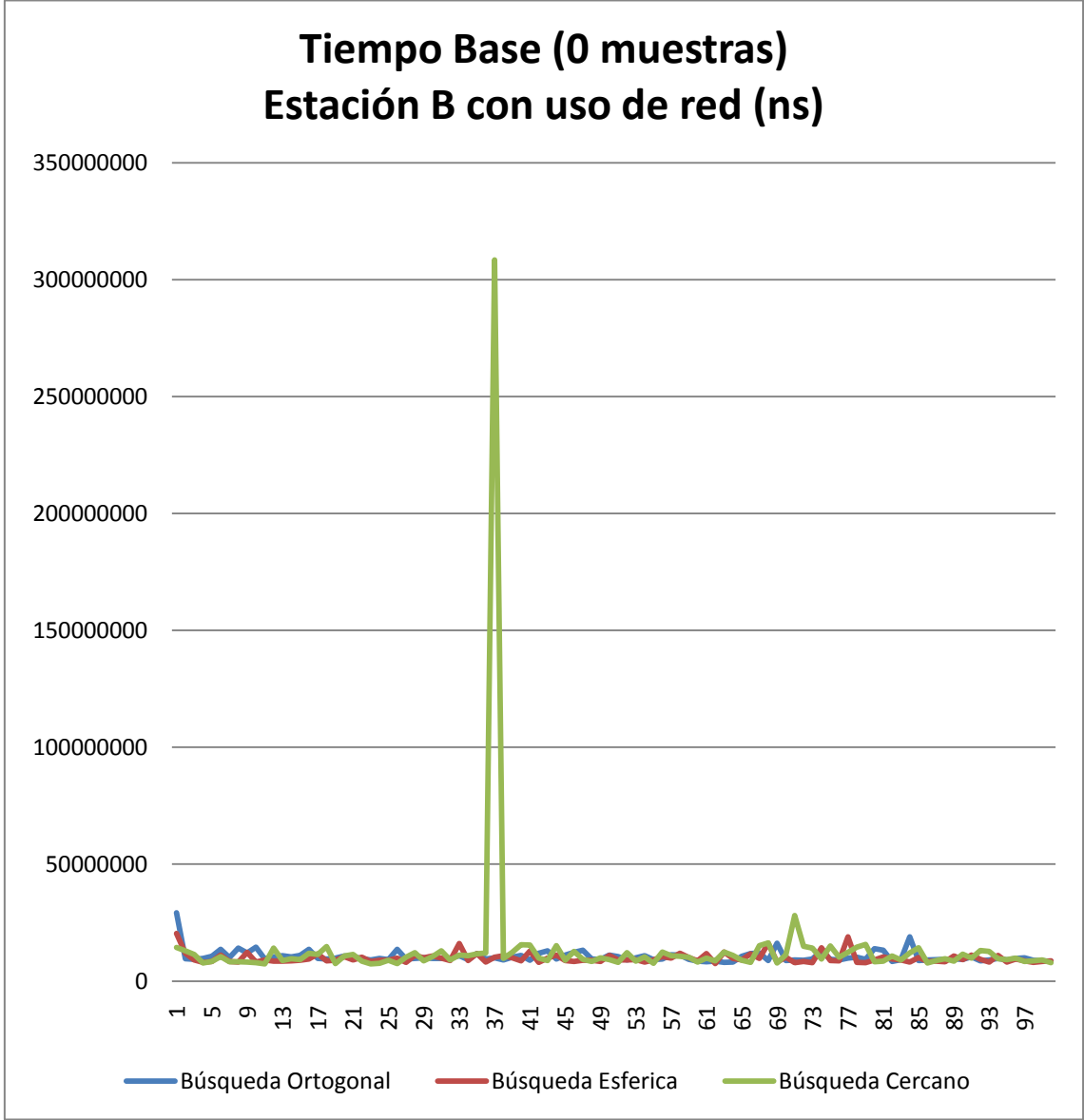


Ilustración 39 Tiempo base de la estación B con uso de protocolos de red en configuración remota

Descripción

Aquí se puede ver cómo serían 100 simulaciones de 0 muestras. Esto nos da una idea del tiempo perdido en tareas que no son de cálculo para la estación A (con operaciones de red y enrutamiento por un enrutador). Los picos los achacamos a momentos de ocupación de CPU empleados por otros programas ajenos a la simulación y ocupación de la red por parte de otras estaciones ajenas a la experimentación.

Tabla 44 Tiempo base de la estación B con uso de protocolos de red en configuración remota

Análisis del cálculo del tiempo base de la estación B con uso de protocolos de red en configuración remota

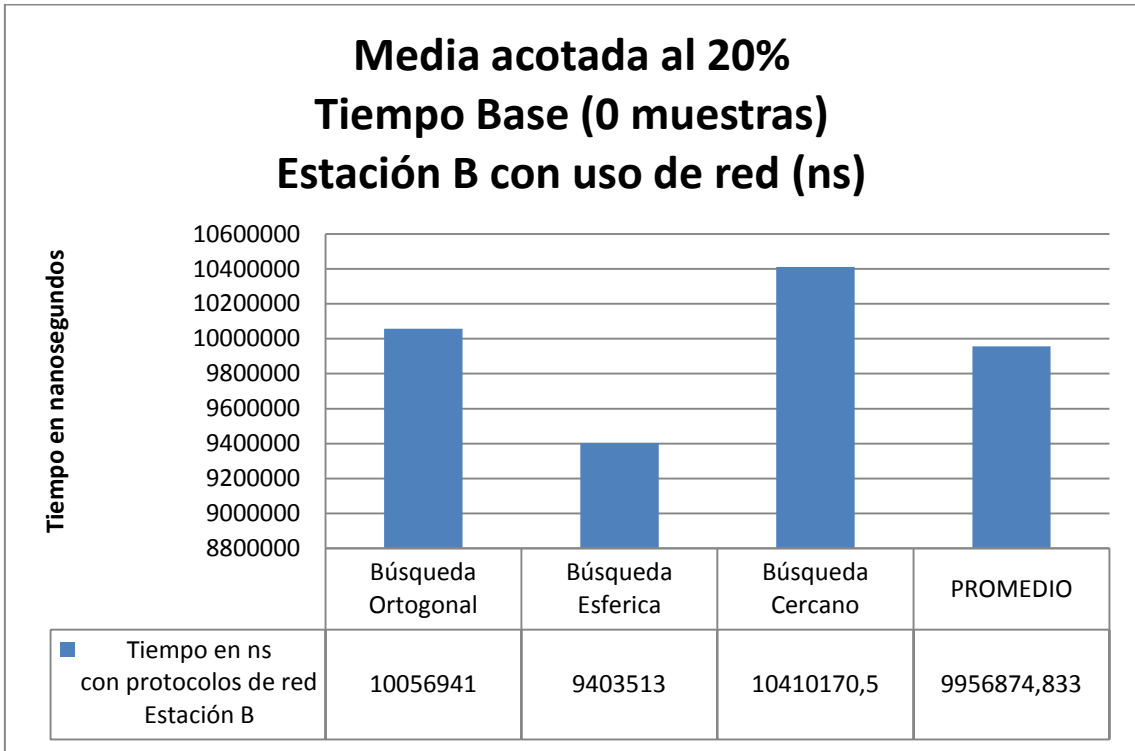


Ilustración 40 Tiempo base de la estación B con uso de protocolos de red en configuración remota

Descripción

Para poder obtener el tiempo base lo que haremos es calcular la media acotada al 20%, con ello eliminamos los picos de CPU achacables a otros procesos externos a las pruebas y los picos de inicialización de los protocolos de red y mapeo de estaciones por parte del enrutador. Como podemos observar las tres búsquedas tienen una media acotada similar, alrededor de los 9956000 nanosegundos, esto es debido a que los tiempos aquí reflejados son debidos a operaciones no relacionadas con los algoritmos.

Tabla 45 Tiempo base de la estación B con uso de protocolos de red en configuración remota

Análisis del cálculo del tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto

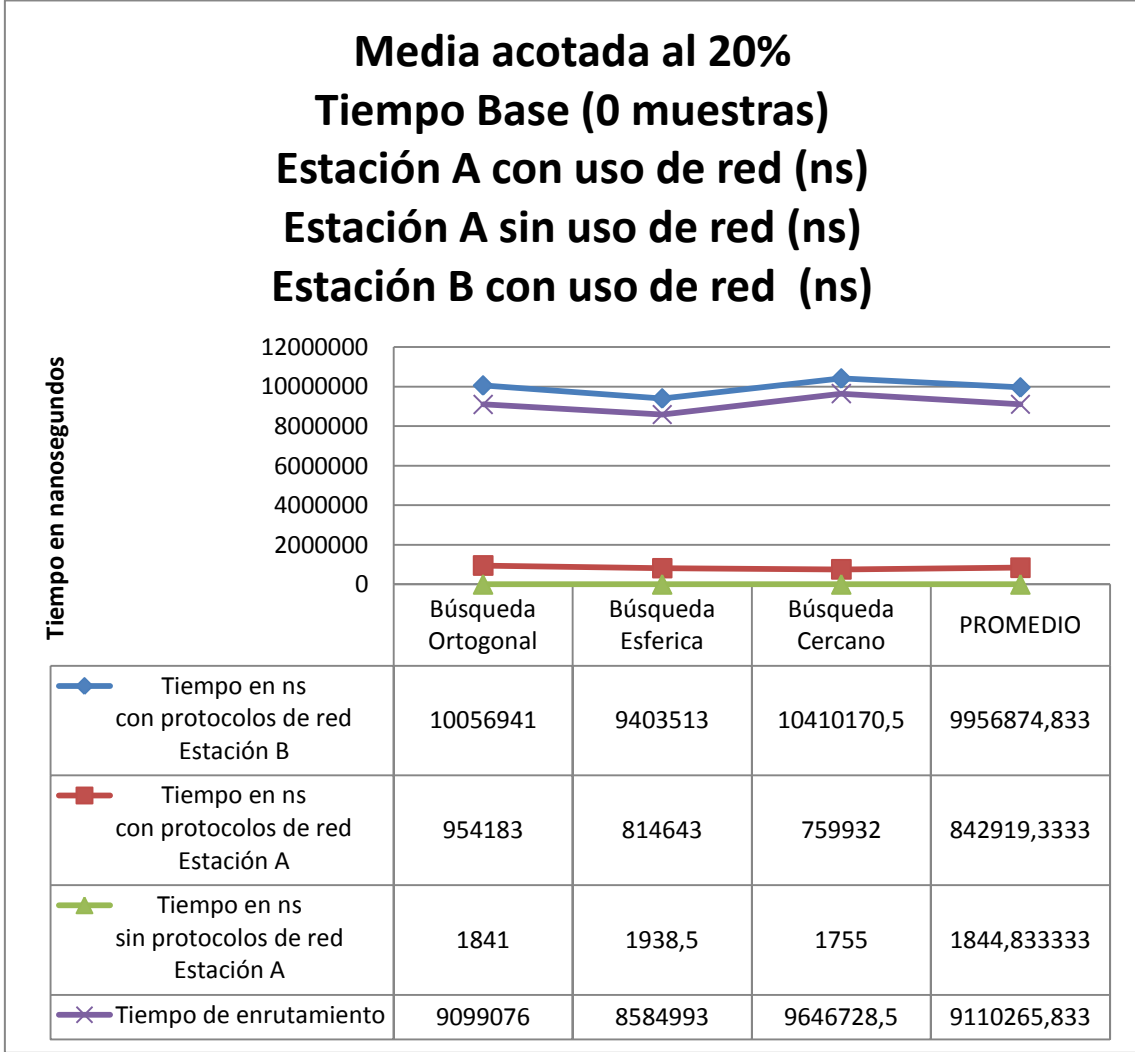


Ilustración 41 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto

Descripción

Si comparamos los tiempos anteriores descubriremos que el tiempo perdido en protocolos de red es bastante elevado.

9956874,833 ns

– 842919,3333 ns

+ –1844,833333 ns

–1844,833333 ns

9110265,833 ns

Como podemos observar el tiempo perdido en enrutamiento de red hace insignificante el tiempo perdido en otras operaciones como en configuración de protocolos..

Tabla 46 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto

Resultados del cálculo del tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

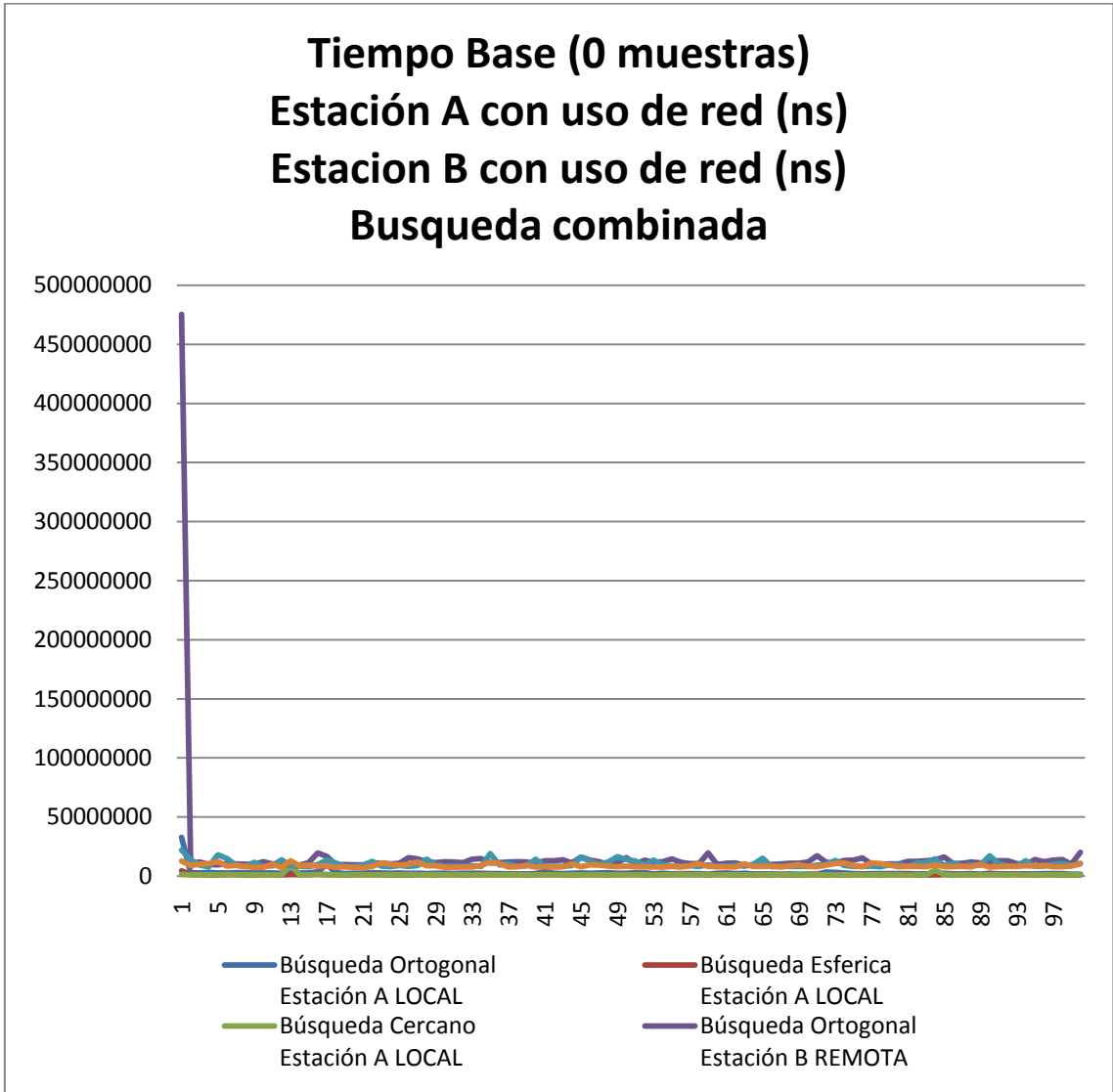


Ilustración 42 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

Descripción

Aquí se puede ver cómo serían 100 simulaciones de 0 muestras. Esto nos da una idea del tiempo perdido en tareas que no son de cálculo para la estación A (con operaciones de red y enrutamiento por un enrutador). Los picos los achacamos a momentos de ocupación de CPU empleados por otros programas ajenos a la simulación y ocupación de la red por parte de otras estaciones ajenas a la experimentación.

Tabla 47 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

Análisis del cálculo del tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

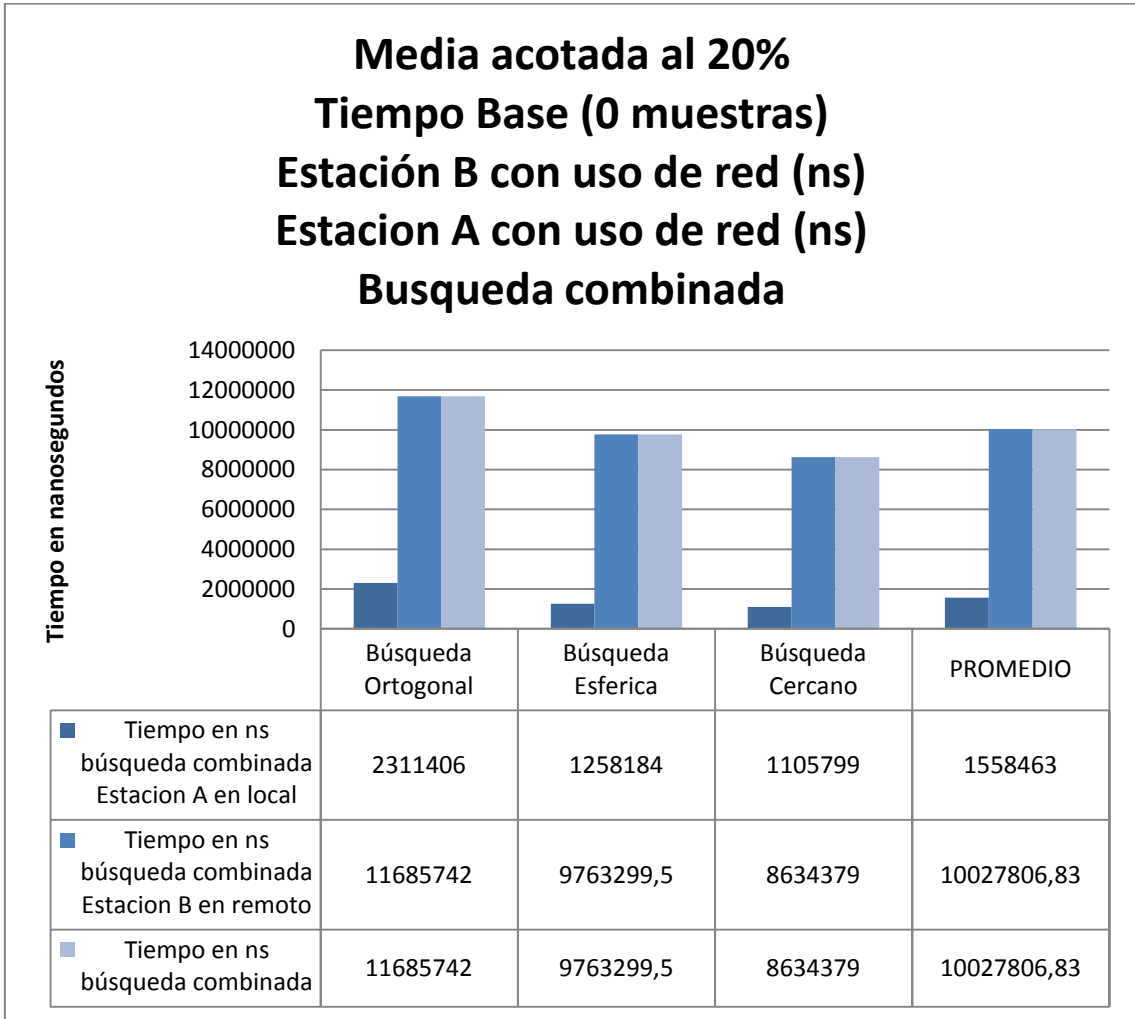


Ilustración 43 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

Descripción

Para poder obtener el tiempo base lo que haremos es calcular la media acotada al 20%, con ello eliminamos los picos de CPU achacables a otros procesos externos a las pruebas y los picos de inicialización de los protocolos de red y mapeo de estaciones por parte del enrutador. Como podemos observar las tres búsquedas tienen una media acotada similar, alrededor de los 10000000 nanosegundos, esto es debido a que los tiempos aquí reflejados son debidos a operaciones no relacionadas con los algoritmos. También he de explicar que el tiempo de búsqueda se limita a la estación más lenta.

$$\max\left(\frac{1558463\text{ ns}}{10027806,83\text{ ns}}\right) = 10027806,83\text{ ns}$$

Tabla 48 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

Análisis del cálculo del tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto y la búsqueda combinada

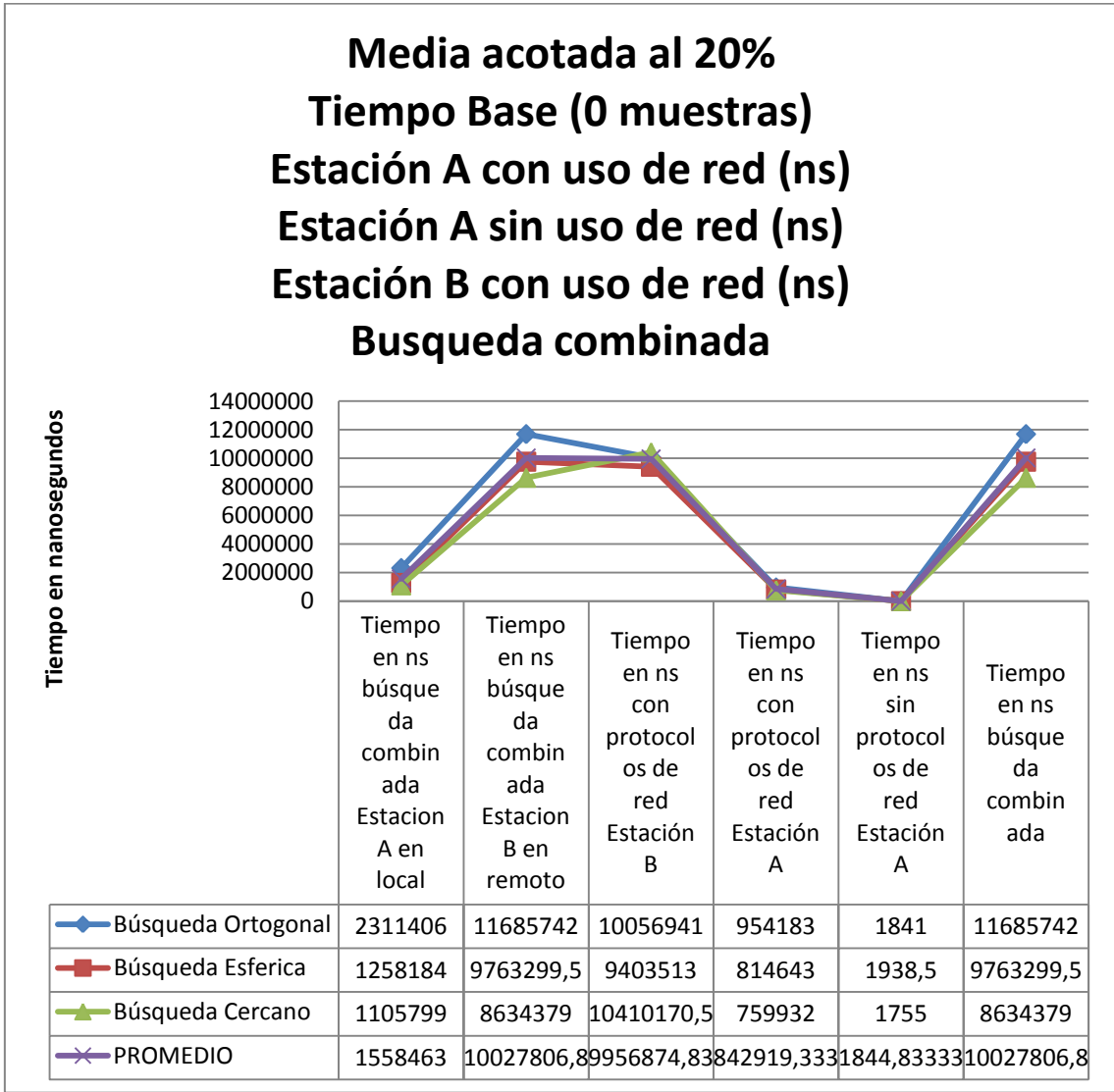


Ilustración 44 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto y la búsqueda combinada

Descripción

Podemos observar que en la búsqueda combinado los resultados son similares a la operación de los mismos por separado. Por ejemplo el tiempo de búsqueda promedio de la estación A es similar en una búsqueda combinar al de la ejecución por separado, dada la escala manejada de tiempos:

$1558463\text{ ns} \cong 842919,333\text{ ns}$ (ver grafica para ver similitud)

Lo mismo pasa para la búsqueda de la estación B:

$10027806,8\text{ ns} \cong 9956874,83\text{ ns}$ (ver grafica para ver similitud)

Tabla 49 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto y la búsqueda combinada

Tiempo de procesamiento de los algoritmos

Análisis del algoritmo del más cercano

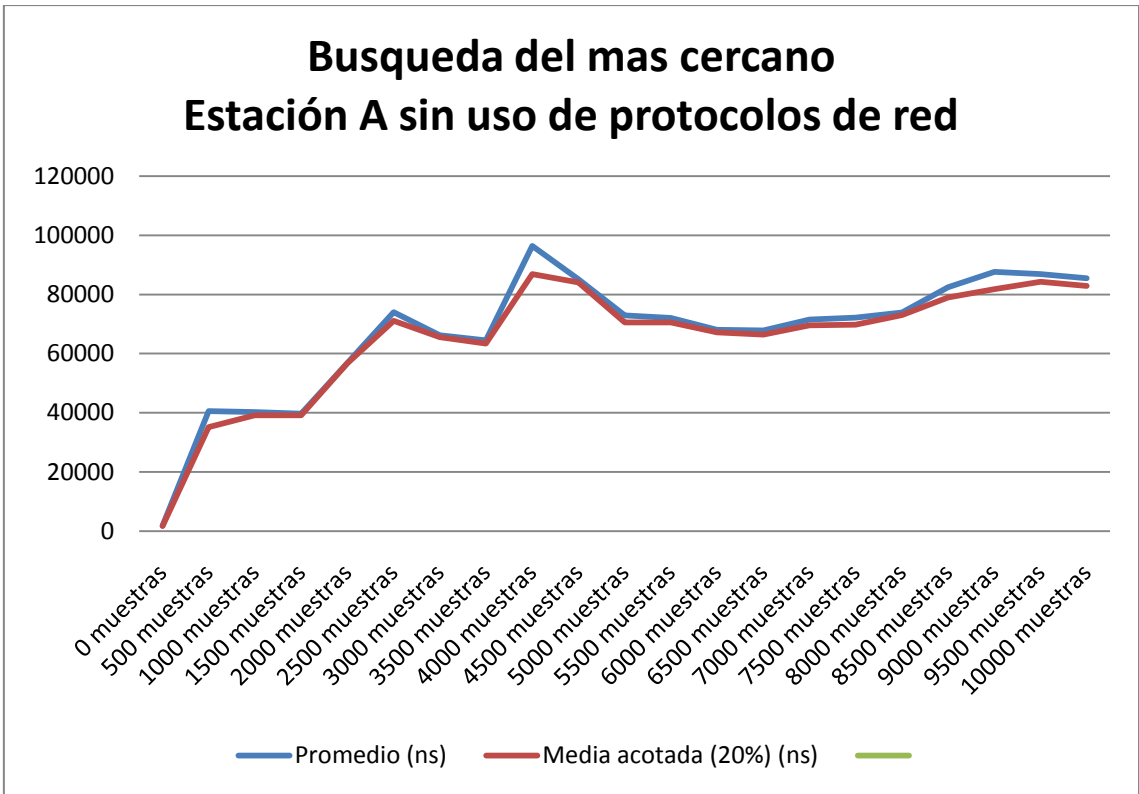


Ilustración 45 Análisis del algoritmo del más cercano

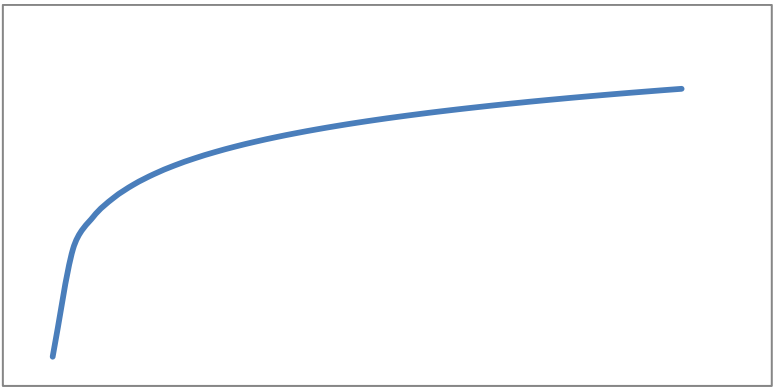


Ilustración Tendencia de la simulación O(Log N)

Descripción

Se puede ver una clara tendencia a aumentar el tiempo de proceso en una escala $O(\log N)$. Se puede observar que la media se dispara en ciertas partes de la gráfica. Esto es debido a momentos puntuales en que la CPU está ocupada por otras aplicaciones no relacionadas con la simulación. Sin embargo esto es así porque en la simulación se evaluaron tan solo puntos, si evaluamos volúmenes esto variara, siendo en el mejor de los casos de escala logarítmica y en el peor de los casos lineal.

Tabla 50 Simulación local de la búsqueda del más cercano Máquina 1

Análisis del algoritmo de búsqueda ortogonal

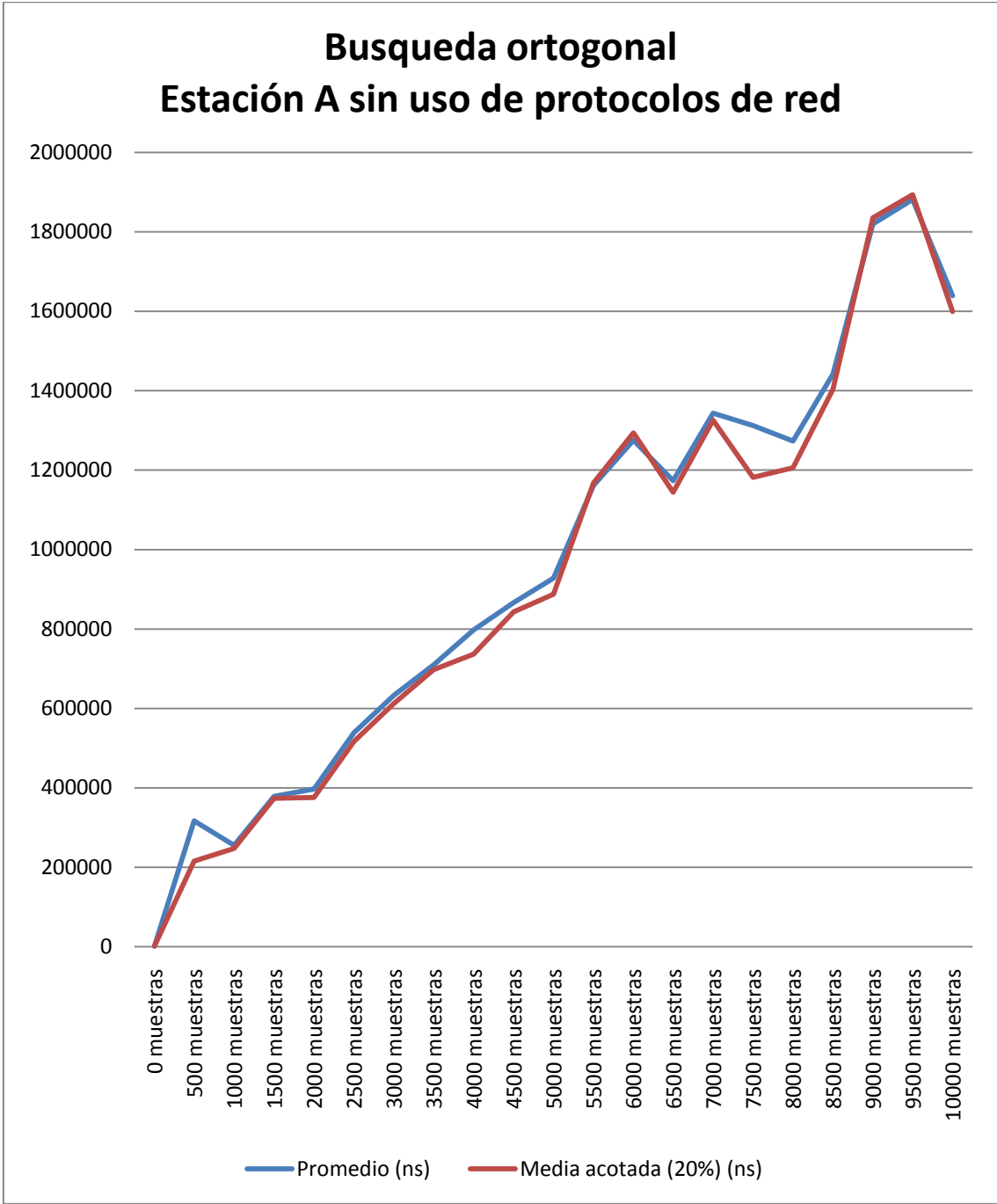


Ilustración 46 Análisis del algoritmo de búsqueda ortogonal

Descripción

Debido al algoritmo llevado en el recorrido del árbol, la tendencia variara de una búsqueda totalmente lineal a una búsqueda logarítmica. Esto es debido a que en ciertas partes las condiciones del algoritmo obligan a evaluar ambas ramas del árbol. Por lo tanto en el mejor de los casos el algoritmo tendrá una tendencia logarítmica y en el peor de los casos lineal. En este caso los puntos que sus proyecciones intersecciones con la esfera, obligaran a evaluar ambas ramas.

Tabla 51 Análisis del algoritmo de búsqueda ortogonal

Análisis del algoritmo de búsqueda esférica

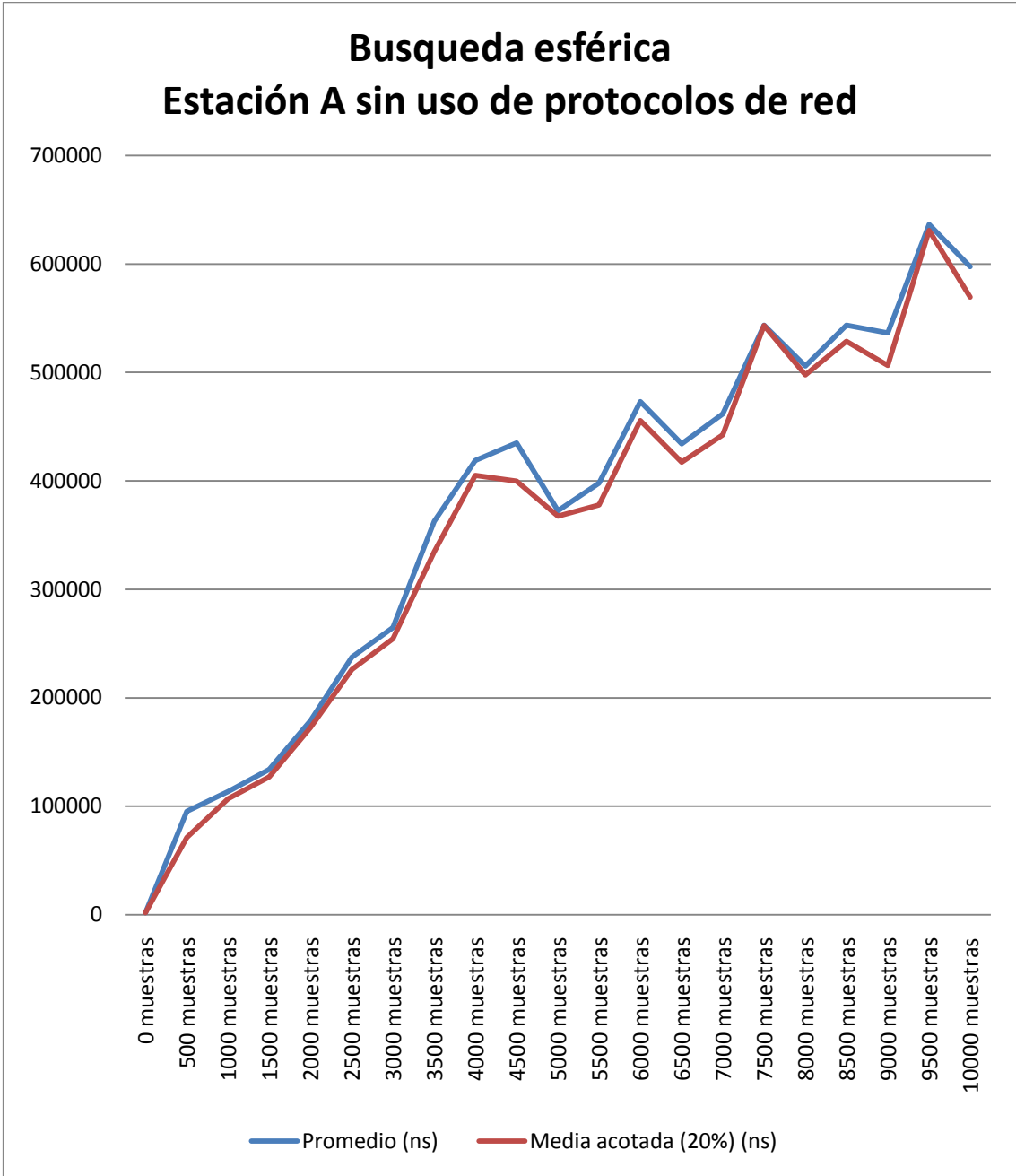


Ilustración 47 Análisis del algoritmo de búsqueda esférica

Descripción
Debido al algoritmo llevado en el recorrido del árbol, la tendencia variara de una búsqueda totalmente lineal a una búsqueda logarítmica. Esto es debido a que en ciertas partes las condiciones del algoritmo obligan a evaluar ambas ramas del árbol. Por lo tanto en el mejor de los casos el algoritmo tendrá una tendencia logarítmica y en el peor de los casos lineal. Los picos observados en la grafica son debidos a momentos de ocupación de la CPU por procesos externos a la simulación. En este caso los puntos que sus proyecciones intersecciones con el cubo, obligaran a evaluar ambas ramas.

Tabla 52 Análisis del algoritmo de búsqueda esférica

Anexo IV. ALGORITMOS

IV.I. Generación del kd tree de una nube de puntos

IV.I.I. Explicación

Este algoritmo elabora la creación de un árbol kd en el que se indexará una nube de puntos.

IV.I.II. Desarrollo

Primero cogeremos todas las coordenadas y las ordenaremos según su valor de la proyección sobre el eje X. En caso de que coincidan pasaremos a ordenarlos por la proyección sobre el eje Y, si también coincide, la ordenaremos también por la proyección sobre el eje Z. Así tendremos el orden como {XYZ}, {YZX} y {ZXY}. Por ejemplo:

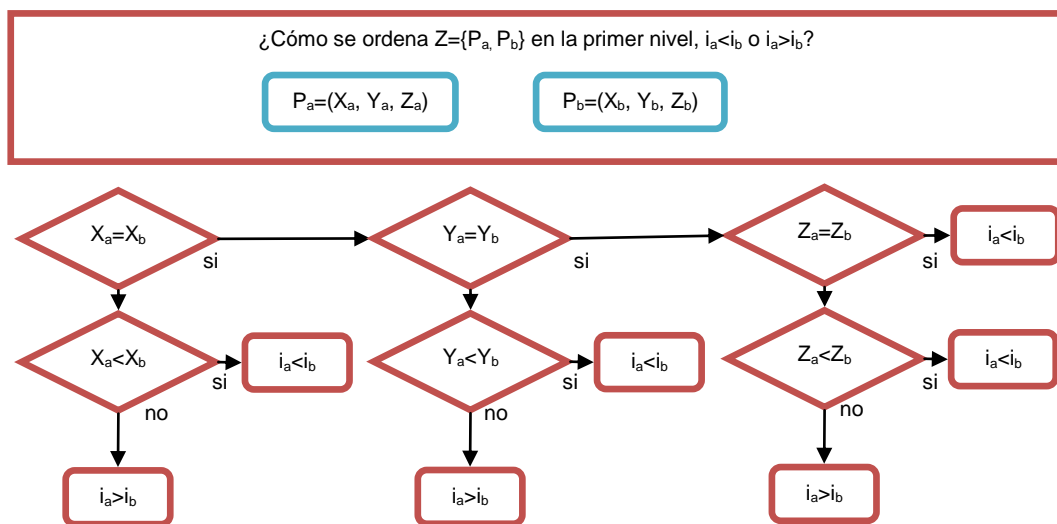


Ilustración 48 Algoritmo de ordenación

$K = \{(0,6,1), (2,1,6), (6,8,2), (3,1,4), (2,7,8), (4,3,4), (6,5,0), (3,7,9)\}$

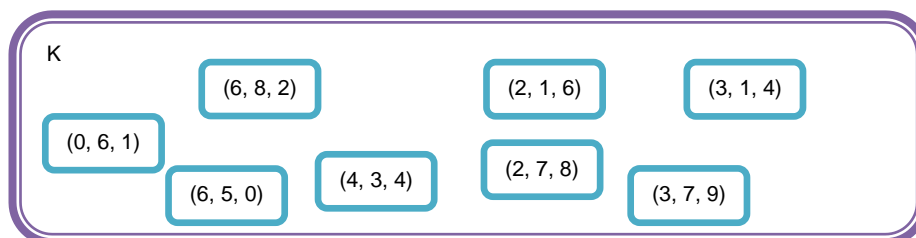


Ilustración 49 Coordenadas de prueba

Una vez hemos ordenado el listado de coordenadas, seleccionaremos la coordenada que se halla en el centro del listado, la mediana, en caso de ser el número de elementos impar:



Ilustración 50 Ordenación 1 paso

En caso de ser impar habrá dos coordenadas centrales, con lo que la mediana se situará entre estos, por lo tanto se seleccionara la coordenada con un índice menor a la mediana:

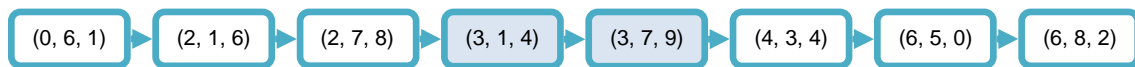


Ilustración 51 Ordenación 2 paso

Esta coordenada separará el árbol en un nodo y dos ramas, en cada una de las cuales, estarán las coordenadas de la lista de la izquierda de dicho nodo y a la derecha de dicho nodo respectivamente.

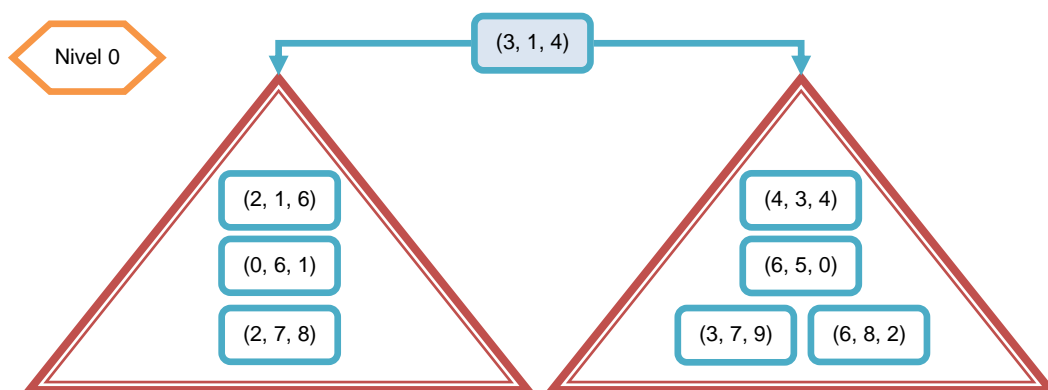


Ilustración 52 Construcción de árbol paso 1

Pasamos a realizar el mismo algoritmo con la rama de la izquierda ordenando los puntos por la coordenada Y.

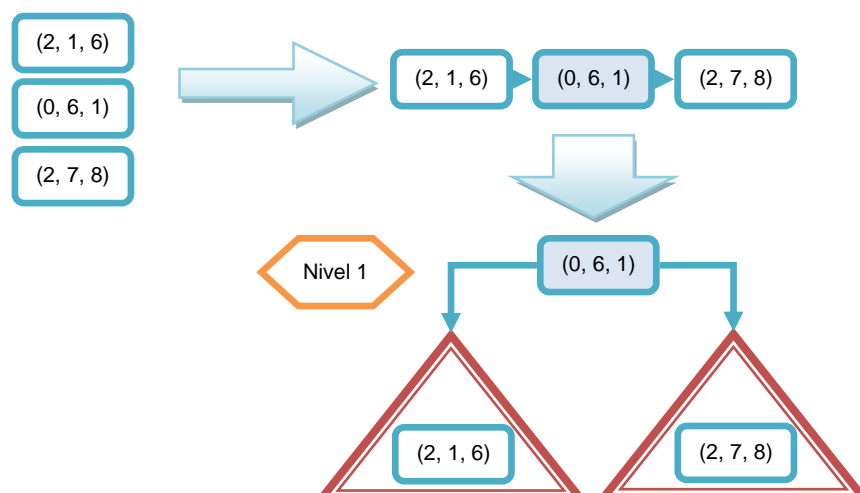


Ilustración 53 Construcción de árbol paso 2

Hacemos el mismo proceso para la rama derecha del árbol:

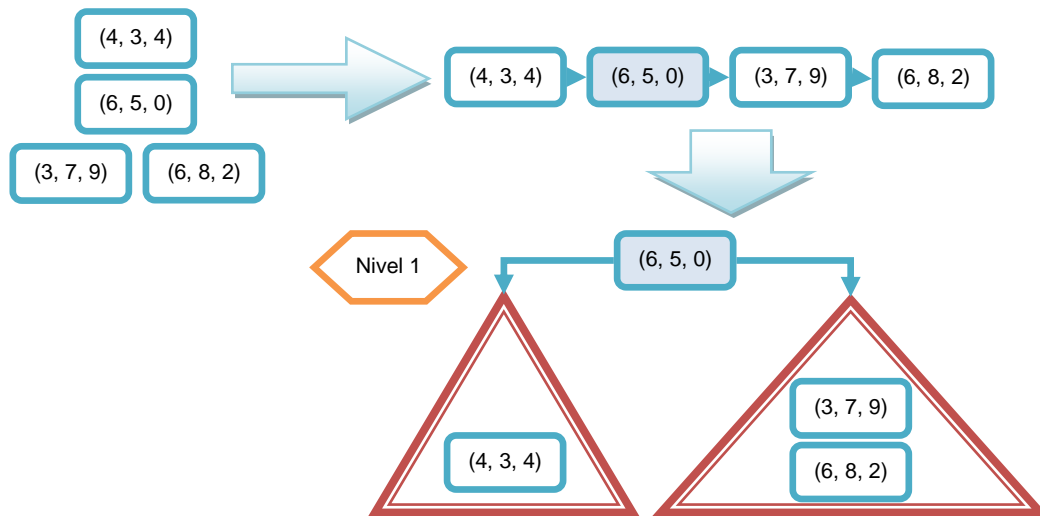


Ilustración 54 Construcción de árbol paso 3

Y continuamos con el mismo algoritmo para los subárboles generados alternando la coordenada, en caso de tan solo haber un elemento lo asignamos a la rama izquierda de su padre.

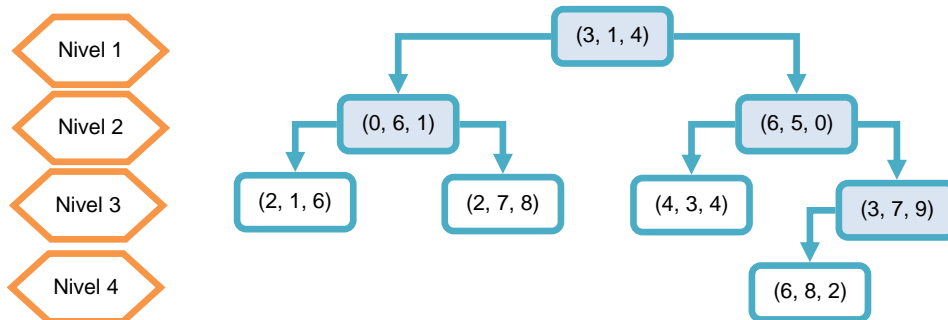


Ilustración 55 Construcción de árbol paso 4

Cabe mencionar que la generación del árbol se puede paralelizar al generar las ramas por separado.

IV.I.III. Pseudocódigo de la construcción de un kdtree en una nube de puntos

El algoritmo de generación en pseudocódigo sería:

```
INICIO CONSTRUIR (LISTADO): KDTREE
  RETORNA CONSTRUIR (LISTADO, 0)
FIN CONSTRUIR
```

```
INICIO CONSTRUIR (LISTADO, NIVEL): KDTREE
  TAMAÑO = NUMERO_ELEMENTOS (LISTADO)
  SELECCIONAR TAMAÑO
  CASO 0
    RETORNA NULO
  CASO 1
    NODO = LISTADO [0]
    RETORNA NODO
  CASO 2
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [0]
    NODO.LEFT = LISTADO [1]
    RETORNA NODO
  EN OTRO CASO
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [TAMAÑO/2]
    PARALELIZAR
      NODO.LEFT= CONSTRUIR (LISTADO[0 .. TAMAÑO/2], NIVEL+1)
      NODO.RIGH= CONSTRUIR (LISTADO[(TAMAÑO/2)+1 .. TAMAÑO], NIVEL+1)
      ESPERAR FIN PARALELIZACION
    FIN PARALELIZAR
    RETORNA NODO
  FIN SELECCIONAR
FIN CONSTRUIR
```

```
INICIO ORDENA (LISTADO, NIVEL): LISTADO
  SELECCIONAR MODULO (3, NIVEL)
  CASO 1
    RETORNA QUICKSORT_POR_X_Y_Z (LISTADO)
  CASO 1
    RETORNA QUICKSORT_POR_Y_Z_X (LISTADO)
  CASO 2
    RETORNA QUICKSORT_POR_Z_X_Y (LISTADO)
  FIN SELECCIONAR
FIN ORDENA
```


IV.II. Generación del un kd tree con elementos con volumen

IV.II.I. Explicación

Este algoritmo elabora la creación de un árbol kd, el cual es utilizado para multitud de aplicaciones, por ser una solución óptima.

IV.II.II. Desarrollo

Para poder estudiar cómo podríamos distribuir en el kdtree los volúmenes vamos a estudiarlos más a fondo.

Hay que tener en cuenta que cuando nos referiremos a un volumen. Este ha de tener todas sus partes conectadas de él.

Así por ejemplo un toroide cumpliría con esta definición a pesar de tener un gran agujero en el interior. Pero una "i" en 3D no la cumpliría al no estar conectadas el punto con el cuerpo.

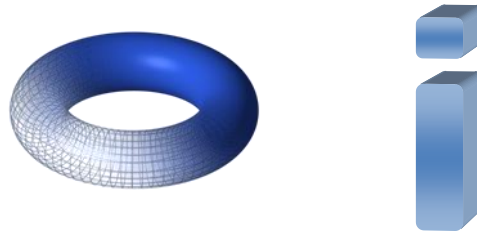


Ilustración 56 Volúmenes válidos

Si en cada nivel tenemos en cuenta un eje de coordenadas X, Y y Z llegamos a la conclusión de que si estudiamos el caso de un sola coordenada podremos extrapolarlo a las otras dos coordenadas.

En el caso de un punto cuando generamos el árbol tenemos en cuenta la proyección del punto sobre el eje.

Si $P = (x, y, z) \Rightarrow$ es un punto en el espacio

$P'_x = (x, 0, 0) \Rightarrow$ es la proyección del punto sobre el eje x

Que gráficamente se traduciría:

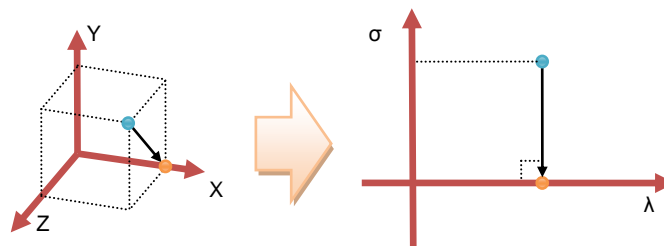


Ilustración 57 Proyección del punto sobre el eje X

Si tenemos un volumen, la proyección del mismo tendrá un intervalo sobre cada eje.

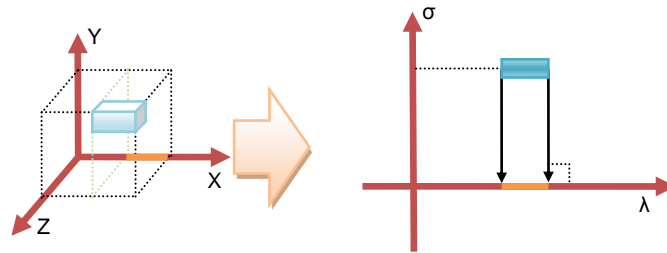


Ilustración 58 Intervalo sobre cada eje

Dicho intervalo tendrá un valor máximo y otro mínimo

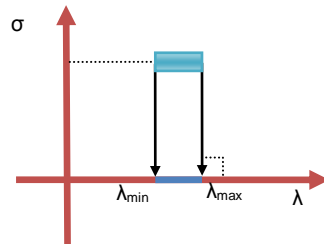


Ilustración 59 Valor máximo y mínimo

Teniendo esto en cuenta podremos utilizarlo para ordenar todos los elementos respecto a su valor mínimo en un árbol.

Veamos un ejemplo, con el fin de simplificarlo lo más posible imaginaremos tan solo dos ejes.

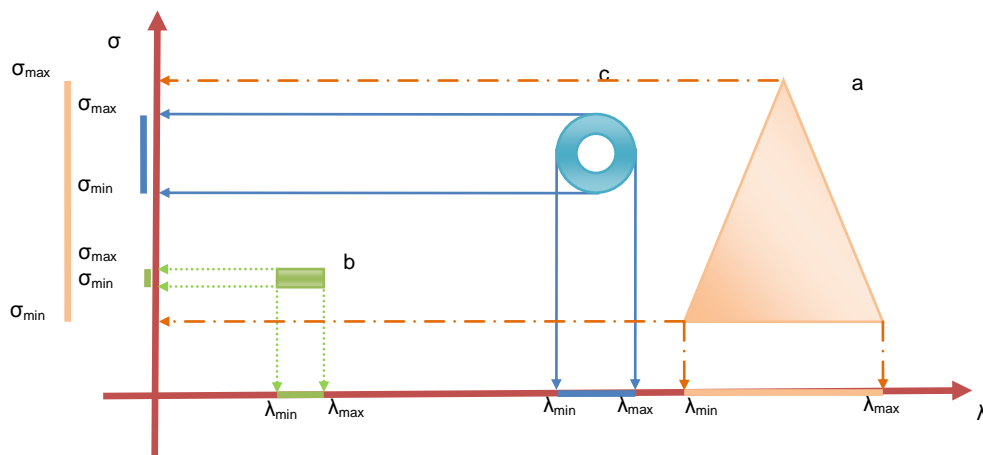


Ilustración 60 Valores máximos y mínimos

Ahora tomamos el valor mínimo de cada una de las figuras y lo utilizamos para generar el árbol como si fuera una nube de puntos:

$$\lambda_{min}^b \leq \lambda_{min}^c \leq \lambda_{min}^a$$

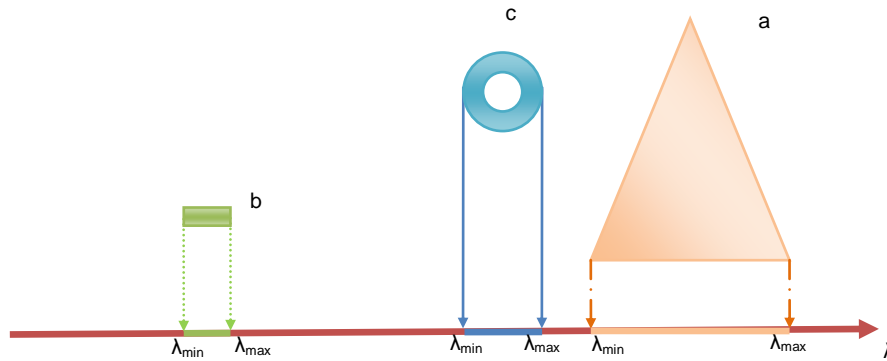


Ilustración 61 Valores máximos y mínimos sobre un eje

Cabe destacar que también podríamos realizar el mismo planteamiento usando los valores máximos o incluso combinarlos para realizar uno u otro según el nivel que nos encontremos en el árbol bajo un criterio como si es par o no el nivel.

IV.II.III. Pseudocódigo de construcción de un árbol kdtree de una nube de volúmenes

Versión simple

El algoritmo de generación en pseudocódigo sería:

```
INICIO CONSTRUIR (LISTADO): KDTREE
  RETORNA CONSTRUIR (LISTADO, 0)
FIN CONSTRUIR
```

```
INICIO CONSTRUIR (LISTADO, NIVEL): KDTREE
  TAMAÑO = NUMERO_ELEMENTOS (LISTADO)
  SELECCIONAR TAMAÑO
  CASO 0
    RETORNA NULO
  CASO 1
    NODO = LISTADO [0]
    RETORNA NODO
  CASO 2
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [0]
    NODO.LEFT = LISTADO [1]
    RETORNA NODO
  EN OTRO CASO
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [TAMAÑO/2]
    PARALELIZAR
      NODO.LEFT= CONSTRUIR (LISTADO[0 .. TAMAÑO/2], NIVEL+1)
      NODO.RIGTH= CONSTRUIR (LISTADO[(TAMAÑO/2)+1 .. TAMAÑO], NIVEL+1)
      ESPERAR FIN PARALELIZACION
    FIN PARALELIZAR
    RETORNA NODO
  FIN SELECCIONAR
FIN CONSTRUIR
```

```
INICIO ORDENA (LISTADO, NIVEL): LISTADO
  SELECCIONAR MODULO (3, NIVEL)
  CASO 1
    RETORNA QUICKSORT_POR_X_Y_Z_MINIMOS (LISTADO)
  CASO 1
    RETORNA QUICKSORT_POR_Y_Z_X_MINIMOS (LISTADO)
  CASO 2
    RETORNA QUICKSORT_POR_Z_X_Y_MINIMOS (LISTADO)
  FIN SELECCIONAR
FIN ORDENA
```

Versión alternada (Para búsqueda con horquilla)

```
INICIO CONSTRUIR (LISTADO): KDTREE
  RETORNA CONSTRUIR (LISTADO, 0)
FIN CONSTRUIR
```

```
INICIO CONSTRUIR (LISTADO, NIVEL): KDTREE
  SELECCIONAR MODULO (2, NIVEL)
  CASE 0
    RETORNA CONSTRUIR_MIN (LISTADO, NIVEL)
  EN OTRO CASO
    RETORNA CONSTRUIR_MAX (LISTADO, NIVEL)
  END SELECCIONAR
  RETORNA CONSTRUIR_MIN (LISTADO, NIVEL)
FIN CONSTRUIR
```

```
INICIO CONSTRUIR_MIN (LISTADO, NIVEL): KDTREE
  TAMAÑO = NUMERO_ELEMENTOS (LISTADO)
  SELECCIONAR TAMAÑO
  CASO 0
    RETORNA NULO
  CASO 1
    NODO = LISTADO [0]
    RETORNA NODO
  CASO 2
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [0]
    NODO.LEFT = LISTADO [1]
    RETORNA NODO
  EN OTRO CASO
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [TAMAÑO/2]
    PARALELIZAR
      NODO.LEFT= CONSTRUIR (LISTADO[0 .. TAMAÑO/2], NIVEL+1)
      NODO.RIGTH= CONSTRUIR (LISTADO[(TAMAÑO/2)+1 .. TAMAÑO], NIVEL+1)
      ESPERAR FIN PARALELIZACION
    FIN PARALELIZAR
    RETORNA NODO
  FIN SELECCIONAR
FIN CONSTRUIR
```

```
INICIO CONSTRUIR_MAX (LISTADO, NIVEL): KDTREE
  TAMAÑO = NUMERO_ELEMENTOS (LISTADO)
  SELECCIONAR TAMAÑO
  CASO 0
    RETORNA NULO
  CASO 1
    NODO = LISTADO [0]
    RETORNA NODO
  CASO 2
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [0]
    NODO.RIGHT = LISTADO [1]
    RETORNA NODO
  EN OTRO CASO
    ORDENA (LISTADO, NIVEL)
    NODO = LISTADO [TAMAÑO/2]
    PARALELIZAR
      NODO.RIGH = CONSTRUIR (LISTADO[0 .. TAMAÑO/2]+1, NIVEL+1)
      NODO.LEFT= CONSTRUIR (LISTADO[(TAMAÑO/2) .. TAMAÑO], NIVEL+1)
      ESPERAR FIN PARALELIZACION
    FIN PARALELIZAR
    RETORNA NODO
  FIN SELECCIONAR
FIN CONSTRUIR
```

```
INICIO ORDENA (LISTADO, NIVEL): LISTADO
  SELECCIONAR MODULO (3, NIVEL)
  CASO 1
    RETORNA QUICKSORT_POR_X_Y_Z_MINIMOS (LISTADO)
  CASO 1
    RETORNA QUICKSORT_POR_Y_Z_X_MINIMOS (LISTADO)
  CASO 2
    RETORNA QUICKSORT_POR_Z_X_Y_MINIMOS (LISTADO)
  FIN SELECCIONAR
FIN ORDENA
```

IV.III. Búsqueda de puntos contenidos en una esfera de una nube de puntos

IV.III.I. Explicación

Buscaremos los puntos contenidos en una esfera. Una posible aplicación podría ser funciones de colisiones.

IV.III.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

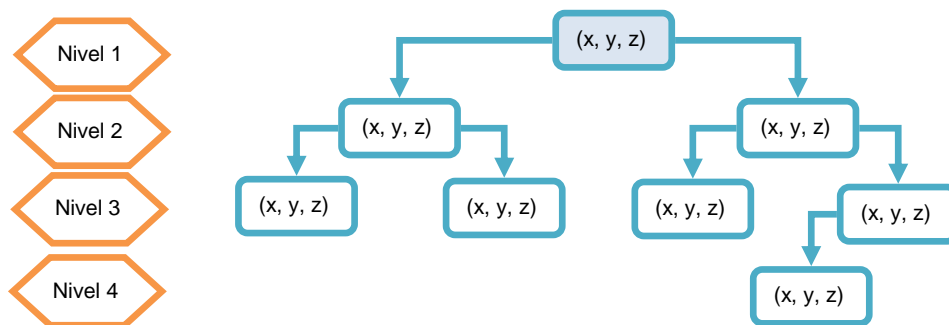


Ilustración 62 Árbol candidato

También tenemos los datos de dicha esfera:

$$P_o = \{\text{Punto de origen de la esfera}\} = (0, 0, 0)$$

$$r = \{\text{Radio de la esfera}\} = 5$$

Primero buscamos la distancia entre el origen de la esfera, P_o y el nodo raíz $P_{raíz}$ por lo tanto si:

$$P_{raíz} = \{\text{Punto del nodo raíz del kd tree}\}$$

Entonces

$$d_o = |P_{raíz} - P_o|$$

A dicha distancia al origen de la esfera d_o , la comparamos con el radio de la esfera si es menor la distancia a la esfera d_e , es cero sino le restamos el radio y obtendremos la distancia desde la esfera al punto.

$$\text{Si } d_o > r \Rightarrow d_e = d_o - r$$

$$\text{Si } d_o \leq r \Rightarrow d_e = 0$$

Si la distancia a la esfera d_e , es cero nos indicará que el nodo cumple con la condición para añadirlo al listado de resultados.

Para poder proseguir con la evaluación del resto de nodos evaluaremos su proyección según el nivel del árbol en el que nos encontremos. Para saber que proyección que se debe evaluar, utilizaremos el módulo del nivel del nodo en el árbol para poder realizar el algoritmo de descarte.

Pueden darse tres casos.

$$m \equiv \text{nivel} \bmod(3) \Rightarrow \begin{cases} \text{si } m = 0 \text{ entonces la proyección sobre } X \Rightarrow P_\lambda = P_x \\ \text{si } m = 1 \text{ entonces la proyección sobre } Y \Rightarrow P_\lambda = P_y \\ \text{si } m = 2 \text{ entonces la proyección sobre } Z \Rightarrow P_\lambda = P_z \end{cases}$$

Si P_λ representa la proyección de punto a evaluar en el eje λ y $P_{o\lambda}$ la proyección del origen sobre el eje λ , calculamos la distancia d_λ entre P_λ y $P_{o\lambda}$:

- Si d_λ menos el radio sale negativo, el punto está contenido en la proyección de la esfera.
- Si d_λ menos el radio sale positivo, el punto no está contenido en la proyección de la esfera.
- Si d_λ menos el radio sale cero, el punto está contenido en los límites de la proyección de la esfera.

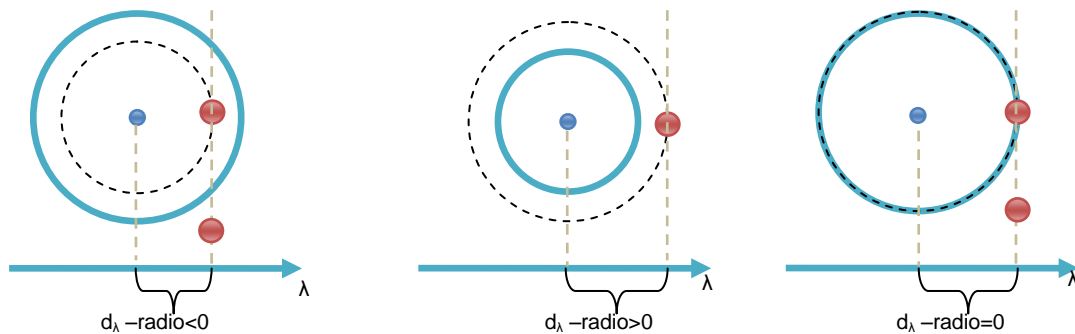


Ilustración 63 Representación de los tres casos

Para el primer caso, esto nos indicará que deberemos buscar en ambas ramas, ya que puede haber puntos en una órbita inferior o superior que estén contenidos en la esfera. Por lo que repetiríamos el proceso con los subárboles derecho e izquierdo.

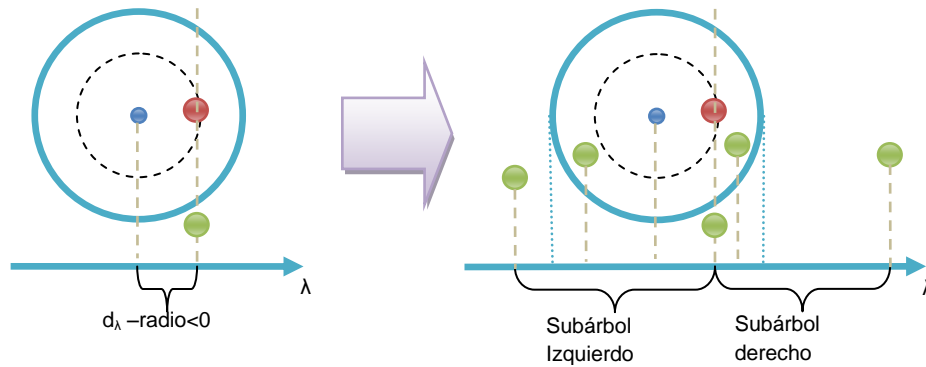


Ilustración 64 Selección de subárbol caso 1

En el segundo caso ya sabemos que la distancia de la proyección menos el radio sale positiva. Por lo que se puede descartar una de las ramas del árbol, según la proyección del punto tenga un valor menor o mayor que la proyección del origen de la esfera en el eje. Si es menor eliminaremos la rama izquierda, y si es mayor eliminaremos la rama derecha.

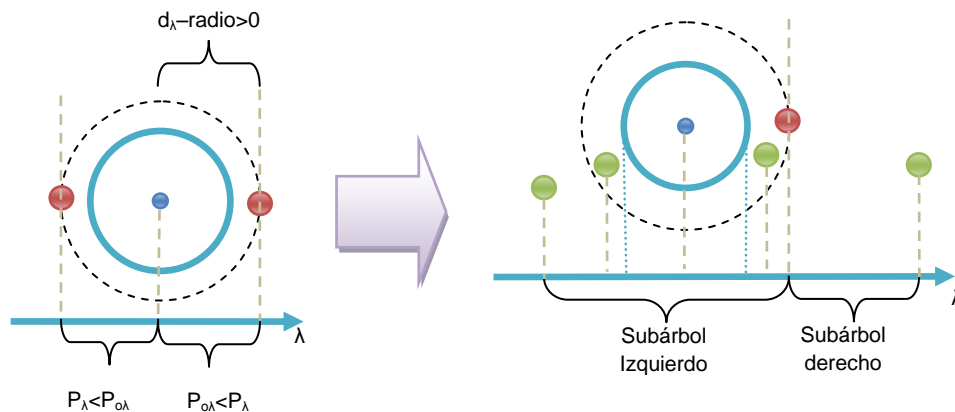


Ilustración 65 Selección de subárbol caso 2

En el tercer caso coincide el punto con la superficie de la esfera por lo que, según el valor de la de la proyección en el eje λ del punto P_λ y la proyección del origen de la esfera, $P_{o\lambda}$ podremos:

Si $P_\lambda \geq P_{o\lambda} \Rightarrow$ No eliminaríamos ninguna de las dos ramas.

Si $P_\lambda < P_{o\lambda} \Rightarrow$ Eliminaríamos la rama de la izquierda.

Esto es así porque en la rama de la derecha se almacenan los puntos que en λ coinciden sus proyecciones.

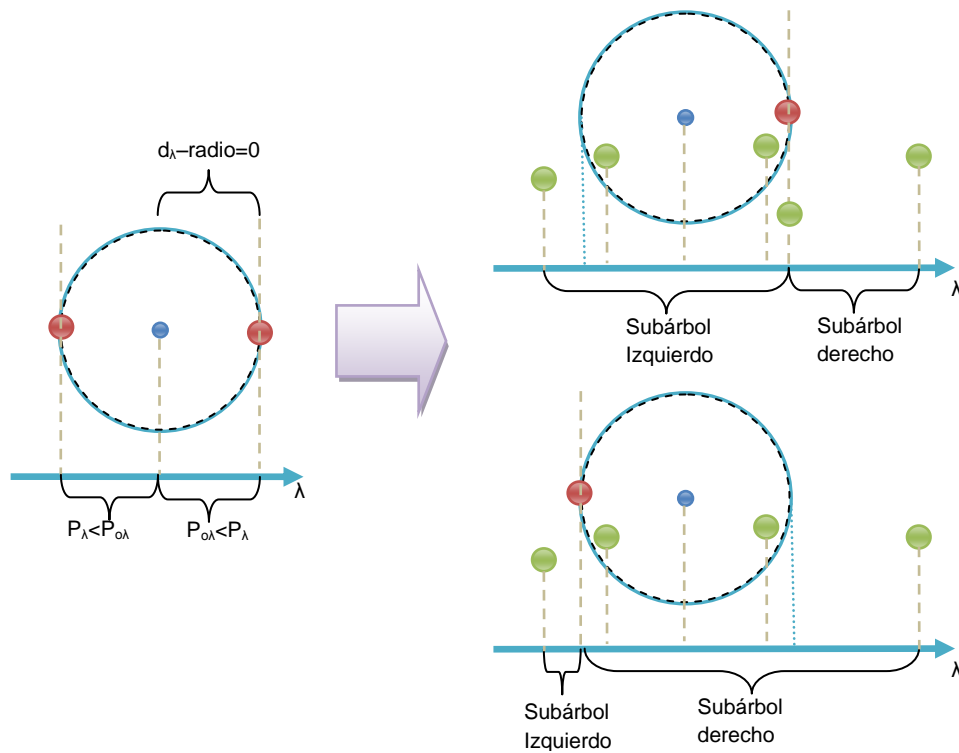


Ilustración 66 Selección de subárbol caso 3

Si resumimos los tres casos obtenemos que sabiendo que:

$S_{\min}(\lambda) = \{\text{Valor mínimo de la proyección de la esfera en el eje } \lambda\}$

$S_{\max}(\lambda) = \{\text{Valor máximo de la proyección de la esfera en el eje } \lambda\}$

$P_{\lambda} = \{\text{Valor de la proyección del punto en el eje } \lambda\}$

Entonces si

$P_{\lambda} \leq S_{\min}(\lambda) \Rightarrow \text{Descartamos la rama Izquierda}$

$P_{\lambda} > S_{\max}(\lambda) \Rightarrow \text{Descartamos la rama Derecha}$

Otro caso \Rightarrow No podemos descartar ninguna de las dos ramas.

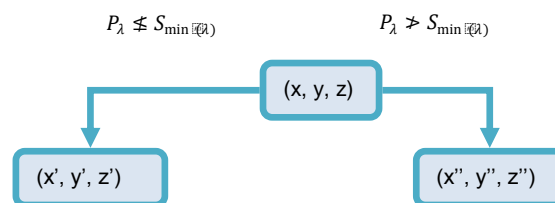


Ilustración 67 Elección de rama

IV.III.III. Estudiémoslo con un ejemplo:

Supongamos que tenemos el siguiente árbol kd:

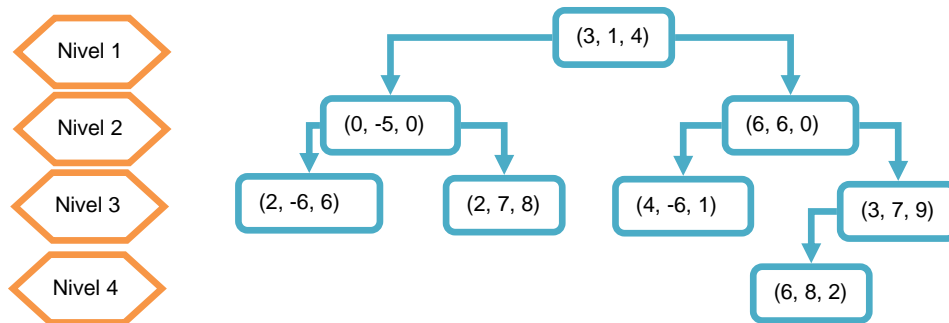


Ilustración 68 Ejemplo

De la esfera sabemos:

$$P_o = \{\text{Punto de origen de la esfera}\} = (0, 0, 0)$$

$$r = \{\text{Radio de la esfera}\} = 5$$

Primero buscamos la distancia entre el origen de la esfera, P_o y el nodo raíz P_{raiz} por lo tanto si:

$$P_{raiz} = (3, 1, 4)$$

Entonces

$$d_o = |P_{raiz} - P_o| = |(3, 1, 4)| = 5,099$$

A dicha distancia le restamos el radio y obtenemos la distancia desde la esfera al punto, d_e .

$$d_e = d_o - r = 5,099 - 5 = 0,099$$

Como podemos observar este nodo no está dentro de la esfera.

Pasemos a ver por qué rama continuar, recordemos:

$$P_\lambda \leq S_{\min(\lambda)} \Rightarrow \text{Descartamos la rama Izquierda}$$

$$P_\lambda > S_{\max(\lambda)} \Rightarrow \text{Descartamos la rama Derecha}$$

Otro caso \Rightarrow No podemos descartar ninguna de las dos ramas.

Por lo que tenemos que si $\lambda=X$:

$$S_{\min(x)} = -5$$

$$S_{\max(x)} = 5$$

$$P_x = 3$$

Entonces

$$P_x \not\leq S_{\min}(x)$$

$$P_x \not\geq S_{\max}(x)$$

Resulta que nos encontramos en el tercer caso, por lo que tenemos que evaluar ambas ramas del árbol en el siguiente nivel.

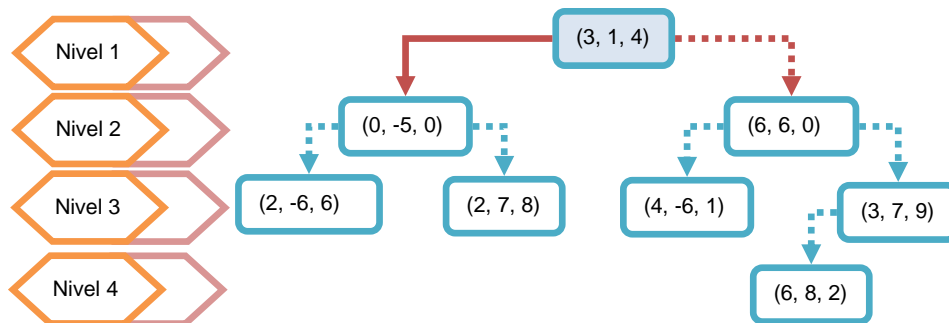


Ilustración 69 Ejemplo proceso 1

En el nivel 2 la rama de la izquierda tenemos que si $\lambda=Y$:

$$S_{\min}(y) = -5$$

$$S_{\max}(y) = 5$$

$$P_y = -5$$

Entonces

$$P_y \leq S_{\min}(y) \Rightarrow \text{Eliminamos la rama de la izquierda}$$

$$P_y \not\geq S_{\max}(y)$$

Representa el primer caso, además este nodo cumple la condición de estar dentro con lo que lo agregaremos al listado de nodos encontrados.

Con esto pasamos a evaluar el nivel 3 de esta rama con lo que al calcular la distancia observamos que no está en el interior de la esfera, este nodo no contiene ramas con lo que daremos por finalizada esta rama.

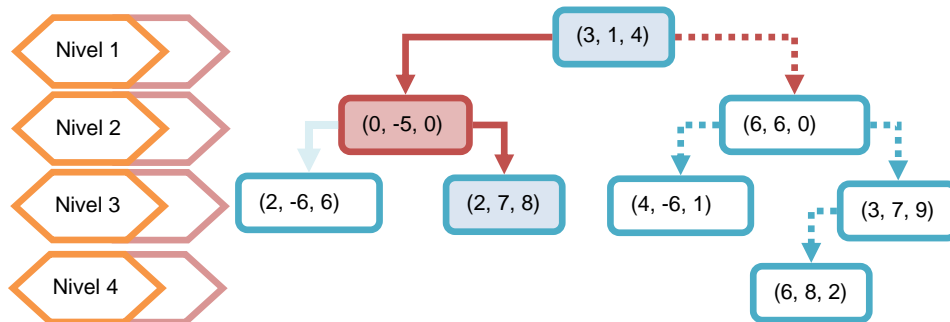


Ilustración 70 Ejemplo proceso 2

Pasemos a evaluar la rama de la derecha del nodo raíz:

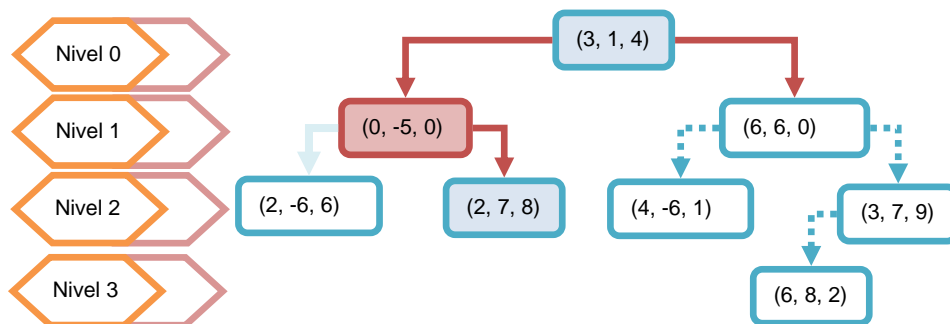


Ilustración 71 Ejemplo proceso 3

En el nivel 2 la rama de la derecha tenemos que si $\lambda=Y$:

$$S_{\min}(y) = -5$$

$$S_{\max}(y) = 5$$

$$P_y = 6$$

Entonces

$$P_y \leq S_{\min}(y)$$

$$P_y > S_{\max}(y) \Rightarrow \text{Eliminamos la rama de la derecha.}$$

Proseguimos evaluando el nodo de nivel 3 por la derecha.

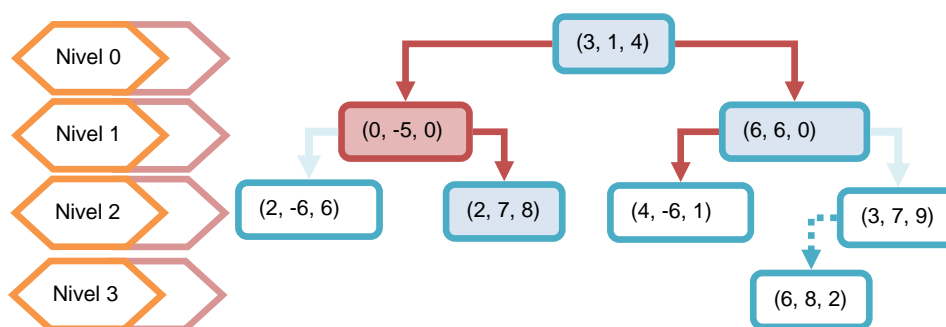


Ilustración 72 Ejemplo proceso 4

El nivel 2 por la derecha no está dentro del árbol y no tiene ramas con lo que habríamos terminado el recorrido del árbol.

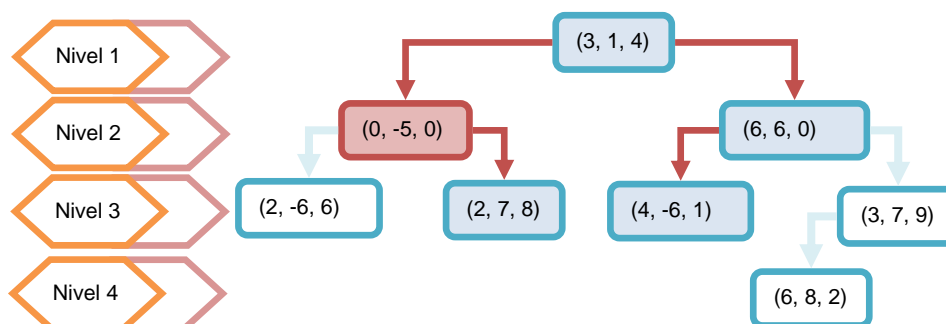


Ilustración 73 Ejemplo proceso 5

Como podemos apreciar hemos recortado bastante el árbol kd y hemos evaluado solo cuatro nodos de los 7 nodos.

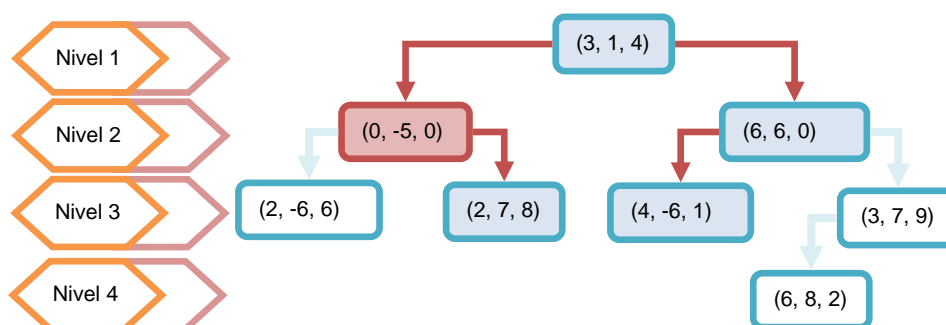


Ilustración 74 Ejemplo proceso 6

Otro dato que es interesante es que existe la posibilidad de paralelizar la ejecución de distintas ramas, por ejemplo si se no da el caso de tener que evaluar la rama de la derecha y en la izquierda estas pueden ser ejecutadas en paralelo. Por ejemplo por una red de ordenadores.

IV.III.IV. *Pseudocódigo de la búsqueda esférica en una nube de puntos*

El algoritmo de generación en pseudocódigo sería:

```

INICIO FIND_ESFERICA (ESFERA, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ESFERICA (LISTADO, ROOT, SPHERE)
  RETORNA LISTADO
FIN FIND_ESFERICA

```

```

INICIO FIND_ESFERICA (LISTADO, NODO, ESFERA)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    PC = NODO.SHAPE
    SI PC INTERSECT ESFERA ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI PC.IS_LESSER_X (ESFERA.MIN_X ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.X = ESFERA.MIN_X ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.IS_GREATER_X (ESFERA.MAX_X ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        INICIO PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI PC.IS_LESSER_Y (ESFERA.MIN_Y ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.Y = ESFERA.MIN_Y ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.IS_GREATER_Y (ESFERA.MAX_Y ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        INICIO PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI PC.IS_LESSER_Z (ESFERA.MIN_Z ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.Z = ESFERA.MIN_Z ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
      SINO SI PC.IS_GREATER_Z (ESFERA.MAX_Z ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        INICIO PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    FIN SI
  END SELECCIONAR
FIN FIND_ESFERICA

```

IV.IV. Búsqueda de volúmenes contenidos en una esfera de una nube de volúmenes

IV.IV.I. Explicación

Buscaremos los volúmenes contenidos ya sea en parte o en totalidad en una esfera. Una posible aplicación podría ser funciones de colisión.

IV.IV.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

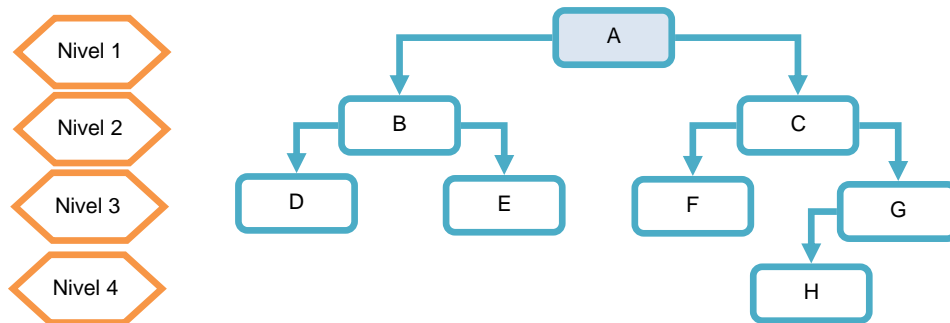


Ilustración 75 Árbol de volúmenes

También tenemos los datos de dicha esfera:

$P_o = \{\text{Punto de origen de la esfera}\}$

$r = \{\text{Radio de la esfera}\}$

Primero buscamos la menor distancia entre el origen de la esfera, P_o y el nodo raíz $N_{raíz}$ por lo tanto si:

$V_{raíz} = \{\text{Volumen del nodo raíz del kd tree}\}$

Entonces

$$d_o = distancia(P_o, V_{raíz})$$

A dicha distancia al origen de la esfera, d_o , la comparamos con el radio de la esfera si es menor la distancia es cero sino le restamos el radio y obtendremos la distancia desde la esfera al volumen.

$$Si\ d_o > r \Rightarrow d_e = d_o - r$$

$$Si\ d_o \leq r \Rightarrow d_e = 0$$

Si la distancia a la esfera, d_e , es cero nos indicará que el nodo cumple con la condición para añadirlo al listado de resultados.

Para poder proseguir con la evaluación del resto de nodos evaluaremos su proyección según el nivel del árbol en el que nos encontremos. Para saber que proyección que se debe evaluar, utilizaremos el modulo del nivel del nodo en el árbol para poder realizar el algoritmo de descarte.

$$m \equiv \text{nivel} \bmod(3) \Rightarrow \begin{cases} \text{si } m = 0 \text{ entonces la proyeccion sobre } X \Rightarrow P_\lambda = P_x \\ \text{si } m = 1 \text{ entonces la proyeccion sobre } Y \Rightarrow P_\lambda = P_y \\ \text{si } m = 2 \text{ entonces la proyeccion sobre } Z \Rightarrow P_\lambda = P_z \end{cases}$$

Un volumen proyectara en eje un intervalo de.

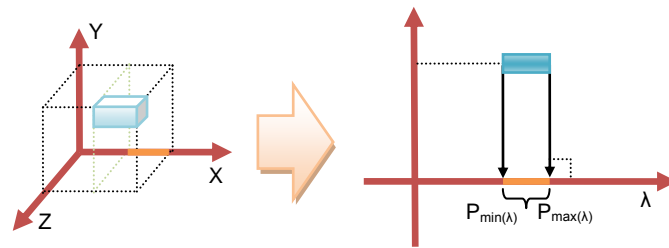


Ilustración 76 Proyección de intervalo

Sabiendo que tenemos:

$$P_{\min(\lambda)} = \{\text{Valor menor del intervalo proyectado sobre el eje } \lambda\}$$

$$P_{\max(\lambda)} = \{\text{Valor mayor del intervalo proyectado sobre el eje } \lambda\}$$

$$P_{\lambda_0} = \{\text{Valor del origen de la esfera proyectado sobre el eje } \lambda\}$$

Pueden darse tres casos.

Si $P_{\min(\lambda)}$ representa la proyección del volumen y tenemos a evaluar en el eje λ y $P_{o\lambda}$ la proyección del origen sobre el eje λ , calculamos la distancia d_λ entre P_λ y $P_{o\lambda}$:

- Si $d_{\min(\lambda)}$ menos el radio sale negativo, el volumen intersecciona en la proyección de la esfera.
- Si $d_{\min(\lambda)}$ menos el radio sale positivo, el punto no intersecciona en la proyección de la esfera.
- Si $d_{\min(\lambda)}$ menos el radio sale cero, el punto intersecciona en los límites de la proyección de la esfera.

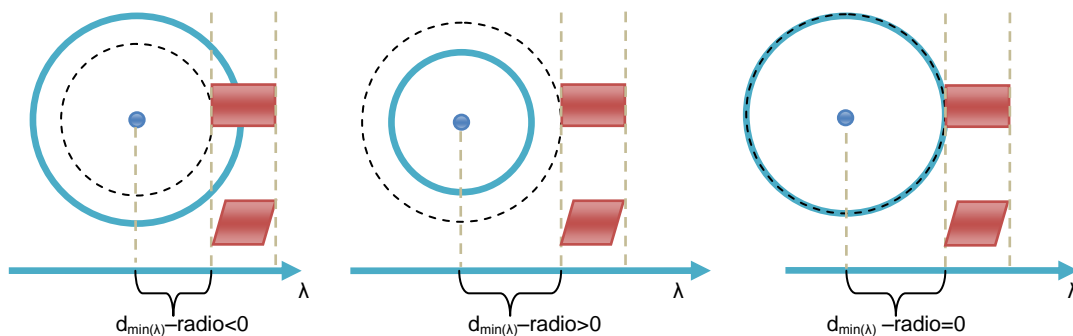


Ilustración 77 Tres casos

Para el primer caso, esto nos indicará que deberemos buscar en ambas ramas, ya que puede haber volúmenes en una órbita inferior o superior que estén en intersección con la esfera. Por lo que repetiríamos el proceso con los subárboles derecho e izquierdo.

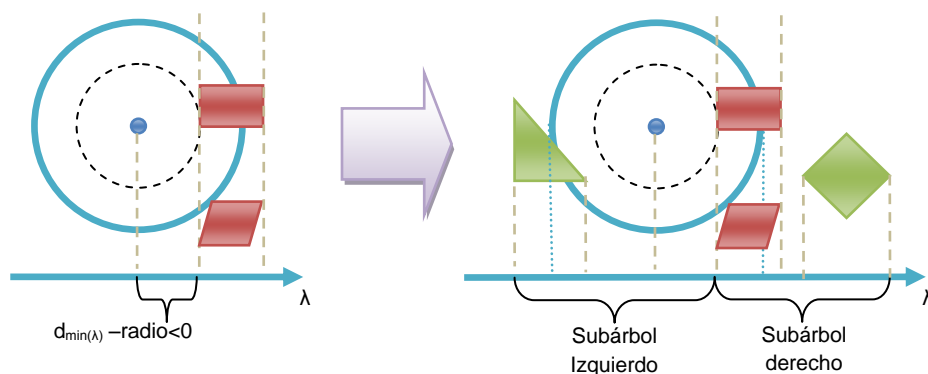


Ilustración 78 Repetiríamos el proceso con los subárboles derecho e izquierdo

En el segundo caso ya sabemos que la distancia de $P_{\min(\lambda)}$ menos el radio sale positiva. Por lo que se puede descartar una de las ramas del árbol, según la proyección del volumen tenga un valor menor o mayor que la proyección del origen de la esfera. Si es menor no eliminaremos ninguna de las ramas, y si es mayor eliminaremos la rama derecha.

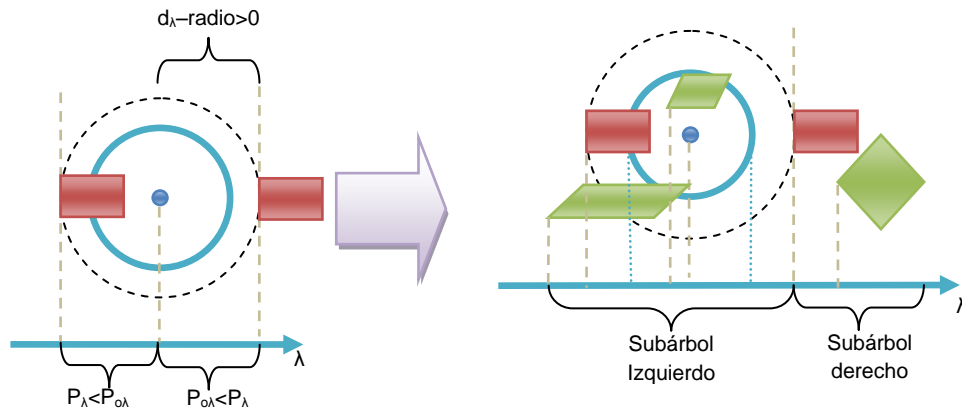


Ilustración 79 Eliminación de la rama

En el tercer caso coincide $P_{\min(\lambda)}$ con la superficie de la esfera por lo que, según el valor de la de $P_{\min(\lambda)}$ en el eje λ y la proyección del origen de la esfera, $P_{o\lambda}$ podremos:

Si $P_{\min(\lambda)} \geq P_{o\lambda} \Rightarrow$ No eliminaremos ninguna de las dos ramas.

Si $P_{\min(\lambda)} < P_{o\lambda} \Rightarrow$ No eliminaremos ninguna de las dos ramas.

Esto es así porque en la rama de la derecha se almacenan los puntos que en λ coinciden sus proyecciones.

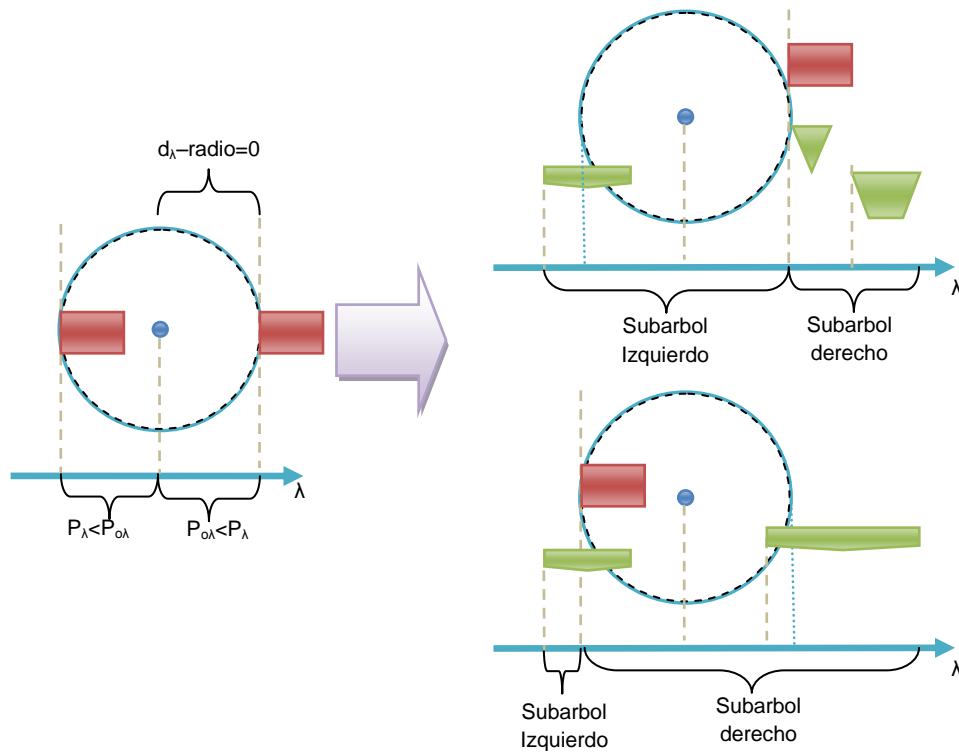


Ilustración 80 Elección de subárbol

Si resumimos los tres casos obtenemos que sabiendo que:

$S_{\min}(\lambda) = \{\text{Valor mínimo de la proyección de la esfera en el eje } \lambda\}$

$S_{\max}(\lambda) = \{\text{Valor máximo de la proyección de la esfera en el eje } \lambda\}$

$P_{\min(\lambda)} = \{\text{Valor mínimo de la proyección del volumen en el eje } \lambda\}$

Entonces si

$$P_{\min(\lambda)} > S_{\max(\lambda)} \Rightarrow \text{Descartamos la rama Derecha}$$

Otro caso \Rightarrow No podemos descartar ninguna de las dos ramas.

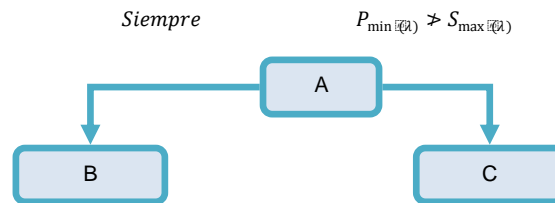


Ilustración 81 Algoritmo de selección de rama

Cabe mencionar que la lectura del árbol puede hacerse de forma paralela en las intersecciones, y sabiendo que el lado izquierdo siempre se evalúa deberíamos paralelizar este.

Una posibilidad que se nos presenta de cara a mejorar la búsqueda frente a espacios grandes es la de en vez de usar siempre el valor $P_{\min(\lambda)}$ también usar el valor $P_{\max(\lambda)}$, por ejemplo utilizar el valor $P_{\min(\lambda)}$ en los niveles impares y el valor $P_{\max(\lambda)}$ en los niveles pares.

Esto acotaría el espacio de búsqueda no tendiendo porque evaluarse la rama de la izquierda siempre.

Por lo tanto para los niveles impares tendríamos:

$$P_{\min(\lambda)} > S_{\max(\lambda)} \Rightarrow \text{Descartamos la rama Derecha}$$

Otro caso \Rightarrow No podemos descartar ninguna de las dos ramas.

Y para los niveles pares:

$$P_{\max(\lambda)} < S_{\min(\lambda)} \Rightarrow \text{Descartamos la rama Izquierda}$$

Otro caso \Rightarrow No podemos descartar ninguna de las dos ramas.

Para que esto sea válido también tendríamos que alternar en la generación del árbol el parámetro con el que se ordena y hacia donde van a parar los valores con un parámetro equivalente.

IV.IV.III. Pseudocódigo de la búsqueda esférica en una nube de volúmenes

Versión simple

El algoritmo de generación en pseudocódigo sería:

```
INICIO FIND_ESFERICA (ESFERA, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ESFERICA (LISTADO, ROOT, SPHERE)
  RETORNA LISTADO
FIN FIND_ESFERICA
```

```
INICIO FIND_ESFERICA (LISTADO, NODO, ESFERA)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT ESFERA ENTONCES
      LISTADO.ADD (VC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI VC.IS_GREATER_X (ESFERA.MAX_X ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_GREATER_Y (ESFERA.MAX_Y ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_GREATER_Z (ESFERA.MAX_Z ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ESFERICA
```

Versión con horquilla

```

INICIO FIND_ESFERICA (ESFERA, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ESFERICA (LISTADO, ROOT, SPHERE)
  RETORNA LISTADO
FIN FIND_ESFERICA

```

```

INICIO FIND_ESFERICA (LISTADO, NODO, ESFERA)
  SELECCIONAR MODULO (2, NODO.LEVEL)
  CASE 0
    FIND_ESFERICA_BY_MIN (LISTADO, NODO, ESFERA)
  EN OTRO CASO
    FIND_ESFERICA_BY_MAX (LISTADO, NODO, ESFERA)
  END SELECCIONAR
FIN FIND_ESFERICA

```

```

INICIO FIND_ESFERICA_BY_MIN (LISTADO, NODO, ESFERA)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT ESFERA ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI VC.IS_GREATER_X (ESFERA.MAX_X ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_GREATER_Y (ESFERA.MAX_Y ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_GREATER_Z (ESFERA.MAX_Z ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ESFERICA_BY_MIN

```

```
INICIO FIND_ESFERICA_BY_MAX (LISTADO, NODO, ESFERA)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT ESFERA ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO (3,NODO.LEVEL)
    CASE 0
      SI VC.IS_LESSER_X (ESFERA.MIN_X ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_LESSER_Y (ESFERA.MIN_Y ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_LESSER_Z (ESFERA.MIN_Z ()) ENTONCES
        FIND_ESFERICA (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ESFERICA (LISTADO, NODO.LEFT, ESFERA)
          FIND_ESFERICA (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ESFERICA_BY_MAX
```


IV.V. Búsqueda del punto más cercano

IV.V.I. Explicación

Buscaremos el punto más cercano. Una posible aplicación podría el obtener los puntos más cercanos con el fin de poder sintetizar un escena.

IV.V.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

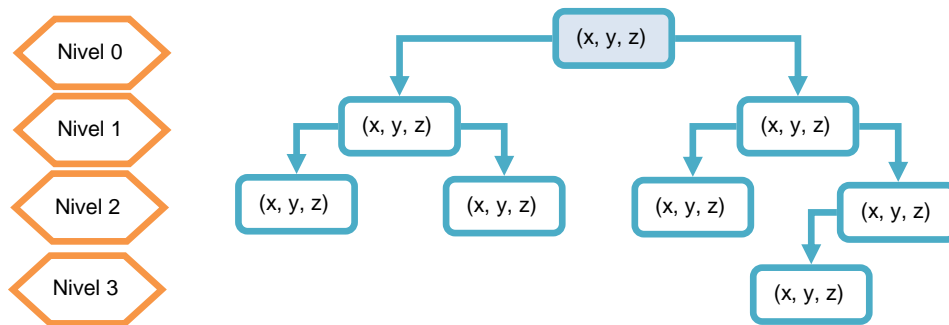


Ilustración 82 Búsqueda del punto más cercano

Como datos tenemos:

$$P_r = \{\text{Punto de referencia}\}$$

$$P_{raiz} = \{\text{Punto raíz del kd tree}\}$$

$$P_c = \{\text{Punto en evaluación del kd tree}\}$$

$$d_{raiz} = |P_{raiz} - P_r|$$

$$d_r = \{\text{Distancia referencia}\}$$

$$d_c = |P_c - P_r|$$

Primero buscamos la distancia entre el punto de referencia, P_r y el punto de la raíz P_{raiz} , esta será la distancia de referencia:

$$d_r = d_{raiz}$$

Con esta distancia iniciamos la búsqueda del punto más cercano.

Si la distancia de referencia d_r , es igual a la distancia candidata d_c , entonces añadiremos el punto como candidato.

Si la distancia de referencia d_r , es menor a la distancia candidata d_c , entonces añadiremos el punto como único candidato.

Una vez hemos evaluado la distancia procedemos a evaluar el siguiente nivel del árbol.

Si la distancia es menor según como hayamos construido el árbol. Deberemos evaluar una rama u otra, esto depende del nivel del árbol, el algoritmo usado en la generación del mismo, etc.

Procederemos a repetir el algoritmo usando de nodo candidato el nodo contenido en el siguiente nivel del árbol.

Suponiendo un algoritmo sencillo de construcción en el cual los niveles estén ordenados según la versión simple de construcción del árbol tendríamos:

Si el nivel está evaluado por la coordenada λ , a la izquierda están los puntos cuyo valor en dicha coordenada sea menor y a la derecha puntos cuyo valor en dicha coordenada sean las mayores o iguales.

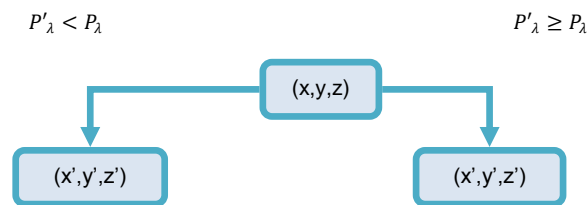


Ilustración 83 Modo de construcción del árbol

Se podrían dar los siguientes casos:

Evaluaríamos la rama de la izquierda si la coordenada del punto candidato $P_{c\lambda}$ es mayor a la del punto de referencia $P_{r\lambda}$.

$$P_{c\lambda} > P_{r\lambda}$$

En otro caso evaluaríamos la rama de la derecha.

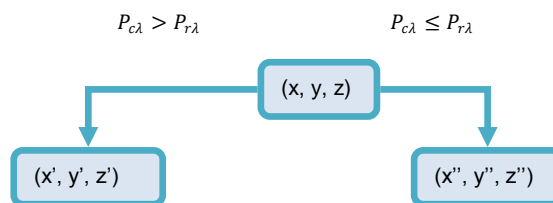


Ilustración 84 Selección de rama

IV.V.III. Pseudocódigo de la búsqueda del más cercano en una nube de puntos

Versión simple

El algoritmo de generación en pseudocódigo sería:

```

INICIO FIND_NEAR (Po, ROOT)
  DR=-1
  SEMAFORO
  LISTADO=EMPTY_LISTADO
  DR= Po.DISTANCE (ROOT.ELEMENTO)
  FIND_NEAR (LISTADO, c, Po, DR, SEMAFORO)
  RETORNA LISTADO
FIN FIND_NEAR

```

```

INICIO FIND_NEAR (LISTADO, NODO, Po, DR, SEMAFORO)
  SI NODO NOT NULO ENTONCES
    PC = NODO.ELEMENTO
    DC = Po.DISTANCE (PC)
    SYNCODE(SEMAFORO)
    IS_EQUAL = (DC = DR)
    SI IS_EQUAL ENTONCES
      LISTADO.ADD(PC)
    FIN SYNCODE
    SYNCODE(SEMAFORO)
    IS_LESSER = (DC < DR)
    SI IS_LESSER ENTONCES
      LISTADO.CLEAR()
      LISTADO.ADD(PC)
      DR=DC
    FIN SYNCODE
  SELECCIONAR (NODO.NIVEL mod 3)
  EN CASO 0:
    SI PC.X<Po.X ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    SI PC.X>Po.X ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
    RETORNAR;
  EN CASO 1:
    SI PC.Y<Po.Y ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    SI PC.Z>Po.Z ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
    RETORNAR;
  EN CASO 2:
    SI PC.Y<Po.Y ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    SI PC.Y>Po.Y ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)

```

```
    RETORNAR;  
  Fin SELECCIONAR  
Fin FIND_NEAR
```

```
INICIO GET_DISTANCE (LISTADO, RANGO, Po)  
  POR CADA ELEMENTO EN LISTADO HAZ  
    RANGO_CANDIDATE = Po.DISTANCE (ELEMENTO)  
    SI RANGO_CANDIDATE < RANGO ENTONCES  
      RETURN RANGO_CANDIDATE  
    FIN SI  
  END POR  
  RETURN -1  
Fin GET_DISTANCE
```

IV.VI. Búsqueda del volumen más cercano

IV.VI.I. Explicación

Buscaremos el volumen más cercano. Una posible aplicación podría el obtener los volumen más cercanos con el fin de poder sintetizar un escena.

IV.VI.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

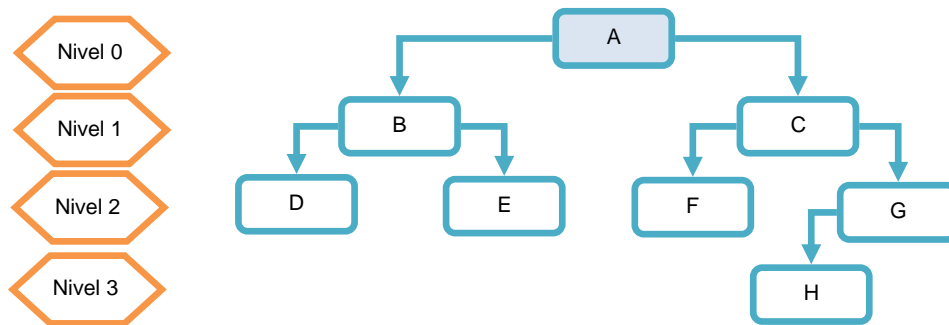


Ilustración 85

Como datos tenemos:

$$P_r = \{\text{Punto de referencia}\}$$

$$V_{raiz} = \{\text{Volúmen raíz del kd tree}\}$$

$$V_c = \{\text{Volumen en evaluacion del kd tree}\}$$

$$d_{raiz} = |V_{raiz} - V_r|$$

$$d_r = \{\text{Distancia referencia}\}$$

$$d_c = |V_c - V_r|$$

Primero buscamos la distancia entre el punto de referencia, P_r y el volumen de la raíz V_{raiz} , esta será la distancia de referencia:

$$d_r = d_{raiz}$$

Con esta distancia iniciamos la búsqueda del volumen más cercano.

Si la distancia de referencia d_r , es igual a la distancia candidata d_c , entonces añadiremos el volumen como candidato.

Si la distancia de referencia d_r , es menor a la distancia candidata d_c , entonces añadiremos el volumen como único candidato.

Una vez hemos evaluado la distancia procedemos a evaluar el siguiente nivel del árbol.

Si la distancia es menor según como hayamos construido el árbol. Deberemos evaluar una rama u otra, esto depende del nivel del árbol, el algoritmo usado en la generación del mismo, etc.

Procederemos a repetir el algoritmo usando de nodo candidato el nodo contenido en el siguiente nivel del árbol.

Suponiendo un algoritmo sencillo de construcción en el cual los niveles estén ordenados según la versión simple de construcción del árbol tendríamos:

Si el nivel está evaluado por la coordenada λ , a la izquierda están los volúmenes cuyo valor en dicha coordenada mínimo sea menor y a la derecha volúmenes cuyo valor en dicha coordenada mínimo sean las mayores o iguales.

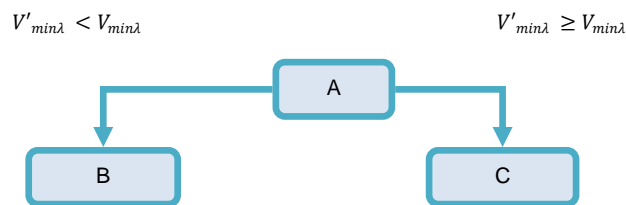


Ilustración 86 Modo de construcción del árbol

Se podrían dar los siguientes casos:

Evaluaríamos la rama de la izquierda si el valor menor de coordenada del volumen candidato $V_{c\ min\ \lambda}$ es menor o igual a la del punto de referencia $P_{r\lambda}$.

$$V_{c\ min\ \lambda} > P_{r\lambda}$$

La rama de la derecha se evaluaría siempre.

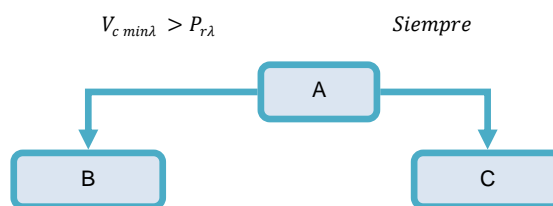


Ilustración 87 Selección de rama

En caso de usar el método de horquilla se irán alternando los niveles impares con los pares la lectura del valor máximo y mínimo.

IV.VI.III. *Pseudocódigo de la búsqueda del más cercano en una nube de volúmenes*

Versión simple

El algoritmo de generación en pseudocódigo sería:

```

INICIO FIND_NEAR (Po, ROOT)
  DR=-1
  SEMAFORO
  LISTADO=EMPTY_LISTADO
  DR= Po.DISTANCE (ROOT.ELEMENTO)
  FIND_NEAR (LISTADO, ROOT, Po, DR, SEMAFORO)
  RETORNA LISTADO
FIN FIND_NEAR

```

```

INICIO FIND_NEAR (LISTADO, NODO, Po, DR, SEMAFORO)
  SI NODO NOT NULO ENTONCES
    VC = NODO.ELEMENTO
    DC = Po.DISTANCE (VC)
    SYNCODE(SEMAFORO)
    IS_EQUAL = (DC = DR)
    SI IS_EQUAL ENTONCES
      LISTADO.ADD(VC)
    FIN SYNCODE
    SYNCODE(SEMAFORO)
    IS_LESSER = (DC < DR)
    SI IS_LESSER ENTONCES
      LISTADO.CLEAR()
      LISTADO.ADD(VC)
      DR=DC
    FIN SYNCODE
    SI IS_EQUAL O IS_LESSER ENTONCES
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    FIN SI
    SELECCIONAR (NODO.NIVEL mod 3)
    EN CASO 0:
      SI VC.MIN_X<Po.X ENTONCES
        FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
        RETORNAR;
      FIN SI
      SI VC.MIN_X>Po.X ENTONCES
        FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
        RETORNAR;
      FIN SI
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    EN CASO 1:
      SI VC.MIN_Y<Po.Y ENTONCES
        FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
        RETORNAR;
      FIN SI
      SI VC.MIN_Y>Po.Y ENTONCES
        FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
        RETORNAR;
      FIN SI
      FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)
      FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)
      RETORNAR;
    EN CASO 2:
      SI VC.MIN_Z<Po.Z ENTONCES
        FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)

```

```
        RETORNAR;  
    FIN SI  
    SI VC.MIN_Z>Po.Z ENTONCES  
        FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)  
        RETORNAR;  
    FIN SI  
    FIND_NEAR(LISTADO, NODO.NODE_LEFT, Po, DR, SEMAFORO)  
    FIND_NEAR(LISTADO, NODO.NODE_RIGHT, Po, DR, SEMAFORO)  
    RETORNAR;  
Fin SELECCIONAR  
FIN FIND_NEAR
```

```
INICIO GET_DISTANCE (LISTADO, RANGO, Po)  
    POR CADA ELEMENTO EN LISTADO HAZ  
        RANGO_CANDIDATE = Po.DISTANCE (ELEMENTO)  
        SI RANGO_CANDIDATE < RANGO ENTONCES  
            RETURN RANGO_CANDIDATE  
        FIN SI  
    END POR  
    RETURN -1  
FIN GET_DISTANCE
```


IV.VII. Búsqueda ortogonal en una nube de puntos

IV.VII.I. Explicación

Buscaremos todos los puntos contenidos en un cubo. Una posible aplicación sería implementar un algoritmo de detección de colisiones.

IV.VII.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

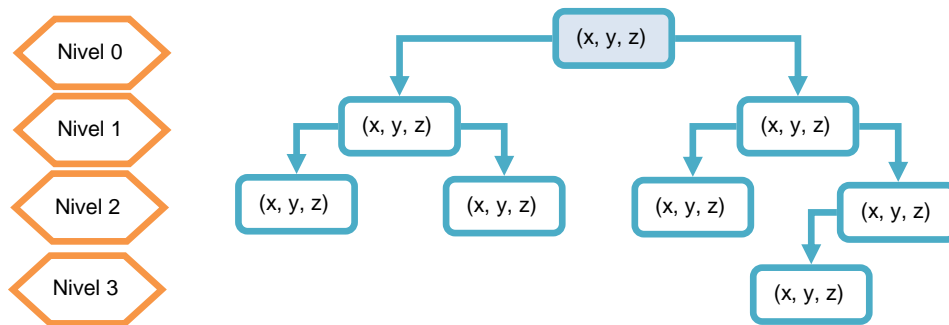


Ilustración 88 Búsqueda ortogonal en una nube de puntos

Como datos tenemos:

$$C = \{\text{Cubo ortogonal}\}$$

$$P_{raiz} = \{\text{Punto raíz del kd tree}\}$$

$$P_c = \{\text{Punto en evaluación del kd tree}\}$$

Iniciamos la evaluación de primer nivel

Si el punto candidato hace intersección con el cubo añadimos a la lista de puntos encontrados.

Deberemos evaluar una rama u otra, esto depende del nivel del árbol y el algoritmo usado en la generación del árbol.

Procederemos a repetir el algoritmo usando de nodo candidato el nodo contenido en el siguiente nivel del árbol.

Suponiendo un algoritmo sencillo de construcción en el cual los niveles estén ordenados según la versión simple de construcción del árbol tendríamos:

Si el nivel está evaluado por la coordenada λ , a la izquierda están los puntos cuyo valor en dicha coordenada sea menor y a la derecha puntos cuyo valor en dicha coordenada sean las mayores o iguales.

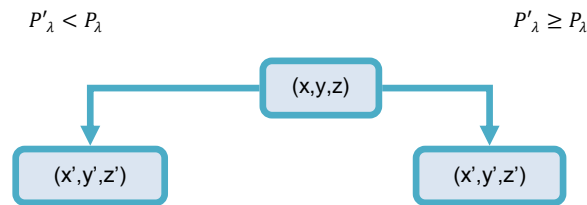


Ilustración 89 Modo de construcción del árbol

Se podrían dar los siguientes casos:

Evaluaríamos el árbol de la izquierda si la coordenada del punto candidato $P_{c\lambda}$ es mayor al mayor valor en λ de la proyección del cubo:

$$V_{c\lambda} > C_{\max\lambda}$$

Evaluaríamos el árbol de la derecha si la coordenada del punto candidato $P_{c\lambda}$ es menor al menor valor en λ de la proyección del cubo:

$$P_{c\lambda} < C_{\min\lambda}$$

En otro caso evaluaríamos las dos.

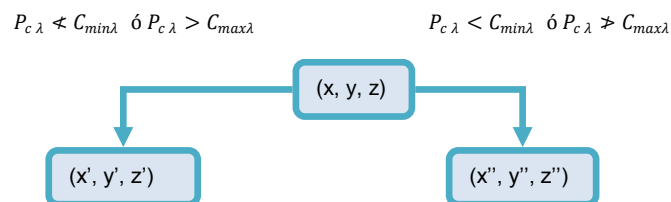


Ilustración 90 Elección de rama

En caso de usar el método de horquilla se irán alternando los niveles impares con los pares la lectura del valor máximo y mínimo.

IV.VII.III. *Pseudocódigo de la búsqueda ortogonal en una nube de puntos*

El algoritmo de generación en pseudocódigo sería:

```

INICIO FIND_ORTOGONAL (CUBE, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ORTOGONAL (LISTADO, ROOT, CUBE)
  RETORNA LISTADO
FIN FIND_ESFERICA

```

```

INICIO FIND_ORTOGONAL (LISTADO, NODO, CUBE)
  SI NODO NO NULO Y CUBE NO NULA ENTONCES
    PC = NODO.SHAPE
    SI PC INTERSECT CUBE ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI PC.IS_LESSER_X (CUBE.MIN_X ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.X = CUBE.MIN_X ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.IS_GREATER_X (CUBE.MAX_X ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        INICIO PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI PC.IS_LESSER_Y (CUBE.MIN_Y ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.Y = CUBE.MIN_Y ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.IS_GREATER_Y (CUBE.MAX_Y ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        INICIO PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI PC.IS_LESSER_Z (CUBE.MIN_Z ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.Z = CUBE.MIN_Z ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
      SINO SI PC.IS_GREATER_Z (CUBE.MAX_Z ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        INICIO PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    FIN SI
  END SELECCIONAR
FIN FIND_ORTOGONAL

```

IV.VIII. Búsqueda ortogonal en una nube de volúmenes

IV.VIII.I. Explicación

Buscaremos todos los volúmenes contenidos en un cubo. Una posible aplicación sería implementar un algoritmo de detección de colisiones.

IV.VIII.II. Desarrollo

Partimos que generamos un árbol kd con todos los candidatos.

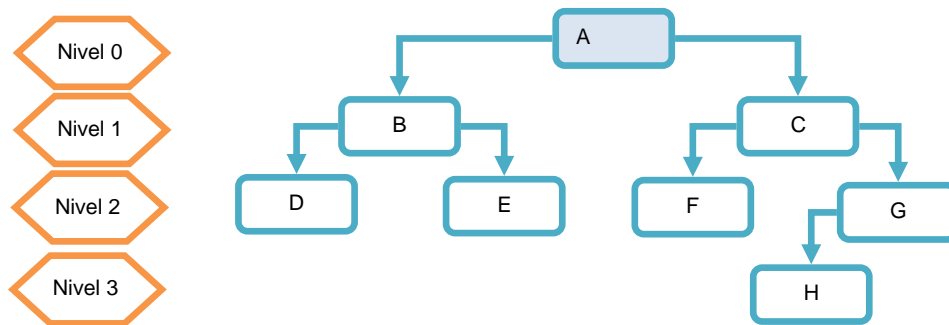


Ilustración 91 Búsqueda ortogonal en una nube de volúmenes

Como datos tenemos:

$C = \{\text{Cubo ortogonal}\}$

$V_{raiz} = \{\text{Volumen raíz del kd tree}\}$

$V_c = \{\text{Volumen en evaluación del kd tree}\}$

Iniciamos la evaluación de primer nivel

Si el volumen candidato hace intersección con el cubo añadimos a la lista de volúmenes encontrados.

Deberemos evaluar una rama u otra, esto depende del nivel del árbol y el algoritmo usado en la generación del árbol.

Procederemos a repetir el algoritmo usando de nodo candidato el nodo contenido en el siguiente nivel del árbol.

Suponiendo un algoritmo sencillo de construcción en el cual los niveles estén ordenados según la versión simple de construcción del árbol tendríamos:

Si el nivel está evaluado por la coordenada λ , a la izquierda están los volúmenes cuyo valor en dicha coordenada mínimo sea menor y a la derecha volúmenes cuyo valor en dicha coordenada mínimo sean las mayores o iguales.

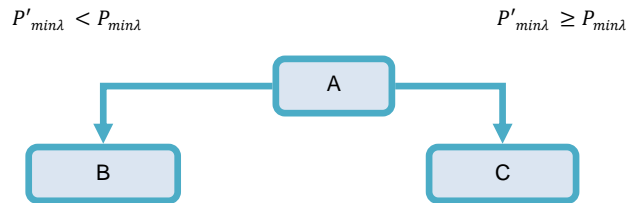


Ilustración 92 Modo de construcción del árbol

Se podrían dar los siguientes casos:

Evaluaríamos el árbol de la izquierda si la coordenada menor del volumen candidato $V_{c\ min\ \lambda}$ es mayor al mayor valor en λ de la proyección del cubo:

$$V_{c\ min\ \lambda} > C_{max\ \lambda}$$

Evaluaríamos el árbol de la derecha si la coordenada mayor del volumen candidato $P_{c\ max\ \lambda}$ es menor al menor valor en λ de la proyección del cubo:

$$V_{c\ max\ \lambda} < C_{min\ \lambda}$$

En otro caso evaluaríamos las dos.

$$V_{c\ max\ \lambda} \nless C_{min\ \lambda} \text{ ó } V_{c\ min\ \lambda} > C_{max\ \lambda} \qquad V_{c\ max\ \lambda} < C_{min\ \lambda} \text{ ó } V_{c\ min\ \lambda} \nless C_{max\ \lambda}$$

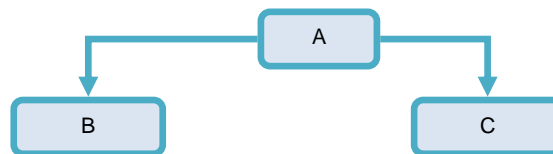


Ilustración 93 Elección de rama

En caso de usar el método de horquilla se irán alternando los niveles impares con los pares la lectura del valor máximo y mínimo.

IV.VIII.III. Pseudocódigo de la búsqueda ortogonal en una nube de volúmenes

Versión simple

El algoritmo de generación en pseudocódigo sería:

```
INICIO FIND_ORTOGONAL (CUBE, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ORTOGONAL (LISTADO, ROOT, CUBE)
  RETORNA LISTADO
FIN FIND_ORTOGONAL
```

```
INICIO FIND_ORTOGONAL (LISTADO, NODO, CUBE)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT CUBE ENTONCES
      LISTADO.ADD (VC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI VC.IS_GREATER_X (CUBE.MAX_X ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_GREATER_Y (CUBE.MAX_Y ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_GREATER_Z (CUBE.MAX_Z ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ORTOGONAL
```

Versión con horquilla

```

INICIO FIND_ORTOGONAL (CUBE, ROOT): LISTADO
  LISTADO=EMPTY_LISTADO
  FIND_ORTOGONAL (LISTADO, ROOT, CUBE)
  RETORNA LISTADO
FIN FIND_ORTOGONAL

```

```

INICIO FIND_ORTOGONAL (LISTADO, NODO, CUBE)
  SELECCIONAR MODULO (2, NODO.LEVEL)
  CASE 0
    FIND_ORTOGONAL_BY_MIN (LISTADO, NODO, CUBE)
  EN OTRO CASO
    FIND_ORTOGONAL_BY_MAX (LISTADO, NODO, CUBE)
  END SELECCIONAR
FIN FIND_ORTOGONAL

```

```

INICIO FIND_ORTOGONAL_BY_MIN (LISTADO, NODO, CUBE)
  SI NODO NO NULO Y CUBE NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT CUBE ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO(3,NODO.LEVEL)
    CASE 0
      SI VC.IS_GREATER_X (CUBE.MAX_X ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_GREATER_Y (CUBE.MAX_Y ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_GREATER_Z (CUBE.MAX_Z ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, CUBE)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, CUBE)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ORTOGONAL_BY_MIN

```

```
INICIO FIND_ORTOGONAL_BY_MAX (LISTADO, NODO, ESFERA)
  SI NODO NO NULO Y ESFERA NO NULA ENTONCES
    VC = NODO.SHAPE
    SI VC INTERSECT ESFERA ENTONCES
      LISTADO.ADD (PC)
    FIN SI
    SELECCIONAR MODULO (3,NODO.LEVEL)
    CASE 0
      SI VC.IS_LESSER_X (ESFERA.MIN_X ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, ESFERA)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    CASE 1
      SI VC.IS_LESSER_Y (ESFERA.MIN_Y ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, ESFERA)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    EN OTRO CASO
      SI VC.IS_LESSER_Z (ESFERA.MIN_Z ()) ENTONCES
        FIND_ORTOGONAL (LISTADO, NODO.RIGTH, ESFERA)
      SINO
        PARALELIZAR
          FIND_ORTOGONAL (LISTADO, NODO.LEFT, ESFERA)
          FIND_ORTOGONAL (LISTADO, NODO.RIGHT, ESFERA)
        FIN PARALELIZAR
      FIN SI
    END SELECCIONAR
  FIN FIND_ORTOGONAL_BY_MAX
```


Anexo V. TABLA DE HASH DISTRIBUIDO

V.I. Preámbulo

Este es el sistema que vamos a emplear para encontrar cada uno de los nodos de la red.

Para poder explicar brevemente e introducirnos en el funcionamiento, nada mejor que citar una definición que me pareció muy acertada:

V.II. Definición

Las tablas de hash distribuidas (en inglés, Distributed Hash Tables, DHT) son una clase de sistemas distribuidos descentralizados que proveen un servicio de búsqueda similar al de las tablas de hash, donde pares (clave, valor) son almacenados en el DHT, y cualquier nodo participante puede recuperar de forma eficiente el valor asociado con una clave dada. La responsabilidad de mantener el mapeo de las claves a los valores está distribuida entre los nodos, de forma que un cambio en el conjunto de participantes causa una cantidad mínima de interrupción. Esto permite que las DHTs puedan escalar a cantidades de nodos extremadamente grandes, y que puedan manejar constantes errores, llegadas y caídas de nodos.

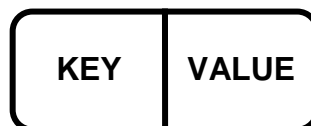


Ilustración 94 Pares (Clave, Valor)

Las DHTs forman una infraestructura que puede ser usada para construir servicios más complejos, como sistemas de archivos distribuidos, compartición de archivos peer-to-peer, sistemas de distribución de contenido, caché web cooperativo, multicast, anycast, servicios de DNS, y mensajería instantánea. Redes distribuidas importantes que usan DHT incluyen los trackers distribuidos de BitTorrent, la red Kad, el Storm botnet, YaCy, y la Coral Content Distribution Network.

V.III. Historia

Las búsquedas por medio de DHTs fueron motivadas originalmente por los sistemas peer-to-peer como Napster, Gnutella, y Freenet, que aprovechaban recursos distribuidos en Internet para proveer una única aplicación. En particular aprovechaban el creciente ancho de banda y capacidad de disco, para brindar un servicio de compartición de archivos.

La diferencia entre estos sistemas estaba en como encontraban los datos que tenían sus pares:

Napster tenía un servicio de indexado central donde cada vez que un nodo se incorporaba, mandaba al servidor una lista de los archivos que poseía a nivel local. El servidor era el encargado de realizar las búsquedas y direccionarlas a los nodos que contenían los resultados. La desventaja estaba en que éste componente central hacía al sistema vulnerable a ataques.

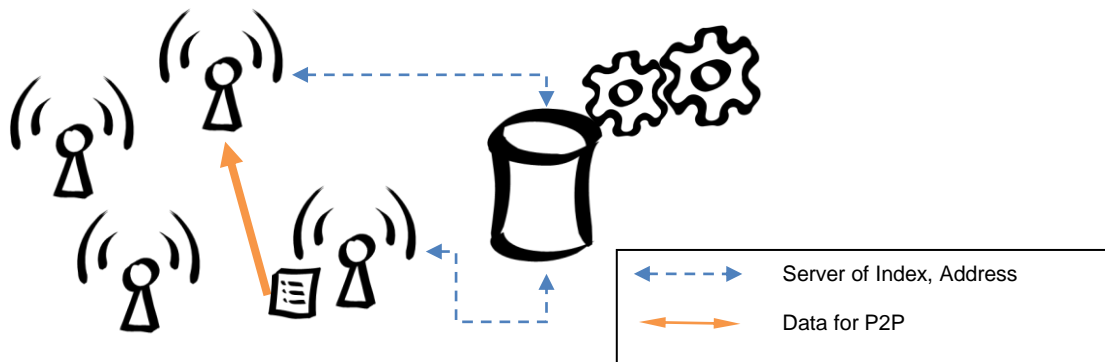


Ilustración 95 Napster

Gnutella y redes similares usaban un modelo de inundaciones, donde cada búsqueda resultaba en un mensaje que se repetía a todas las máquinas en la red. Éste método evitaba el problema de tener un único punto de falla, pero era significativamente menos eficiente que *Napster*.

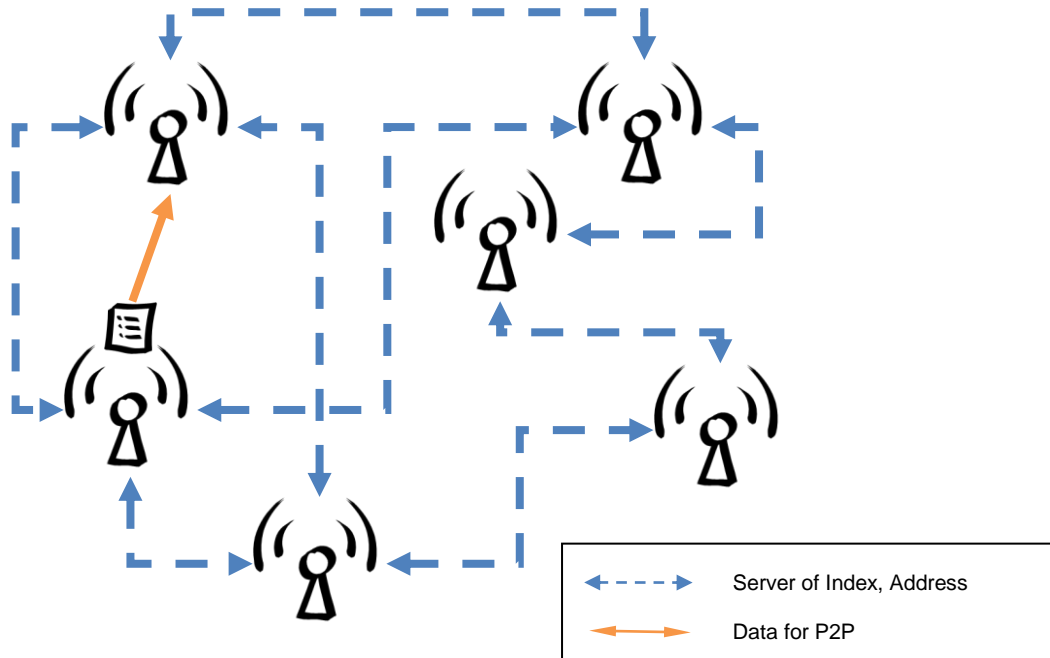


Ilustración 96 Gnutella

Freenet también era totalmente distribuido, y empleaba una clave heurística de ruteo, en donde cada archivo estaba asociado a una clave. Los archivos con claves similares tendían a agruparse en conjuntos de nodos similares. De esta forma era probable que las búsquedas fueran ruteadas a esos conjuntos, sin tener que visitar muchos nodos. Sin embargo no se garantizaba que se iba a encontrar los datos.

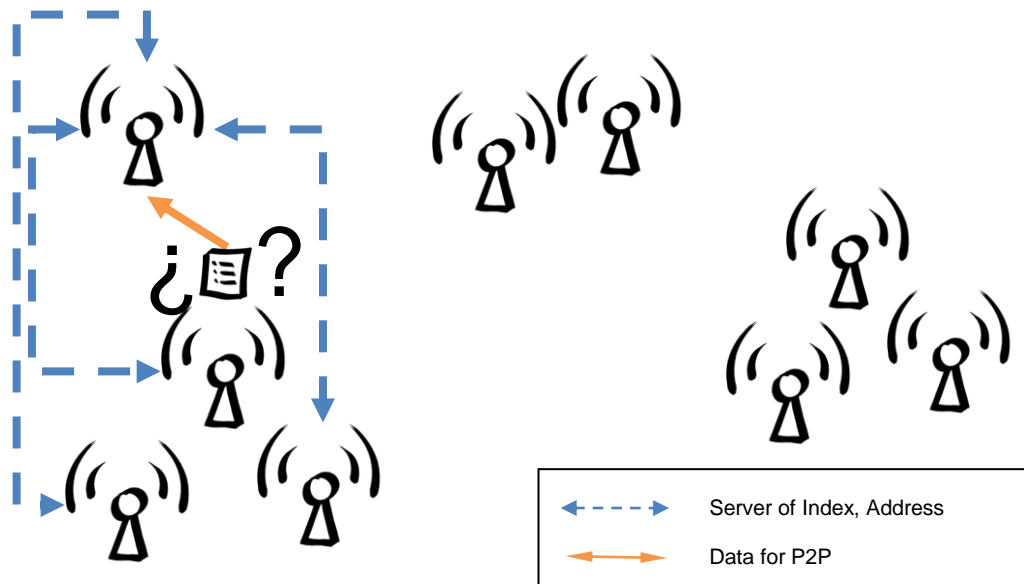


Ilustración 97 Freenet

Las DHTs usan un ruteo basado en claves que incorpora los beneficios de la descentralización de Gnutella y Freenet, y la eficiencia y resultados garantizados de Napster. Una desventaja es que las DHTs así como Freenet, solo soportan búsquedas de coincidencia exacta, no obstante es posible implementar una funcionalidad de búsquedas por palabra clave como una capa superior a las DHTs.

Las primeras cuatro DHTs - CAN, Chord, Pastry y Tapestry- surgieron en la misma época durante el 2001. Desde entonces esta forma de búsqueda ha sido muy usada, principalmente desde que BitTorrent las incorporó.

V.IV. Estructura

La estructura de una DHT puede ser descompuesta en una gran cantidad de componentes. La base es un espacio de claves abstracto. Un esquema de particionamiento de espacio de claves divide entre los nodos este espacio de claves. Una red de overlay conecta los nodos, permitiendo encontrar al titular de cualquier llave en el espacio de claves.

Una vez que estos componentes están en su lugar, el DHT puede ser usado para almacenamiento y recuperación de la siguiente manera:

Supongamos que el espacio de claves es una serie de strings de 160 bits. Para almacenar un archivo con un nombre y datos en el DHT, se aplica el hash SHA1 sobre el nombre del archivo, obteniendo una llave “K” de 160 bits.

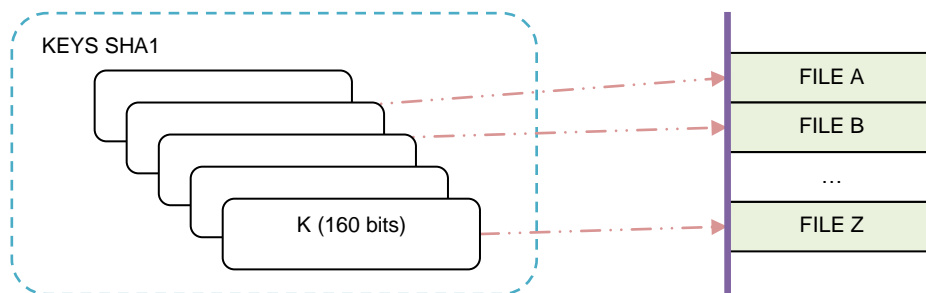


Ilustración 98 Obteniendo una llave “K” de 160 bits

Luego se envía un mensaje $\text{put}(K, \text{data})$ a los nodos participantes en la DHT. El mensaje es enviado de nodo en nodo a través de la red hasta que alcanza al nodo responsable de la llave “K”, especificado en el espacio de claves. En este nodo se almacena el par (K, data) .

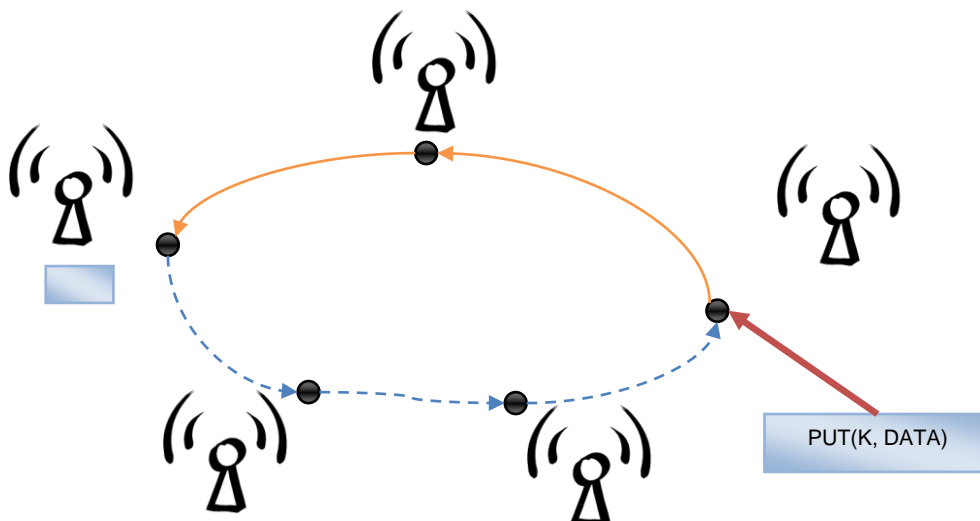


Ilustración 99 El mensaje es enviado de nodo en nodo a través de la red

Si cualquier cliente quiere obtener el contenido del archivo, debe hacer un hash del nombre del archivo, lo cual produce la llave “K”; con ésta se genera un mensaje `get(K)` que es enrutado hasta llegar al nodo responsable, el cual responderá con los datos almacenados.

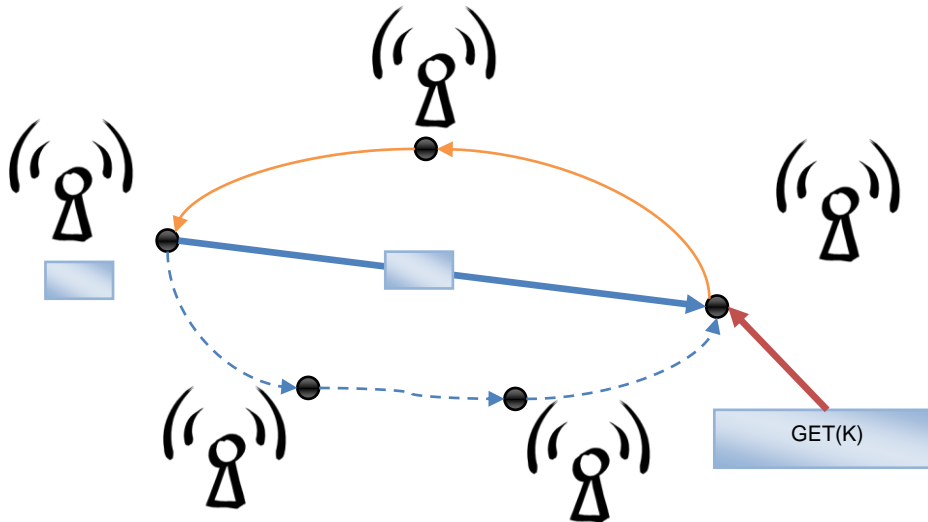


Ilustración 100 Enrutado hasta llegar al nodo responsable

A continuación se describen los componentes del espacio de claves y de la red, con el objetivo de capturar la idea principal de las DHTs; muchos diseños difieren en detalles.

V.V. Particionamiento del espacio de claves

La mayoría de las DHTs utilizan alguna variante de dispersión hash para mapear las claves con los nodos. Esta técnica implementa una función $\delta(k_1, k_2)$ que define una noción abstracta de la distancia entre la clave k_1 y k_2 , la cual no está relacionada con la distancia geográfica o la latencia de la red. A cada nodo se le asigna una única clave denominada ID. Un nodo con el ID "i" posee todas las claves para las cuales "i" es el ID más cercano, medido con la función δ .

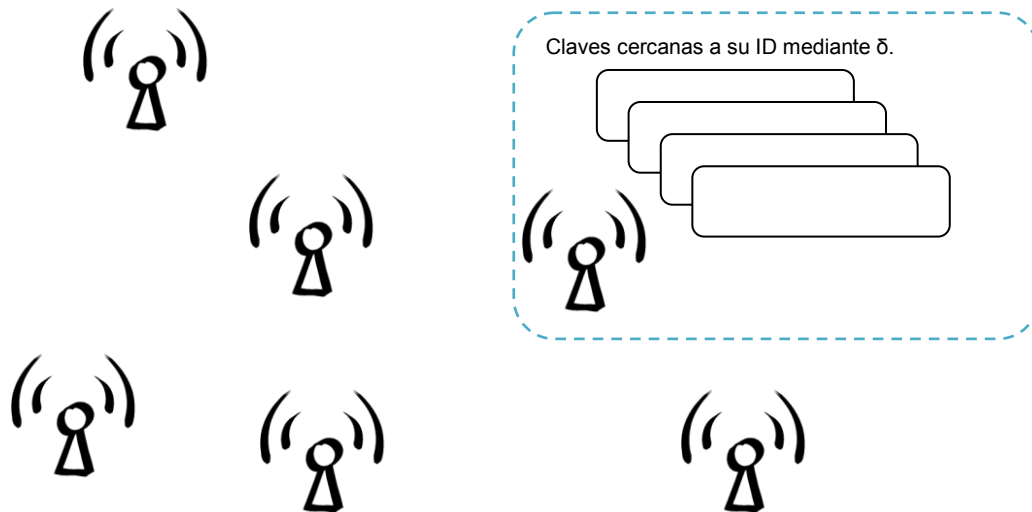


Ilustración 101 Particionamiento del espacio de claves

Ejemplo: la DHT Chord trata las claves como puntos en un círculo y $\delta(k_1, k_2)$ es la distancia alrededor del círculo desde k_1 a k_2 en sentido horario. Así, el espacio de claves circular se divide en segmentos contiguos cuyos extremos son los identificadores del nodo. Si i_1 e i_2 son dos identificadores adyacentes, el nodo con ID i_2 posee todas las llaves que se encuentren entre i_1 e i_2 .

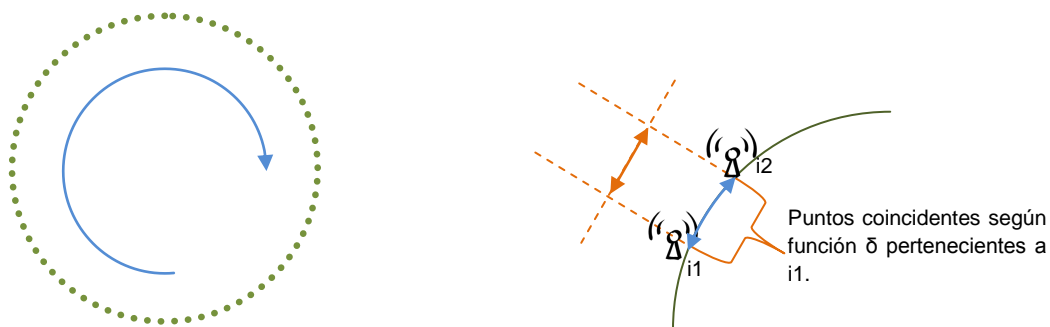


Ilustración 102 $\delta(k_1, k_2)$ es la distancia alrededor del círculo

El hashing tiene la propiedad que la remoción o adición de un nodo cambia únicamente las claves de los nodos con IDs adyacentes, y los demás nodos no son afectados. En una tabla hash tradicional la adición o eliminación de un nodo implica que casi la totalidad del espacio de claves sea reasignada. Dado que cualquier cambio generalmente es debido a un intenso uso del ancho de banda ocasionado por el movimiento de un nodo a otro de objetos almacenados en el DHT, es necesario minimizar esa reorganización para soportar de forma eficiente altas tasas de llegada y falla de nodos.

V.VI. Red de overlay

Cada nodo mantiene una serie de links a otros nodos (sus vecinos o tabla de enrutamiento). En conjunto, estos enlaces forman la red.

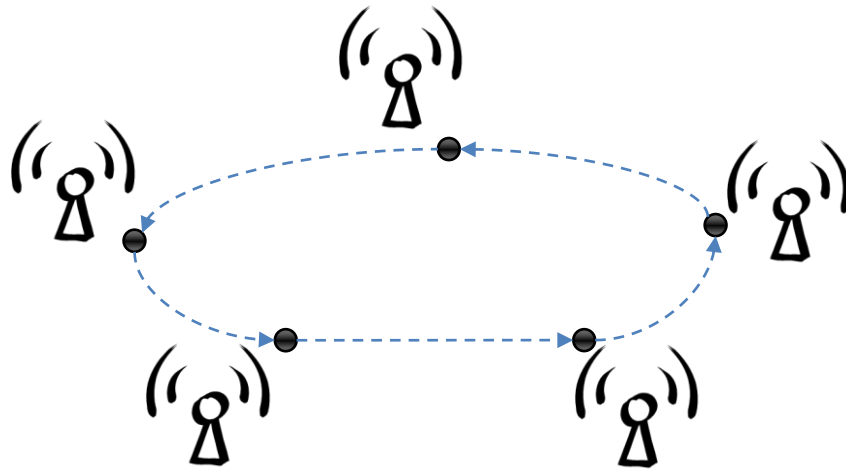


Ilustración 103 Red de overlay

Un nodo escoge sus vecinos de acuerdo a una estructura específica, llamada “topología de la red”. Todas las topologías DHT comparten alguna variante de la siguiente propiedad: para cualquier clave “k”, pasa que el nodo tiene un ID que posee “k” o tiene un link a un nodo cuyo ID es más cercano a “k”, en términos de la distancia definida en el espacio de claves. De esta forma es fácil enrutar un mensaje hacia el dueño de cualquier clave “k” usando un algoritmo “greedy” (algoritmo voraz), que no necesariamente es óptimo a nivel global. El algoritmo consiste en reenviar el mensaje al vecino cuyo ID es más cercano a “k” de forma sucesiva, y cuando no existe ese vecino, pasa que se llegó al nodo más cercano el cual posee “k”. Este estilo de enrutamiento se suele llamar “key based routing”. Más allá de la exactitud del enrutamiento básico, dos limitaciones importantes en la topología son garantizar que el número máximo de saltos en cualquier ruta (longitud de la ruta) sea bajo, de forma que las solicitudes se completen rápidamente; y que el número máximo de vecinos de cualquier nodo sea mínimo, de forma que la sobrecarga de mantenimiento no sea excesiva.

La longitud máxima de la red es el número de saltos del camino más largo, entre los caminos más cortos entre los nodos. La longitud de ruteo de la red es al menos tan grande como su diámetro y puede ser mayor dado que el algoritmo de enrutamiento codicioso puede no encontrar el camino más corto.

V.VII. Algoritmos para redes de overlay

Además del enrutamiento, existen muchos algoritmos que aprovechan la estructura de la red de overlay para el envío de un mensaje a todos los nodos, o un subconjunto de los nodos, en una DHT. Estos algoritmos se utilizan en aplicaciones para hacer multicast, consultas, o para recoger estadísticas.

V.VIII. Propiedades

Las DHTs destacan las siguientes propiedades:

Descentralización: los nodos en conjunto forman el sistema sin ningún tipo de coordinación central.

Escalabilidad: el sistema debe funcionar de manera eficiente, incluso con miles o millones de nodos.

Tolerancia a fallas: el sistema debe ser fiable (en cierto sentido), incluso con nodos continuamente uniéndose, yéndose y fallando.

Una técnica clave utilizada para lograr estos objetivos es que cualquier nodo debe coordinar con sólo unos pocos nodos en el sistema - la más común, $O(\log n)$ de los n participantes (ver abajo) - de manera que ante un cambio en la composición sólo es necesario una cantidad limitada de trabajo. Algunos diseños de DHT buscan ser seguros contra los participantes maliciosos y permiten a los participantes permanecer en el anonimato, aunque esto es menos común que en muchos otros sistemas peer-to-peer (sobre todo para compartir archivos). Por último, las DHTs deben ocuparse de cuestiones más tradicionales de sistemas distribuidos, como sistemas de balanceo de carga, integridad de datos y rendimiento (en particular, garantizar que las operaciones como el enrutamiento y almacenamiento de datos o la recuperación de manera completa se haga de forma rápida).

V.IX. Implementaciones de DHT

Las diferencias más destacables encontradas en casos prácticos de implementaciones de DHT incluyen las siguientes:

El espacio de direcciones es un parámetro de DHT. Varias DHTs utilizan un espacio de direcciones de 128 o 160 bits.

Algunas DHT utilizan funciones de hash distintas a SHA1.

La clave k puede ser un hash del contenido de un archivo en lugar de un hash de su nombre. De esta forma el cambio de nombre del archivo no impide a los usuarios poder encontrarlo.

Algunas DHTs también pueden publicar objetos de diferentes tipos. Por ejemplo, la clave k podría ser el ID del nodo y los datos asociados podrían describir cómo ponerse en contacto con este nodo. Esto permite la publicación

de información de presencia frecuentemente usada en aplicaciones de mensajería instantánea. El caso más simple de identificación es un número aleatorio que se utiliza directamente como clave k (de esta manera en DHTs de 160 bits el ID será un número de 160 bits, usualmente elegido al azar). En algunas DHTs, la publicación de los IDs de los nodos también es usado para optimizar las operaciones de DHT.

Un clave k podría ser almacenada en más de un nodo para mejorar la fiabilidad de DHT por medio de la redundancia. Por lo general, en lugar de seleccionar un solo nodo, los algoritmos DHT seleccionan los i nodos adecuados, siendo i un parámetro específico de implementación del DHT. En estos diseños de DHT, los nodos acuerdan manejar cierto espacio de claves, donde el tamaño del espacio debe ser elegido de forma dinámica.

Algunas DHTs avanzadas como Kademlia realizan en primer lugar búsquedas iterativas de la DHT para seleccionar un conjunto de nodos adecuados y enviar mensajes put (k , data) únicamente a dichos nodos. Esto reduce drásticamente el tráfico inútil, ya que los mensajes publicados sólo se envían a los nodos adecuados para almacenar la clave k y las búsquedas iterativas cubren un conjunto pequeño de nodos en lugar de todo el DHT. En dicha transmisión, los mensajes put (k , data) pueden aparecer como parte de un algoritmo de auto-sanación: si un nodo de destino recibe un mensaje put (k , data), pero considera que k está fuera de su área y conoce un nodo más cercano (en términos del espacio de claves DHT), el mensaje se reenvía a ese nodo. De lo contrario, los datos se indizan a nivel local. Esto conduce a sí mismo a un auto-balanceo de comportamiento. Estos algoritmos requieren que los nodos publiquen sus datos de presencia en la DHT, de forma que se puedan realizar búsquedas iterativas.

Anexo VI. ESTIMACIÓN DE COSTES

VI.I. Presupuesto

Concepto	Precio hora	Cantidad de horas	Total
Análisis	85€	65	5525€
Diseño	75€	75	5625€
Programación	30€	250	7500€
Diseño gráfico	30€	100	3000€
Pruebas	20€	25	500€
SUBTOTAL		515	22150€
IVA			18%
TOTAL			26137€

Tabla 53 Calculo del Presupuesto

El presupuesto es de 22150€ sin IVA y 26137€ con IVA.

EL PRESUPUESTO ES DE **VENTIDOS MIL CIENTO CINCUENTA EUROS SIN IVA Y VENTISEIS MIL CIENTO TRENTA Y SIETE EUROS CON IVA.**

VI.II. Comunicado del desarrollador:

Este presupuesto es el que se estima que se hubiese acordado por la venta del software. Queremos hacer constar que nos atenemos a la licencia GNU para la mejora de la comunidad de usuarios de la red, además de la promoción de los desarrolladores.

VI.III. Ofertas de formación

VI.III.I. Curso de desarrollo de aplicaciones RMI:

Cursillo en el cual se imparte los protocolos RMI y su implementación en java, reseñando con ejemplos.

Mínimo de siete plazas para la realización del cursillo.

Foro máximo de quince plazas.

PRECIO/HORA POR PERSONA: 10€ (**DIEZ** EUROS IVA INCLUIDO)

VI.III.II. Curso de desarrollo de un servicio de mensajería:

Cursillo en el cual se imparte como desarrollar la arquitectura y el modelado en java de un gestor de mensajería.

Mínimo de cinco plazas para la realización del cursillo.

Foro máximo de diez plazas.

PRECIO/HORA POR PERSONA: 15€ (**QUINCE** EUROS IVA INCLUIDO)

VI.III.III. Curso de desarrollo de un programa P2P:

Cursillo en el cual se imparte como desarrollar la arquitectura y el modelado en java de un gestor de archivos P2P.

Mínimo de cinco plazas para la realización del cursillo.

Foro máximo de diez plazas.

PRECIO/HORA POR PERSONA: 20€ (**VEINTE** EUROS IVA INCLUIDO)

Anexo VII. PLIEGO DE CONDICIONES

VII.I. Condiciones Generales

VII.I.I. Documentación del proyecto

Se entregará al fabricante una copia del diseño y pliego de condiciones del proyecto, así como cuanta información necesite para la completa realización de la aplicación.

El fabricante podrá tomar nota o sacar copia por su cuenta y coste de la memoria, presupuesto y anexos del proyecto, así como segundas copias de todos los documentos, si son autorizadas por el proyectista.

El fabricante se hace responsable de la buena conservación de los originales, los cuales, serán devueltos al ingeniero proyectista después de su utilización.

No se harán por parte del fabricante alteraciones, correcciones, omisiones, adiciones o variaciones en los datos fijados en el proyecto salvo aprobación previa y escrita del proyectista.

VII.I.II. Documentación del propietario

La aplicación deberá acompañarse como mínimo de un manual de instalación y uso de la aplicación y la licencia de uso. El soporte de dichos documentos podrá ser tanto en formato digital como impreso.

Dichos documentos deberán considerarse como un componente de la aplicación.

VII.II. Instrucciones de utilización

Deberán contener toda la información necesaria para hacer funcionar la aplicación de acuerdo con su especificación.

VII.II.I. Descripción técnica

La descripción técnica deberá proporcionar las características necesarias para la correcta utilización y funcionamiento de la aplicación. La descripción técnica será distribuida como parte de la solución.

VII.II.II. Plazo de garantía

No hay plazo de garantía. Queremos dejar bien claro por ello citamos textualmente las cláusulas que hablan en la licencia de la inexistencia de garantía.

“11. DEBIDO A QUE EL PROGRAMA SE OTORGA LIBRE DE CARGOS Y REGALIAS, NO EXISTE NINGUNA GARANTIA PARA EL MISMO HASTA DONDE LO PERMITA LA LEY APLICABLE. A EXCEPCION DE QUE SE INDIQUE OTRA COSA, LOS PROPIETARIOS DE LOS DERECHOS DE AUTOR PROPORCIONAN EL PROGRAMA "COMO ES" SIN NINGUNA GARANTIA DE NINGUN TIPO, YA SEA EXPLICITA O IMPLICITA, INCLUYENDO, PERO NO LIMITADA A, LAS GARANTIAS QUE IMPLICA EL MERCADEO Y EJERCICIO DE UN PROPOSITO EN PARTICULAR. CUALQUIER RIESGO DEBIDO A LA CALIDAD Y DESEMPEÑO DEL PROGRAMA ES TOMADO COMPLETAMENTE POR USTED. SI EL SOFTWARE MUESTRA ALGUN DEFECTO, USTED CUBRIRA LOS COSTOS DE CUALQUIER SERVICIO, REPARACION O CORRECCION DE SUS EQUIPOS Y/O SOFTWARE QUE REQUIERA

12. EN NINGUN CASO NI BAJO NINGUNA CIRCUNSTANCIA EXCEPTO BAJO SOLICITUD DE LA LEY O DE COMUN ACUERDO POR ESCRITO, NINGUN PROPIETARIO DE LOS DERECHOS DE AUTOR NI TERCERAS PERSONAS QUE PUDIERAN MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA COMO SE PERMITE ARRIBA, SERAN RESPONSABLES DE LOS DAÑOS CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA, SIN IMPORTAR SI SON DAÑOS GENERALES, ESPECIALES, INCIDENTALES O CONSECUENTES CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA (INCLUYENDO PERO NO LIMITADO A LA PERDIDA DE INFORMACION O DETERIORO DE LA MISMA AFECTANDOLO A USTED, A TERCERAS PERSONAS QUE SEA POR FALLAS EN LA OPERACION DEL PROGRAMA O SU INTERACCION CON OTROS PROGRAMAS) INCLUSIVE SI TAL PROPIETARIO U OTRAS PERSONAS HAYAN SIDO NOTIFICADAS DE TALES FALLAS Y DE LA POSIBILIDAD DE TALES DAÑOS.”

VII.II.III. Exclusión de responsabilidad

El proyectista, fabricante e instalador no se harán responsables de las posibles reclamaciones u daños ocasionados por el programa.

VII.II.IV. Formalización del contrato.

El contrato debe ser redactado, leído y firmado por fabricante y comprador antes del comienzo de la realización de las tareas.

Ambas partes han de estar conformes en el contenido del contrato, así como en las responsabilidades que se deriven de tal contrato las cuales están expuestas en el pliego de condiciones del presente proyecto.

VII.II.V. Rescisión de contrato.

El contrato puede ser rescindido por cualquiera de las causas reconocidas como válidas en las cláusulas del mismo o en la vigente legislación. Toda diferencia o falta de acuerdo en el cumplimiento del contrato será resuelta por vía judicial, pudiendo no obstante, si ambas partes convienen en ello, acatar el fallo dictado por un tercer perito o tribunal nombrado a tal efecto.

Podrán ser causas de resolución del contrato unilateralmente por parte del comprador, sin que medie indemnización ninguna al proyectista, fabricante ni instalador en el caso de que estos cometiesen reincidencia en alguna de las faltas que a continuación se exponen:

- Si no se respetasen las prescripciones de la oferta.
- Si no mantuvieran sus compromisos en la realización de la aplicación proyectada.
- Si no cumpliera cualquiera de las restantes especificaciones acordadas.
- Desatención de las medidas de seguridad en el trabajo.
- El incumplimiento de las leyes laborales vigentes, en especial, el impago de impuestos y seguros sociales.
- Incumplimiento de la licencia software.

VII.II.VI. Plazo de ejecución

No hay plazo de ejecución establecido aunque se podrá acordar uno en cuyo caso se hará constar por escrito como parte del contrato los términos del acuerdo.

VII.II.VII. Limitación de responsabilidad

El fabricante no se hace cargo de cualquier perjuicio que pueda acarrear el mal uso del software y la documentación que le acompaña. Así como tampoco el perjuicio o daño a terceros. Se quiere dejar constancia así pues de que el mal uso de esta solución no es responsabilidad del fabricante.

VII.III. Normas, leyes y reglamentos

Este proyecto se suministrara con un contrato de licencia o un contrato de confidencialidad. Será ilícita la copia del software por cualquier medio distinto al específicamente permitido en el contrato. Por ello el cliente solo puede tener una copia de seguridad del software.

El sistema se adapta a la normativa vigente por la Ley Orgánica sobre Tratamiento Automatizado de Datos y sobre la Ley de Internet, manteniendo la seguridad y confidencialidad con respecto a la información que se encuentra almacenada en la base de datos de la aplicación de estar la base de datos implementada.

Es el motivo de citar dicha ley por si en un futuro se hiciera necesaria la incorporación de una base de datos en la aplicación.

VII.IV. Condiciones contractuales.

VII.IV.I. Compatibilidad y relación entre documentos.

Todo lo indicado en el documento Pliego de condiciones va a prevalecer sobre lo señalado en cualquier otro documento del proyecto aunque este omita, duplique o contradiga cualquier contenido del resto de dicho documento.

VII.IV.II. Representante del contratista.

El contratista deberá designar un técnico que asuma la dirección y control de la ejecución de este proyecto, para lo cual deberá tener poder de decisión.

VII.IV.III. Interpretación del proyecto.

Se entiende en el presente Pliego que el contratista estará capacitado para la perfecta interpretación y ejecución del proyecto en todas sus partes, o que, en su defecto, dispondrá de personal a su servicio para tal fin.

VII.IV.IV. Modificaciones del diseño del proyecto.

En ninguno de los casos el contratista podrá introducir modificaciones de ningún tipo al ejecutar el diseño señalado en este proyecto sin tener la debida aprobación y correspondiente autorización para ejecutarlas, debiendo ser ésta extendida por escrito y debidamente firmada en el libro de órdenes por el cliente a quien vaya destinado el diseño del proyecto.

VII.IV.V. Condiciones del proyecto.

Para la ejecución del proyecto, antes del comienzo del mismo deberá existir un contrato por escrito entre la propiedad y el contratante.

Se sobreentiende que contratista y cliente admiten íntegramente el contenido de este pliego de condiciones.

VII.IV.VI. Revisión o alteración del proyecto.

El contrato podrá ser rescindido por cualquiera de las causas que se especifiquen en la actual legislación.

Toda diferencia o falta de acuerdo en el cumplimiento del contrato se regirá por lo determinado en el Reglamento General de Contratación del Estado.

VII.V. Condiciones técnicas y materiales

El software deberá ser almacenado en un dispositivo digital no haciéndose responsable el fabricante ni los desarrolladores del mal almacenamiento. Por ello es recomendación la creación de una copia de seguridad.

De haber un soporte manuscrito es responsabilidad del usuario la preservación del mismo.

El funcionamiento solo está garantizado para las condiciones indicadas en el manual de usuario y no es responsabilidad el no acatamiento de las mismas.

VII.VI. Exclusividad del proyecto

La exclusividad del proyecto esta acotada por los términos de la licencia los cuales se citan aquí para su mejor comprensión.

“1. *Usted puede copiar y distribuir copias fieles del código fuente del programa tal como lo recibió, en cualquier medio, siempre que proporcione de manera consciente y apropiada una nota de derechos de autor y una declaración de no garantía, además de mantener intactas todas las notas que se refieran a esta licencia y a la ausencia de garantía, y que le proporcione a las demás personas que reciban el programa una copia de esta licencia junto con el Programa. Usted puede aplicar un cargo por el acto físico de transferir una copia, y ofrecer protección de garantía por una cuota, lo cual no compromete a que el autor original del Programa responda por tal efecto.*

2. *Usted puede modificar su copia del Programa o de cualquier parte de él, formando así un trabajo basado en el Programa, y copiar y distribuir tales modificaciones o bien trabajar bajo los términos de la sección 1 arriba descrita, siempre que cumpla con las siguientes condiciones:*

- *Usted debe incluir en los archivos modificados notas declarando que modificó dichos archivos y la fecha de los cambios.*
- *Usted debe notificar que ese trabajo que distribuye contiene totalmente o en partes al Programa, y que debe ser licenciado como un conjunto sin cargo alguno a cualquier otra persona que reciba sus modificaciones bajo los términos de esta Licencia.*
- *Si el programa modificado lee normalmente comandos interactivamente cuando es ejecutado, usted debe presentar un aviso, cuando el programa inicie su ejecución en ese modo interactivo de la forma más ordinaria, que contenga una noticia de derechos de autor y un aviso de que no existe garantía alguna (o que sí existe si es que usted la proporciona) y que los usuarios pueden redistribuir el programa bajo esas condiciones, e informando al usuario como puede ver una copia de esta Licencia. (Excepción: si el programa en sí es interactivo pero normalmente no muestra notas, su trabajo basado en el Programa no tiene la obligación de mostrar tales notas)*

Estos requerimientos aplican al trabajo modificado como un todo. Si existen secciones identificables de tal trabajo que no son derivadas del Programa original, y pueden ser razonablemente consideradas trabajos separados e independientes como tal, entonces esta Licencia y sus términos no aplican a dichas secciones cuando usted las distribuye como trabajos separados. Pero cuando usted distribuye las mismas secciones como parte de un todo que es un trabajo basado en el Programa, la distribución del conjunto debe ser bajo los términos de esta Licencia, cuyos permisos para otras personas que obtengan

el software se extienden para todo el software, así como para cada parte de él, independientemente de quién lo escribió

No es la intención de esta sección de reclamar derechos o pelear sus derechos sobre trabajos hechos enteramente por usted, en lugar de eso, la intención es ejercer el derecho de controlar la distribución de los trabajos derivados o colectivos basados en el Programa. Adicionalmente, el simple agregado de otro trabajo NO basado en el Programa al Programa en cuestión (o a un trabajo basado en el Programa) en algún medio de almacenamiento no pone el otro trabajo bajo el alcance de esta Licencia

3. *Usted puede copiar y distribuir el Programa (o un trabajo basado en él, bajo la Sección 2) en código objeto o en forma de ejecutable bajo los términos de las secciones 1 y 2 arriba descritas siempre que cumpla los siguientes requisitos:*

- *Acompañarlo con el correspondiente código fuente legible por la máquina, que debe ser distribuido bajo los términos de las secciones 1 y 2 y en un medio comúnmente utilizado para el intercambio de software, o*
- *Acompañarlo con una oferta escrita, válida por al menos 3 años y para cualquier persona, por un cargo no mayor al costo que conlleve la distribución física del código fuente correspondiente en un medio comúnmente utilizado para el intercambio de software, o*
- *Acompañarlo con la información que usted recibió sobre la oferta de distribución del código fuente correspondiente. (Esta alternativa está permitida sólo para distribución no-comercial y sólo si usted recibió el Programa en código objeto o en forma de ejecutable con tal oferta de acuerdo a la subsección b anterior*

El código fuente de un trabajo significa la forma preferida de hacer modificaciones al mismo. Para un trabajo ejecutable, un código fuente completo significa todo el código fuente de todos los módulos que contiene, mas cualquier archivo de definición de interfases, mas los programas utilizados para controlar la compilación y la instalación del ejecutable.

Sin embargo, como excepción especial, no se requiere que el código fuente distribuido incluya cualquier cosa que no sea normalmente distribuida con las componentes mayores (compilador, kernel, etc.) del sistema operativo en el cual el ejecutable corre, a menos de que una componente en particular acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace ofreciendo acceso a copiar desde un lugar designado, entonces el ofrecer acceso equivalente para copiar el código fuente desde el mismo lugar se considera distribución del

código fuente, aunque las demás personas no copien el código fuente junto con el código objeto.

4. *Usted no puede copiar, modificar, sub-licenciar ni distribuir el Programa a menos que sea expresamente bajo esta Licencia, de otra forma cualquier intento de copiar, modificar, sub-licenciar o distribuir el programa es nulo, y automáticamente causará la pérdida de sus derechos bajo esta Licencia. Sin embargo, cualquier persona que haya recibido copias o derechos de usted bajo esta Licencia no verá terminadas sus Licencias ni sus derechos perdidos mientras ellas continúen cumpliendo los términos de esta Licencia.”*

VII.VII. Plazos de entrega

No hay compromiso en los plazos de entrega así como penalizaciones. Excepto que se llegue a un acuerdo por parte de las partes en cuyo caso se hará constar por escrito como parte del contrato.

VII.VIII. Derechos de autor

El autor se reserva todos los derechos así como la libertad de realizar modificaciones sin aviso a los usuarios.

VII.IX. Licencia de software

TERMINOS Y CONDICIONES PARA COPIA, MODIFICACION Y DISTRIBUCION

0. Esta licencia aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta "GPL General Public License". El "Programa", utilizado en lo subsiguiente, se refiere a cualquier programa o trabajo original, y el "trabajo basado en el Programa" significa ya sea el Programa o cualquier trabajo derivado del mismo bajo la ley de derechos de autor: es decir, un trabajo que contenga el Programa o alguna porción de él, ya sea íntegra o con modificaciones y/o traducciones a otros idiomas. De aquí en adelante "traducción" estará incluida (pero no limitada a) en el término "modificación", y la persona a la que se aplique esta licencia será llamado "usted". Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance. El acto de ejecutar el programa no está restringido, y la salida de información del programa está cubierto sólo si su contenido constituye un trabajo basado en el Programa (es independiente de si fue resultado de ejecutar el programa). Si esto es cierto o no depende de la función del programa.

1. Usted puede copiar y distribuir copias fieles del código fuente del programa tal como lo recibió, en cualquier medio, siempre que proporcione de manera consciente y apropiada una nota de derechos de autor y una declaración de no garantía, además de mantener intactas todas las notas que se refieran a esta licencia y a la ausencia de garantía, y que le proporcione a las demás personas que reciban el programa una copia de esta licencia junto con el Programa. Usted puede aplicar un cargo por el acto físico de transferir una copia, y ofrecer protección de garantía por una cuota, lo cual no compromete a que el autor original del Programa responda por tal efecto.

2. Usted puede modificar su copia del Programa o de cualquier parte de él, formando así un trabajo basado en el Programa, y copiar y distribuir tales modificaciones o bien trabajar bajo los términos de la sección 1 arriba descrita, siempre que cumpla con las siguientes condiciones:

- Usted debe incluir en los archivos modificados notas declarando que modificó dichos archivos y la fecha de los cambios.
- Usted debe notificar que ese trabajo que distribuye contiene totalmente o en partes al Programa, y que debe ser licenciado como un conjunto sin cargo alguno a cualquier otra persona que reciba sus modificaciones bajo los términos de esta Licencia.
- Si el programa modificado lee normalmente comandos interactivamente cuando es ejecutado, usted debe presentar un aviso, cuando el programa

inicie su ejecución en ese modo interactivo de la forma más ordinaria, que contenga una noticia de derechos de autor y un aviso de que no existe garantía alguna (o que sí existe si es que usted la proporciona) y que los usuarios pueden redistribuir el programa bajo esas condiciones, e informando al usuario como puede ver una copia de esta Licencia. (Excepción: si el programa en sí es interactivo pero normalmente no muestra notas, su trabajo basado en el Programa no tiene la obligación de mostrar tales notas)

Estos requerimientos aplican al trabajo modificado como un todo. Si existen secciones identificables de tal trabajo que no son derivadas del Programa original, y pueden ser razonablemente consideradas trabajos separados e independientes como tal, entonces esta Licencia y sus términos no aplican a dichas secciones cuando usted las distribuye como trabajos separados. Pero cuando usted distribuye las mismas secciones como parte de un todo que es un trabajo basado en el Programa, la distribución del conjunto debe ser bajo los términos de esta Licencia, cuyos permisos para otras personas que obtengan el software se extienden para todo el software, así como para cada parte de él, independientemente de quién lo escribió

No es la intención de esta sección de reclamar derechos o pelear sus derechos sobre trabajos hechos enteramente por usted, en lugar de eso, la intención es ejercer el derecho de controlar la distribución de los trabajos derivados o colectivos basados en el Programa. Adicionalmente, el simple agregado de otro trabajo NO basado en el Programa al Programa en cuestión (o a un trabajo basado en el Programa) en algún medio de almacenamiento no pone el otro trabajo bajo el alcance de esta Licencia

3. *Usted puede copiar y distribuir el Programa (o un trabajo basado en él, bajo la Sección 2) en código objeto o en forma de ejecutable bajo los términos de las secciones 1 y 2 arriba descritas siempre que cumpla los siguientes requisitos:*

- *Acompañarlo con el correspondiente código fuente legible por la máquina, que debe ser distribuido bajo los términos de las secciones 1 y 2 y en un medio comúnmente utilizado para el intercambio de software, o*
- *Acompañarlo con una oferta escrita, válida por al menos 3 años y para cualquier persona, por un cargo no mayor al costo que conlleve la distribución física del código fuente correspondiente en un medio comúnmente utilizado para el intercambio de software, o*
- *Acompañarlo con la información que usted recibió sobre la oferta de distribución del código fuente correspondiente. (Esta alternativa está permitida sólo para distribución no-comercial y sólo si usted recibió el*

Programa en código objeto o en forma de ejecutable con tal oferta de acuerdo a la subsección b anterior

El código fuente de un trabajo significa la forma preferida de hacer modificaciones al mismo. Para un trabajo ejecutable, un código fuente completo significa todo el código fuente de todos los módulos que contiene, mas cualquier archivo de definición de interfases, mas los programas utilizados para controlar la compilación y la instalación del ejecutable.

Sin embargo, como excepción especial, no se requiere que el código fuente distribuido incluya cualquier cosa que no sea normalmente distribuida con las componentes mayores (compilador, kernel, etc.) del sistema operativo en el cual el ejecutable corre, a menos de que una componente en particular acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace ofreciendo acceso a copiar desde un lugar designado, entonces el ofrecer acceso equivalente para copiar el código fuente desde el mismo lugar se considera distribución del código fuente, aunque las demás personas no copien el código fuente junto con el código objeto.

4. *Usted no puede copiar, modificar, sub-licenciar ni distribuir el Programa a menos que sea expresamente bajo esta Licencia, de otra forma cualquier intento de copiar, modificar, sub-licenciar o distribuir el programa es nulo, y automáticamente causará la pérdida de sus derechos bajo esta Licencia. Sin embargo, cualquier persona que haya recibido copias o derechos de usted bajo esta Licencia no verá terminadas sus Licencias ni sus derechos perdidos mientras ellas continúen cumpliendo los términos de esta Licencia.*

5. *Usted no está obligado a aceptar esta Licencia, dado que no la ha firmado. Sin embargo, nada le otorga el permiso de modificar o distribuir el Programa ni sus trabajos derivados. Estas acciones están prohibidas por la ley si usted no acepta esta Licencia. Sin embargo, modificando o distribuyendo el Programa (o cualquier trabajo basado en el Programa) indica su aceptación de esta Licencia y de todos sus términos y condiciones para copiar, distribuir o modificar el Programa y/o trabajos basados en el.*

6. *Cada vez que usted redistribuye el Programa (o cualquier trabajo basado en el Programa), la persona que lo recibe automáticamente recibe una licencia del autor original para copiar, distribuir o modificar el Programa sujeto a estos términos y condiciones. Usted no puede imponer ninguna restricción adicional a las personas que reciban el Programa sobre los derechos que en esta Licencia se les otorga. Usted no es responsable de forzar a terceras personas en el cumplimiento de esta Licencia.*

7. Si como consecuencia de un veredicto de un juzgado o por el alegato de infringir una patente o por cualquier otra razón (no limitado solo a cuestiones de patentes) se imponen condiciones sobre usted que contradigan los términos y condiciones de esta Licencia, éstas no le excusan de los términos y condiciones aquí descritas. Si usted no puede distribuir el producto cumpliendo totalmente con las obligaciones concernientes a la resolución oficial y al mismo tiempo con las obligaciones que se describen en este contrato de Licencia, entonces no podrá distribuir más este producto. Por ejemplo, si una licencia de patente no permitirá la distribución del Programa de forma libre de regalías (sin pago de regalías) por parte de quienes lo reciban directa o indirectamente, entonces la única forma de cumplir con ambas obligaciones es renunciar a la distribución del mismo.

Si cualquier parte de esta sección resulta inválida, inaplicable o no obligatoria bajo cualquier circunstancia en particular, la tendencia de esta es a aplicarse, y la sección completa se aplicará bajo otras circunstancias.

La intención de esta sección no es la de inducirlo a infringir ninguna ley de patentes, ni tampoco infringir algún reclamo de derechos, ni discutir la validez de tales reclamos; esta sección tiene el único propósito de proteger la integridad del sistema de distribución del software libre, que está implementado por prácticas de licencia pública. Mucha gente ha hecho generosas contribuciones a la amplia gama de software distribuido bajo este sistema favoreciendo así la constante aplicación de este sistema de distribución; es decisión del autor/donador si su Programa será distribuido utilizando este u otro sistema de distribución, y la persona que recibe el software no puede obligarlo a hacer ninguna elección en particular

Esta sección pretende dejar muy en claro lo que se cree que será una consecuencia del resto de esta Licencia.

8. Si la distribución y/o el uso del Programa se restringe a algunos países ya sea por patentes, interfaces protegidas por derechos de autor, el propietario original de los derechos de autor que ubica su Programa bajo esta Licencia puede agregar una restricción geográfica de distribución explícita excluyendo los países que aplique, dando como resultado que su distribución sólo se permita en los países no excluidos. En tal caso, esta Licencia incorpora la limitación como si hubiera sido escrita en el cuerpo de esta misma Licencia.

9. La "FSF Free Software Foundation" puede publicar versiones nuevas o revisadas de la "GPL General Public License" de uno a otro momento. Estas nuevas versiones mantendrán el espíritu de la presente versión, pero pueden diferir en la inclusión de nuevos problemas o en la manera de tocar los problemas o aspectos ya presentes.

Cada versión tendrá un número de versión que la distinga. Si el Programa especifica un número de versión para esta Licencia que aplique a él y "cualquier versión subsiguiente", usted tiene la opción de seguir los términos y condiciones de dicha versión o de cualquiera de las posteriores versiones publicadas por la "FSF". Si el programa no especifica una versión en especial de esta Licencia, usted puede elegir entre cualquiera de las versiones que han sido publicadas por la "FSF".

10. Si usted desea incorporar partes del Programa en otros Programas de software libre cuyas condiciones de distribución sean distintas, deberá escribir al autor solicitando su autorización. Para programas de software protegidas por la "FSF Free Software Foundation", deberá escribir a la "FSF" solicitando autorización, en ocasiones hacemos excepciones. Nuestra decisión será guiada por dos metas principales:

- Mantener el estado de libertad de todos los derivados de nuestro software libre.
- Promover el uso comunitario y compartido del software en general.

NO EXISTE GARANTIA ALGUN

11. DEBIDO A QUE EL PROGRAMA SE OTORGA LIBRE DE CARGOS Y REGALIAS, NO EXISTE NINGUNA GARANTIA PARA EL MISMO HASTA DONDE LO PERMITA LA LEY APLICABLE. A EXCEPCION DE QUE SE INDIQUE OTRA COSA, LOS PROPIETARIOS DE LOS DERECHOS DE AUTOR PROPORCIONAN EL PROGRAMA "COMO ES" SIN NINGUNA GARANTIA DE NINGUN TIPO, YA SEA EXPLICITA O IMPLICITA, INCLUYENDO, PERO NO LIMITADA A, LAS GARANTIAS QUE IMPLICA EL MERCADEO Y EJERCICIO DE UN PROPOSITO EN PARTICULAR. CUALQUIER RIESGO DEBIDO A LA CALIDAD Y DESEMPEÑO DEL PROGRAMA ES TOMADO COMPLETAMENTE POR USTED. SI EL SOFTWARE MUESTRA ALGUN DEFECTO, USTED CUBRIRA LOS COSTOS DE CUALQUIER SERVICIO, REPARACION O CORRECCION DE SUS EQUIPOS Y/O SOFTWARE QUE REQUIERA

12. EN NINGUN CASO NI BAJO NINGUNA CIRCUNSTANCIA EXCEPTO BAJO SOLICITUD DE LA LEY O DE COMUN ACUERDO POR ESCRITO, NINGUN PROPIETARIO DE LOS DERECHOS DE AUTOR NI TERCERAS PERSONAS QUE PUDIERAN MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA COMO SE PERMITE ARRIBA, SERAN RESPONSABLES DE LOS DAÑOS CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA, SIN IMPORTAR SI SON DAÑOS GENERALES, ESPECIALES, INCIDENTALES O CONSECUENTES CORRESPONDIENTES

AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA (INCLUYENDO PERO NO LIMITADO A LA PERDIDA DE INFORMACION O DETERIORO DE LA MISMA AFECTANDOLO A USTED, A TERCERAS PERSONAS QUE SEA POR FALLAS EN LA OPERACION DEL PROGRAMA O SU INTERACCION CON OTROS PROGRAMAS) INCLUSIVE SI TAL PROPIETARIO U OTRAS PERSONAS HAYAN SIDO NOTIFICADAS DE TALES FALLAS Y DE LA POSIBILIDAD DE TALES DAÑOS.

FIN DE TERMINOS Y CONDICIONE

Anexo VIII. RAZONES GEOMÉTRICAS

VIII.I. Preámbulo

Aquí se explicaran las razones geométricas empleadas en los cálculos.

VIII.II. Distancia de un punto a una recta

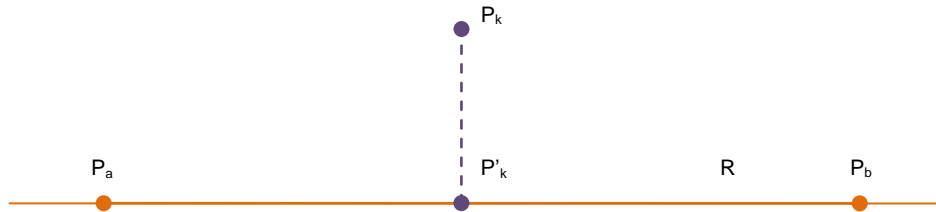


Ilustración 104 Distancia de un punto a una recta

VIII.II.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el segmento}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un punto de la recta}\}$

$P_b = (x_b, y_b, z_b) = \{\text{Otro punto de la recta distinto a } P_a\}$

$R_{ab} = \{\text{Recta que pasa por } P_a \text{ y } P_b\}$

VIII.II.II. Cálculos

$\vec{v} = \overrightarrow{P_{ab}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_b\}$

$\vec{u} = \overrightarrow{P_{ka}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_k \text{ con } P_a\}$

$$\text{Distancia}(L, P_k) = \frac{|\vec{v} \times \vec{u}|}{|\vec{u}|}$$

VIII.III. Distancia de un punto a un segmento

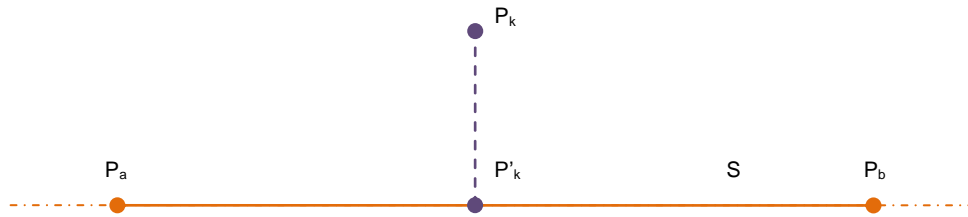


Ilustración 105 Distancia de un punto a un segmento

VIII.III.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el segmento}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un extremos del segmento}\}$

$P_b = (x_b, y_b, z_b) = \{\text{El otro extremo del segmento}\}$

$S_{ab} = \{\text{Segmento que tiene como extremos a } P_a \text{ y } P_b\}$

VIII.III.II. Cálculos

$\vec{v} = \overrightarrow{P_{ab}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_b\}$

$\vec{u} = \overrightarrow{P_{ka}} = (x_u, y_u, z_u) = \{\text{Vector que une } P_k \text{ con } P_a\}$

$\vec{w} = \overrightarrow{P_{kb}} = (x_w, y_w, z_w) = \{\text{Vector que une } P_k \text{ con } P_b\}$

t

$$= \frac{\vec{v} \cdot \vec{u}}{\vec{v} \cdot \vec{v}} \begin{cases} \text{Si } t < 0 \Rightarrow P_k \text{ cae a la izquierda de } P_a \text{ por lo que la distancia sería } \Rightarrow |\overrightarrow{P_{ka}}| \\ \text{Si } t > 0 \Rightarrow P_k \text{ cae a la derecha de } P_b \text{ por lo que la distancia sería } \Rightarrow |\overrightarrow{P_{kb}}| \\ \text{Si } \left\{ \begin{matrix} t \leq 1 \\ t \geq 0 \end{matrix} \right\} \text{ cae en el segmento calculamos la distancia a la recta } \Rightarrow \frac{|\vec{v} \times \vec{u}|}{|\vec{u}|} \end{cases}$$

VIII.IV. Distancia de un punto a un plano

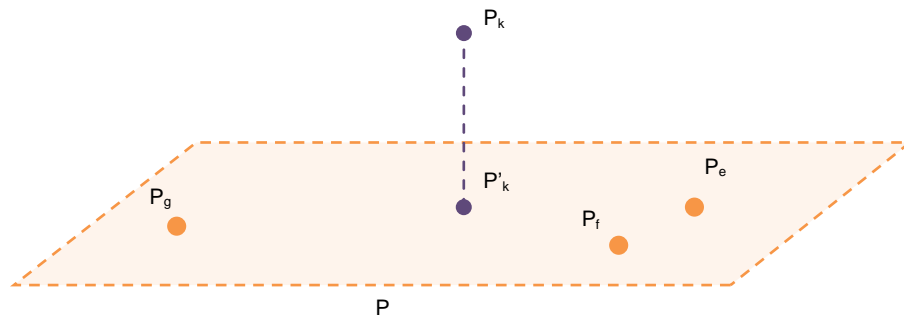


Ilustración 106 Distancia de un punto a un plano

VIII.IV.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el plano}\}$

$a, b, c, d = \{\text{Coeficientes de la ecuación implícita del plano}\}$

$P_e = (x_e, y_e, z_e) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_g\}$

$P_f = (x_f, y_f, z_f) = \{\text{Un punto cualquiera del plano distinto de } P_e \text{ y } P_g\}$

$P_g = (x_g, y_g, z_g) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_e\}$

$P_{efgh} = \left\{ \begin{array}{l} \text{Plano que contiene los puntos } P_e, P_f \text{ y } P_g \text{ cuya ecuación implícita} \\ \text{de coeficientes } a, b, c, d \text{ es } \Rightarrow 0 = ax + by + cz + d \end{array} \right\}$

VIII.IV.II. Cálculos

$$\text{Distancia}(P_k, P) = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}}$$

Sustituimos las coordenadas x, y, z por las del punto:

$$\text{Distancia}(P_k, P) = \frac{ax_k + by_k + cz_k + d}{\sqrt{a^2 + b^2 + c^2}}$$

VIII.V. Distancia de un punto a un triángulo

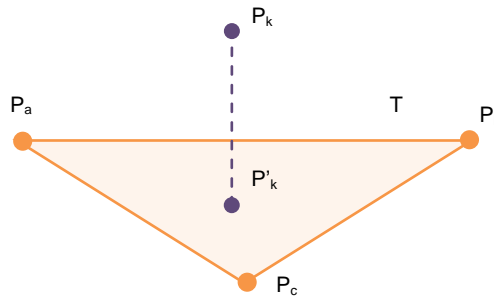


Ilustración 107 Distancia de un punto a un triángulo

VIII.V.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el triángulo}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un vertice del triángulo distinto de } P_b \text{ y } P_c\}$

$P_b = (x_b, y_b, z_b) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_c\}$

$P_c = (x_c, y_c, z_c) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_b\}$

$T_{abc} = \widehat{P_a P_b P_c} = \{\text{Triángulo cuyos vertices son } P_a, P_b \text{ y } P_c\}$

VIII.V.II. Cálculos

$$\text{Distancia}(P_k, T) = \left| \overrightarrow{P_k P'_k} \right|$$

Para el cálculo de P'_k ver el cálculo de la proyección de un punto sobre un triángulo en este mismo apartado.

VIII.VI. Distancia de un punto a un cuadrado

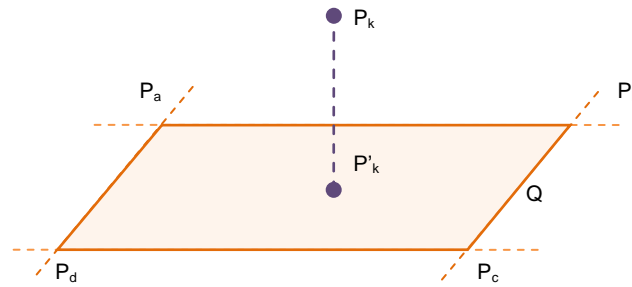


Ilustración 108 Distancia de un punto a un cuadrado

VIII.VI.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el triángulo}\}$

$P_a, P_b, P_c, P_d = \{\text{Los vertices del cuadrado}\}$

$Q_{abcd} = \{\text{Cuadrado cuyos vertices son } P_a, P_b, P_c \text{ y } P_d\}$

VIII.VI.II. Cálculos

$\text{Distancia}(P_k, Q) = \text{Min}(\text{Distancia}(P_k, \widehat{P_a P_b P_c}), \text{Distancia}(P_k, \widehat{P_a P_d P_c}))$

Para el cálculo de la distancia de los triángulos ver el cálculo de la distancia a un triángulo.

VIII.VII. Distancia de un punto a un cubo

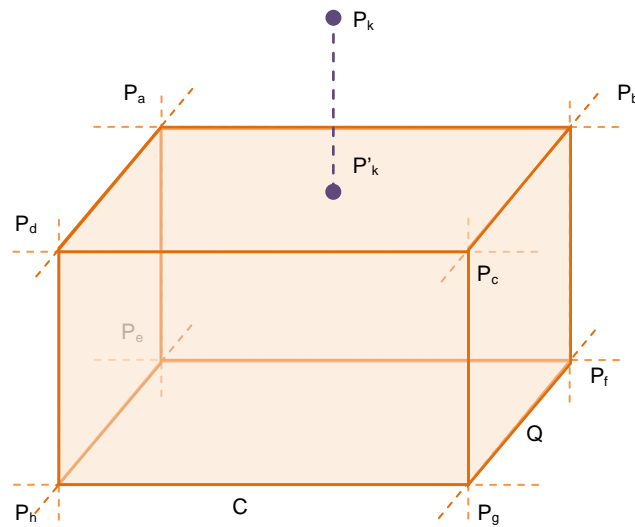


Ilustración 109 Distancia de un punto a un cubo

VIII.VII.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el Cuadrado}\}$

$P_a, P_b, P_c, P_d, P_e, P_f, P_g, P_h = \{\text{Los vertices del cuadrado}\}$

$C = \{\text{Cubocuyos vertices son } P_a, P_b, P_c, P_d, P_e, P_f, P_g \text{ y } P_h\}$

VIII.VII.II. Cálculos

Primero comprobamos si el punto está en el interior del cubo.

Para saber si el punto está contenido en el cubo ver el cálculo de la intersección de un punto con un cubo.

Si el punto esta contenido la distancia será cero, sino calculamos las distancias a cada una de sus caras y nos quedamos con la distancia menor.

$$Distancia(P_k, C) = \text{Min} \begin{pmatrix} Distancia(P_k, Q_{abcd}), \\ Distancia(P_k, Q_{abfe}), \\ Distancia(P_k, Q_{aedh}), \\ Distancia(P_k, Q_{gcbf}), \\ Distancia(P_k, Q_{gcdh}), \\ Distancia(P_k, Q_{gfeh}) \end{pmatrix}$$

Para el cálculo de la distancia de los triángulos ver el cálculo de la distancia a un triángulo.

VIII.VIII. Proyección de un punto a un plano

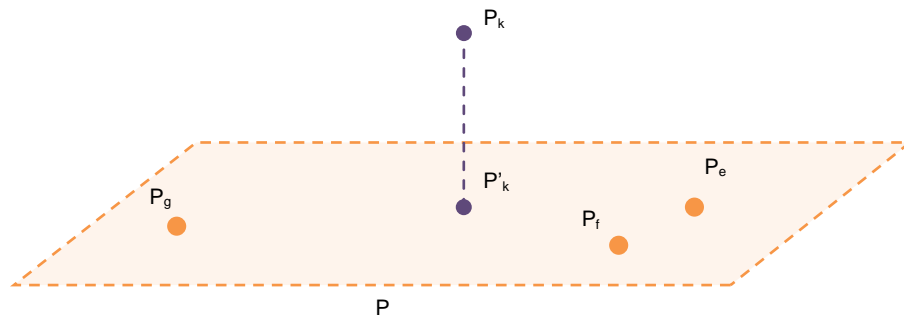


Ilustración 110 Proyección de un punto a un plano

VIII.VIII.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el plano}\}$

$a, b, c, d = \{\text{Coeficientes de la ecuación implícita del plano}\}$

$P_e = (x_e, y_e, z_e) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_g\}$

$P_f = (x_f, y_f, z_f) = \{\text{Un punto cualquiera del plano distinto de } P_e \text{ y } P_g\}$

$P_g = (x_g, y_g, z_g) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_e\}$

$P_{efgh} = \left\{ \begin{array}{l} \text{Plano que contiene los puntos } P_e, P_f \text{ y } P_g \text{ cuya ecuación implícita} \\ \text{de coeficientes } a, b, c, d \text{ es } \Rightarrow 0 = ax + by + cz + d \end{array} \right\}$

VIII.VIII.II. Cálculos

$\vec{n} = (a, b, c) = \{\text{Vector normal al plano}\}$

$\vec{v}_{nk} = (a - x_k, b - y_k, c - z_k) = \{\text{Vector normal al plano con origen en } P_k\}$

Obtenemos la ecuación de la recta que pasa por ese punto perpendicular al plano:

$$0 = ax + by + cz + d \Rightarrow 0 = ax_k + by_k + cz_k + d$$

$$\vec{P}' = \vec{P} + t\vec{v} \Rightarrow \vec{P}'_k = \vec{P}_k + t\vec{v}_{nk}$$

Si despejamos obtenemos:

$$t = -\frac{x_k + y_k + z_k + d}{a + b + c}$$

Sustituimos y obtendremos P'_k .

VIII.IX. Proyección de un punto en un segmento

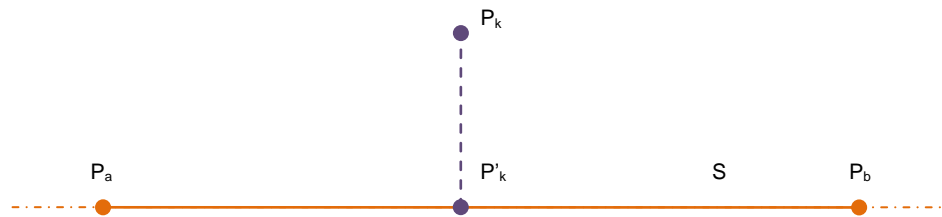


Ilustración 111 Proyección de un punto en un segmento

VIII.IX.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la proyección}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el segmento}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un extremos del segmento}\}$

$P_b = (x_b, y_b, z_b) = \{\text{El otro extremo del segmento}\}$

$S_{ab} = \{\text{Segmento que tiene como extremos a } P_a \text{ y } P_b\}$

VIII.IX.II. Cálculos

$\vec{v} = \overrightarrow{P_{ab}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_b\}$

$\vec{u} = \overrightarrow{P_{ak}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_k\}$

$\vec{w} = \overrightarrow{P_{kb}} = (x_w, y_w, z_w) = \{\text{Vector que une } P_k \text{ con } P_b\}$

$$t = \frac{\vec{v} \cdot \vec{u}}{\vec{v} \cdot \vec{v}} \begin{cases} \text{Si } t < 0 \Rightarrow P_k \text{ cae a la izquierda de } P_a \text{ la proyeccion sería } P_a. \\ \text{Si } t > 1 \Rightarrow P_k \text{ cae a la derecha de } P_b \text{ la proyeccion sería } P_b. \\ \text{Si } \begin{cases} t \leq 1 \\ t \geq 0 \end{cases} \text{ cae en el segmento calculamos la distancia a la recta} \end{cases}$$

$$\Rightarrow P'_k = P_a + \frac{\vec{v} \cdot \vec{u}}{\vec{v} \cdot \vec{v}} \cdot \vec{v} \Rightarrow P_a + t\vec{v}$$

VIII.X. Proyección de un punto en un triángulo

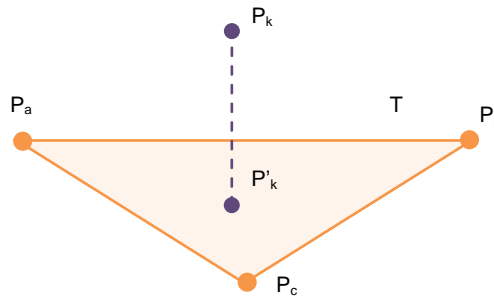


Ilustración 112 Proyección de un punto en un triángulo

VIII.X.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la proyección}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el triángulo}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un vertice del triángulo distinto de } P_b \text{ y } P_c\}$

$P_b = (x_b, y_b, z_b) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_c\}$

$P_c = (x_c, y_c, z_c) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_b\}$

$T_{abc} = \widehat{P_a P_b P_c} = \{\text{Triángulo cuyos vertices son } P_a, P_b \text{ y } P_c\}$

VIII.X.II. Cálculos

$$\text{Distancia}(P_k, T) = \left| \overrightarrow{P_k P'_k} \right|$$

Primero calcularemos la proyección del punto sobre el plano donde esta contenido el triángulo.

Para calcular la proyección de un punto sobre el plano ver el apartado en este mismo capítulo.

Luego veremos si dicha proyección sobre el plano que contiene el triángulo está contenida en el triángulo.

Para ver si la proyección de un punto en el plano hace intersección con el triángulo ver apartado de este mismo capítulo.

Si hace intersección ya tendremos el punto. Si no hace intersección tendremos que ver distancia a cada uno de sus lados.

Para calcular la distancia de un punto en el lado ver apartado de distancia de un punto con un segmento de este mismo capítulo.

Nos quedaremos con la proyección en el lado con menor distancia.

Para calcular la proyección de un punto en el lado ver apartado de proyección de un punto con un segmento de este mismo capítulo.

VIII.XI. Intersección de un punto con una recta

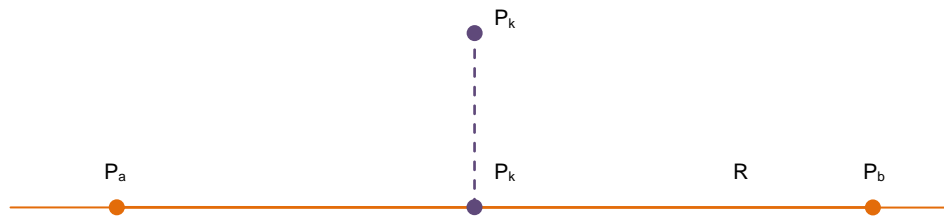


Ilustración 113 Intersección de un punto con una recta

VIII.XI.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la intersección}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el segmento}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un punto de la recta}\}$

$P_b = (x_b, y_b, z_b) = \{\text{Otro punto de la recta distinto a } P_a\}$

$R_{ab} = \{\text{Recta que pasa por } P_a \text{ y } P_b\}$

VIII.XI.II. Cálculos

$\vec{v} = \overrightarrow{P_{ab}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_b\}$

$\vec{u} = \overrightarrow{P_{ka}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_k \text{ con } P_a\}$

$$\text{Distancia}(L, P_k) = \frac{|\vec{v} \times \vec{u}|}{|\vec{u}|}$$

Si la distancia es 0 por lo que $P_k = P'_k$, hace intersección. Si no, no habrá intersección.

VIII.XII. Intersección de un punto con un plano

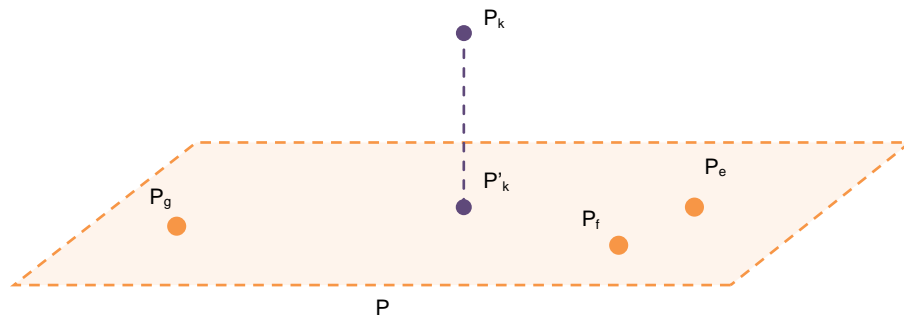


Ilustración 114 Intersección de un punto con un plano

VIII.XII.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la intersección}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el plano}\}$

$a, b, c, d = \{\text{Coeficientes de la ecuación implícita del plano}\}$

$P_e = (x_e, y_e, z_e) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_g\}$

$P_f = (x_f, y_f, z_f) = \{\text{Un punto cualquiera del plano distinto de } P_e \text{ y } P_g\}$

$P_g = (x_g, y_g, z_g) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_e\}$

$P_{efgh} = \left\{ \begin{array}{l} \text{Plano que contiene los puntos } P_e, P_f \text{ y } P_g \text{ cuya ecuación implícita} \\ \text{de coeficientes } a, b, c, d \text{ es } \Rightarrow 0 = ax + by + cz + d \end{array} \right\}$

VIII.XII.II. Cálculos

$$\text{Distancia}(P_k, P) = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}}$$

Sustituimos las coordenadas x, y, z por las del punto:

$$\text{Distancia}(P_k, P) = \frac{ax_k + by_k + cz_k + d}{\sqrt{a^2 + b^2 + c^2}}$$

Si la distancia es 0 por lo que $P_k = P'_k$, hace intersección. Si no, no habrá intersección.

VIII.XIII. Intersección de un punto con un segmento

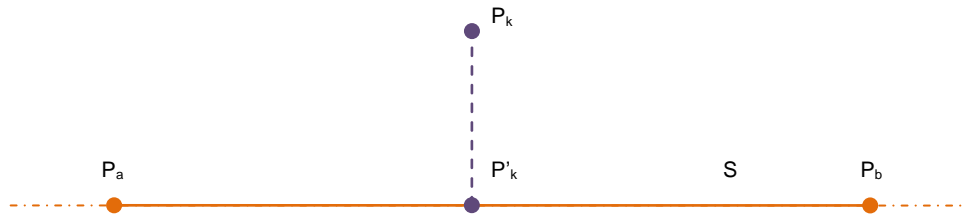


Ilustración 115 Intersección de un punto con un segmento

VIII.XIII.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el segmento}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un extremos del segmento}\}$

$P_b = (x_b, y_b, z_b) = \{\text{El otro extremo del segmento}\}$

$S_{ab} = \{\text{Segmento que tiene como extremos a } P_a \text{ y } P_b\}$

VIII.XIII.II. Cálculos

$\vec{v} = \overrightarrow{P_{ab}} = (x_v, y_v, z_v) = \{\text{Vector que une } P_a \text{ con } P_b\}$

$\vec{u} = \overrightarrow{P_{ka}} = (x_u, y_u, z_u) = \{\text{Vector que une } P_k \text{ con } P_a\}$

$\vec{w} = \overrightarrow{P_{kb}} = (x_w, y_w, z_w) = \{\text{Vector que une } P_k \text{ con } P_b\}$

t

$$= \frac{\vec{v} \cdot \vec{u}}{\vec{v} \cdot \vec{v}} \begin{cases} \text{Si } t < 0 \Rightarrow P_k \text{ cae a la izquierda de } P_a \text{ por lo que la distancia sería } \Rightarrow |\overrightarrow{P_{ka}}| \\ \text{Si } t > 0 \Rightarrow P_k \text{ cae a la derecha de } P_b \text{ por lo que la distancia sería } \Rightarrow |\overrightarrow{P_{kb}}| \\ \text{Si } \left\{ \begin{matrix} t \leq 1 \\ t \geq 0 \end{matrix} \right\} \text{ cae en el segmento calculamos la distancia a la recta } \Rightarrow \frac{|\vec{v} \times \vec{u}|}{|\vec{u}|} \end{cases}$$

Si la distancia es 0 por lo que $P_k = P'_k$, hace intersección. Si no, no habrá intersección.

VIII.XIV. Intersección de un punto con un triángulo

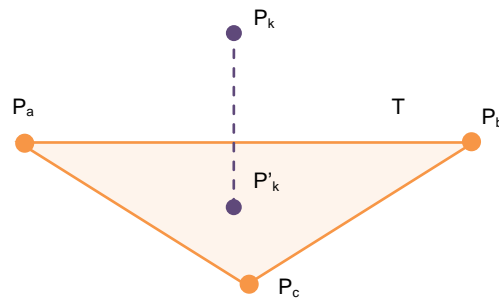


Ilustración 116 Intersección de un punto con un triángulo

VIII.XIV.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la distancia}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el triángulo}\}$

$P_a = (x_a, y_a, z_a) = \{\text{Un vertice del triángulo distinto de } P_b \text{ y } P_c\}$

$P_b = (x_b, y_b, z_b) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_c\}$

$P_c = (x_c, y_c, z_c) = \{\text{Otro vertice del triángulo distinto de } P_a \text{ y } P_b\}$

$T_{abc} = \widehat{P_a P_b P_c} = \{\text{triángulo cuyos vértices son } P_a, P_b \text{ y } P_c\}$

VIII.XIV.II. Cálculos

$$Distancia(P_k, T) = |\overrightarrow{P_k P'_k}|$$

Dado un triángulo comprobamos que la proyección del punto sobre cada uno de los planos formados por los ejes este contenida sobre la proyección del triángulo sobre cada uno de los planos formado por los ejes.

Para ello calculamos la orientación del triángulo proyectado sobre el plano a evaluar.

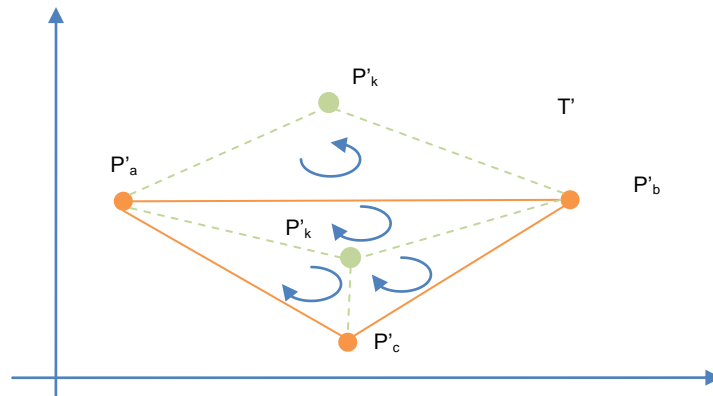


Ilustración 117 Orientación del triángulo

Si la orientación del triángulo **abc** es contraria a la orientación del triángulo **abk** tendremos que el punto se encuentra fuera del interior del triángulo, si es igual este caerá dentro.

Para calcular la orientación calcularemos el área de los triángulos manteniendo el signo para ver si el cálculo sale positivo o negativo.

$$area = |orientacion| = |\overrightarrow{v_{ab}} \times \overrightarrow{u_{bk}}|$$

Si en cada uno de los planos cae en el interior es porque $P_k = P'_k$, hace intersección. Si no, no habrá intersección.

VIII.XV. Intersección de un punto con un cubo ortogonal

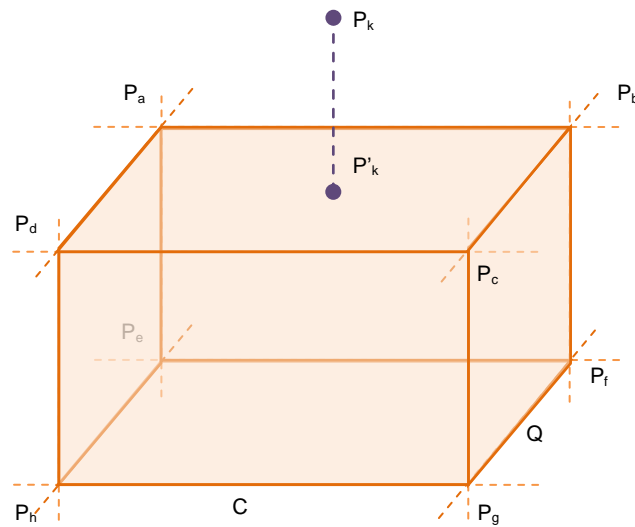


Ilustración 118 Intersección de un punto con un cubo ortogonal

VIII.XV.I. Datos

$P_k = (x_k, y_k, z_k) = \{\text{Punto del que hay que buscar la intersección}\}$

$P'_k = (x'_k, y'_k, z'_k) = \{\text{Proyección de } P_k \text{ sobre el Cuadrado}\}$

$P_a, P_b, P_c, P_d, P_e, P_f, P_g, P_h = \{\text{Los vértices del cuadrado}\}$

$C = \{\text{Cubo cuyos vértices son } P_a, P_b, P_c, P_d, P_e, P_f, P_g \text{ y } P_h\}$

$x_{\max}(C), y_{\max}(C), z_{\max}(C)$

$= \{\text{Valores máximos del cubo para la coordenadas } x, y, z\}$

$x_{\min}(C), y_{\min}(C), z_{\min}(C)$

$= \{\text{Valores mínimos del cubo para la coordenadas } x, y, z\}$

VIII.XV.II. Cálculos

$$\text{si } \left\{ \begin{array}{l} x_k \leq x_{\max}(c) \\ x_k \geq x_{\min}(c) \\ y_k \leq y_{\max}(c) \\ y_k \geq y_{\min}(c) \\ z_k \leq z_{\max}(c) \\ z_k \geq z_{\min}(c) \end{array} \right\} \text{ entonces esta dentro.}$$

Si cae en el interior es porque $P_k = P'_k$, y hace intersección.

VIII.XVI. Intersección de una recta con una recta

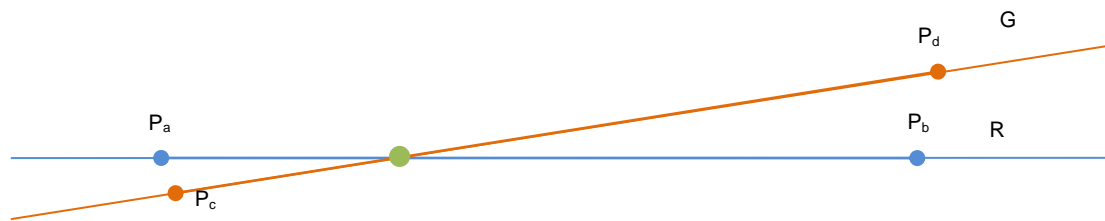


Ilustración 119 Intersección de una recta con una recta

VIII.XVI.I. Datos

$P_a, P_b, = \{\text{Dos puntos distintos de la recta } R\}$

$P_c, P_d, = \{\text{Dos puntos distintos de la recta } G\}$

$G = \{\text{Recta que pasa por } P_c, P_d\}$

$R = \{\text{Recta que pasa por } P_a, P_b\}$

VIII.XVI.II. Cálculos

Pueden darse tres casos:

Que se corten en un punto.

Que no se corten, con lo que las rectas serían coplanarias.

Que coincidan en más de un punto, con lo que todos los puntos de una recta estarían contenidos en la otra.

Para poder dilucidar en cuál de los casos nos encontramos igualaremos las ecuaciones intrínsecas de la recta.

$\overrightarrow{v_{ab}} = \{\text{Vector director de la recta } R\}$

$\overrightarrow{u_{cd}} = \{\text{Vector director de la recta } G\}$

$\overrightarrow{w_{ac}} = \{\text{Vector con un extremo en cada recta}\}$

Estos tres vectores son coplanarios si el producto mixto entre ellos es nulo (volumen del cuerpo que generan). El producto mixto se calcula como un determinante de grado tres en que cada línea corresponde con las coordenadas de cada vector.

$$\det = \begin{vmatrix} x_v & y_v & z_v \\ x_u & y_u & z_u \\ x_w & y_w & z_w \end{vmatrix} \Rightarrow \det = 0 \text{ son coplanarias}$$

Si no fueran coplanarios podríamos averiguar si los puntos de una recta están contenidos en la otra y así saber si hacen intersección en una recta o un punto. O igualar sus ecuaciones intrínsecas y ver en qué punto coinciden.

VIII.XVII. Intersección de una segmento con un segmento

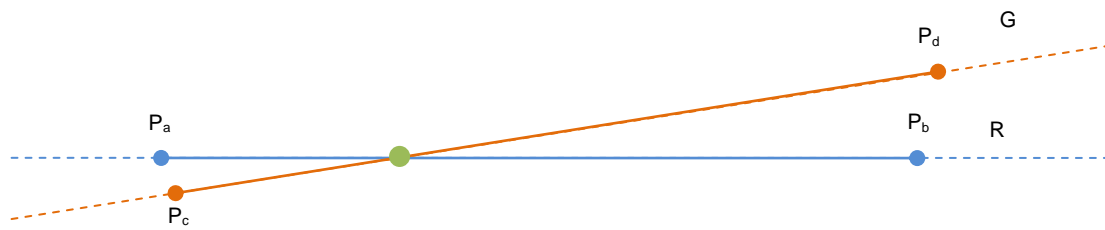


Ilustración 120 Intersección de una segmento con un segmento

VIII.XVII.I. Datos

$P_a, P_b, = \{\text{Dos puntos distintos del segmento } R\}$

$P_c, P_d, = \{\text{Dos puntos distintos del segmento } G\}$

$G = \{\text{Segmento de extremos } P_c, P_d\}$

$R = \{\text{Segmento de extremos } P_a, P_b\}$

VIII.XVII.II. Cálculos

Pueden darse tres casos:

- Que se corten en un punto.
- Que no se corten.
- Que coincidan en más de un punto, con lo que harían intersección en un segmento.

Para poder dilucidar en cuál de los casos nos encontramos igualaremos las ecuaciones intrínsecas de la recta que contienen los segmentos.

En caso de ser coplanarias esto implicaría que a su vez que los segmentos no harían intersección.

Si coinciden en solo un punto, si dicho punto cae fuera de cualquiera de los dos segmentos significaría que no habría intersección. Por otro lado, si el punto coincide pues este sería la intersección de los segmentos.

En caso de que las rectas coincidan, comprobaremos si alguno de los extremos de los segmentos está contenido en el otro, de ser así eso significaría que ambos segmentos coinciden en alguna parte, dando como resultado un segmento de intersección.

VIII.XVIII. Intersección de una recta con un plano

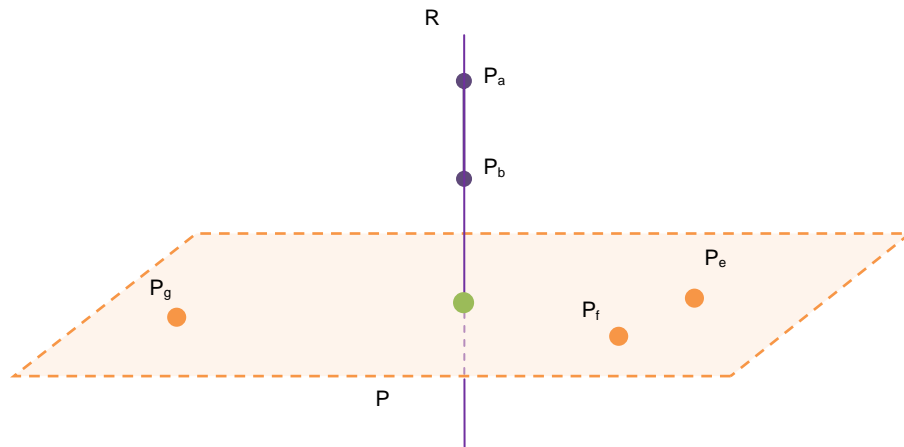


Ilustración 121 Intersección de una recta con un plano

VIII.XVIII.I. Datos

$P_a = \{\text{Punto de la recta } R\}$

$P_b = \{\text{Otro punto de la recta } R, \text{ distinto de } P_a\}$

$P_e = (x_e, y_e, z_e) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_g\}$

$P_f = (x_f, y_f, z_f) = \{\text{Un punto cualquiera del plano distinto de } P_e \text{ y } P_g\}$

$P_g = (x_g, y_g, z_g) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_e\}$

$P_{efgh} = \left\{ \begin{array}{l} \text{Plano que contiene los puntos } P_e, P_f \text{ y } P_g \text{ cuya ecuación implícita} \\ \text{de coeficientes } a, b, c, d \text{ es } \Rightarrow 0 = ax + by + cz + d \end{array} \right\}$

VIII.XVIII.II. Cálculos

$\vec{n} = (a, b, c) = \{\text{Vector normal al plano}\}$

$\vec{v}_{na} = (a - x_a, b - y_a, c - z_a) = \{\text{Vector normal al plano con origen en } P_a\}$

Obtenemos la ecuación de la recta que pasa por ese punto perpendicular al plano:

$$0 = ax + by + cz + d \Rightarrow 0 = ax_a + by_a + cz_a + d$$

$$\vec{P'} = \vec{P} + t\vec{v} \Rightarrow \vec{P'_a} = \vec{P_a} + t\vec{v}_{na}$$

Si despejamos obtenemos:

$$t = -\frac{x_a + y_a + z_a + d}{a + b + c}$$

Evaluamos la ecuación de la recta sustituyendo el parámetro t con el valor obtenido.

Para obtener el punto de corte entre la recta y el plano, si llamamos P_k al punto de intersección con el plano, la ecuación para encontrar dicho punto debe cumplir:

$$0 = ax + by + cz + d \Rightarrow 0 = ax_k + by_k + cz_k + d$$

$$\vec{P'} = \vec{P} + t\vec{v} \Rightarrow \vec{P_k} = \vec{P_a} + t\vec{v_{na}}$$

Si t se encuentra entre el rango comprendido entre 0 y 1, significa que el punto de intersección se encuentra dentro del segmento definido por P_a y P_k .

VIII.XIX. Intersección de un segmento con un plano

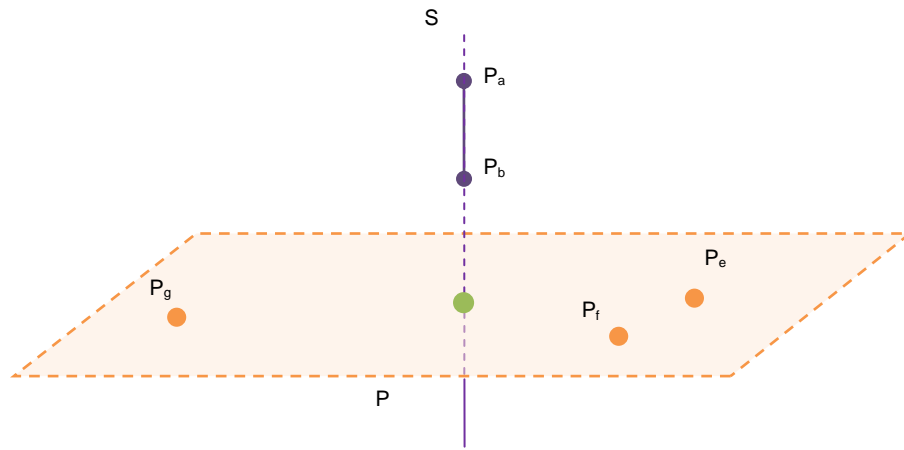


Ilustración 122 Intersección de un segmento con un plano

VIII.XIX.I. Datos

$P_a = \{\text{Exremo del segmento } S\}$

$P_b = \{\text{Otro extremo del segmento } S, \text{ distinto de } P_a\}$

$P_e = (x_e, y_e, z_e) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_g\}$

$P_f = (x_g, y_g, z_f) = \{\text{Un punto cualquiera del plano distinto de } P_e \text{ y } P_g\}$

$P_g = (x_b, y_b, z_b) = \{\text{Un punto cualquiera del plano distinto de } P_f \text{ y } P_e\}$

$P_{efgh} = \left\{ \begin{array}{l} \text{Plano que contiene los puntos } P_e, P_f \text{ y } P_g \text{ cuya ecuación implícita} \\ \text{de coeficientes } a, b, c, d \text{ es } \Rightarrow 0 = ax + by + cz + d \end{array} \right\}$

VIII.XIX.II. Cálculos

Primero buscamos el punto de corte entre la recta y el plano, P_k . La ecuación que defina este punto debe cumplir:

$\vec{n} = (a, b, c) = \{\text{Vector normal al plano}\}$

$\vec{v}_{ab} = \{\text{Vector con origen en } P_a \text{ y fin en } P_b\}$

$$P_k = (x_k, y_k, z_k) \Rightarrow \begin{cases} 0 = ax_k + by_k + cz_k + d \\ \vec{P}_k = \vec{P}_a + t\vec{v}_{ab} \end{cases}$$

Si despejamos obtenemos:

$$t = -\frac{ax_a + by_a + cz_a + d}{ax_v + by_v + cz_v}$$

Si el denominador

$$0 \approx ax_a + by_a + cz_a + d$$

Entonces el segmento es perpendicular al plano o esta sobre el plano, podemos verificar si la línea esta sobre el plano si sustituyendo P_a y P_b obtenemos cero.

$$ax_a + by_a + cz_a + d = 0$$

$$ax_a + by_a + cz_a + d = 0$$

Si t se encuentra entre el rango comprendido entre 0 y 1, significa que el punto de intersección se encuentra dentro del segmento definido por P_a y P_b .

El punto se obtiene sustituyendo t en la ecuación de la recta.

VIII.XX. Intersección de un segmento con un cuadrado

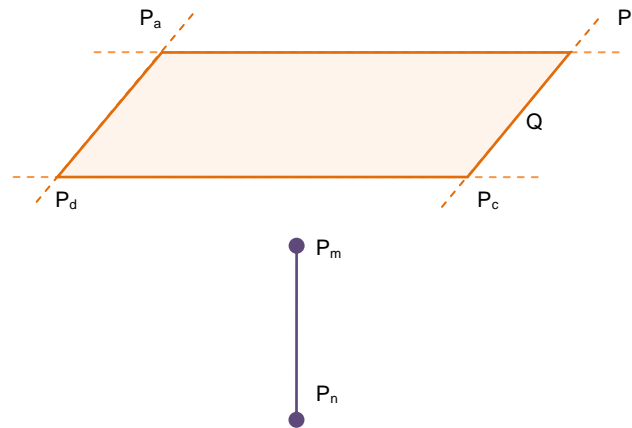


Ilustración 123 Intersección de un segmento con un cuadrado

VIII.XX.I. Cálculos

Comprobamos si hay intersección con los triángulos formados por sus vértices ABC y DBC

VIII.XXI. Intersección de un cubo con un cuadrado

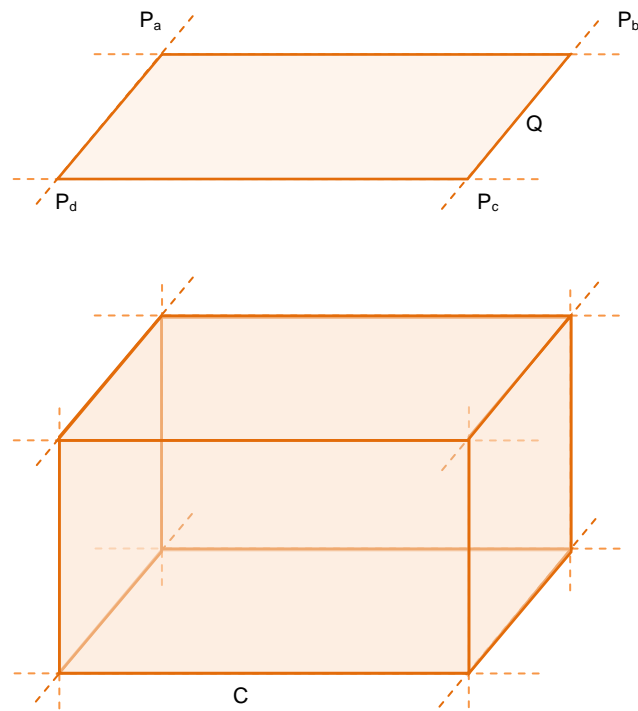


Ilustración 124 Intersección de un cubo con un cuadrado

VIII.XXI.I. Cálculos

Comprobamos si hay intersección con los triángulos formados por sus vértices ABC y DBC

VIII.XXII. Intersección de un triángulo con un cubo

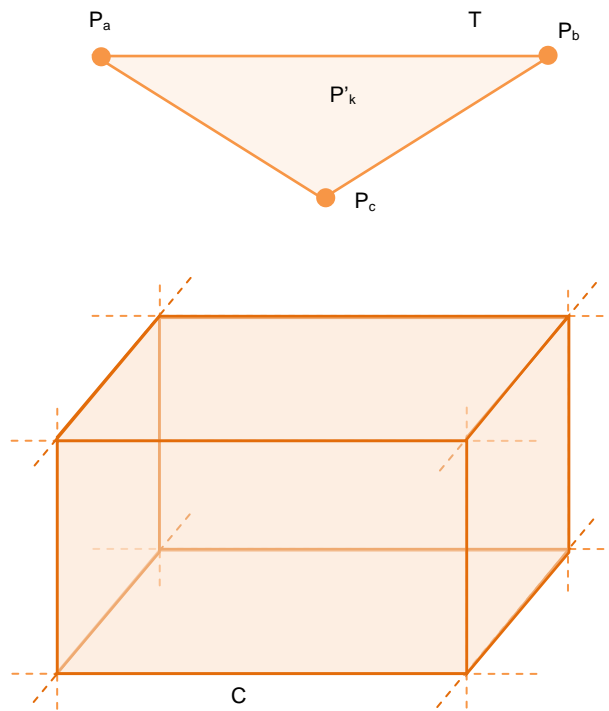


Ilustración 125 Intersección de un triángulo con un cubo

VIII.XXII.I. Cálculos

Calcula si el cuadrado hace intersección con el cubo ortogonal. Para ello comprobamos si alguno de los puntos está contenido en el cubo. Si no es así comprobamos si alguno de los lados del triángulo está cortado por el cubo. Si no es así finalmente comprobamos si alguna de las aristas del cubo corta al triángulo.

VIII.XXIII. Intersección de un triángulo con un segmento

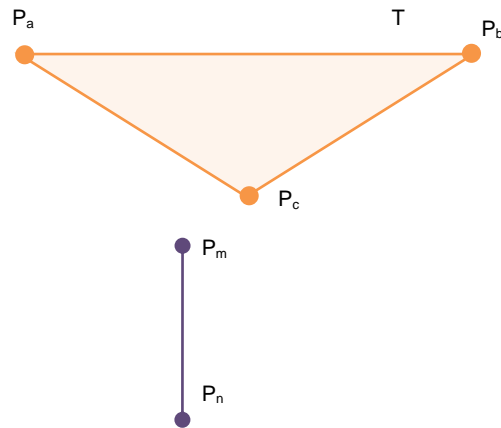


Ilustración 126 Intersección de un triángulo con un segmento

VIII.XXIII.I. Cálculos

Comprueba si el segmento hace intersección con el plano que contiene el triángulo

Si no interacciona con el plano, no interacciona con el triángulo.

Si interacciona en un punto comprobamos si el punto de intersección esta dentro del triángulo.

Si esta contenido comprobamos si el segmento hace intersección con uno de los lados del triángulo o uno de los extremos del segmento esta dentro del triángulo.

VIII.XXIV. Intersección de un segmento con un cubo

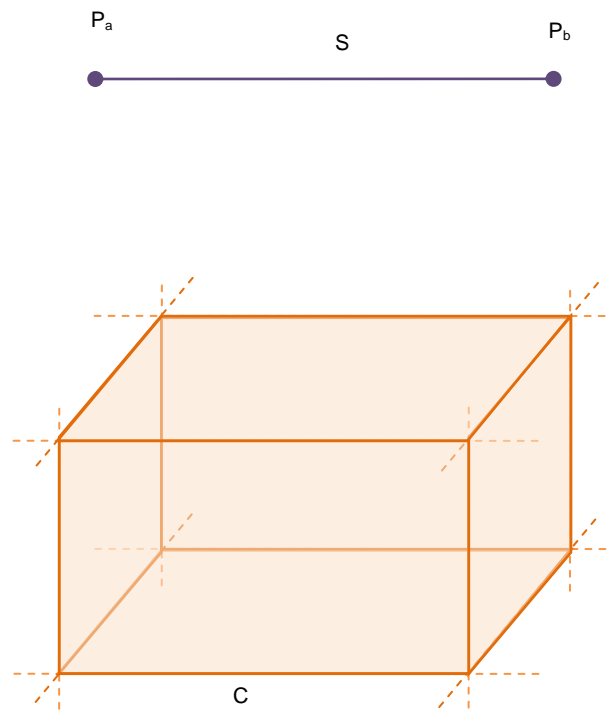
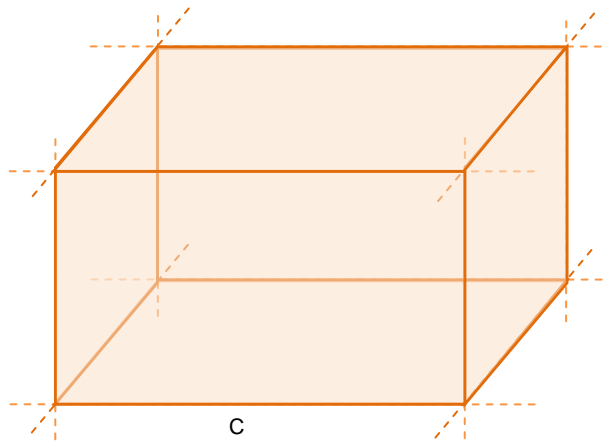
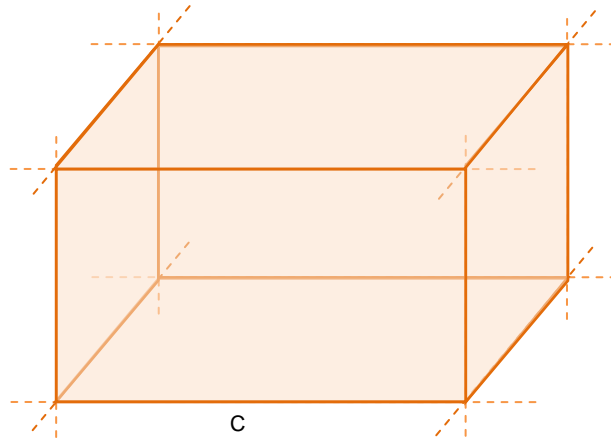


Ilustración 127 Intersección de un segmento con un cubo

VIII.XXIV.I. Cálculos

Calcula si un segmento hace intersección con el cubo, para ello comprobamos si alguno de los extremos del segmento hace intersección con el cubo, sino comprobamos si alguno de los segmentos hace intersección con alguna de sus caras.

VIII.XXV. Intersección de un cubo con un cubo**Ilustración 128 Intersección de un cubo con un cubo****VIII.XXV.I. Cálculos**

Calculamos si alguna de sus caras está contenida o hace intersección con el otro cubo.

TABLAS E ILUSTRACIONES

Índice de Tablas

Tabla 1 Objetivos del primer ciclo	2
Tabla 2 Riesgos analizados en el primer ciclo.....	2
Tabla 3 Contratiempos del primer ciclo.....	2
Tabla 4 Objetivos del segundo ciclo.....	4
Tabla 5 Riesgos del segundo ciclo.....	4
Tabla 6 Objetivos del tercer ciclo.....	5
Tabla 7 Riesgos del tercer ciclo.....	5
Tabla 8 Contratiempos del tercer ciclo	5
Tabla 9 Objetivos del cuarto ciclo.....	7
Tabla 10 Riesgos del cuarto ciclo.....	7
Tabla 11 Estructura de los datos del modelo.....	10
Tabla 12 Estructura del modelo.....	10
Tabla 13 Modelo del listado de estaciones de render	11
Tabla 14 Modelo del listado de formas	12
Tabla 15 Modelo del árbol kd.....	13
Tabla 16 Estructura del controlador.....	14
Tabla 17 Módulo de servicios del controlador.....	15
Tabla 18 Módulo de estaciones de render del controlador	16
Tabla 19 Módulo de formas del controlador	16
Tabla 20 Módulo del árbol del controlador	17
Tabla 21 Módulo de simulación del controlador	17
Tabla 22 Módulo del monitor del controlador.....	18

Tabla 23 Módulo de tareas del controlador	18
Tabla 24 Estructura del Interfaz del Usuario	19
Tabla 25 Diagrama de Componentes	20
Tabla 26 Diagrama de Objetos.....	21
Tabla 27 Diagrama de Despliegue.....	22
Tabla 28 Diagrama de Paquetes	22
Tabla 29 Casos de uso de aplicación	23
Tabla 30 Casos de uso relacionados con los servicios	23
Tabla 31 Casos de uso relacionados con las estaciones de render	24
Tabla 32 Casos de uso relacionados con las formas.....	25
Tabla 33 Casos de uso relacionados con el árbol kd.....	26
Tabla 34 Casos de uso relacionados con la simulación	26
Tabla 35 Estados del servicio de red.....	27
Tabla 36 Inicio de vida de un módulo cualquiera	27
Tabla 37 Secuencia de cierre de la aplicación	28
Tabla 38 Ejecución de una funcionalidad cualquiera	29
Tabla 39 Tiempo Base Estación A sin uso de red.....	31
Tabla 40 Análisis del cálculo del tiempo base de la estación A sin uso de red	32
Tabla 41 Tiempo base de la estación A con uso de red en local.....	33
Tabla 42 Tiempo base de la estación A con uso de red en local.....	34
Tabla 43 Tiempo base de la estación A con uso de red en local y sin uso de red	35
Tabla 44 Tiempo base de la estación B con uso de protocolos de red en configuración remota	36
Tabla 45 Tiempo base de la estación B con uso de protocolos de red en configuración remota	37
Tabla 46 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto.....	38
Tabla 47 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada	39
Tabla 48 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada	40
Tabla 49 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto y la búsqueda combinada	41
Tabla 50 Simulación local de la búsqueda del más cercano Máquina 1.....	42
Tabla 51 Análisis del algoritmo de búsqueda ortogonal	43
Tabla 52 Análisis del algoritmo de búsqueda esférica.....	44
Tabla 53 Calculo del Presupuesto.....	101

Índice de Ilustraciones

Ilustración 1 Primera estimación de la planificación	1
Ilustración 2 Ajuste en la planificación	3
Ilustración 3 Ajuste en la planificación	4
Ilustración 4 Ajuste en la planificación	6
Ilustración 5 Ajuste en la planificación	7
Ilustración 6 Estructura del sistema	9
Ilustración 7 Estructura del modelo	10
Ilustración 8 Modelo del listado de estaciones de render	11
Ilustración 9 Modelo del listado de formas	12
Ilustración 10 Modelo del árbol kd	13
Ilustración 11 Estructura del controlador	14
Ilustración 12 Módulo de servicios del controlador	15
Ilustración 13 Módulo de estaciones de render del controlador	16
Ilustración 14 Módulo de formas del controlador	16
Ilustración 15 Módulo del árbol del controlador	17
Ilustración 16 Módulo de simulación del controlador	17
Ilustración 17 Módulo del monitor del controlador	18
Ilustración 18 Módulo de tareas del controlador	18
Ilustración 19 Estructura del Interfaz del Usuario	19
Ilustración 20 Diagrama de Componentes	20
Ilustración 21 Diagrama de Objetos	21
Ilustración 22 Diagrama de Despliegue	22
Ilustración 23 Diagrama de Paquetes	22
Ilustración 24 Casos de uso de aplicación	23
Ilustración 25 Casos de uso relacionados con los servicios	23
Ilustración 26 Casos de uso relacionados con las estaciones de render	24
Ilustración 27 Casos de uso relacionados con las formas	25
Ilustración 28 Casos de uso relacionados con el árbol kd	26
Ilustración 29 Casos de uso relacionados con la simulación	26
Ilustración 30 Estados del servicio de red	27
Ilustración 31 Inicio de vida de un módulo cualquiera	27
Ilustración 32 Secuencia de cierre de la aplicación	28
Ilustración 33 Ejecución de una funcionalidad cualquiera	29
Ilustración 34 Tiempo Base Estación A sin uso de red	31
Ilustración 35 Análisis del cálculo del tiempo base de la estación A sin uso de red	32
Ilustración 36 Tiempo base de la estación A con uso de red en local	33
Ilustración 37 Tiempo base de la estación A con uso de red en local	34
Ilustración 38 Tiempo base de la estación A con uso de red en local y sin uso de red	35
Ilustración 39 Tiempo base de la estación B con uso de protocolos de red en configuración remota	36
Ilustración 40 Tiempo base de la estación B con uso de protocolos de red en configuración remota	37
Ilustración 41 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto	38
Ilustración 42 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada	39
Ilustración 43 Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada	40

<i>Ilustración 44 Tiempo base de la estación A con uso de red en local y sin uso de red y la estación B con uso de red en remoto y la búsqueda combinada.....</i>	<i>41</i>
<i>Ilustración 45 Análisis del algoritmo del más cercano</i>	<i>42</i>
<i>Ilustración 46 Análisis del algoritmo de búsqueda ortogonal</i>	<i>43</i>
<i>Ilustración 47 Análisis del algoritmo de búsqueda esférica.....</i>	<i>44</i>
<i>Ilustración 48 Algoritmo de ordenación</i>	<i>45</i>
<i>Ilustración 49 Coordenadas de prueba.....</i>	<i>45</i>
<i>Ilustración 50 Ordenación 1 paso.....</i>	<i>46</i>
<i>Ilustración 51 Ordenación 2 paso.....</i>	<i>46</i>
<i>Ilustración 52 Construcción de árbol paso 1.....</i>	<i>46</i>
<i>Ilustración 53 Construcción de árbol paso 2.....</i>	<i>46</i>
<i>Ilustración 54 Construcción de árbol paso 3.....</i>	<i>47</i>
<i>Ilustración 55 Construcción de árbol paso 4.....</i>	<i>47</i>
<i>Ilustración 56 Volúmenes válidos</i>	<i>49</i>
<i>Ilustración 57 Proyección del punto sobre el eje X</i>	<i>49</i>
<i>Ilustración 58 Intervalo sobre cada eje.....</i>	<i>50</i>
<i>Ilustración 59 Valor máximo y mínimo</i>	<i>50</i>
<i>Ilustración 60 Valores máximos y mínimos</i>	<i>50</i>
<i>Ilustración 61 Valores máximos y mínimos sobre un eje.....</i>	<i>51</i>
<i>Ilustración 62 Árbol candidato</i>	<i>55</i>
<i>Ilustración 63 Representación de los tres casos</i>	<i>56</i>
<i>Ilustración 64 Selección de subárbol caso 1</i>	<i>57</i>
<i>Ilustración 65 Selección de subárbol caso 2</i>	<i>57</i>
<i>Ilustración 66 Selección de subárbol caso 3</i>	<i>58</i>
<i>Ilustración 67 Elección de rama.....</i>	<i>58</i>
<i>Ilustración 68 Ejemplo</i>	<i>59</i>
<i>Ilustración 69 Ejemplo proceso 1.....</i>	<i>60</i>
<i>Ilustración 70 Ejemplo proceso 2.....</i>	<i>61</i>
<i>Ilustración 71 Ejemplo proceso 3.....</i>	<i>61</i>
<i>Ilustración 72 Ejemplo proceso 4.....</i>	<i>62</i>
<i>Ilustración 73 Ejemplo proceso 5.....</i>	<i>62</i>
<i>Ilustración 74 Ejemplo proceso 6.....</i>	<i>62</i>
<i>Ilustración 75 Árbol de volúmenes</i>	<i>64</i>
<i>Ilustración 76 Proyección de intervalo.....</i>	<i>65</i>
<i>Ilustración 77 Tres casos.....</i>	<i>66</i>
<i>Ilustración 78 Repetiríamos el proceso con los subárboles derecho e izquierdo</i>	<i>66</i>
<i>Ilustración 79 Eliminación de la rama</i>	<i>67</i>
<i>Ilustración 80 Elección de subárbol</i>	<i>68</i>
<i>Ilustración 81 Algoritmo de selección de rama</i>	<i>69</i>
<i>Ilustración 82 Búsqueda del punto más cercano.....</i>	<i>73</i>
<i>Ilustración 83 Modo de construcción del árbol.....</i>	<i>74</i>
<i>Ilustración 84 Selección de rama.....</i>	<i>74</i>
<i>Ilustración 85.....</i>	<i>77</i>
<i>Ilustración 86 Modo de construcción del árbol.....</i>	<i>78</i>
<i>Ilustración 87 Selección de rama</i>	<i>78</i>
<i>Ilustración 88 Búsqueda ortogonal en una nube de puntos.....</i>	<i>81</i>
<i>Ilustración 89 Modo de construcción del árbol.....</i>	<i>82</i>
<i>Ilustración 90 Elección de rama.....</i>	<i>82</i>
<i>Ilustración 91 Búsqueda ortogonal en una nube de volúmenes.....</i>	<i>84</i>
<i>Ilustración 92 Modo de construcción del árbol.....</i>	<i>85</i>

<i>Ilustración 93 Elección de rama.....</i>	<i>85</i>
<i>Ilustración 94 Pares (Clave, Valor).....</i>	<i>89</i>
<i>Ilustración 95 Napster</i>	<i>90</i>
<i>Ilustración 96 Gnutella</i>	<i>91</i>
<i>Ilustración 97 Freenet.....</i>	<i>92</i>
<i>Ilustración 98 Obteniendo una llave “K” de 160 bits</i>	<i>93</i>
<i>Ilustración 99 El mensaje es enviado de nodo en nodo a través de la red</i>	<i>93</i>
<i>Ilustración 100 Enrutado hasta llegar al nodo responsable</i>	<i>94</i>
<i>Ilustración 101 Particionamiento del espacio de claves</i>	<i>95</i>
<i>Ilustración 102 $\delta(k_1, k_2)$ es la distancia alrededor del círculo.....</i>	<i>95</i>
<i>Ilustración 103 Red de overlay</i>	<i>97</i>
<i>Ilustración 104 Distancia de un punto a una recta.....</i>	<i>121</i>
<i>Ilustración 105 Distancia de un punto a un segmento</i>	<i>122</i>
<i>Ilustración 106 Distancia de un punto a un plano</i>	<i>123</i>
<i>Ilustración 107 Distancia de un punto a un triángulo</i>	<i>124</i>
<i>Ilustración 108 Distancia de un punto a un cuadrado</i>	<i>125</i>
<i>Ilustración 109 Distancia de un punto a un cubo</i>	<i>126</i>
<i>Ilustración 110 Proyección de un punto a un plano.....</i>	<i>127</i>
<i>Ilustración 111 Proyección de un punto en un segmento</i>	<i>128</i>
<i>Ilustración 112 Proyección de un punto en un triángulo</i>	<i>129</i>
<i>Ilustración 113 Intersección de un punto con una recta.....</i>	<i>131</i>
<i>Ilustración 114 Intersección de un punto con un plano</i>	<i>132</i>
<i>Ilustración 115 Intersección de un punto con un segmento</i>	<i>133</i>
<i>Ilustración 116 Intersección de un punto con un triángulo</i>	<i>134</i>
<i>Ilustración 117 Orientación del triángulo</i>	<i>135</i>
<i>Ilustración 118 Intersección de un punto con un cubo ortogonal</i>	<i>136</i>
<i>Ilustración 119 Intersección de una recta con una recta.....</i>	<i>137</i>
<i>Ilustración 120 Intersección de una segmento con un segmento.....</i>	<i>138</i>
<i>Ilustración 121 Intersección de una recta con un plano</i>	<i>139</i>
<i>Ilustración 122 Intersección de un segmento con un plano.....</i>	<i>141</i>
<i>Ilustración 123 Intersección de un segmento con un cuadrado</i>	<i>143</i>
<i>Ilustración 124 Intersección de un cubo con un cuadrado</i>	<i>144</i>
<i>Ilustración 125 Intersección de un triángulo con un cubo</i>	<i>145</i>
<i>Ilustración 126 Intersección de un triángulo con un segmento</i>	<i>146</i>
<i>Ilustración 127 Intersección de un segmento con un cubo</i>	<i>147</i>
<i>Ilustración 128 Intersección de un cubo con un cubo</i>	<i>148</i>

Proyecto Final de Carrera

Ingeniería Informática

Curso 2010/2011

Manual del Usuario

ESTRUCTURA DE ACELERACIÓN PARA RENDER 3D

DISTRIBUIDA MEDIANTE COMUNICACIONES PEER TO PEER

2/3

Diego Ignacio Mallea Lobera

Director:

Juan Antonio Magallón Lacarta

Grupo de Informática Gráfica Avanzada

Departamento de Informática e Ingeniería de Sistemas

Centro Politécnico Superior

Universidad de Zaragoza

Junio de 2011

*A mis padres Jorge José y María Pilar Olga, y a mis hermanas Gabriela y
Eugenia,
por insistirme en que termine mis estudios.*

*A mi novia Nadia,
por no reclamar el tiempo empleado.*

*A mis amigos de toda la vida,
por secuestrarme e impedirme que me centre en acabar la carrera.*

Derechos de Autor

Los derechos de la presente obra pertenecen a D. Diego Ignacio Mallea Lobera y al Dr. D. Juan Antonio Magallón Lacarta, del Departamento de Informática e Ingeniería de Sistemas del Centro Politécnico Superior de la Universidad de Zaragoza. Queda prohibida la reproducción total o parcial de esta obra, por cualquier medio, sin el permiso escrito de los autores.

Ficha Técnica

Proyecto fin de carrera

Título:	ESTRUCTURA DE ACCELERACIÓN PARA RENDER 3D DISTRIBUIDA MEDIANTE COMUNICACIONES PEER TO PEER
Autor:	D. Diego Ignacio Mallea Lobera
DNI:	72981973-Y
Promoción:	2004/2011
Especialidad:	Informática
Director:	Juan Antonio Magallón Lacarta
Departamento:	Informática e Ingeniería de Sistemas
Centro:	Centro Politécnico Superior
Universidad:	Universidad de Zaragoza
Fecha	Junio 2011

Resumen

Hoy en día, se puede afirmar que el acceso a computadoras con una gran potencia de cálculo está al alcance en nuestros hogares o puestos de trabajo.

Estas pequeñas estaciones, con las que consultamos el correo o las cuentas de nuestras redes sociales, están la mayoría del tiempo sin aprovechar al máximo su poder de cálculo. Hace tiempo que programas como el SETI, Folding o figthAids entre otros aprovechan estos tiempos de baja actividad de nuestras computadoras para procesar en horas grandes cantidades de datos, las cuales requerirían de supercomputadoras que requieren de costosas inversiones y un mantenimiento tan costoso que solo grandes gobiernos u organismos pueden hacer frente.

Por otro lado tenemos que ámbitos tan dispares como el cine, los videojuegos, en la aeronáutica y un largo etcétera, se hace necesario el computar ingentes cantidades de datos con cálculos increíblemente complejos.

Un ámbito en el cual se tiene una serie de cálculos que se pueden paralelizar y tiene una complejidad tan alta que requiere de más de un computado, es el mundo de la animación 3D.

El presente proyecto busca evaluar los fundamentos básicos tras la producción de una animación 3D, en un sistema distribuido Peer to Peer.

En primera instancia se decidió el evaluar las búsquedas en árboles *kd* mediante el empleo de dicha red, para ello se decidió el implementar una red de dichas características para poder distribuir el cálculo y los nodos de la red.

Posteriormente se implemento una serie de algoritmos comúnmente empleados para la generación de dichas escenas animadas, búsquedas en *kd-tree*, modificando estos para poder realizar los cálculos de una manera distribuida entre los componentes de la red.

Según se avanzo en el estudio se hizo patente un hambre por evaluar e investigar las posibilidades de dicho sistema para nubes de volúmenes incurriendo en unas posibilidades de investigar las posibilidades de las mismas.

Tras el desarrollo del PFC, se puede concluir que el uso de redes P2P abre un sinfín de posibilidades y mejoras. Además la ampliación de dichos algoritmos del procesado de nubes de puntos a nubes de volúmenes abre posibilidades que congenian con técnicas de colisiones.

CONTENIDOS

Capítulo 1. Datos Técnicos de Distribución.....	1
1.1. Modo de distribución	1
1.2. Tamaño de la distribución	1
1.3. Tamaño de la distribución sin comprimir	1
1.4. Directorio de instalación	1
1.5. Espacio mínimo en disco para la distribución.	1
1.6. Estructura de directorios al ser extraída la distribución	2
Capítulo 2. Requisitos Hardware y Software	3
2.1. Requisitos de tipo hardware.....	3
2.1.1. Requerimientos de máquina mínimos.	3
2.1.2. Requisitos de máquina aconsejables	3
2.2. Requisitos de tipo software	3
2.2.1. Software necesario	3
2.3. Como obtener una utilidad de descompresión ZIP.	4
2.4. Como obtener los permisos de escritura en disco:	4
2.5. Como obtener la copia del CD de P2P	4
2.6. Como obtener la Máquina virtual Java 5.0 o superior	5

2.7. Cómo obtener la extensión Java 3D para la Máquina virtual Java 5.0 ó superior	6
Capítulo 3. Instalación de la aplicación	7
3.1. Ejemplo de línea de comando:.....	7
3.2. Ejemplo en un entorno gráfico:	7
Capítulo 4. Manual de uso de la aplicación	11
4.1. ¿Qué es un programa de P2P?	11
4.2. ¿Qué es una red de renderizado?	11
4.3. Descripción del interfaz de usuario	12
4.3.1. Secciones de la aplicación	12
4.3.2. Barra de menús de la aplicación.....	14
4.3.3. Monitor del sistema.....	20
4.3.4. Consola de comandos	22
4.3.5. Sección de configuración de servicios.....	23
4.3.6. Sección de monitoreo de Estaciones de Render	24
4.3.7. Sección de monitoreo de Estaciones de Render	25
4.3.8. Sección de estadísticas de Estaciones de Render	27
4.3.9. Sección de sintetizado de formas	28
4.3.10. Sección de listado de formas	29
4.3.11. Sección de estadísticas de formas	34
4.3.12. Sección de representación del árbol kd.....	35
4.3.13. Sección de estadísticas y configuración del árbol kd	37
4.3.14. Sección de sintetizado de simulación.....	38
4.3.15. Sección de control de simulación local.....	39
4.3.16. Sección de control de simulación distribuida.....	43
4.3.17. Sección de configuración y estadísticas de la simulación.	47
4.4. Iniciando la aplicación	49
4.5. Iniciando la conexión de los servicios	50
4.5.1. Inicio rápido	50
4.5.2. Cambio del puerto o el host de los servicios	50
4.6. Gestión de los servicios	52
4.7. Gestión de Estaciones de Render.....	53
4.7.1. ¿Qué son las estaciones de render?	53

4.7.2.	Monitorizar	53
4.7.3.	Añadir estaciones de render	53
4.7.4.	Eliminar estaciones de render	53
4.7.5.	Ping a estaciones	54
4.8.	Gestión de formas.....	55
4.8.1.	¿Qué son las formas?	55
4.8.2.	Sintetizar formas.....	55
4.8.3.	Añadir formas	55
4.8.4.	Eliminar formas.....	56
4.8.5.	Propagación	56
4.9.	Gestión del árbol kd	57
4.9.1.	¿Qué es un árbol kd?	57
4.9.2.	Generar árbol kd.....	57
4.9.3.	Ver representación del árbol kd.....	57
4.10.	Gestión de la simulación	58
4.10.1.	¿Tipos de simulación?.....	58
4.10.2.	¿Tipos de test?.....	58
Capítulo 5.	Ejemplos de Uso	59
5.1.	Monitoreo del listado de estaciones de render.....	59
5.2.	Monitoreo del listado de formas	60
5.3.	Monitoreo del árbol kd.....	61
5.4.	Simulación local de la búsqueda en una esfera	62
5.5.	Errores típicos	64
5.5.1.	Error de conexión de red	64
5.5.2.	No se pueden almacenar las formas	64
5.5.3.	No autorizados.....	64
Capítulo 6.	Licencia de software	65

Capítulo 1. DATOS TÉCNICOS DE DISTRIBUCIÓN

1.1. Modo de distribución

El programa se distribuye en un directorio llamado **DISTRIBUCION** contenido dentro del directorio **otrosAnexos**. El contenido de esta carpeta es la que sigue:

P2P.ZIP	Archivo que contiene los ejecutables del programa.
DOC.ZIP	Manual del programa.
SRC.ZIP	Código fuente de la aplicación y API de programación.
README.TXT	Archivo con información del contenido de la distribución.
LICENCIA.TXT	Licencia de Software.

Tabla 1 Modo de distribución

1.2. Tamaño de la distribución

P2P.ZIP	9.69 MB
DOC.ZIP	6.41 MB
SRC.ZIP	120 MB
README.TXT	1 KB
LICENCIA.TXT	14 KB

Tabla 2 Tamaño de la distribución

1.3. Tamaño de la distribución sin comprimir

P2P	13 MB
DOC	6.57 MB
SRC	156 MB
README.TXT	1 KB
LICENCIA.TXT	14 KB

Tabla 3 Tamaño de la distribución sin comprimir

1.4. Directorio de instalación

No es necesaria la instalación en ninguna ruta específica. Es necesario tener permisos de lectura y escritura en el directorio de instalación.

1.5. Espacio mínimo en disco para la distribución.

Tipo de Instalación	Tamaño de Disco
Aplicación de P2P sin documentación	13 MB
Aplicación de P2P con documentación	317 MB

Tabla 4 Espacio mínimo en disco para la distribución

1.6. Estructura de directorios al ser extraída la distribución

Directorio raíz:

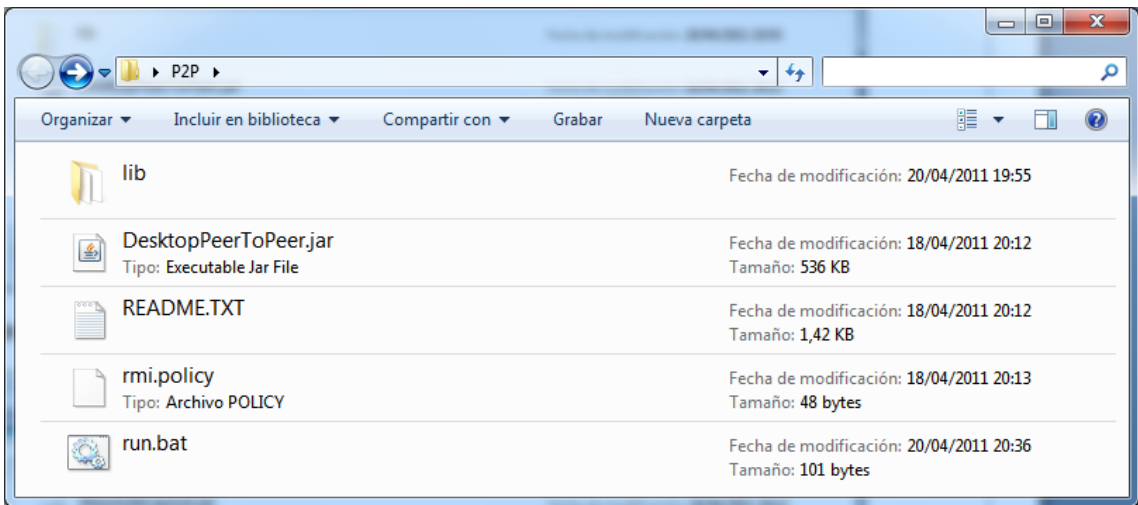


Ilustración 1 Estructura de directorios al ser extraída la distribución

Subdirectorio LIB

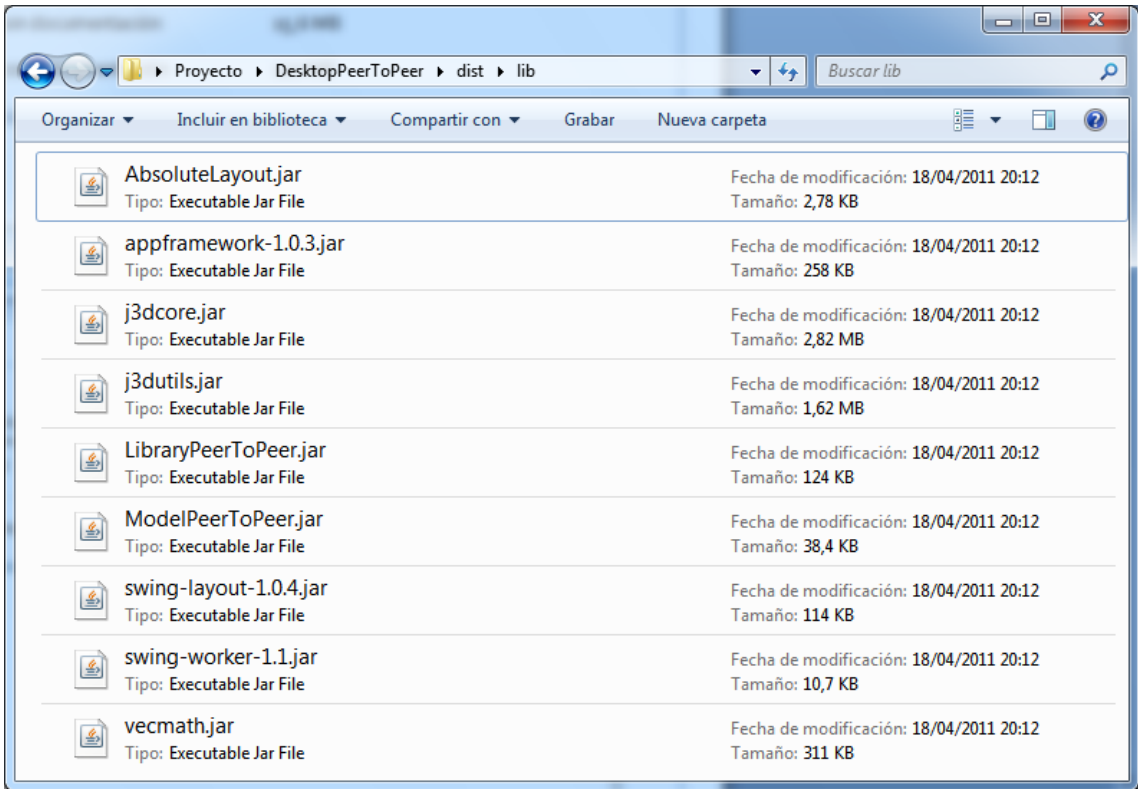


Ilustración 2 Subdirectorio LIB

Capítulo 2. REQUISITOS HARDWARE Y SOFTWARE

2.1. Requisitos de tipo hardware

2.1.1. *Requerimientos de máquina mínimos.*

- Velocidad 1 GHZ.
- Memoria RAM 512 MB
- Almacenamiento de 1GB para la aplicación
- Dispositivo de red de 10 MB

2.1.2. *Requisitos de máquina aconsejables*

- Velocidad 2 GHZ.
- Memoria RAM 2 GB
- Almacenamiento de 10 GB para la aplicación.
- Dispositivo de red de 1024 MB.

2.2. Requisitos de tipo software

2.2.1. *Software necesario*

- Una aplicación de descompresión de archivos ZIP
- Tener permisos de escritura en disco
- Tener una copia del CD de P2P
- Tener la Máquina Virtual Java 5.0 o superior instalada.
- Tener la extensión Java 3D de la Máquina virtual Java.

2.3. Como obtener una utilidad de descompresión ZIP.

Hoy en día el formato ZIP está ampliamente soportado por los sistemas operativos pero de ser necesario podemos acceder a la página web de PeaZIP:

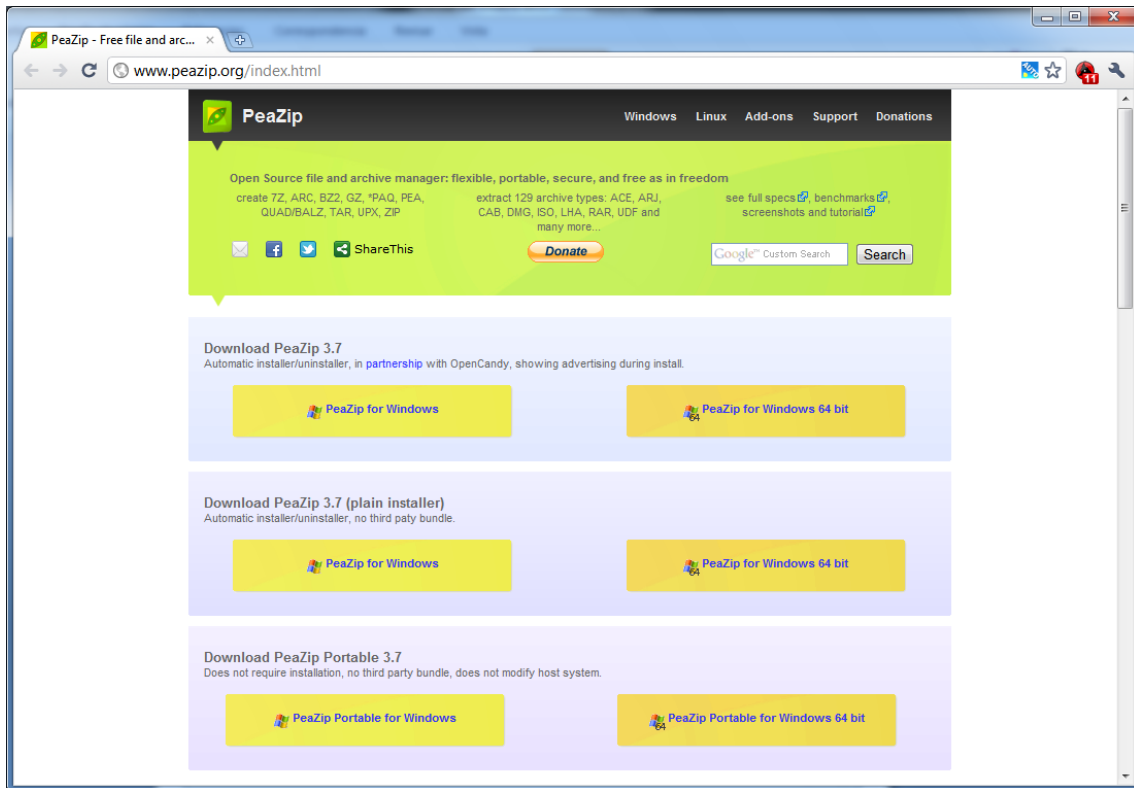


Ilustración 3 Página web de PeaZip

En la página principal podremos descargarnos una versión de la aplicación en la plataforma que queramos, o de ser necesario podremos compilar las fuentes para nuestro sistema.

2.4. Como obtener los permisos de escritura en disco:

Comprueba si tienes permisos, si no lo sabes prueba a copiar algo dentro de la carpeta candidata. De no poder copiarlo entra en el ordenador como administrador. Si no tienes acceso a este nivel dirígete a tu administrador para que te de permisos de escritura.

2.5. Como obtener la copia del CD de P2P

Dirígete al proveedor más cercano y solicítale una copia de este CD.

2.6. Como obtener la Máquina virtual Java 5.0 o superior

Para obtener la máquina virtual Java 5.0 o superior dirigirse a la página oficial de java.



Ilustración 4 Página de descarga de MV Java

Desde allí podrás seleccionar la más indicada para tu plataforma.

2.7. Cómo obtener la extensión Java 3D para la Máquina virtual Java 5.0 ó superior

Dirígete a la sección de tecnologías de java dentro de la página de Oracle.

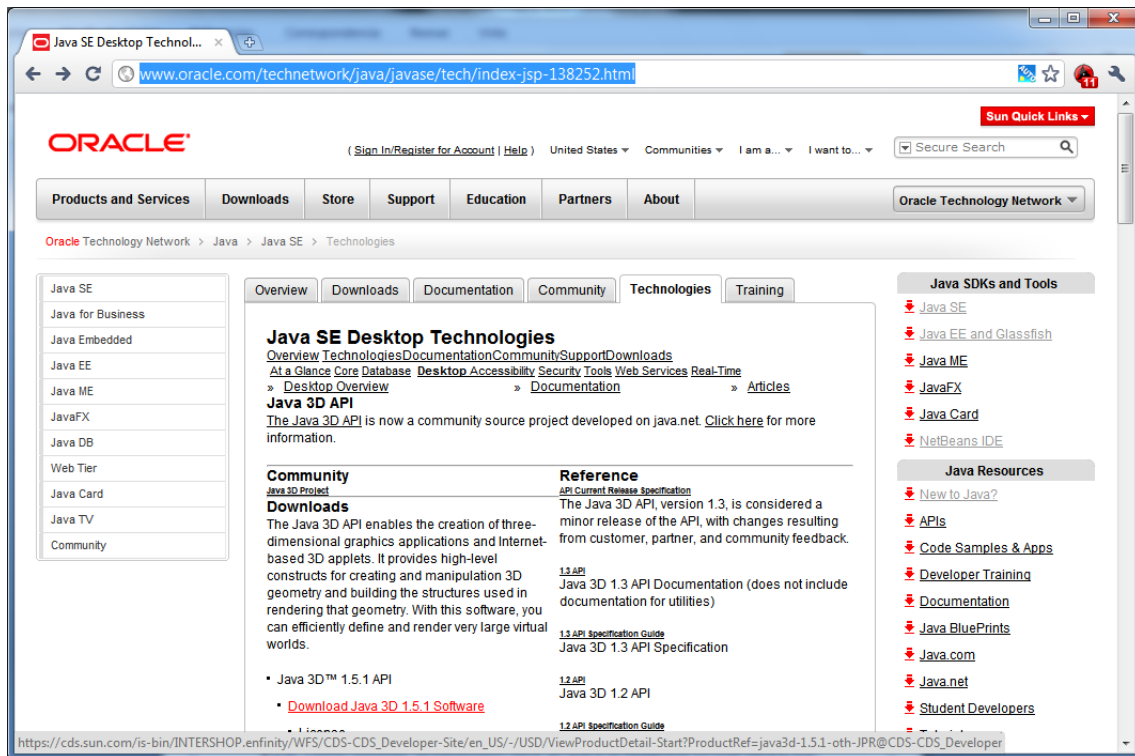


Ilustración 5 Página de descarga de Java3D

Una vez allí podrás acceder a la sección de descarga de la extensión.

Java 3D 1.5.1

Provide Information, then Continue to Download

Select Platform for your download:

Platform:

☐ I agree to the [Software License Agreement](#).

Ilustración 6 Selección de la plataforma Java3D

Capítulo 3. INSTALACIÓN DE LA APLICACIÓN

Dentro del CD de P2P usted encontrará en la carpeta de DISTRIBUCION un archivo llamado P2P.RAR con la herramienta de descompresión extraiga el contenido en una ubicación de disco duro.

3.1. Ejemplo de línea de comando:

```
unzip E:\P2P.ZIP -d C:\P2P\
```

Recuerde que la carpeta seleccionada será la que contenga dentro el archivo de ejecución del programa. En el ejemplo sería la carpeta **C:\P2P**

3.2. Ejemplo en un entorno gráfico:

Se inserta el CD P2P y se va a la unidad que contiene el CD de P2P

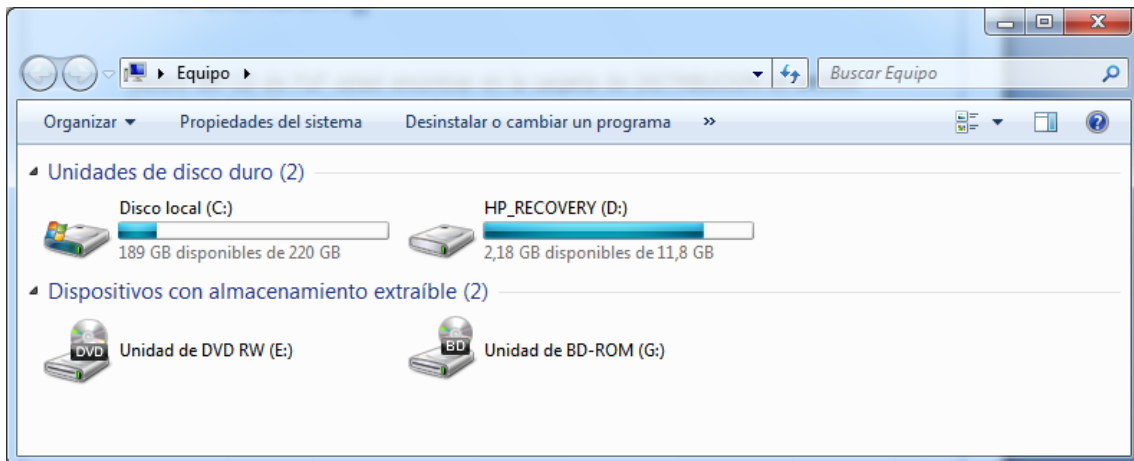


Ilustración 7 Explorador de Archivos, unidad de CD

Se abre la unidad y dentro del directorio **otrosAnexos** obtendremos la siguiente pantalla

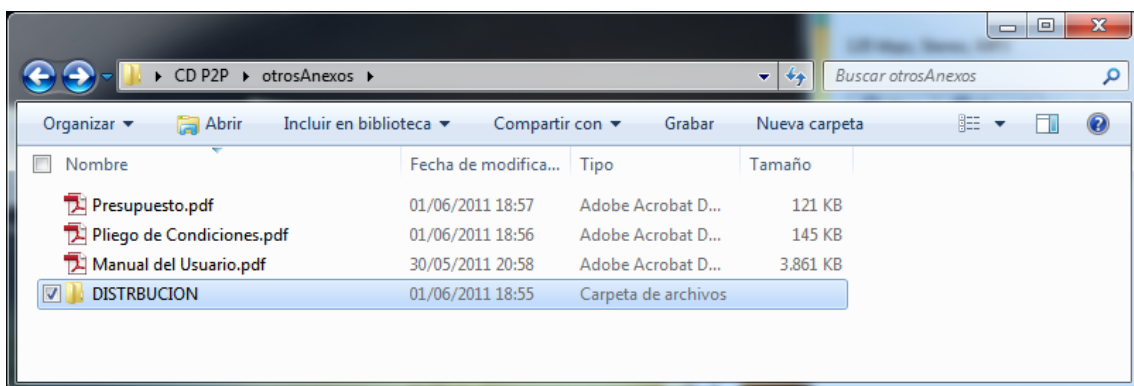


Ilustración 8 Explorador de archivos, distribución

En ella se puede observar un directorio llamado **DISTRIBUCION**.

Se abre el directorio de **DISTRIBUCION** contenido en el directorio **otrosAnexos**:

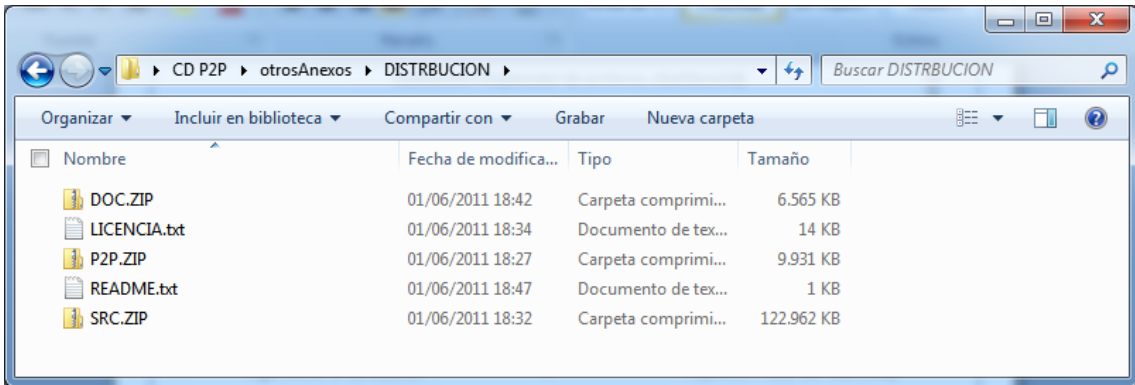


Ilustración 9 Explorador de archivos, dentro del directorio DISTRIBUCION

En él se pueden observar cinco archivos:

P2P.ZIP	Archivo que contiene los ejecutables del programa.
DOC.ZIP	Manual del programa y otra documentación
SRC.ZIP	Código fuente de la aplicación y API.
README.TXT	Archivo con información del contenido de la distribución.
LICENCIA.TXT	Licencia de Software.

Tabla 5 Listado de archivos de la distribución.

Con la herramienta de descompresión ZIP se descomprime el archivo P2P.ZIP

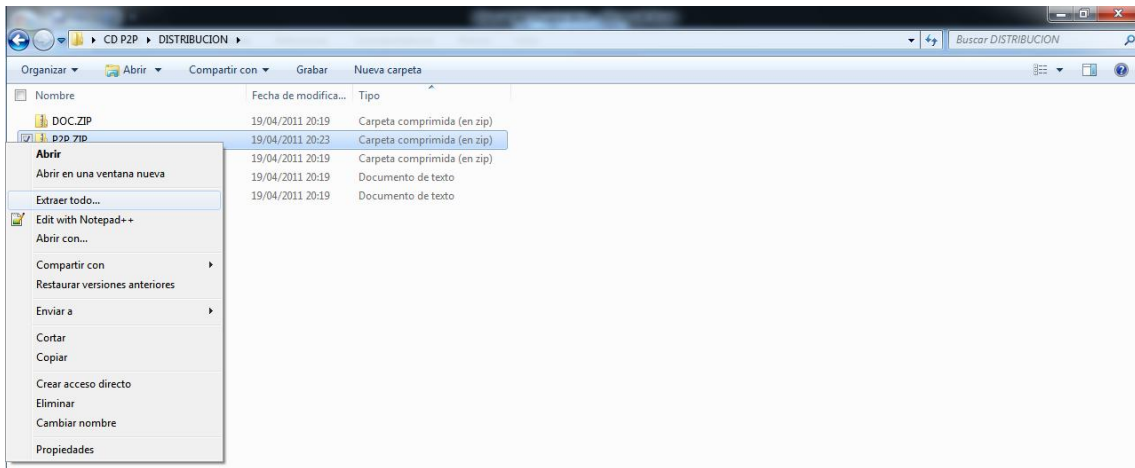


Ilustración 10 Operación de descompresión de la aplicación

Por último elige la ruta donde se quiere extraer el contenido.

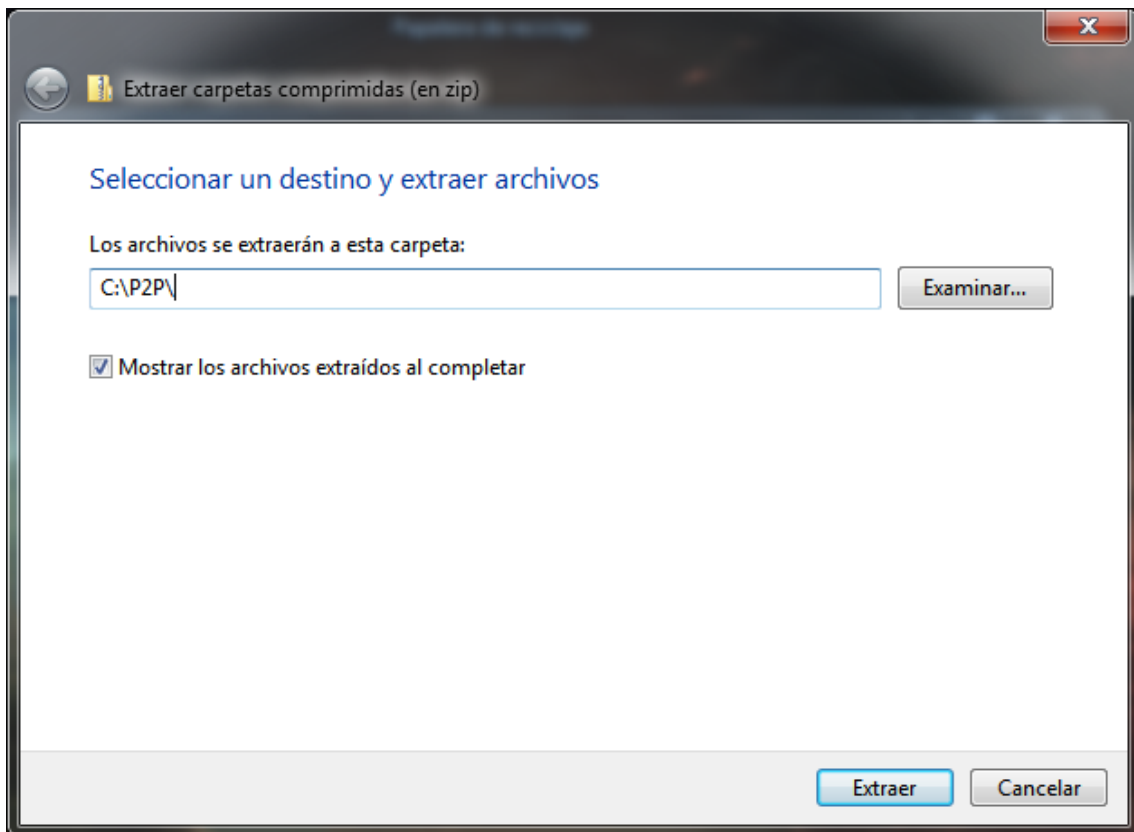


Ilustración 11 Operación de descompresión, destino

Una vez extraído tendrás un directorio con la siguiente estructura.

```
C:\P2P\  
  DesktopPeerToPeer.jar  
  README.TXT  
  rmi.policy  
  run.bat  
C:\P2P\lib\  
  AbsoluteLayout.jar  
  appframework-1.0.3.jar  
  j3dcore.jar  
  j3dutils.jar  
  LibraryPeerToPeer.jar  
  ModelPeerToPeer.jar  
  swing-layout-1.0.4.jar  
  swing-worker-1.1.jar  
vecmatch.jar
```

El archivo run.bat es el que ejecutará la aplicación.

Capítulo 4. MANUAL DE USO DE LA APLICACIÓN

4.1. ¿Qué es un programa de P2P?

Internet, las redes locales y las conexiones de banda ancha han revolucionado el mundo de la comunicación. En apenas unos años los hogares de medio mundo se han equipado con ordenadores, móviles, conexiones basadas en ADSL, en fibra óptica, etc. Cada ordenador conectado es un nodo de una enorme red mundial.

El surgimiento de programas de compartición de archivos y comunicación de usuarios ha disparado el uso de las redes y la alarma debido a la difícil protección de los derechos sobre cualquier tipo de información, que puede ser difundida a cientos de miles de usuarios en apenas unos días.

Internet no es algo estático. Su forma y contenido cambia constantemente, los usuarios pueden ser de cualquier parte del mundo y pueden conectarse en cualquier momento. Sus equipos de conexión pueden ser muy diferentes, habiendo desde servidores hasta estaciones con sistemas operativos diferentes o distintas arquitecturas. Debido a esa heterogeneidad es necesario buscar protocolos y mecanismos con los cuales adaptarse a todos los usuarios y así crear programas más versátiles y eficientes, prestando servicio a todo tipo de sistemas.

Un programa de P2P, es un programa para compartir todo tipo de información y recursos a través de una red. De esta forma podrías intercambiar y compartir todo tipo de archivos con los demás usuarios de la red así como comunicarte con ellos en cualquier momento.

4.2. ¿Qué es una red de renderizado?

Tanto en el mundo del cine como el mundo de los videojuegos se hace necesario el procesar grandes cantidades de información para generar escenas sintéticas. Para poder abaratar costes y tiempos se fragmenta dicha tarea en múltiples procesos independientes. Al haber múltiples procesos independientes se puede operar en diversas máquinas conectadas en una red. Al realizar los cálculos en distintas máquinas se consigue paralelizar el proceso global. De este modo, se consigue un acortamiento en el tiempo necesario para llevar a cabo el proceso completo.

4.3. Descripción del interfaz de usuario

4.3.1. Secciones de la aplicación

Descripción
La aplicación tiene cuatro secciones claramente diferenciadas: <ul style="list-style-type: none">La barra de menú.Los paneles de monitorización.La consola de comandos.Sección principal.

Tabla 6 Secciones de la aplicación

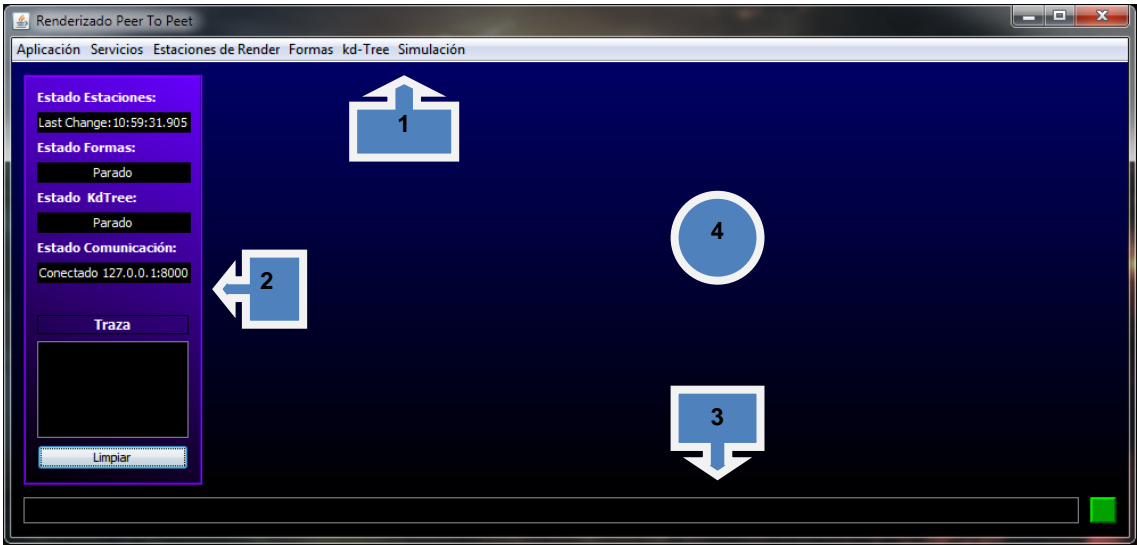


Ilustración 12 Secciones de la aplicación

1	Barra de menú de aplicación
Permite el acceso a los diferentes apartados de la aplicación.	

Tabla 7 Barra de menú de aplicación

2	Paneles de monitorización
Da una breve descripción de lo que está ocurriendo en el sistema.	

Tabla 8 Paneles de monitorización

3 Consola de comandos de la aplicación

Permite una entrada rápida y avanzada de comandos.

[Tabla 9 Consola de comandos de la aplicación](#)

4 Sección principal

Aquí se cargaran las distintas secciones de la aplicación.

[Tabla 10 Sección principal](#)

4.3.2. Barra de menús de la aplicación

Descripción
Desde la barra de menús tendremos acceso a las distintas opciones de la aplicación.

Tabla 11 Barra de menús de la aplicación

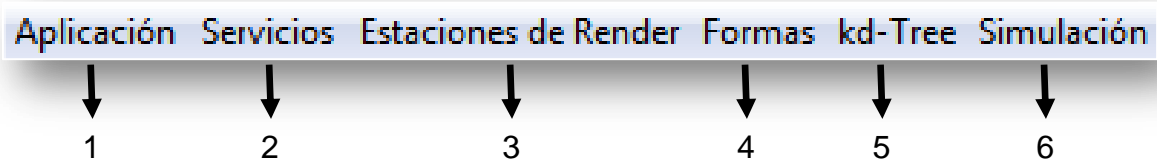


Ilustración 13 Barra de menú

1 Submenú de Aplicación
Abre un submenú que sirve para mostrar opciones relacionadas con la aplicación en general.

Tabla 12 Submenú de aplicación

2 Submenú de Servicios
Abre un submenú que sirve para mostrar opciones relacionadas con los servicios de red.

Tabla 13 Submenú de servicios

3 Submenú de Estaciones de Render
Abre un submenú que sirve para mostrar opciones relacionadas con las estaciones de render.

Tabla 14 Submenú de estaciones de render

4 Submenú de Formas
Abre un submenú que sirve para mostrar opciones relacionadas con las formas a procesar.

Tabla 15 Submenú de formas

5 Submenú de kd-tree
Abre un submenú que sirve para mostrar opciones relacionadas con el árbol kd.

Tabla 16 Submenú de kdtree

6 Submenú de Simulación
Abre un submenú que sirve para mostrar opciones relacionadas con la simulación.

Tabla 17 Submenú de simulación

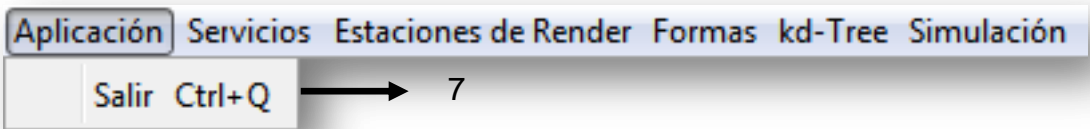


Ilustración 14 Submenú de aplicación

7	Acción del menú Aplicación [Salir]
Cierra la aplicación, se encarga de cerrar correctamente los módulos y salvar la configuración de los mismos.	

Tabla 18 Acción salir

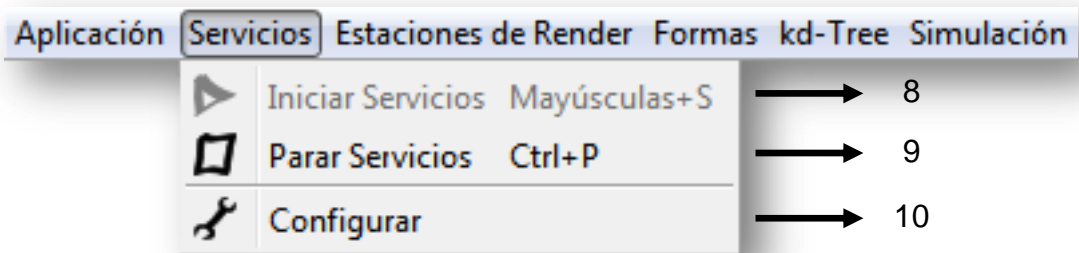


Ilustración 15 Submenú de servicios

8	Acción del menú Servicios [Iniciar Servicios]
Inicia los servicios de red, permitiendo recibir o enviar peticiones de otras estaciones de render.	

Tabla 19 Acción Iniciar

9	Acción del menú Servicios [Parar Servicios]
Para los servicios de red, no permitiendo recibir o enviar peticiones de otras estaciones de render..	

Tabla 20 Acción Parar

10	Acción del menú Servicios [Configurar]
Abre las opciones de configuración de los servicios de red. Desde allí podremos ver el estado de los servicios así como configurar los mismos.	

Tabla 21 Acción Configurar

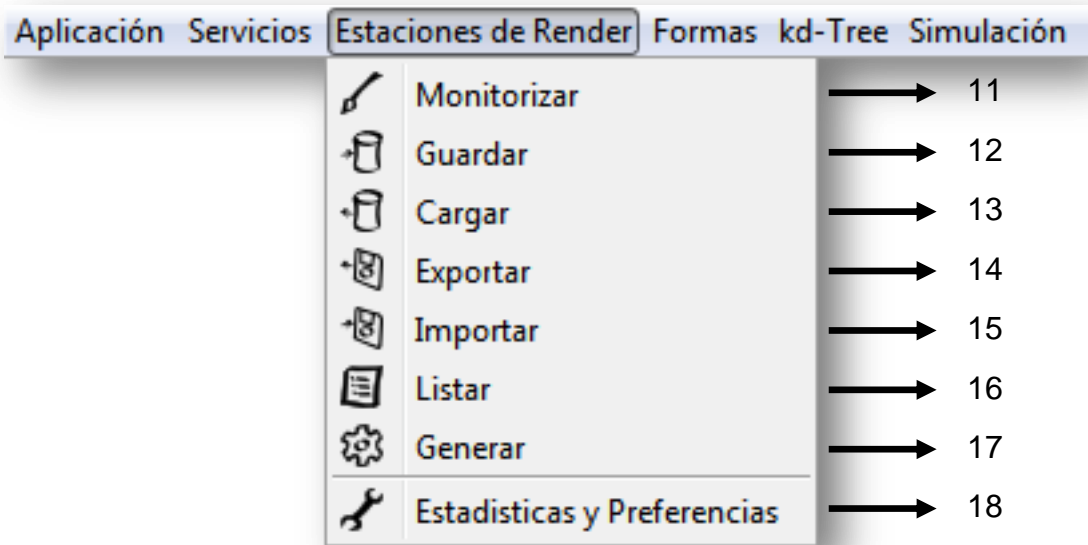


Ilustración 16 Submenú estaciones

11	Acción del menú Estaciones de Render [Monitorizar]
Abre una vista gráfica del estado de las estaciones de render con datos significativos para su monitorización.	

Tabla 22 Acción monitorizar

12	Acción del menú Estaciones de Render [Guardar]
Guarda el listado de estaciones de render en el disco, para que al cargar la próxima vez la aplicación estos se carguen.	

Tabla 23 Acción guardar

13	Acción del menú Estaciones de Render [Cargar]
Carga de disco el listado de estaciones de render.	

Tabla 24 Acción cargar

14	Acción del menú Estaciones de Render [Exportar]
Exporta el listado de estaciones de render con el fin de poder llevarlo a otra estación de render.	

Tabla 25 Acción exportar

15	Acción del menú Estaciones de Render [Importar]
Importa el listado de formas previamente exportado de otra estación de render	

Tabla 26 Acción importar

16	Acción del menú Estaciones de Render [Listar]
Abre un listado de estaciones de render, donde podremos gestionar las mismas.	

Tabla 27 Acción listar

17	Acción del menú Estaciones de Render [Generar]
Genera una lista por defecto de estaciones de render, de esta manera se podrá contactar por defecto con otras estaciones.	

Tabla 28 Acción generar

18	Acción del menú Estaciones de Render [Estadísticas y Preferencias]
Abre una serie de estadísticas y permite la configuración de la gestión de las estaciones de render.	

Tabla 29 Acción estadísticas y preferencias

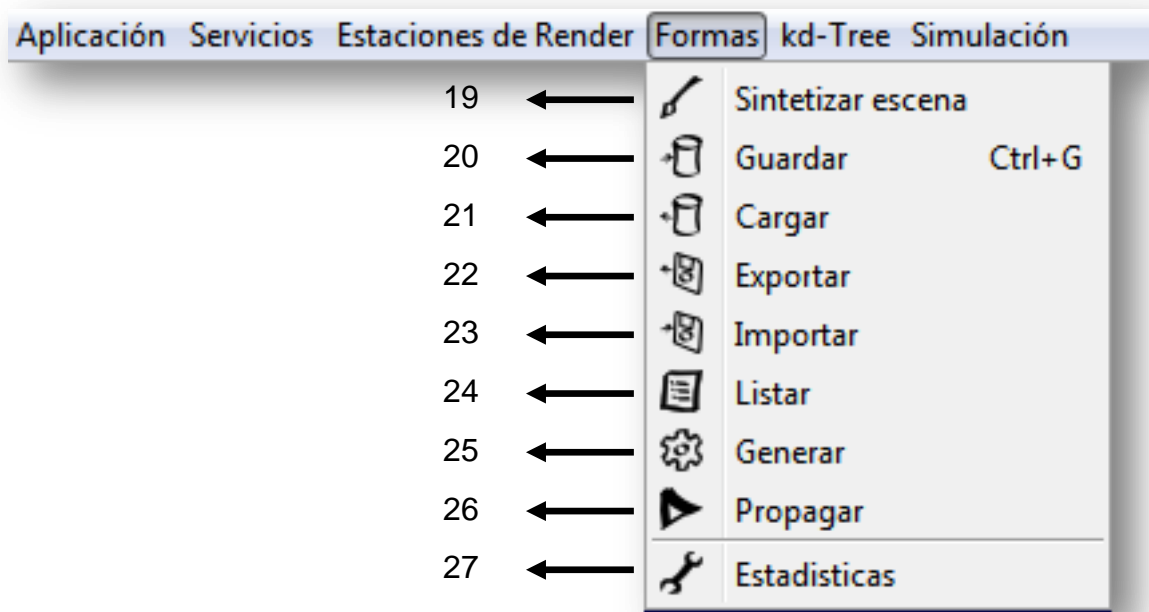


Ilustración 17 Submenú formas

19	Acción del menú Formas [Sintetizar escena]
Abre una vista grafica 3D del escenario para su monitorización.	

Tabla 30 Acción del menú Formas [Sintetizar escena]

20	Acción del menú Formas [Guardar]
Guarda el listado de formas en el disco para que al cargar la próxima vez la aplicación estas se carguen..	

Tabla 31 Acción del menú Formas [Guardar]

21	Acción del menú Formas [Cargar]
Carga de disco el listado de formas.	

Tabla 32 Acción del menú Formas [Cargar]

22	Acción del menú Formas [Exportar]
Exporta el listado de formas con el fin de poder llevarlo a otra estación de render.	

Tabla 33 Acción del menú Formas [Exportar]

23	Acción del menú Formas [Importar]
Importa el listado de formas previamente exportado de otra estación de render.	

Tabla 34 Acción del menú Formas [Importar]

24	Acción del menú Formas [Listar]
Abre un listado de formas, donde podremos gestionar las mismas.	

Tabla 35 Acción del menú Formas [Listar]

25	Acción del menú Formas [Generar]
Genera una lista aleatoria de formas, de esta manera se podrán realizar simulaciones con datos de manera aleatoria.	

Tabla 36 Acción del menú Formas [Generar]

26	Acción del menú Formas [Propagar]
Fuerza la propagación la lista de formas.	

Tabla 37 Acción del menú Formas [Propagar]

27	Acción del menú Formas [Estadísticas]
Abre una serie de estadísticas.	

Tabla 38 Acción del menú Formas [Estadísticas]

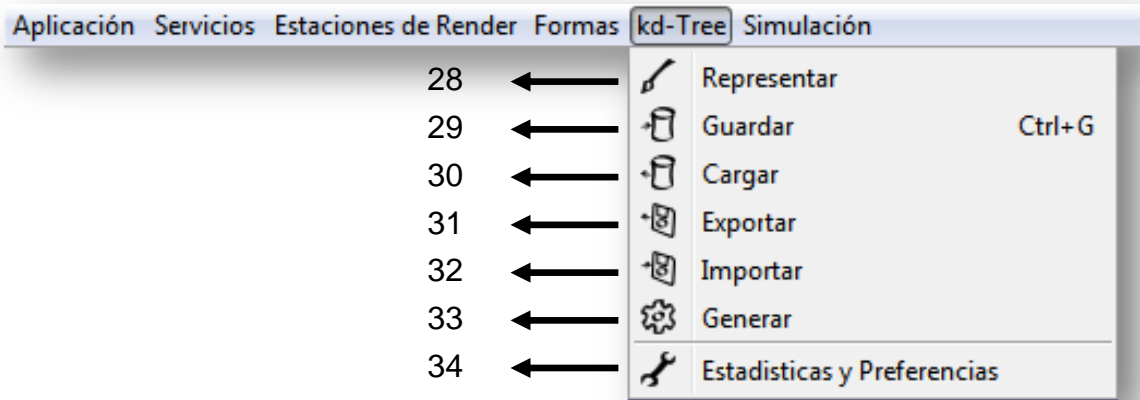


Ilustración 18 Submenú kd-tree

28	Acción del menú kd-tree [Representar]
Dibuja una representación del árbol kd, enseñando sus distintos niveles. Además nos muestra como han sido este recorrido mostrando que ramas han sido truncadas.	

Tabla 39 Acción del menú kd-tree [Representar]

29	Acción del menú kd-tree [Guardar]
Guarda el árbol en el disco para que al cargar la próxima vez la aplicación este se cargue.	

Tabla 40 Acción del menú kd-tree [Guardar]

30	Acción del menú kd-tree [Cargar]
Carga de disco el árbol.	

Tabla 41 Acción del menú kd-tree [Cargar]

31	Acción del menú kd-tree [Exportar]
Exporta el árbol con el fin de poder llevarlo a otra estación de render.	

Tabla 42 Acción del menú kd-tree [Exportar]

32	Acción del menú kd-tree [Importar]
Importa el árbol previamente exportado de otra estación de render.	

Tabla 43 Acción del menú kd-tree [Importar]

33	Acción del menú <i>kd-tree</i> [Generar]
Genera el árbol, para ello recurre al listado de formas con el fin de construirlo.	

Tabla 44 Acción del menú *kd-tree* [Generar]

34	Acción del menú <i>kd-tree</i> [Estadísticas y preferencias]
Abre una serie de estadísticas y preferencias de la gestión del árbol.	

Tabla 45 Acción del menú *kd-tree* [Estadísticas y preferencias]

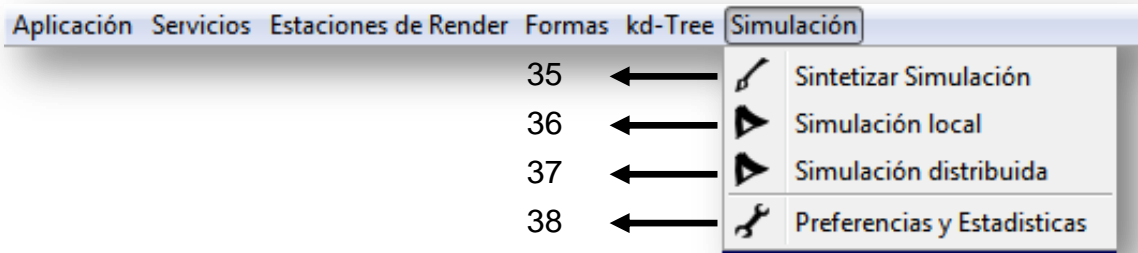


Ilustración 19 Submenú Simulación

35	Acción del menú Simulación [Sintetizar Simulación]
Representa un grafico 3D con el resultado de la simulación..	

Tabla 46 Acción del menú Simulación [Sintetizar Simulación]

36	Acción del menú Simulación [Simulación local]
Abre un control para realizar varios tipos de simulación local.	

Tabla 47 Acción del menú Simulación [Simulación local]

37	Acción del menú Simulación [Simulación distribuida]
Abre un control para realizar varios tipos de simulación distribuida.	

Tabla 48 Acción del menú Simulación [Simulación distribuida]

38	Acción del menú Simulación [Estadísticas y preferencias]
Abre una serie de estadísticas y preferencias de la gestión del árbol.	

Tabla 49 Acción del menú Simulación [Estadísticas y preferencias]

4.3.3. Monitor del sistema

Descripción
Desde el monitor del sistema tendremos un seguimiento rápido y resumido a lo que está sucediendo en la aplicación.

Tabla 50 Monitor del sistema



Ilustración 20 Monitor del sistema

1	Monitor de Estado de las Estaciones de Render Nos muestra de manera rápida los sucesos relacionados con las estaciones de render.
Tabla 51 Monitor de Estado de las Estaciones de Render	
2	Monitor de Estado de las Formas Nos muestra de manera rápida los sucesos relacionados con las formas.
Tabla 52 Monitor de Estado de las Formas	
3	Monitor de Estado del kd-tree Nos muestra de manera rápida los sucesos relacionados con el árbol kd.
Tabla 53 Monitor de Estado del kd-tree	
4	Monitor de Estado de las Comunicaciones de red Nos muestra de manera rápida los sucesos relacionados con las comunicaciones de red.
Tabla 54 Monitor de Estado de las Comunicaciones de red	

5 Monitor informativo general de la aplicación

Nos muestra de manera rápida mensajes informativos, alertas o errores ocurridos en la aplicación.

Tabla 55 Monitor informativo general de la aplicación

6 Acción de limpieza de monitor informativo general [Limpiar]

Elimina los mensajes mostrados en el monitor.

Tabla 56 Acción de limpieza de monitor informativo general [Limpiar]

4.3.4. Consola de comandos

Descripción
La consola comandos de la aplicación nos sirve para poder ejecutar comandos rápidos. Tiene como objetivos los usuarios avanzados. Nos provee de una interfaz más completa y rápida de ejecución.

Tabla 57 Consola de comandos

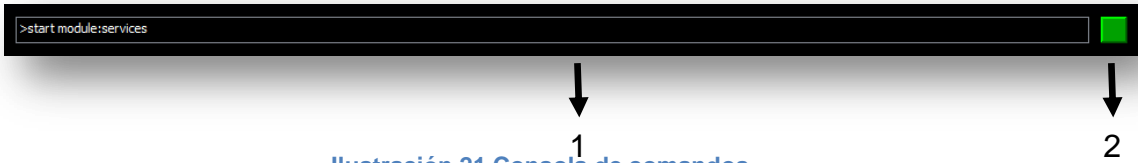


Ilustración 21 Consola de comandos

1	Entrada de comandos
	Entrada de texto para poder ejecutar los comandos.

Tabla 58 Entrada de comandos

2	Control de estado de los comandos
	Nos indicará si la sintaxis del comando escrito en la consola es válida.

Tabla 59 Control de estado de los comandos

4.3.5. Sección de configuración de servicios

Descripción

En esta sección podremos configurar los servicios de red de la aplicación.

Tabla 60 Sección de configuración de servicios

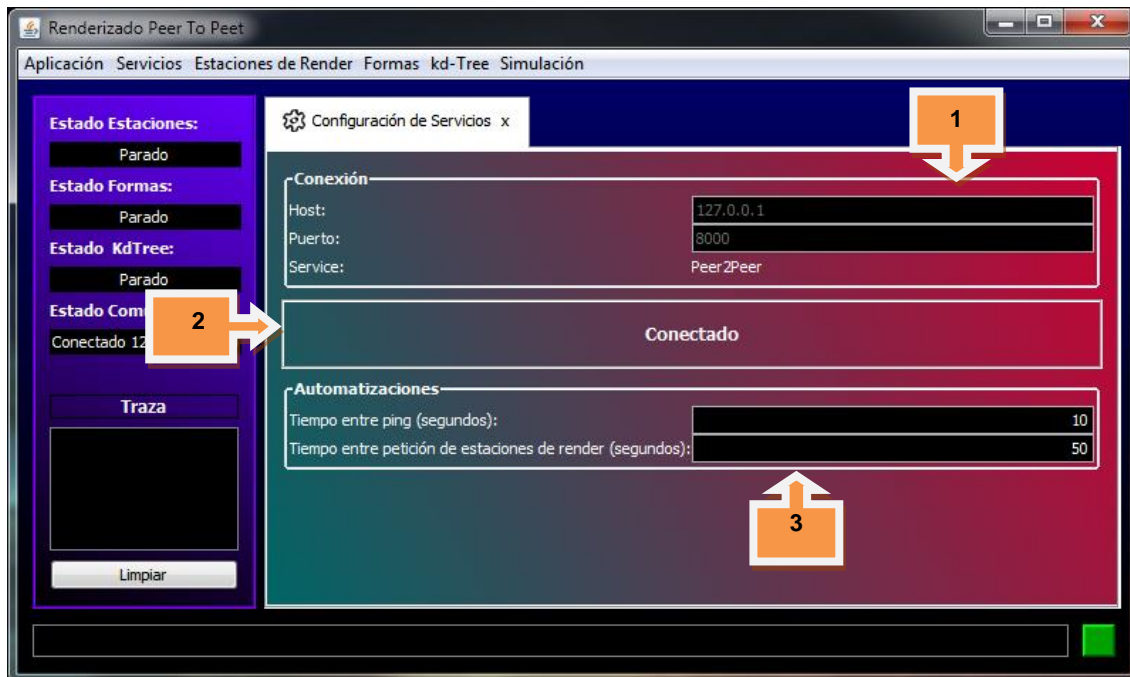


Ilustración 22 Sección de configuración de servicios

1 Datos de conexión de nuestra estación de render

Mediante este apartado podremos seleccionar el interfaz de red con el cual estamos operando.

- HOST: Normalmente será 127.0.0.1 (localhost) pero aquí podríamos escribir otro si tuviéramos más de una tarjeta de red.
- PUERTO: Aquí indicaremos el de salida de nuestros servicios.

Estos datos no podrán ser cambiados mientras los servicios estén activados.

Tabla 61 Datos de conexión de nuestra estación de render

2 Estado de los servicios

Nos indicará el estado actual de los servicios.

Tabla 62 Estado de los servicios

3 Automatizaciones de los servicios

Aquí podremos configurar los ciclos de tiempo de procesamiento de ciertas tareas programadas.

- Tiempo entre ping: Indica el lapso de tiempo que se espera el sistema al testear las distintas estaciones de render, con el fin de comprobar su disponibilidad. Si indicamos un cero el proceso no se ejecutará.
- Tiempo entre petición de estaciones de render: Indica el lapso de tiempo que se espera el sistema al solicitar a las distintas estaciones de render su listado de estaciones. Si indicamos un cero el proceso no se ejecutará.

Tabla 63 Automatizaciones de los servicios

4.3.6. Sección de monitoreo de Estaciones de Render

Descripción
En esta sección podremos monitorear el estado de las estaciones de render.

Tabla 64 Sección de monitoreo de Estaciones de Render

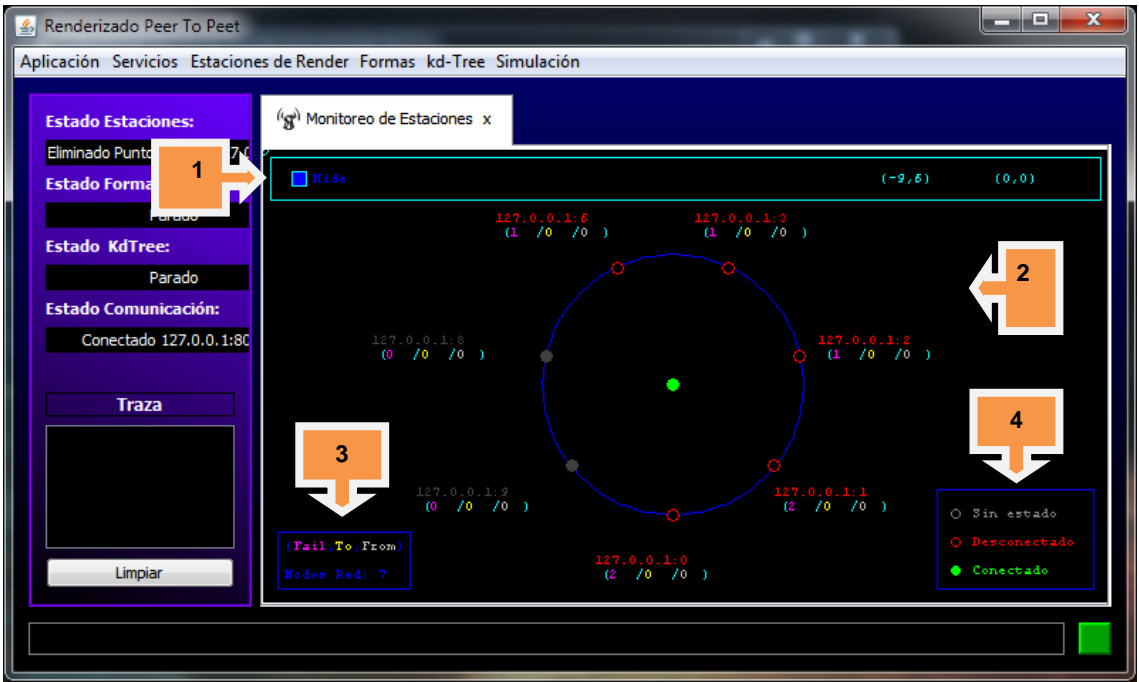


Ilustración 23 Sección de monitoreo de Estaciones de Render

1 Barra de opciones del monitor
Desde aquí podremos ocultar los cuadros de leyenda. Además de aportarnos de información adicional al seleccionar las estaciones.

Tabla 65 Barra de opciones del monitor

2 Estado de los servicios
Mapa de red de las estaciones de render. Aquí se nos indicará el estado de las estaciones así como datos significativos de la interacción con nuestra estación.

Tabla 66 Estado de los servicios

3 Leyenda de interacción
Nos indica una leyenda llevada al mostrar las estaciones.

- Fail: Nos indica el número de fallos detectados al interactuar con la estación.
- To: Indica las peticiones hacia la estación remota procesadas correctamente.
- From. Indica las peticiones desde la estación remota procesadas correctamente
- Nodos de red: Indica el número de nodos conocidos de nuestra red.

Tabla 67 Leyenda de interacción

4 Leyenda de estados
Nos indica una leyenda llevada al mostrar las estaciones.

- Sin estado: Indica que dicha estación no tiene un estado conocido.
- Desconectado: Indica que dicha estación no esta operativa.
- Conectado: Indica que dicha estación esta operativa.

Tabla 68 Leyenda de estados

4.3.7. Sección de monitoreo de Estaciones de Render

Descripción

En esta sección podremos monitorear el estado de las estaciones de render.

Tabla 69 Sección de monitoreo de Estaciones de Render

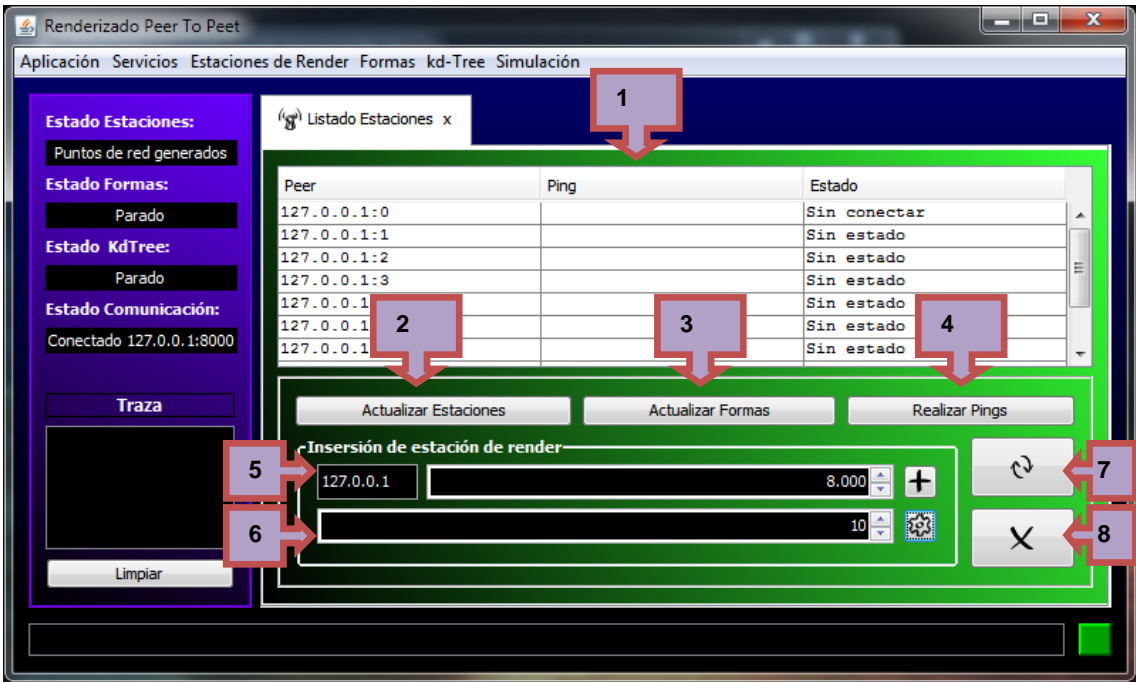


Ilustración 24 Sección de monitoreo de Estaciones de Render

1 Listado de Estaciones de Render

Listado de estaciones de render. Aquí se nos indica la dirección de la estación así como cuando se respondió al ping y su estado actual.

Tabla 70 Listado de Estaciones de Render

2 Acción de actualizar estaciones de render [Actualizar Estaciones]

Fuerza la consulta de estaciones de render a las estaciones seleccionadas.

Tabla 71 Acción de actualizar estaciones de render [Actualizar Estaciones]

3 Acción de actualizar formas [Actualizar formas]

Fuerza la consulta de formas a las estaciones seleccionadas.

Tabla 72 Acción de actualizar formas [Actualizar formas]

4 Acción de realizar ping [Realizar Pings]

Fuerza la consulta del estado actual de las estaciones seleccionadas.

Tabla 73 Acción de realizar ping [Realizar Pings]

5 Acción Insertar estación de render [Icono de añadir]

Añade una estación de render.

Tabla 74 Acción Insertar estación de render [Icono de añadir]

6	Acción generar estaciones de render [Icono de generar]
Genera un listado de estaciones de render.	
Tabla 75 Acción generar estaciones de render [Icono de generar]	
7	Acción de refrescar listado [Icono de refrescar]
Refresca el listado de estaciones de render.	
Tabla 76 Acción de refrescar listado [Icono de refrescar]	
8	Acción de eliminar estaciones de render [Icono de eliminar]
Elimina las estaciones seleccionadas.	
Tabla 77 Acción de eliminar estaciones de render [Icono de eliminar]	

4.3.8. Sección de estadísticas de Estaciones de Render

Descripción
En esta sección podremos ver estadísticas de las estaciones de render.

Tabla 78 Sección de estadísticas de Estaciones de Render

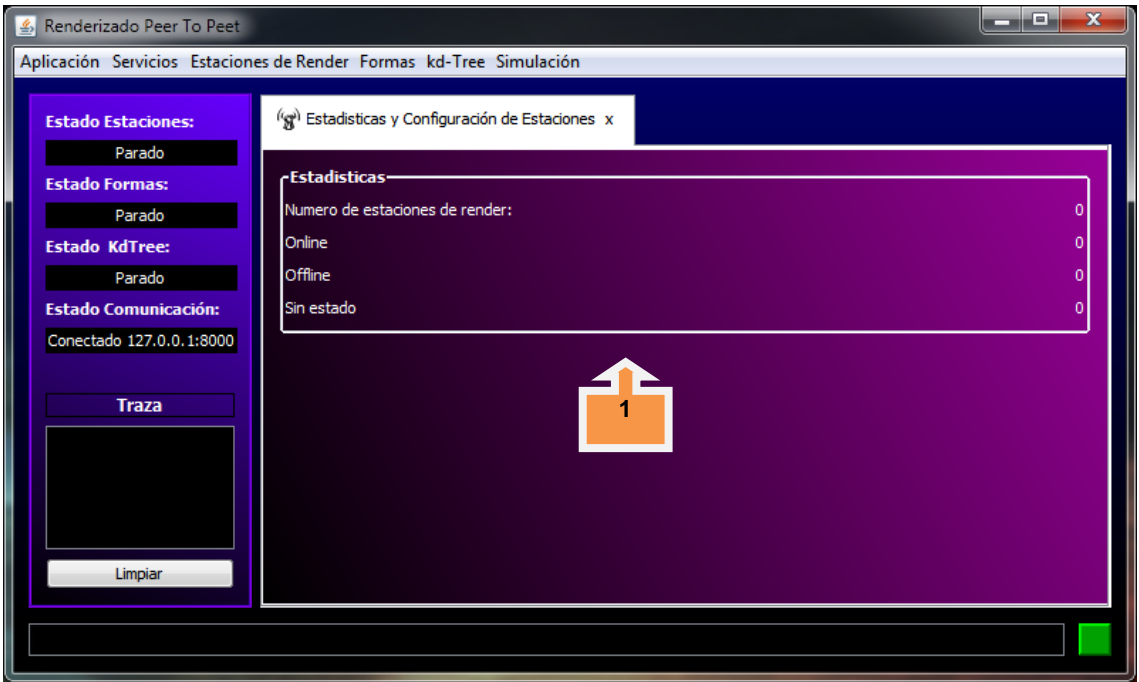


Ilustración 25 Sección de estadísticas de Estaciones de Render

1	Estadísticas de las estaciones de render
Nos muestra una serie de estadísticas sobre las estaciones de render	
<ul style="list-style-type: none">• Número de estaciones de render: Nos indica el total de estaciones conocidas.• Online: nos indica las estaciones conectadas.• Offline: Nos indica el número de estaciones desconectadas.• Sin estado: no indica el número de estaciones cuyo estado es desconocido.	

Tabla 79 Estadísticas de las estaciones de render

4.3.9. Sección de sintetizado de formas

Descripción

En esta sección podremos ver una sintetizado en 3D de las formas.

Tabla 80 Sección de sintetizado de formas

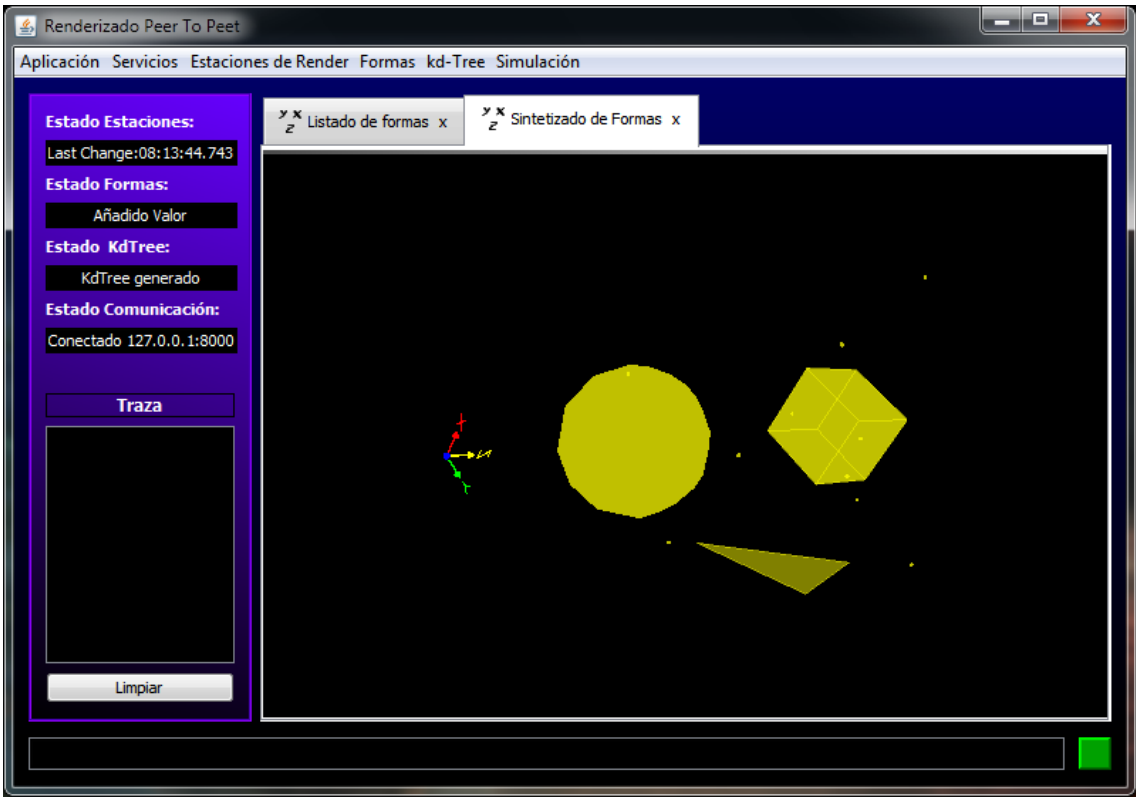


Ilustración 26 Sección de sintetizado de formas

Controles del escenario

Para movernos por el escenario utilizaremos el ratón:

- El botón derecho del ratón nos permitirá rotar el eje de coordenadas de la escena.
- La rueda del ratón nos permitirá desplazar el punto de vista alejándolo o acercándolo.
- El botón izquierdo nos permitirá desplazar el punto de vista arriba-abajo o derecha-izquierda.

Tabla 81 Controles del escenario

4.3.10. Sección de listado de formas

Descripción

En esta sección podremos ver un listado de las formas. También podremos gestionarlas.

Tabla 82 Sección de listado de formas

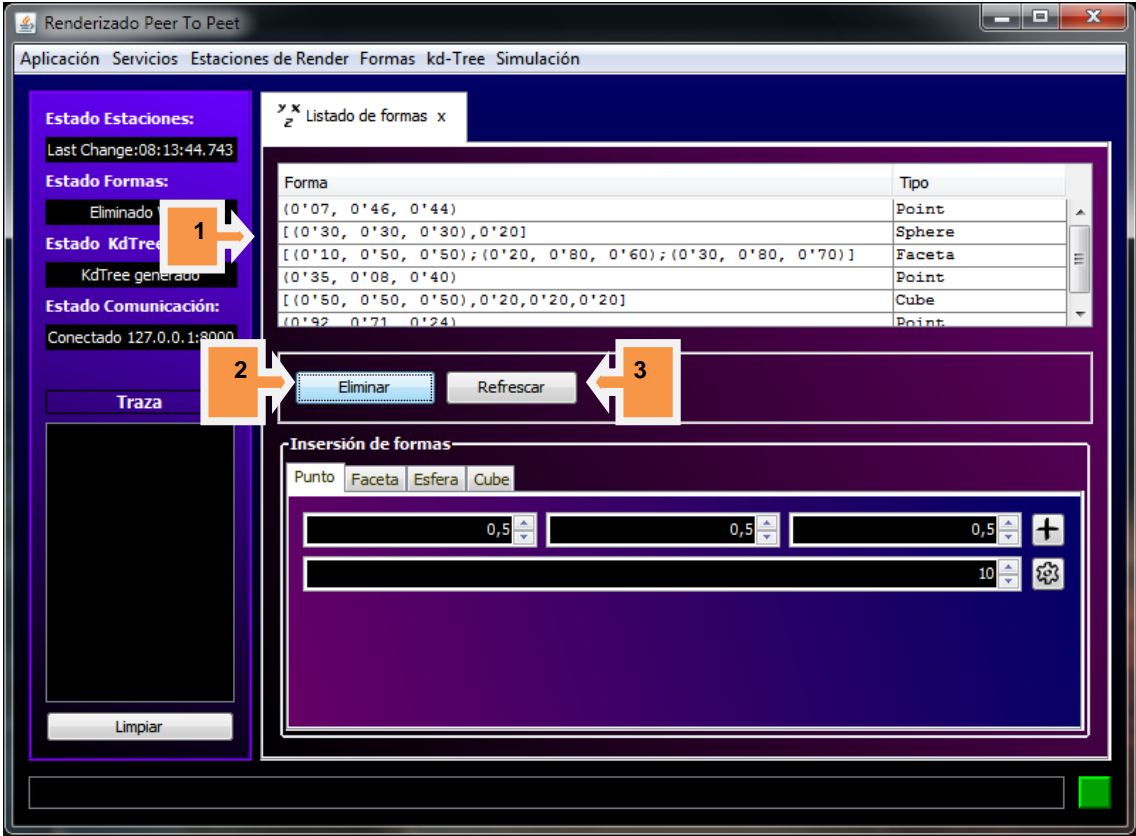


Ilustración 27 Sección de listado de formas

1

Listado de formas

Nos muestra el listado de formas, así como unos datos básicos de las mismas.

Tabla 83 Listado de formas

2

Acción de eliminar [Eliminar]

Elimina las formas seleccionadas..

Tabla 84 Acción de eliminar [Eliminar]

3

Acción refrescar [Refrescar]

Refresca el listado de formas.

Tabla 85 Acción refrescar [Refrescar]

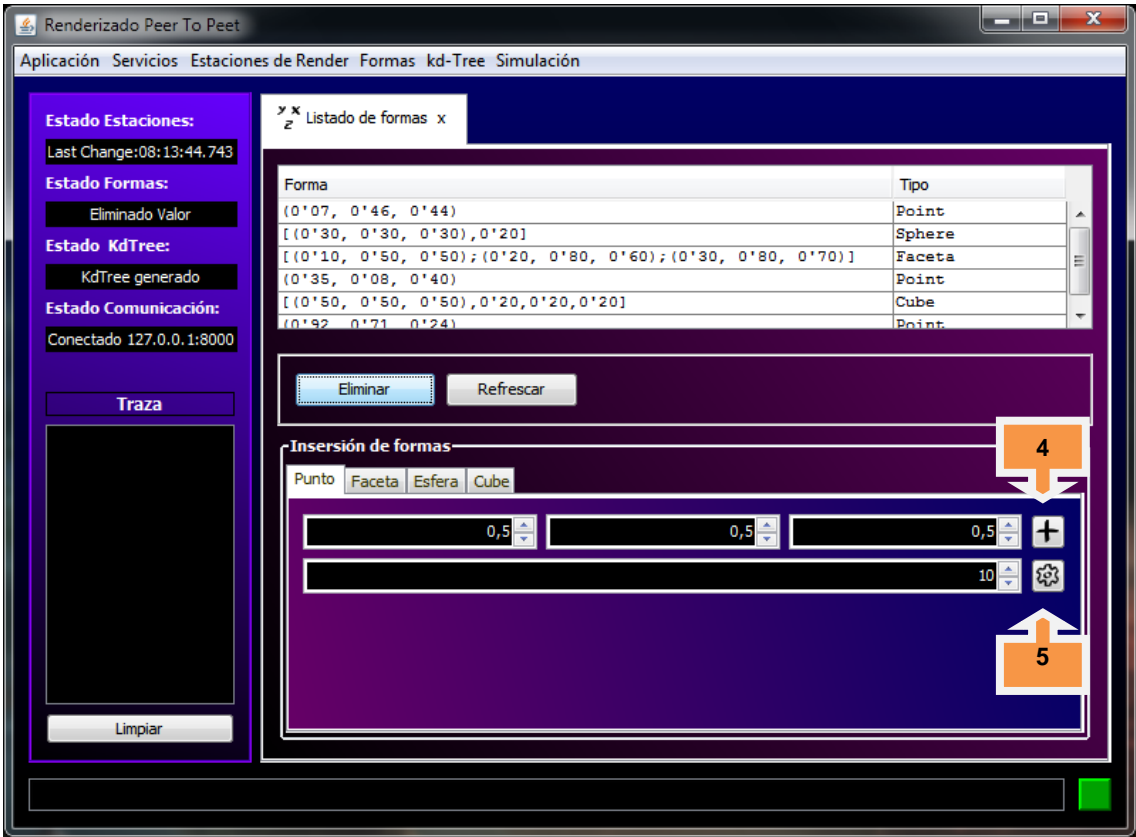


Ilustración 28 Añadir punto

4	Acción añadir punto [Icono añadir]
Añade un punto.	

Tabla 86 Acción añadir punto [Icono añadir]

5	Acción generar puntos [Icono generar]
Genera un listado de puntos aleatorios.	

Tabla 87 Acción generar puntos [Icono generar]

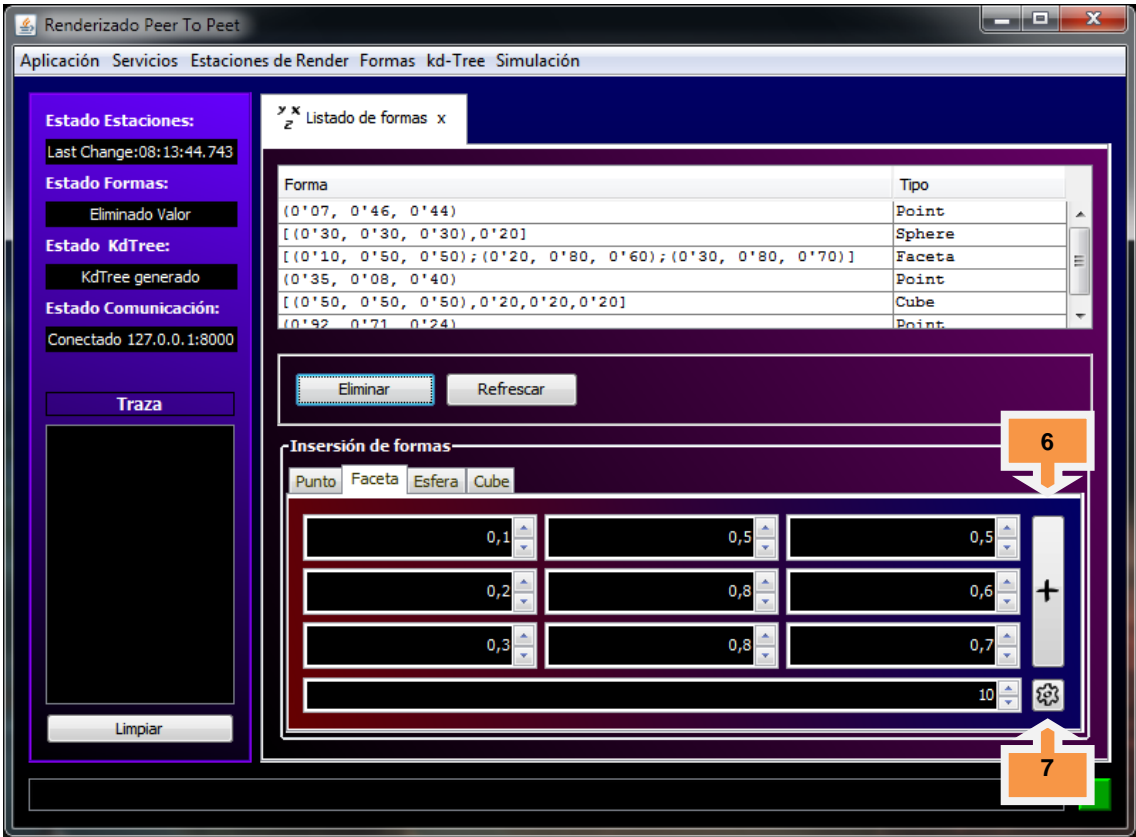


Ilustración 29 Añadir faceta

6	Acción añadir faceta [Icono añadir]
Añade una faceta.	
Tabla 88 Acción añadir faceta [Icono añadir]	
7	Acción generar facetas [Icono generar]
Genera un listado de facetas aleatorias.	
Tabla 89 Acción generar facetas [Icono generar]	

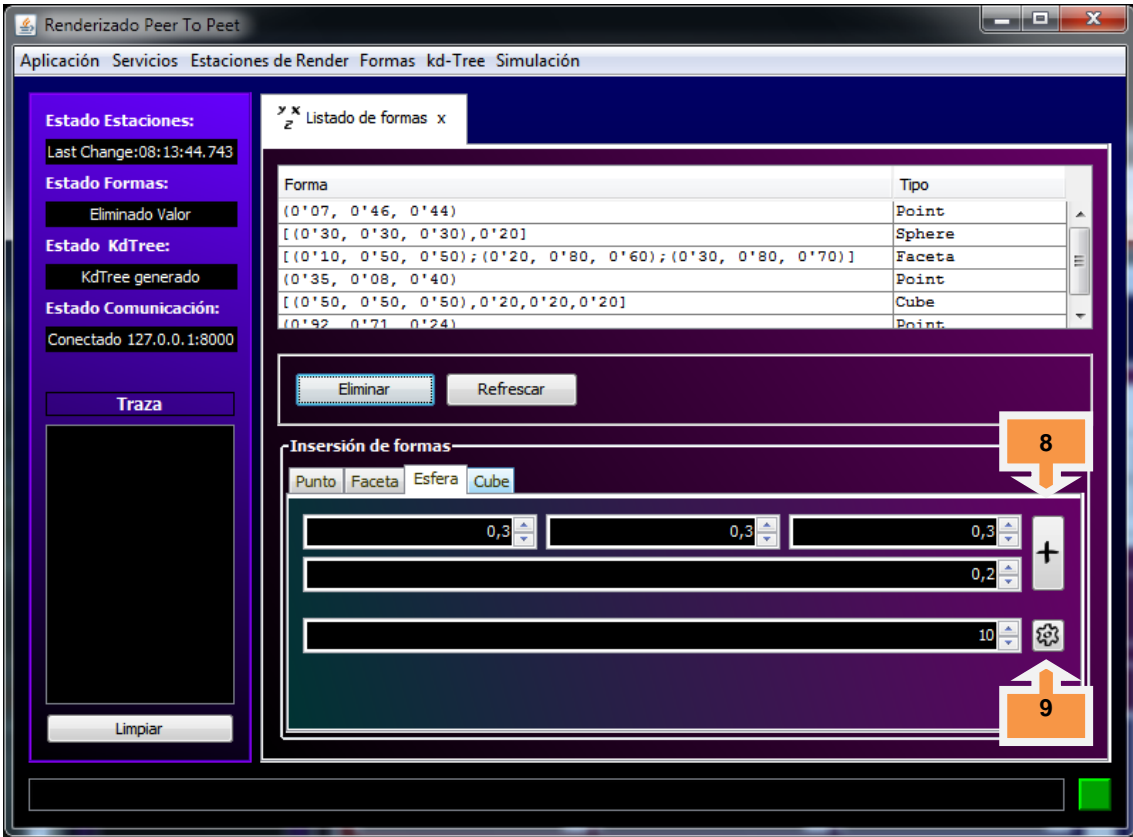


Ilustración 30 Añadir esfera

8	Acción añadir esfera [Icono añadir]
Añade una esfera.	

Tabla 90 Acción añadir esfera [Icono añadir]

9	Acción generar esferas [Icono generar]
Genera un listado de esferas aleatorias.	

Tabla 91 Acción generar esferas [Icono generar]

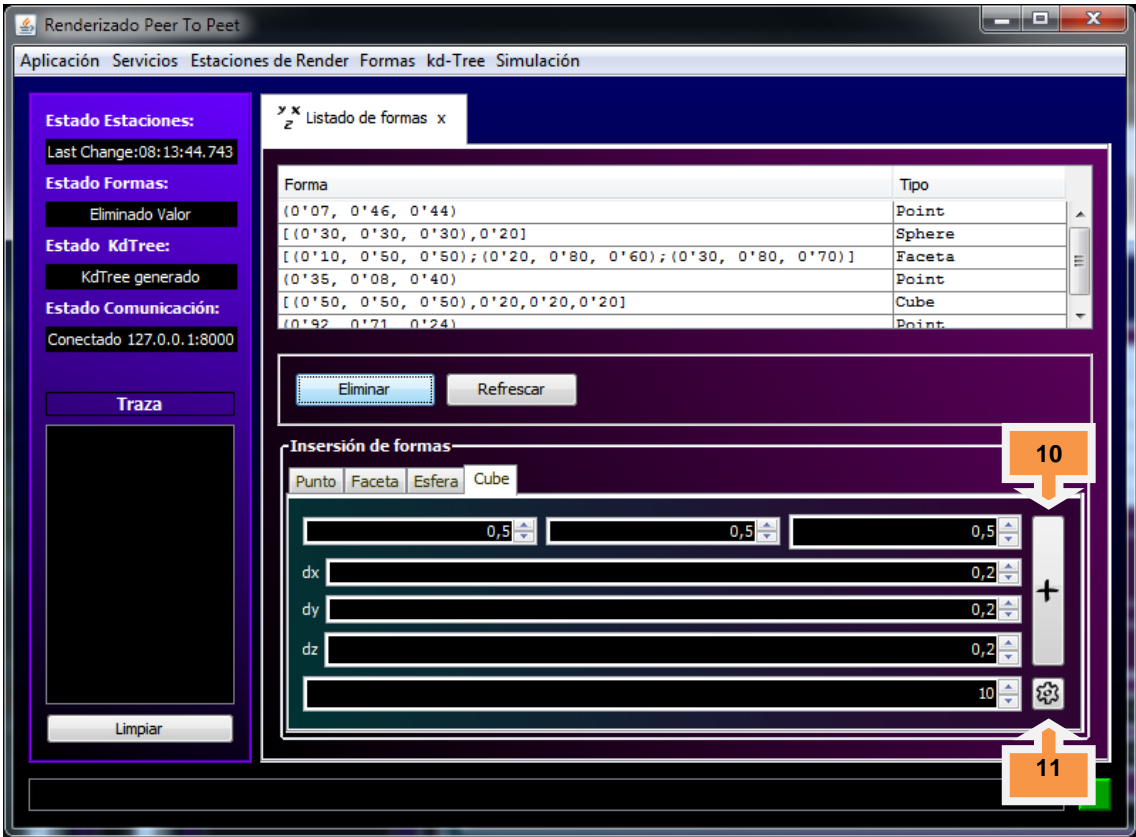


Ilustración 31 Añadir cubo

10	Acción añadir cubo [Icono añadir]
Añade un cubo.	
Tabla 92 Acción añadir cubo [Icono añadir]	
11	Acción generar cubos [Icono generar]
Genera un listado de cubos aleatorios.	
Tabla 93 Acción generar cubos [Icono generar]	

4.3.11. Sección de estadísticas de formas

Descripción
En esta sección podremos ver estadísticas sobre las formas.

Tabla 94 Sección de estadísticas de formas

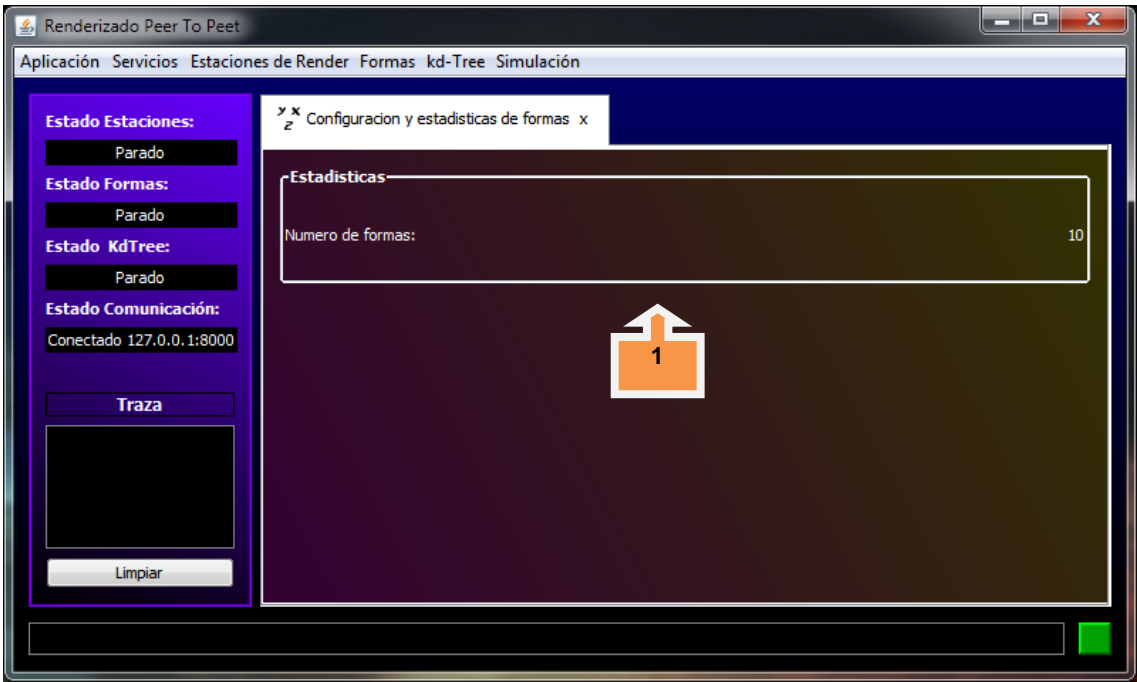


Ilustración 32 Sección de estadísticas de formas

1 Estadísticas
Nos muestra una serie de estadísticas sobre las formas.
<ul style="list-style-type: none">Numero de formas: Nos indica el número de formas.

Tabla 95 Estadísticas

4.3.12. Sección de representación del árbol kd

Descripción
En esta sección podremos ver una representación del árbol kd. Además nos permitirá ver que ramas se ha seguido durante la simulación.

Tabla 96 Sección de representación del árbol kd

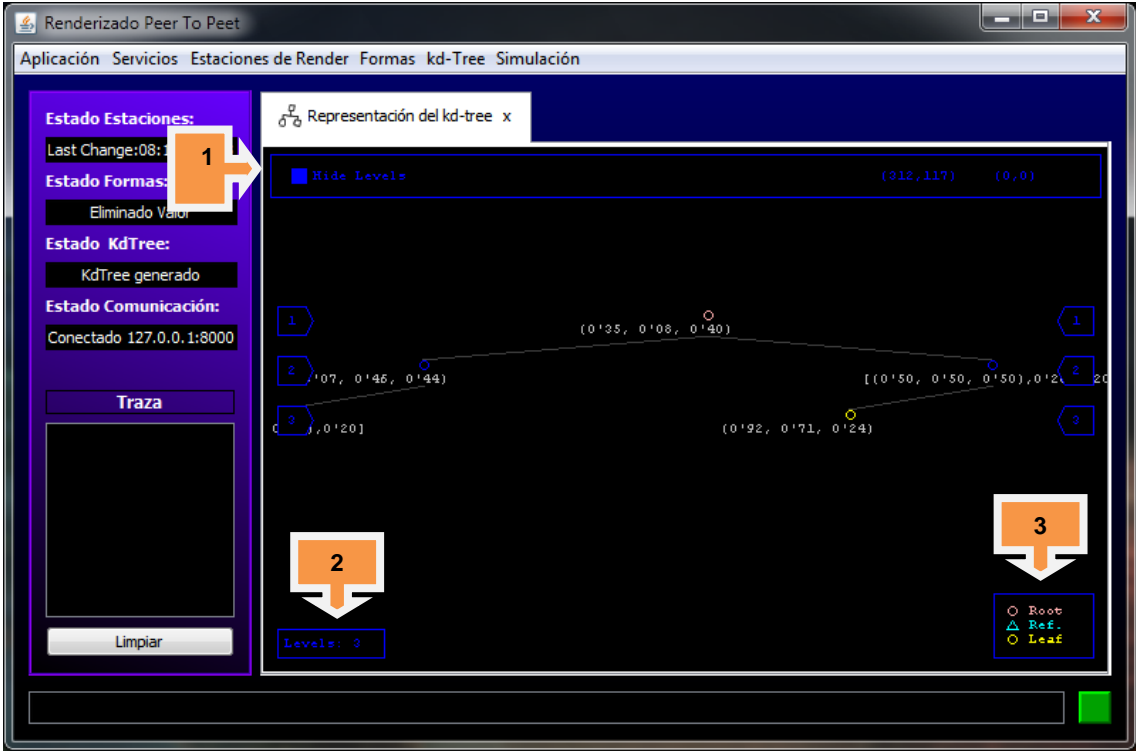


Ilustración 33 Sección de representación del árbol kd

1 Barra de opciones de la representación
Desde aquí podremos ocultar los cuadros de leyenda. Además de aportarnos de información adicional al seleccionar los nodos.

Tabla 97 Barra de opciones de la representación

2 Leyenda de niveles
Nos indica una leyenda con el numero de niveles del árbol.

Tabla 98 Leyenda de niveles

3 Leyenda de interacción
Nos indica el tipo de nodo.

- Root: Nodo raíz del árbol.
- Ref: Nodo que es una referencia.
- Leaf: Nos indica un nodo hoja.

Tabla 99 Leyenda de interacción

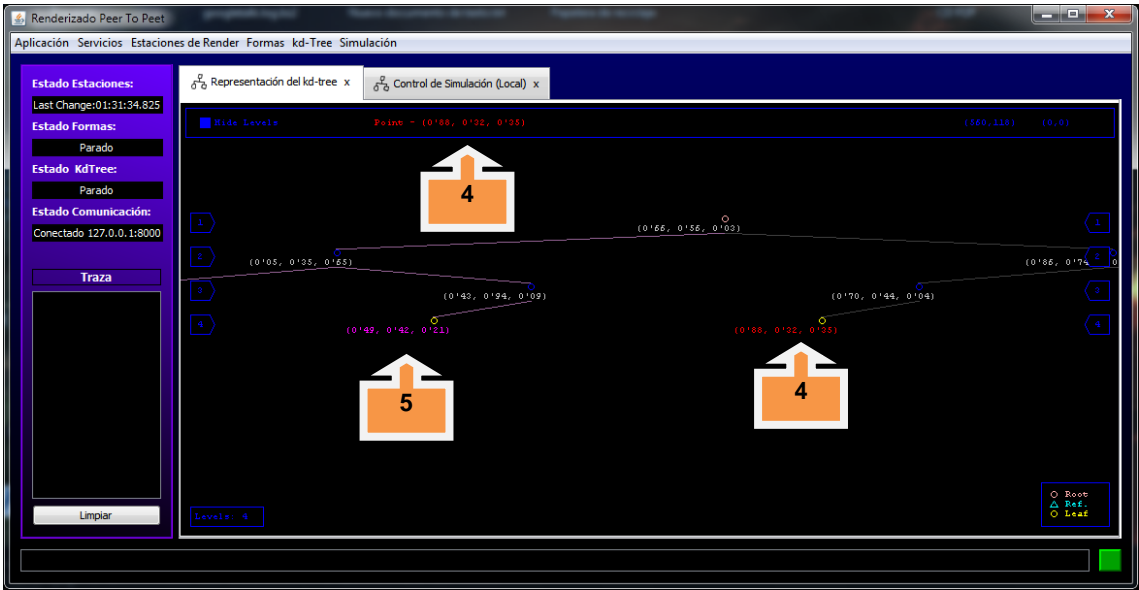


Ilustración 34 Selección de un nodo

4	Acción de selección de un nodo [Nodos dibujados]
Nos indica información de un nodo seleccionado en el árbol.	

Tabla 100 Acción de selección de un nodo [Nodos dibujados]

5	Visualización de una simulación
No indica el recorrido llevado durante la simulación por el árbol y los nodos que contienen una forma que cumplen las condiciones buscadas. Se puede observar los nodos que han sido procesados y las ramas no procesadas.	

Tabla 101 Visualización de una simulación

4.3.13. Sección de estadísticas y configuración del árbol kd

Descripción
En esta sección podremos ver estadísticas relacionadas con el árbol kd y configurar algunas propiedades.

Tabla 102 Sección de estadísticas y configuración del árbol kd

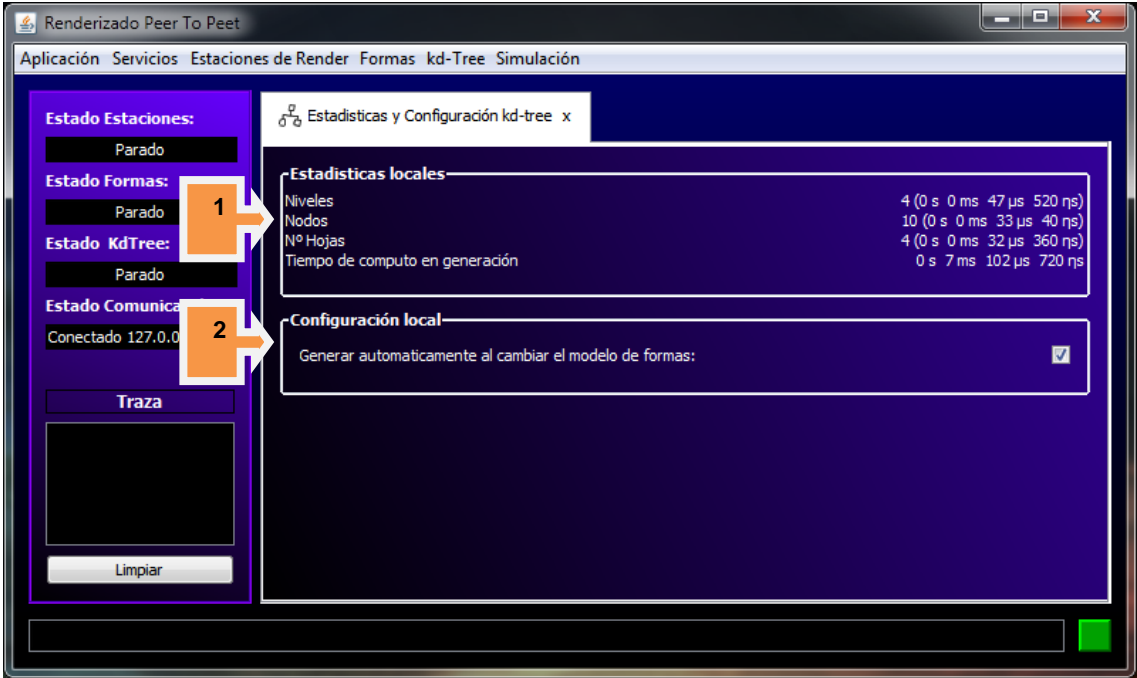


Ilustración 35 Sección de estadísticas y configuración del árbol kd

1	Estadísticas locales
Nos indica información referente a las estadísticas relacionadas con el árbol.	
<ul style="list-style-type: none">Niveles: Nos indica el número de niveles del árbol local.Nodos: Nos indica el número de nodos contenidos en el árbol local.Número de hojas: Nos indica el número de hojas del árbol local.Tiempo de cómputo en generación: El tiempo de proceso en generar el árbol local.	

Tabla 103 Estadísticas locales

2	Configuración local
Las configuraciones relacionadas con el árbol local.	
<ul style="list-style-type: none">Generar automáticamente al cambiar el modelo de formas: Indica si al cambiar el modelo de formas locales se debe volver a generar el árbol.	

Tabla 104 Configuración local

4.3.14. Sección de sintetizado de simulación

Descripción
En esta sección podremos ver un sintetizado en 3D de la simulación.

Tabla 105 Sección de sintetizado de simulación

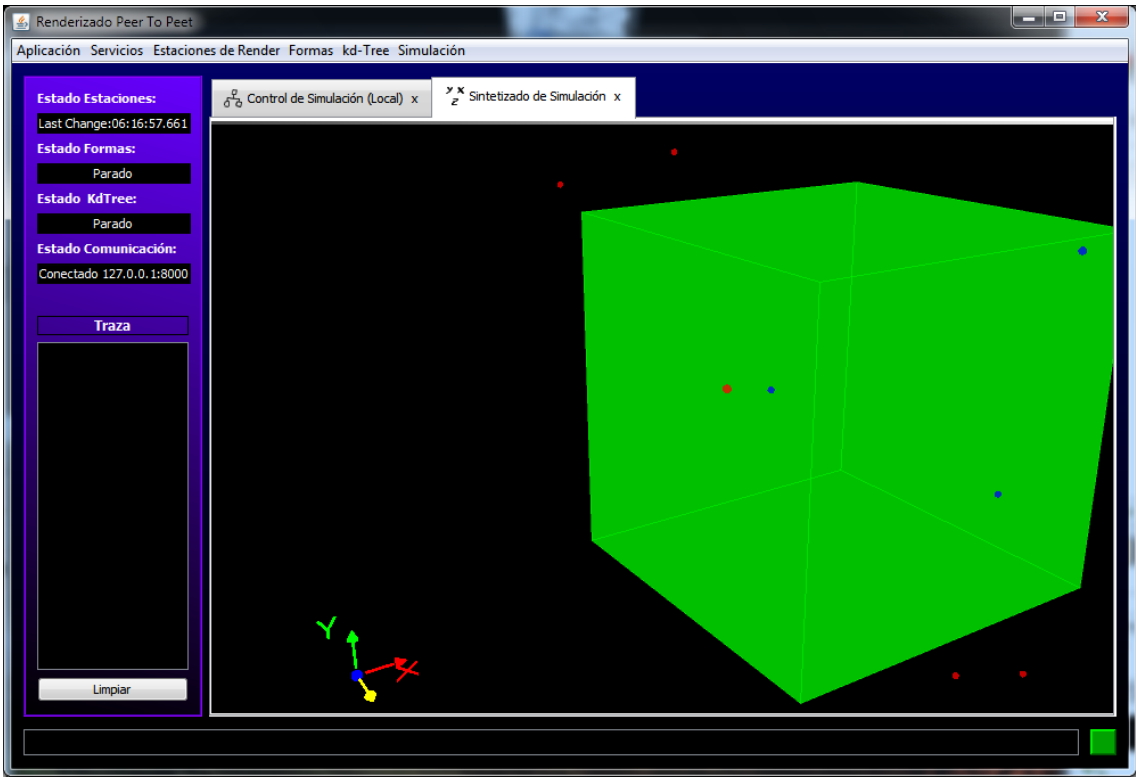


Ilustración 36 Sección de sintetizado de simulación

Descripción del modo de presentación
En color azul se indican las formas buscadas.
En color verde se muestra la referencia de la simulación.
En color rojo el resto de formas.

Tabla 106 Descripción del modo de presentación

Controles del escenario
Para movernos por el escenario utilizaremos el ratón:
<ul style="list-style-type: none">• El botón derecho del ratón nos permitirá rotar el eje de coordenadas de la escena.• La rueda del ratón nos permitirá desplazar el punto de vista alejándolo o acercándolo.• El botón izquierdo nos permitirá desplazar el punto de vista arriba-abajo o derecha-izquierda.

Tabla 107 Controles del escenario

4.3.15. Sección de control de simulación local

Descripción
En esta sección podremos realizar la simulación de distintos tipos de test en un ámbito local.

Tabla 108 Sección de control de simulación local

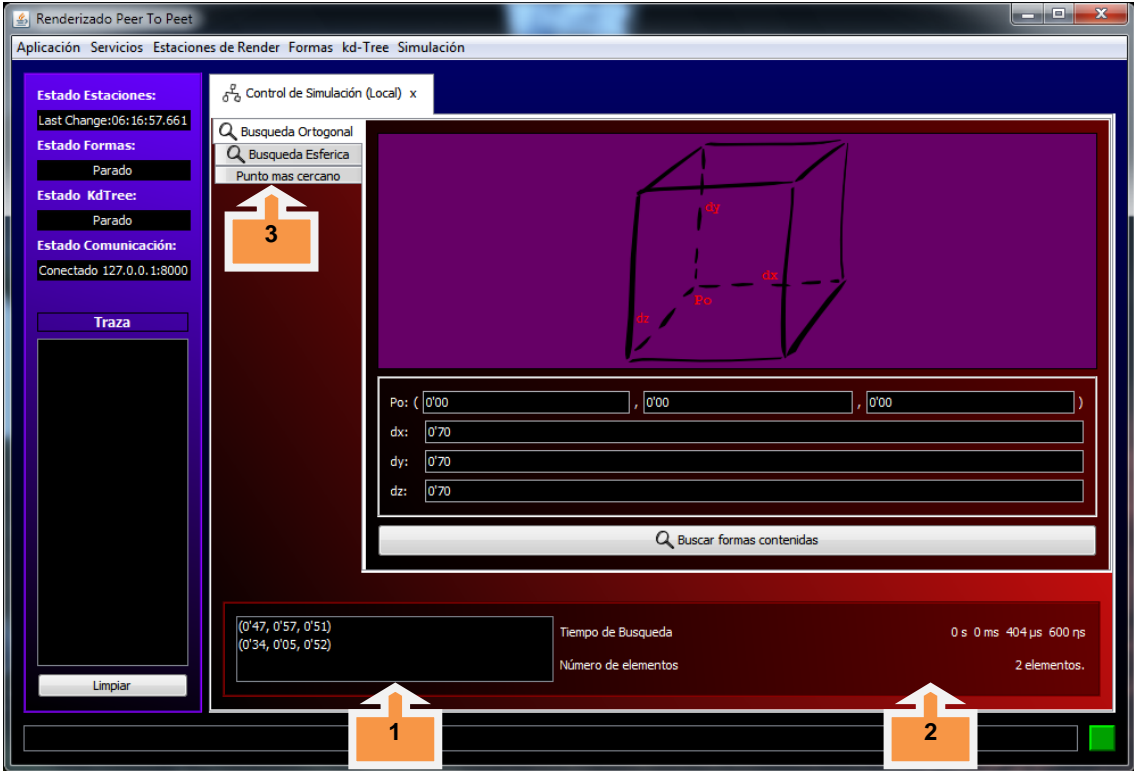


Ilustración 37 Sección de control de simulación local

1 Resultado de elementos encontrados al realizar el test
Muestra el resultado arrojado por el test. Nos mostrar las formas que cumplan con los requisitos.

Tabla 109 Resultado de elementos encontrados al realizar el test

2 Estadísticas de la simulación
Nos indica las estadísticas de la simulación.
Tiempo de búsqueda: Indica el tiempo empleado en el test.
Número de elementos: Nos indica el número de formas que cumplen con los requisitos.

Tabla 110 Estadísticas de la simulación

3 Acción selección del test [Pestañas laterales]
Selecciona el test a realizar.

Tabla 111 Acción selección del test [Pestañas laterales]

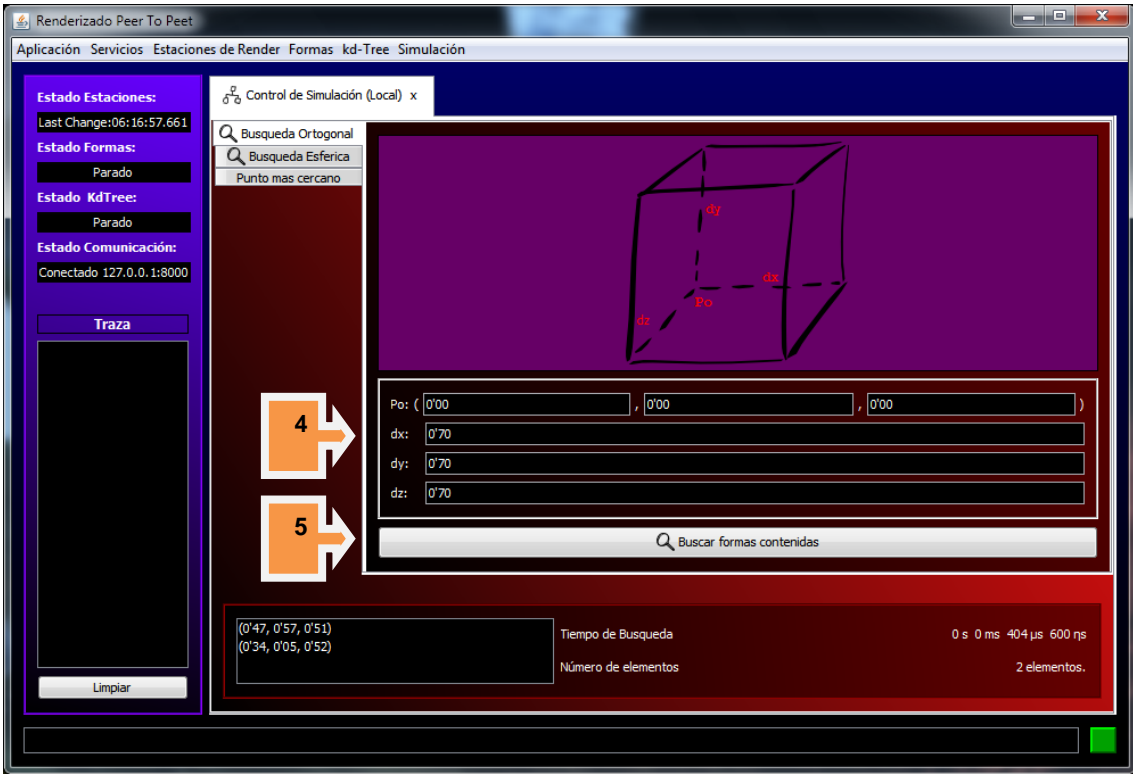


Ilustración 38 Datos de Simulación Ortogonal

4 Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen un cubo ortogonal.

Tabla 112 Datos de Simulación

5 Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca las formas contenidas dentro del cubo. Para ello recorre el árbol kd.

Tabla 113 Acción inicio del test de búsqueda [Buscar formas contenidas]

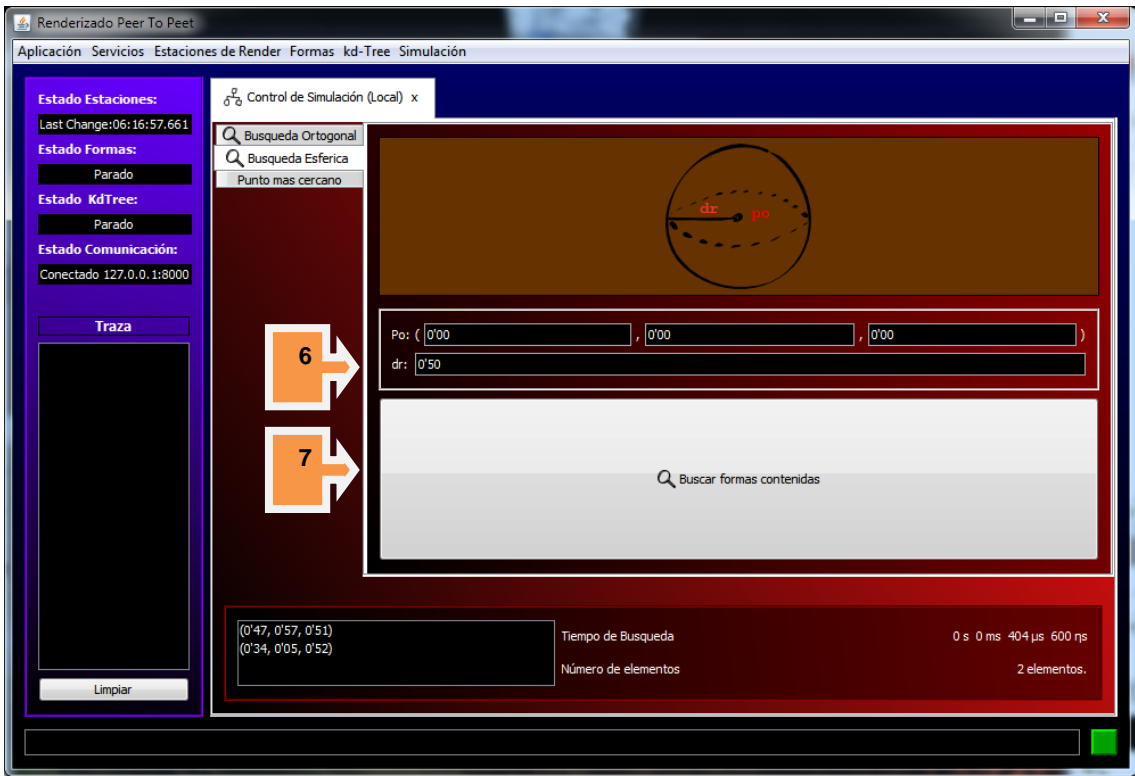


Ilustración 39 Datos de Simulación Esférica

6 Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen una esfera.

Tabla 114 Datos de Simulación

7 Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca las formas contenidas dentro de la esfera. Para ello recorre el árbol kd.

Tabla 115 Acción inicio del test de búsqueda [Buscar formas contenidas]

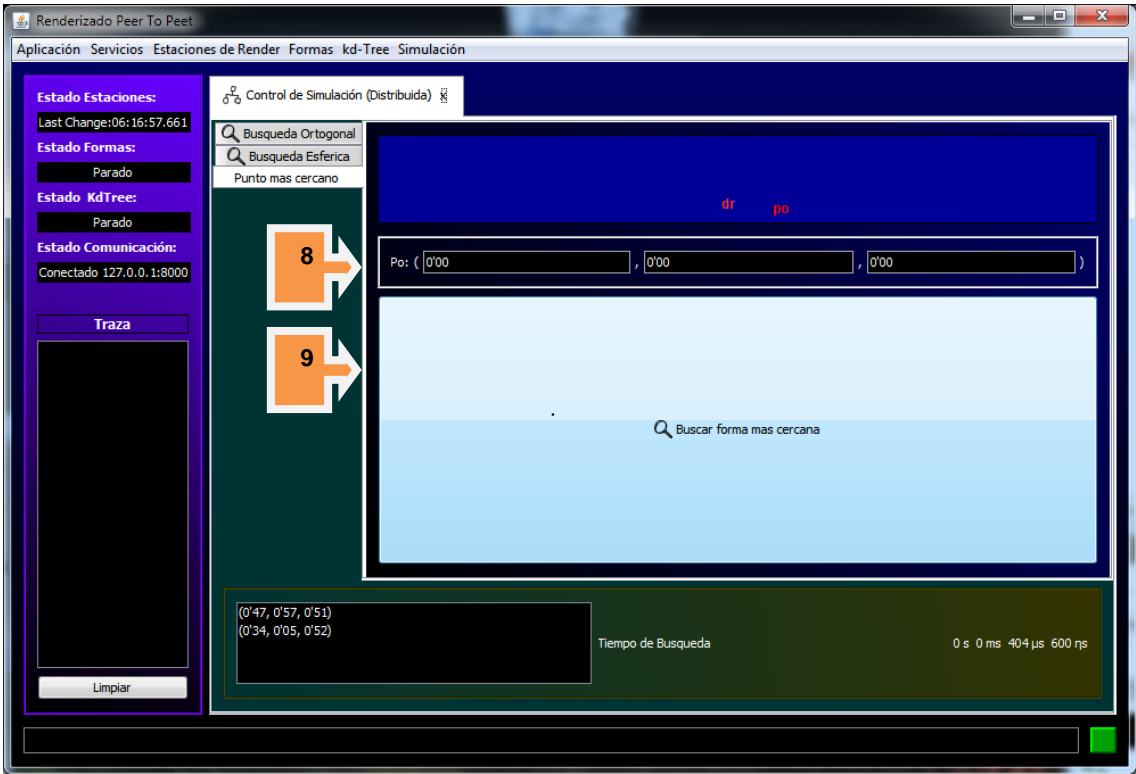


Ilustración 40 Datos de Simulación más cercano

8	Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen una coordenada en el espacio.	

Tabla 116 Datos de Simulación

9	Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca la forma más cercana a una coordenada espacial. Para ello recorre el árbol kd.	

Tabla 117 Acción inicio del test de búsqueda [Buscar formas contenidas]

4.3.16. Sección de control de simulación distribuida

Descripción
En esta sección podremos realizar la simulación de distintos tipos de test en un ámbito distribuido.

Tabla 118 Sección de control de simulación distribuida

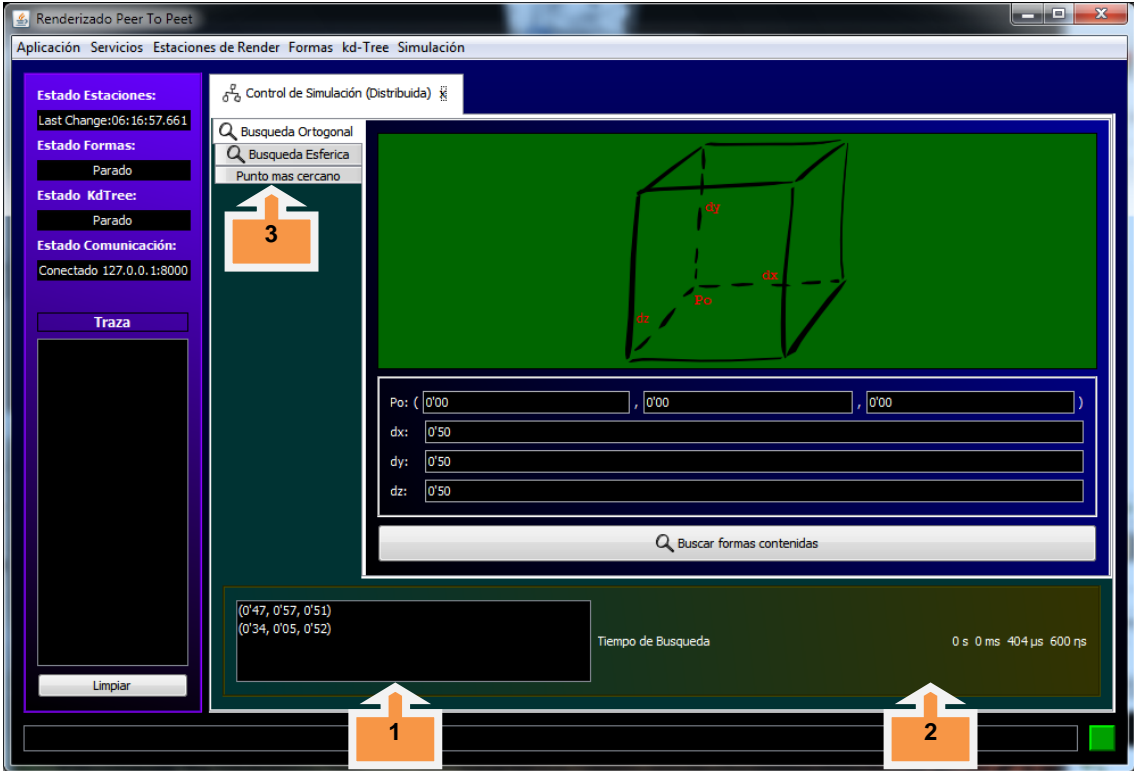


Ilustración 41 Sección de control de simulación distribuida

1 Resultado de elementos encontrados al realizar el test
Muestra el resultado arrojado por el test. Nos mostrar las formas que cumplan con los requisitos.

Tabla 119 Resultado de elementos encontrados al realizar el test

2 Estadísticas de la simulación.
Nos indica las estadísticas de la simulación.
Tiempo de búsqueda: Indica el tiempo empleado en el test.
Número de elementos: Nos indica el número de formas que cumplen con los requisitos.

Tabla 120 Estadísticas de la simulación

3 Acción selección del test [Pestañas laterales]
Selecciona el test a realizar.

Tabla 121 Acción selección del test [Pestañas laterales]

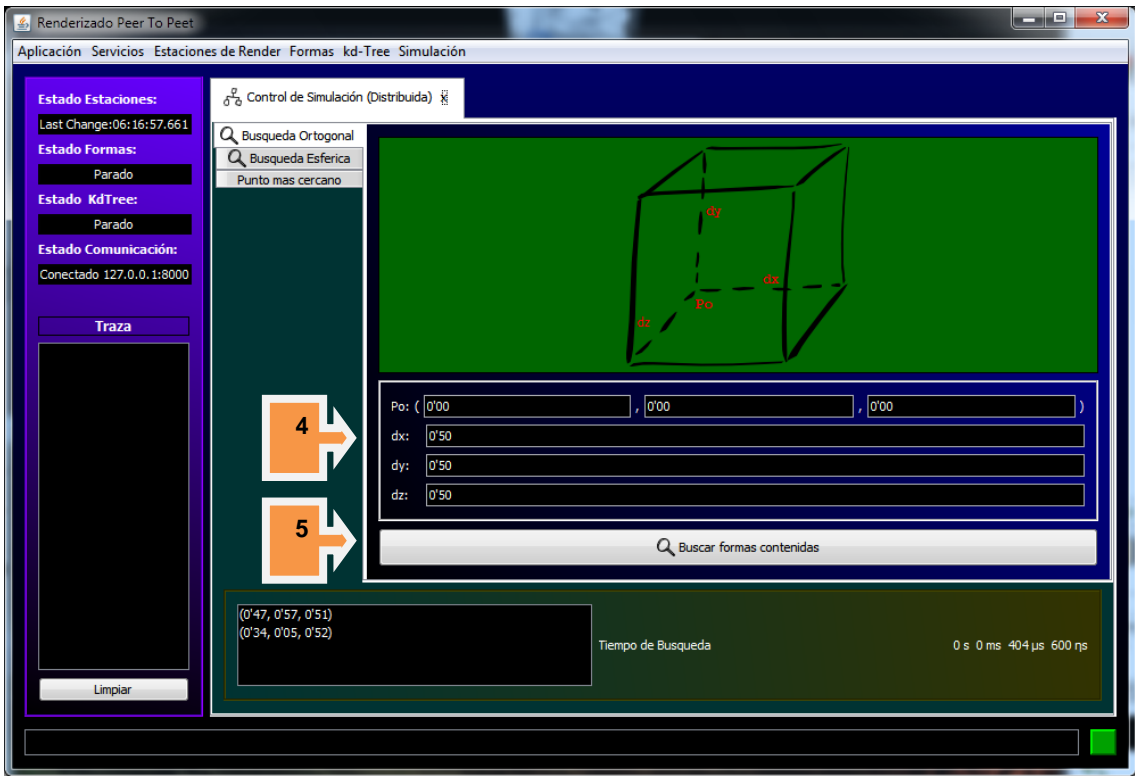


Ilustración 42 Datos de Simulación Ortogonal

4	Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen un cubo ortogonal.	
Tabla 122 Datos de Simulación	
5	Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca las formas contenidas dentro del cubo. Para ello recorre el árbol kd.	
Tabla 123 Acción inicio del test de búsqueda [Buscar formas contenidas]	

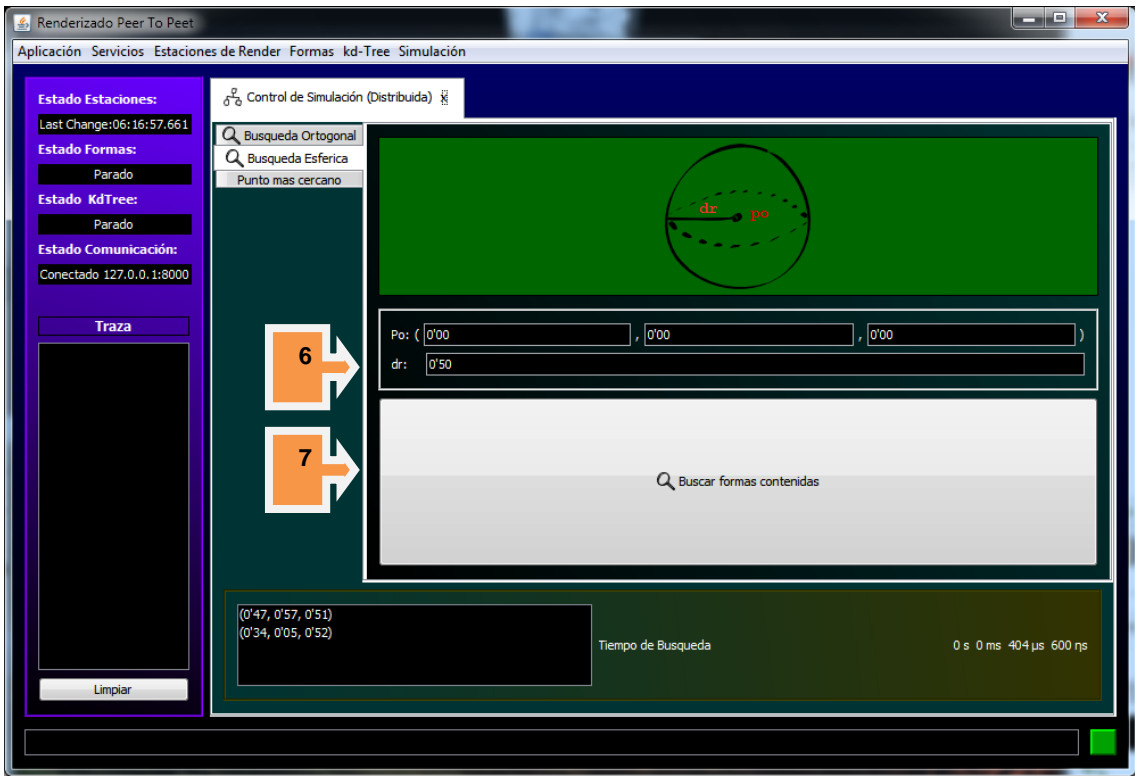


Ilustración 43 Datos de Simulación Esférica

6	Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen una esfera.	

Tabla 124 Datos de Simulación

7	Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca las formas contenidas dentro de la esfera. Para ello recorre el árbol kd.	

Tabla 125 Acción inicio del test de búsqueda [Buscar formas contenidas]

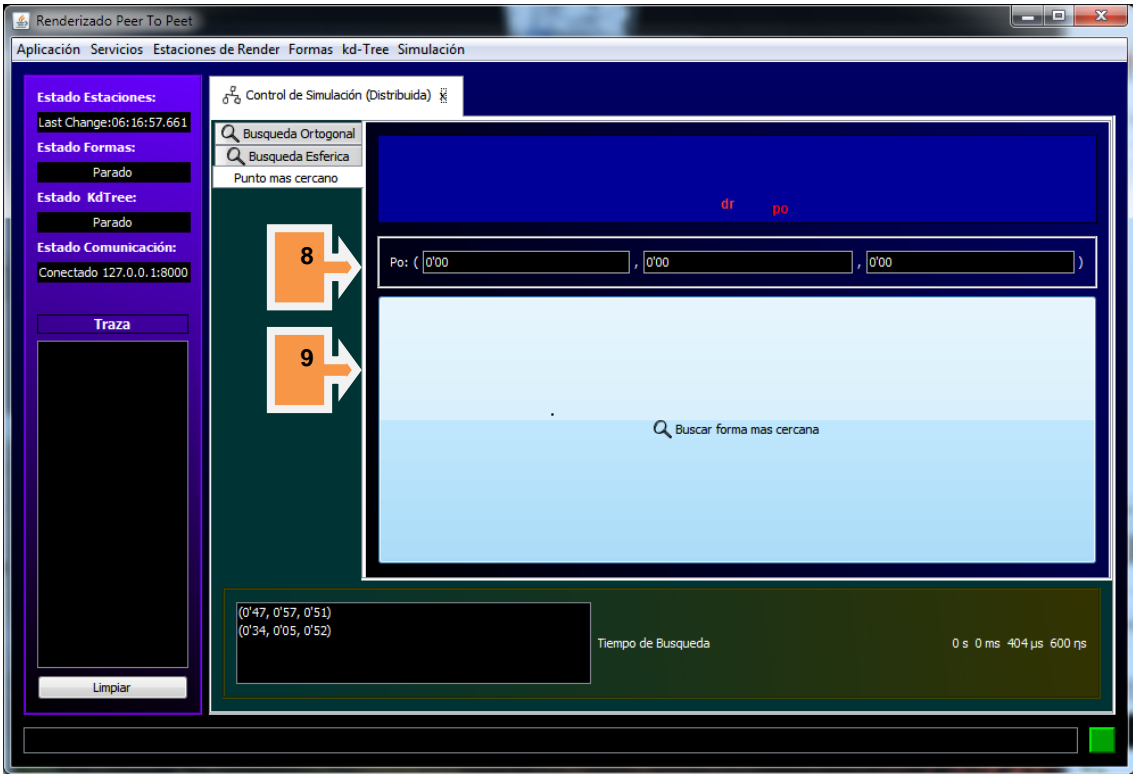


Ilustración 44 Datos de Simulación más cercano

8	Datos de Simulación
Lugar donde se introducen los datos de la simulación. En este caso pondremos los datos básicos que definen una coordenada en el espacio.	

Tabla 126 Datos de Simulación

9	Acción inicio del test de búsqueda [Buscar formas contenidas]
Busca la forma más cercana a una coordenada espacial. Para ello recorre el árbol kd.	

Tabla 127 Acción inicio del test de búsqueda [Buscar formas contenidas]

4.3.17. Sección de configuración y estadísticas de la simulación.

Descripción

En esta sección podremos ver estadísticas de la simulación, así como, configurar datos concernientes a la simulación.

Tabla 128 Sección de configuración y estadísticas de la simulación

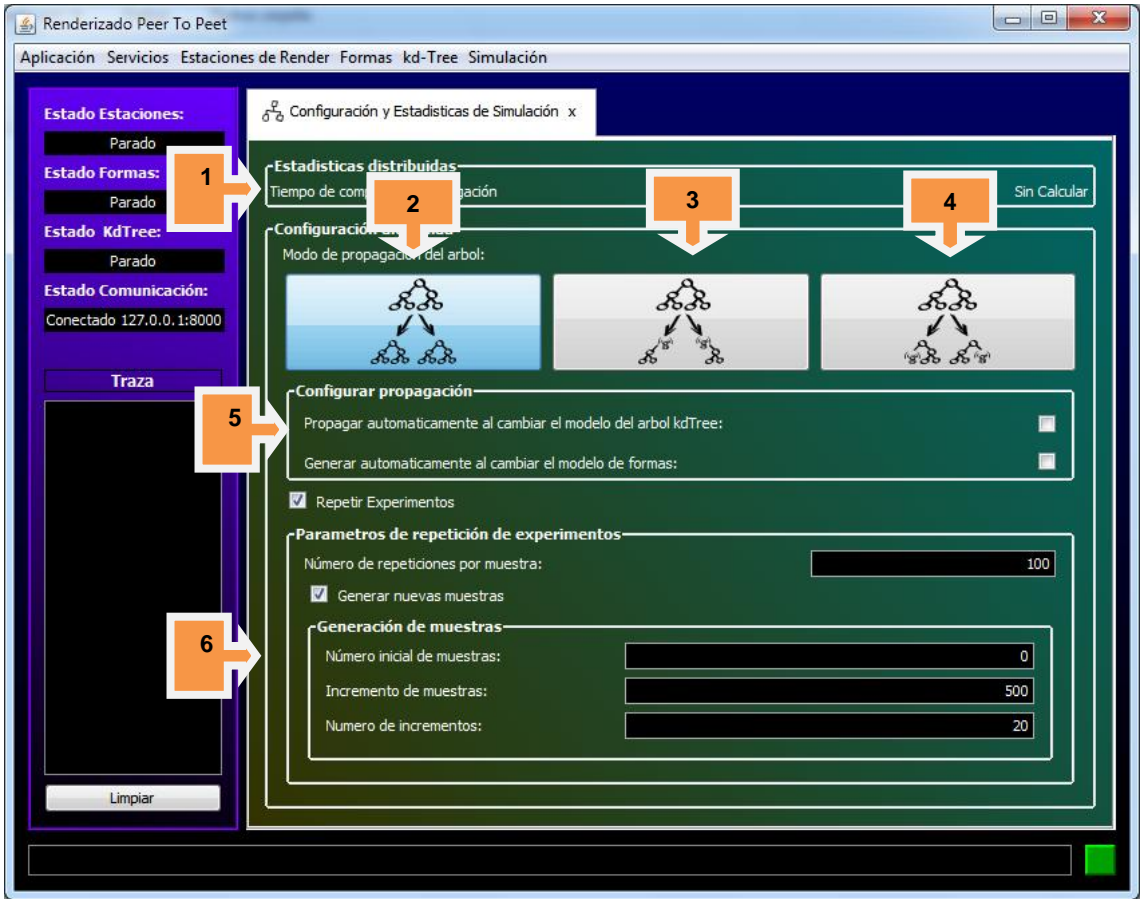


Ilustración 45 Sección de configuración y estadísticas de la simulación

1	Estadísticas distribuidas
Estadísticas relacionadas con la simulación: Tiempo de cómputo de la propagación: Nos indica el tiempo que tarda en propagarse el árbol kd.	

Tabla 129 Estadísticas distribuidas

2	Modo de propagación del árbol [Icono de Propagar todo]
Indica que se propagara todo el árbol.	

Tabla 130 Modo de propagación del árbol [Icono de Propagar todo]

3	Modo de propagación del árbol [Icono de Propagar ramas]
Indica que se propagaran solo ramas del árbol.	

Tabla 131 Modo de propagación del árbol [Icono de Propagar ramas]

4 Modo de propagación del árbol [Icono de Propagar niveles]

Indica que se propagaran ramas del árbol y se mantendrán en común los primeros niveles del árbol.

Tabla 132 Modo de propagación del árbol [Icono de Propagar niveles]

5 Configurar propagación

Opciones de configuración de la propagación:

- Propagar automáticamente al cambiar el modelo del árbol kd: Indica que se realizara la propagación del árbol al cambiar el modelo del árbol kd.
- Generar automáticamente al cambiar el modelo de formas: Indica que se generar automáticamente el árbol al generar el listado de formas.

Tabla 133 Configurar propagación

5 Configurar de repeticiones

Opciones de configuración de las repeticiones:

- Repetir experimentos: Habilita las repeticiones
- Número de repeticiones por muestra: Indica el número de veces que se repite cada muestra.
- Generar Cada muestra: Indica que en cada repetición se generar las muestras de manera aleatoria.
- Número inicial de muestras: indica el número de muestras iniciales.
- Incremento de las muestras: Indica el incremento de las muestras en cada generación:
- Numero de Incrementos. Indica el número de veces que se generan las muestras.

Tabla 134 Configuración de repeticiones

El uso de estas posibilidades tiene sentido cuando se desea realizar tareas de geolocalización del árbol.

4.4. Iniciando la aplicación

Para ejecutar la aplicación debes buscar el directorio donde lo has instalado (Ver apartado de instalación de este manual).

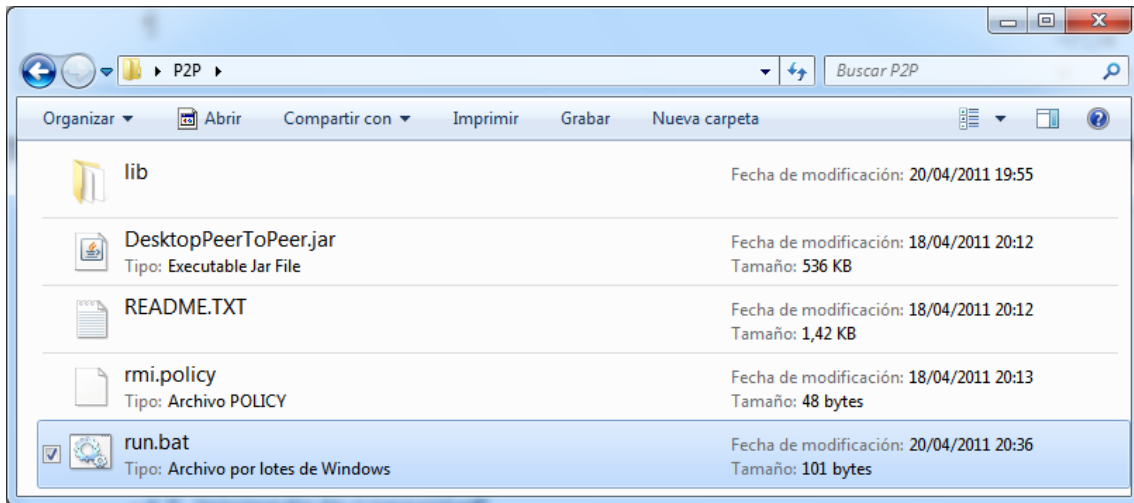


Ilustración 46 Iniciando la aplicación

En el directorio de instalación encontraras un archivo que se llama run.bat.

Lo ejecutas dándole doble clic con el ratón, y se abrirá la aplicación.

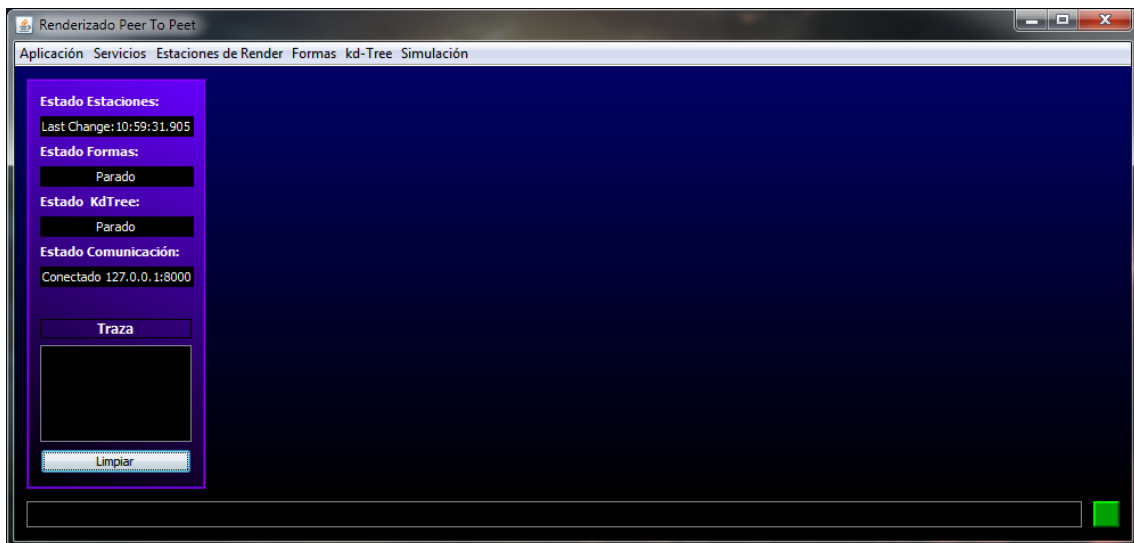


Ilustración 47 Pantalla de inicio de la aplicación

¡En hora buena, has conseguido arrancar la aplicación!

4.5. Iniciando la conexión de los servicios

4.5.1. Inicio rápido

La aplicación nada más arrancar inicia los servicios de red. Para ello automáticamente busca un puerto a partir del 8000 que este libre. Si por algún motivo no pudiéramos iniciar los servicios:

- Asegúrate de que tienes los permisos de acceso a la red.
- Asegúrate de que no tengas otra aplicación trabajando en el sistema apuntando al mismo host y puerto.
- Verifica que no tengas un firewall bloqueando la aplicación.
- Prueba a cambiar el puerto por otro.

4.5.2. Cambio del puerto o el host de los servicios

De querer cambiar su configuración nos dirigiremos al menú de servicios a la opción de configuración y preferencias:

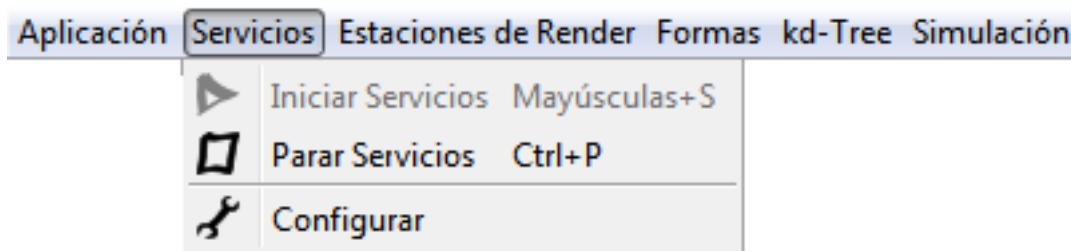


Ilustración 48 Menú de servicios

Una vez allí pararemos los servicios. En ese mismo menú accederemos a la configuración de los servicios y cambiaremos el puerto o el host.

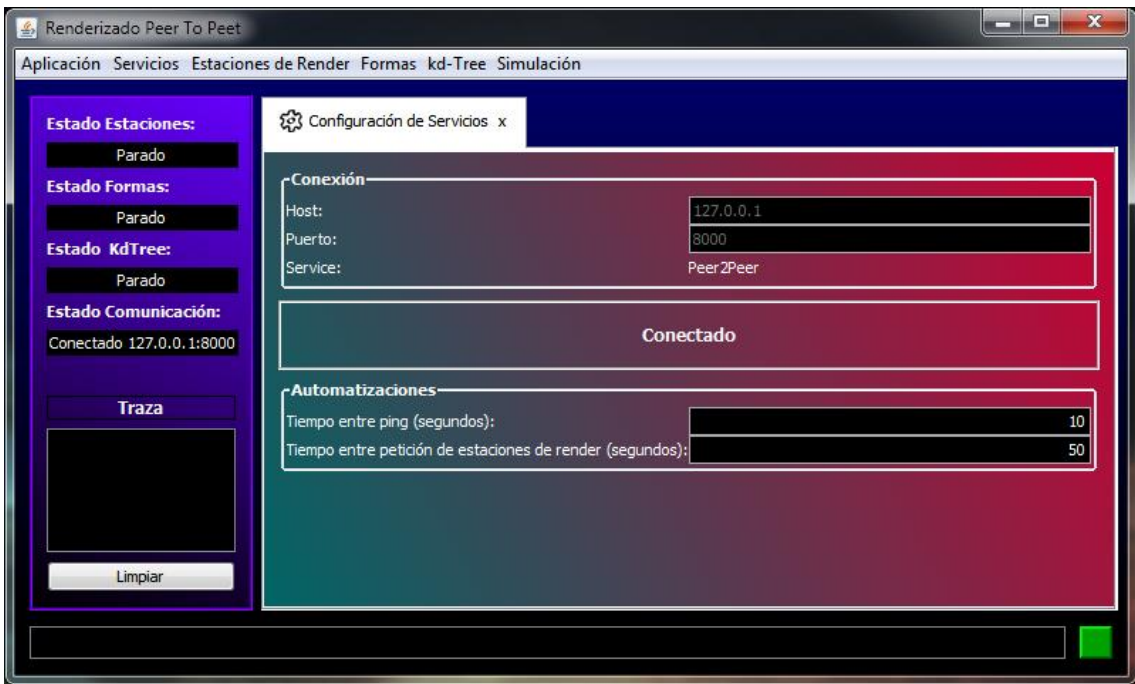


Ilustración 49 Sección de configuración de servicios

4.6. Gestión de los servicios

De querer cambiar la configuración de los servicios iremos al menú de servicios y allí accederemos a la configuración

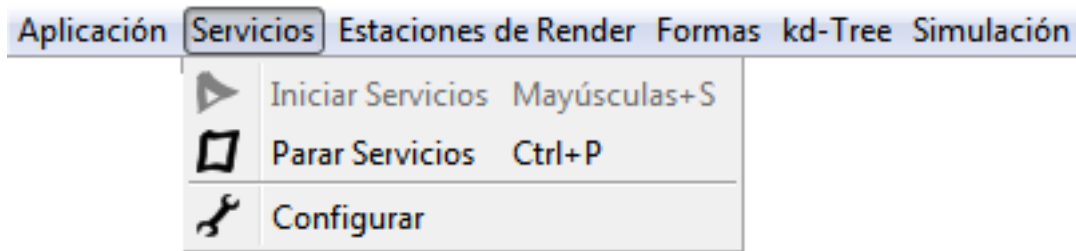


Ilustración 50 Menú de servicios

Para poder cambiar el host o el puerto deberemos tener los servicios parados.

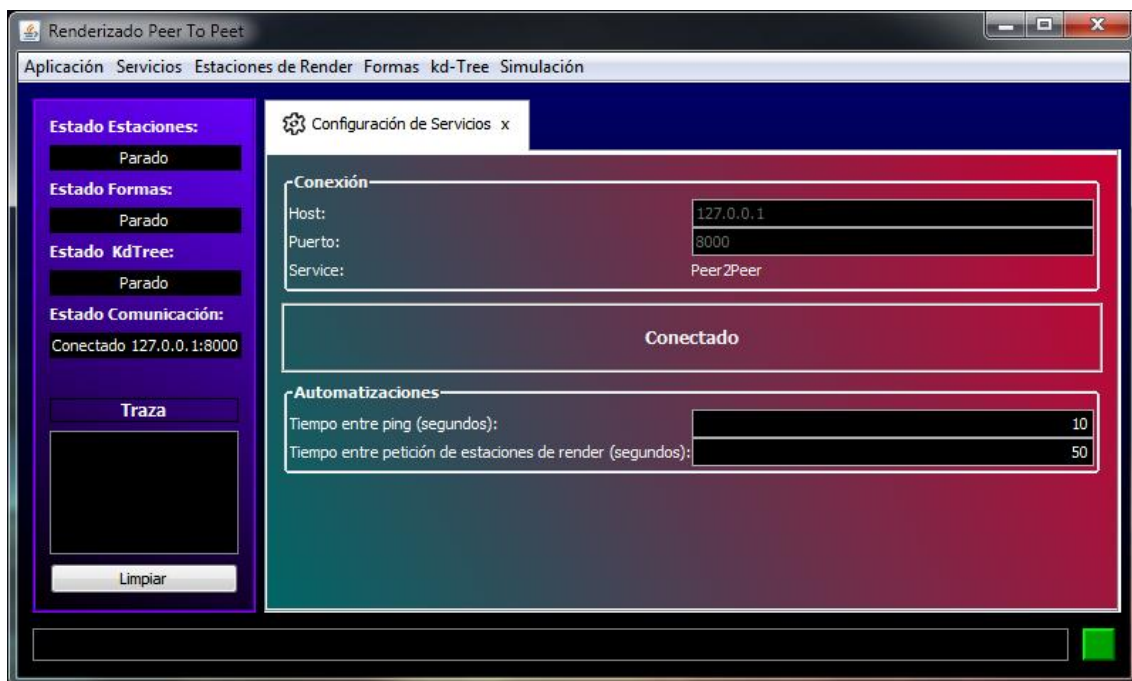


Ilustración 51 Sección de configuración de servicios

Desde aquí podremos configurar distintas automatizaciones del programa, relacionadas con los servicios.

4.7. Gestión de Estaciones de Render

4.7.1. ¿Qué son las estaciones de render?

Las estaciones de render son estaciones como la tuya. Tanto ellos como tu comparten los recursos de su equipo con el fin de compartir recursos abocados a tareas de render.

4.7.2. Monitorizar

En la opción monitorizar del submenú de las estaciones de render tendremos una vista gráfica donde hacer un seguimiento de las estaciones de la red.

4.7.3. Añadir estaciones de render

Existen varias maneras de agregar estaciones de render a nuestra aplicación. Existe un mecanismo que recorre cíclicamente las estaciones para consultar el listado de estaciones de las mismas. Esta operación se puede forzar en el listado de estaciones de render.

Añadir las de manera manual

Simplemente, en el listado de estaciones de render, introduciremos los datos de la estación que queremos añadir y procederemos a su inserción.

Cargarlas de disco

En el submenú de servicios pulsaremos la opción de cargar. Se cargara el listado de estaciones previamente salvadas.

Importar desde otra estación

En el submenú de servicios pulsaremos la opción de importar. Importaremos el listado de estaciones previamente exportadas.

Actualizarlas desde una estación conocida

En el listado de estaciones de render seleccionaremos la estación de la que queremos obtener el listado de estaciones.

Inserción remota

Que una estación se añada en nuestra aplicación.

Generación

Genera un listado estándar de estaciones de render con fines de acelerar operaciones de test.

4.7.4. Eliminar estaciones de render

En el listado de estaciones de render, seleccionaremos las estaciones a eliminar y seleccionaremos la opción de eliminarlas.

4.7.5. *Ping a estaciones*

Existe un mecanismo que recorre cíclicamente las estaciones para consultar el estado de las mismas. Esta operación se puede forzar en el listado de estaciones de render.

4.8. Gestión de formas

4.8.1. ¿Qué son las formas?

Cuando construimos un mundo sintético en 3D empleamos objetos en 3D para crear dicho mundo. Las formas son objetos en 3D. Dichas formas deben cumplir las siguientes condiciones. Una forma siempre tiene todo su volumen conectado.

4.8.2. Sintetizar formas

En el submenú de formas pulsaremos la opción de sintetizar formas. En la ventana podremos ver una representación 3D de las formas.

4.8.3. Añadir formas

Existen varias maneras de agregar formas a nuestra aplicación. Existe un mecanismo que recorre cíclicamente las estaciones para consultar el listado de formas de las mismas. Esta operación se puede forzar en el listado de estaciones de render.

Añadir las de manera manual

Simplemente, en el listado de formas, introduciremos los datos de la forma que queremos añadir y procederemos a su inserción.

Cargar las de disco

En el submenú de formas pulsaremos la opción de cargar. Se cargara el listado de formas previamente salvadas.

Importar desde otra estación

En el submenú de formas pulsaremos la opción de importar. Importaremos el listado de formas previamente exportadas.

Actualizar las desde una estación conocida

En el listado de estaciones de render seleccionaremos la estación de la que queremos obtener el listado de formas.

Inserción remota

Que una estación remota añada una forma en nuestra aplicación.

Generación

Genera un listado estándar de formas con fines de acelerar operaciones de test.

4.8.4. Eliminar formas

En el listado de formas, seleccionaremos las formas a eliminar y seleccionaremos la opción de eliminarlas.

4.8.5. Propagación

Propaga el listado de formas entre las estaciones.

4.9. Gestión del árbol *kd*

4.9.1. ¿Qué es un árbol *kd*?

Un árbol *kd*, *kd-tree* en inglés, es una estructura de ordenación espacial que tiene el fin de organizar el espacio para acelerar las búsquedas en él.

4.9.2. Generar árbol *kd*

Genera el árbol basándose en el listado de formas.

4.9.3. Ver representación del árbol *kd*

En el submenú de *kd-tree* pulsaremos la opción de representar. Desde allí podremos observar una representación del *kd-tree*.

4.10. Gestión de la simulación

4.10.1. ¿Tipos de simulación?

A la hora de ejecutar una simulación podremos optar por una simulación local o una simulación remota.

La simulación local no interactúa con la red. Sólo se usan los medios locales. Esto nos dará una medida de tiempos sin penalizaciones por la red.

La simulación distribuida existe interacción con la red. Se usan los medios locales y remotos. Esto nos dará una medida de tiempos contando penalizaciones por la red y los beneficios de paralelizar el proceso.

4.10.2. ¿Tipos de test?

En la presente aplicación se han realizado tres tipos diferentes de test:

- Búsqueda en un cubo ortogonal: En la búsqueda ortogonal se busca los elementos contenidos en un cubo ortogonal.
- Búsqueda en una esfera: En la búsqueda esférica se busca los elementos contenidos en una esfera.
- Búsqueda del más cercano: en la búsqueda del más cercano se busca el elemento más cercano a una coordenada.

Capítulo 5. EJEMPLOS DE USO

5.1. Monitoreo del listado de estaciones de render

Paso 1: Iniciamos la aplicación

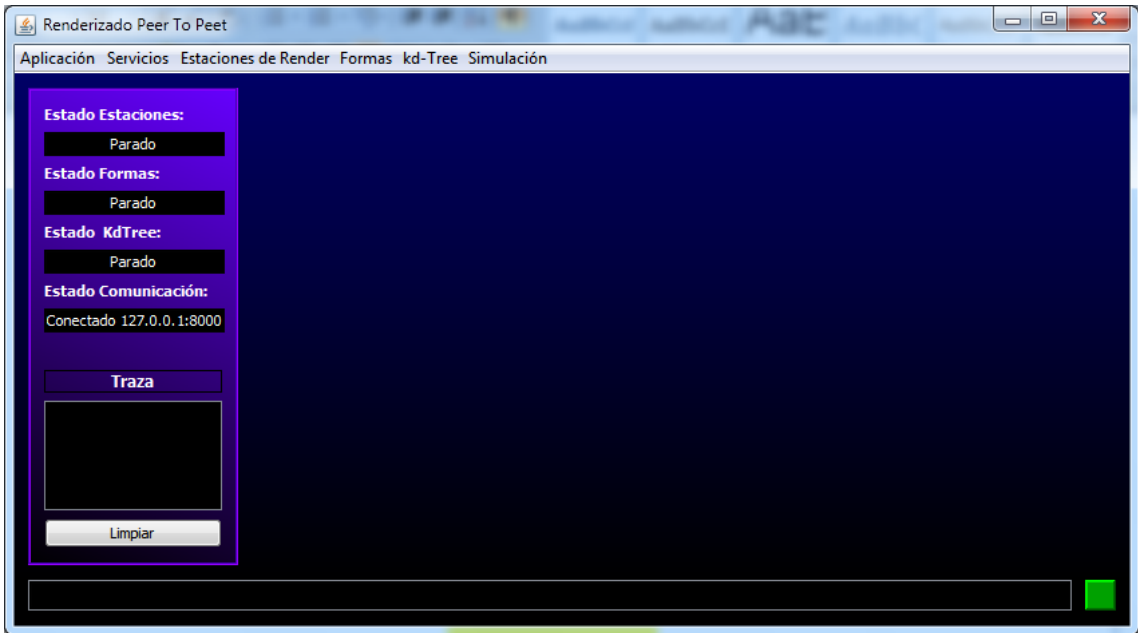


Ilustración 52 Iniciamos la aplicación

Paso 2: Abrimos en el menu de estaciones de render, la opcion de monitorizar

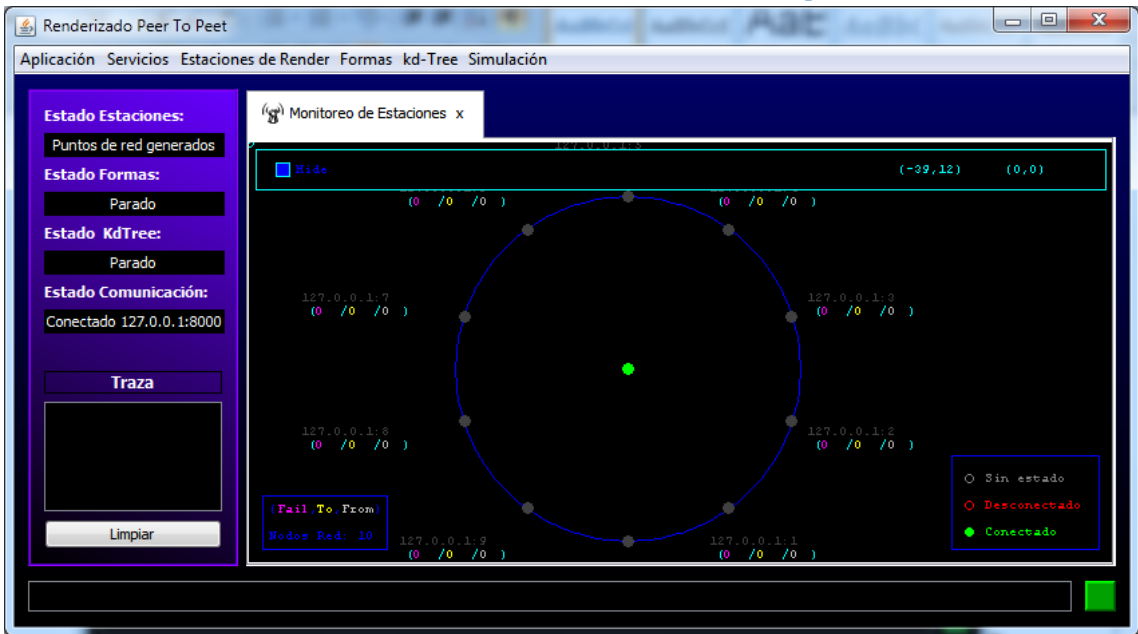


Ilustración 53 Abrimos en el menu de estaciones de render, la opcion de monitorizar

5.2. Monitoreo del listado de formas

Paso 1: Iniciamos la aplicación

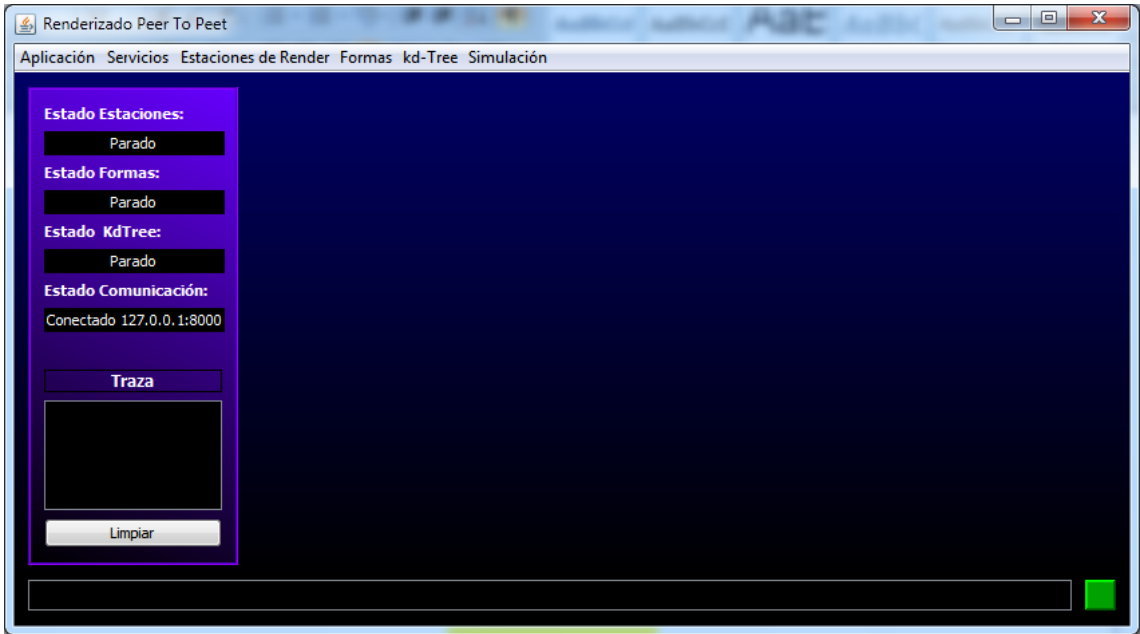


Ilustración 54 Iniciamos la aplicación

Paso 2: Abrimos en el menu de, la opcion de sintetizar escena

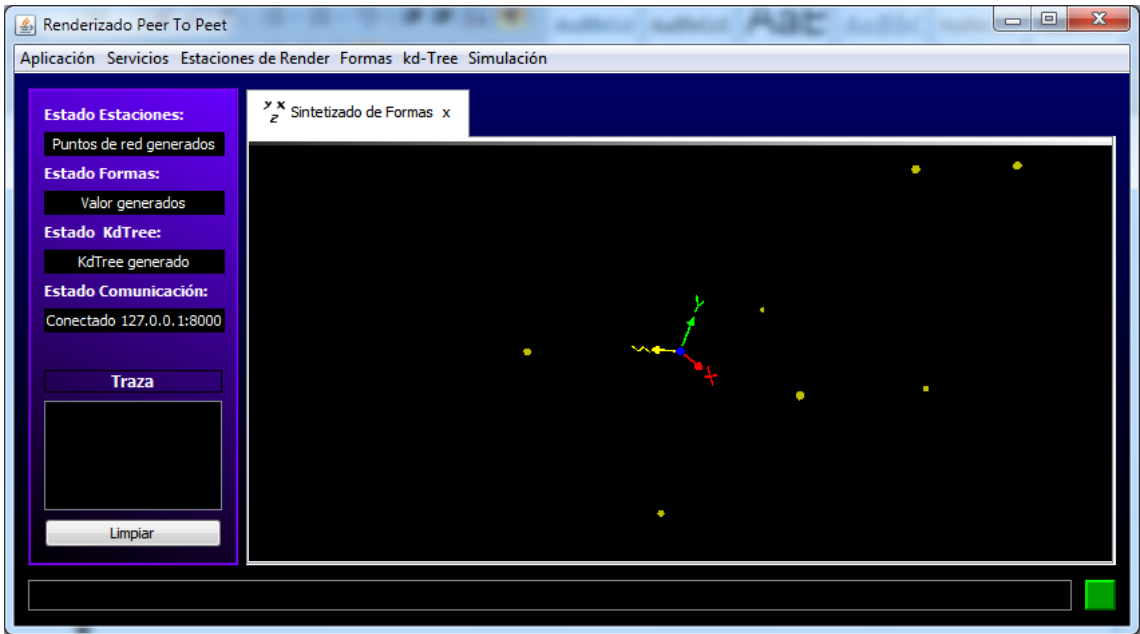


Ilustración 55 Abrimos en el menu de, la opcion de sintetizar escena

5.3. Monitoreo del árbol kd

Paso 1: Iniciamos la aplicación

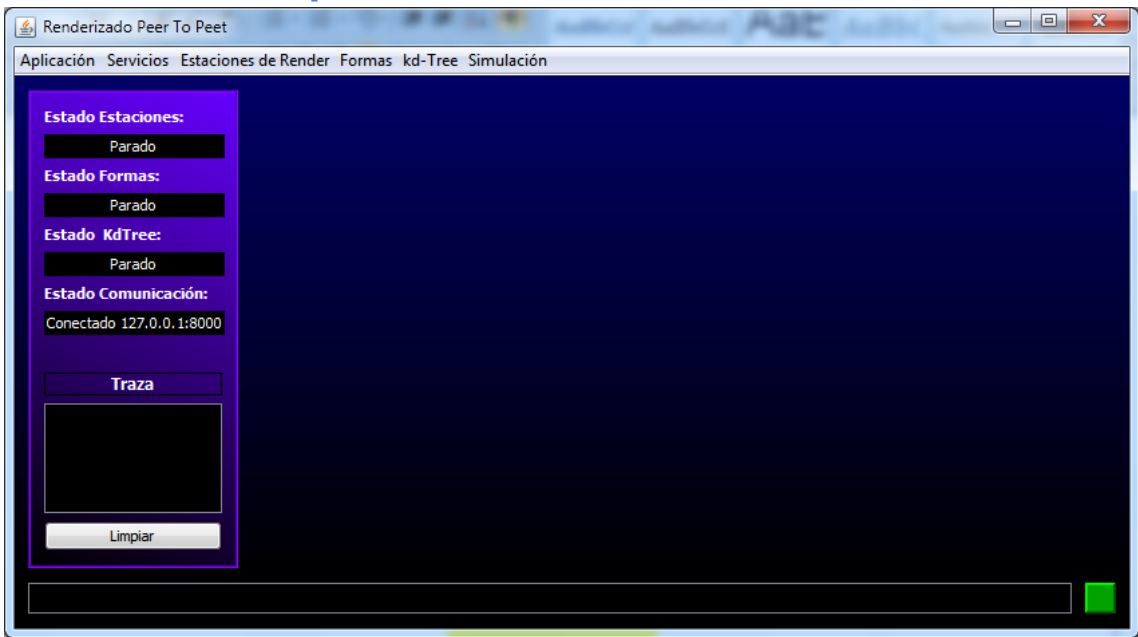


Ilustración 56 Iniciamos la aplicación

Paso 2: Abrimos en el menu del kdtree, la opcion de representar

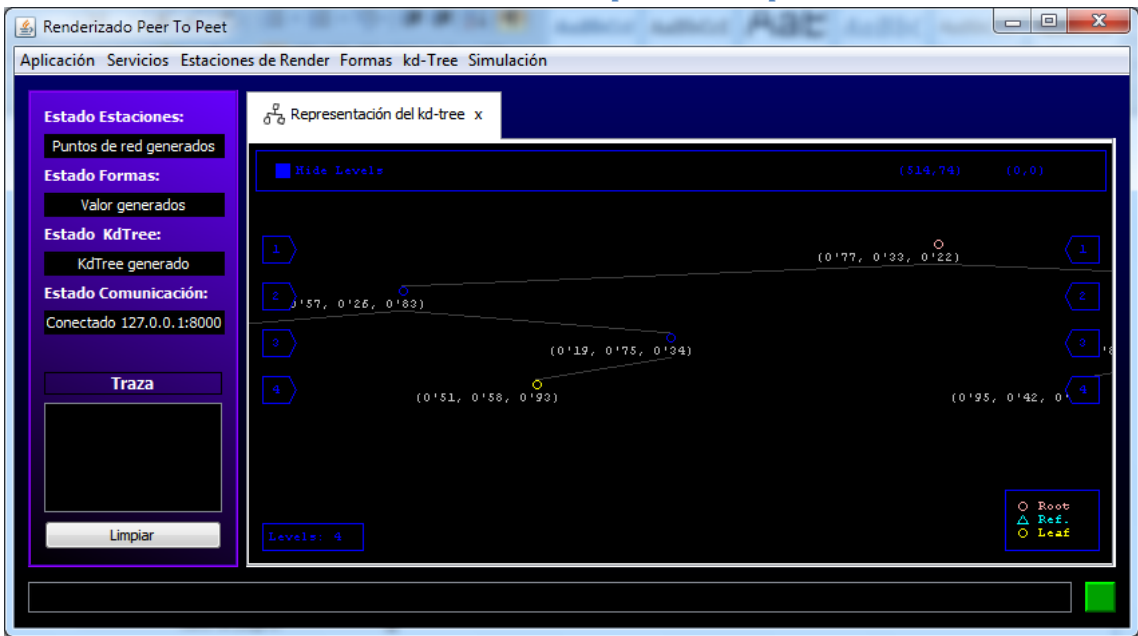


Ilustración 57 Abrimos en el menu del kdtree, la opcion de representar

5.4. Simulación local de la búsqueda en una esfera

Paso 1: Iniciamos la aplicación

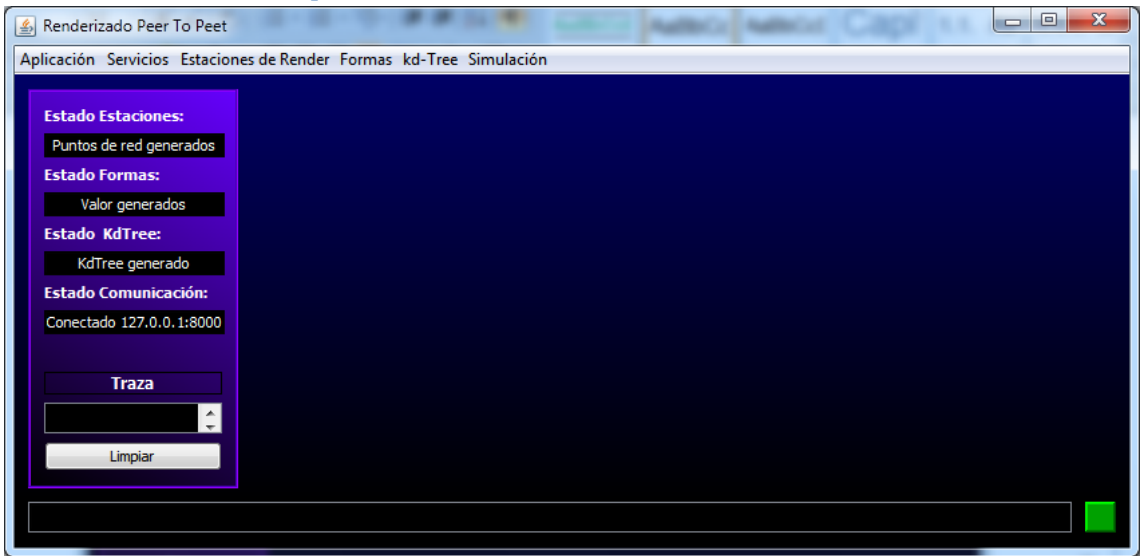


Ilustración 58 Iniciamos la aplicación

Paso 2: Abrimos en el menu del simulacion, la opcion de simulacion local

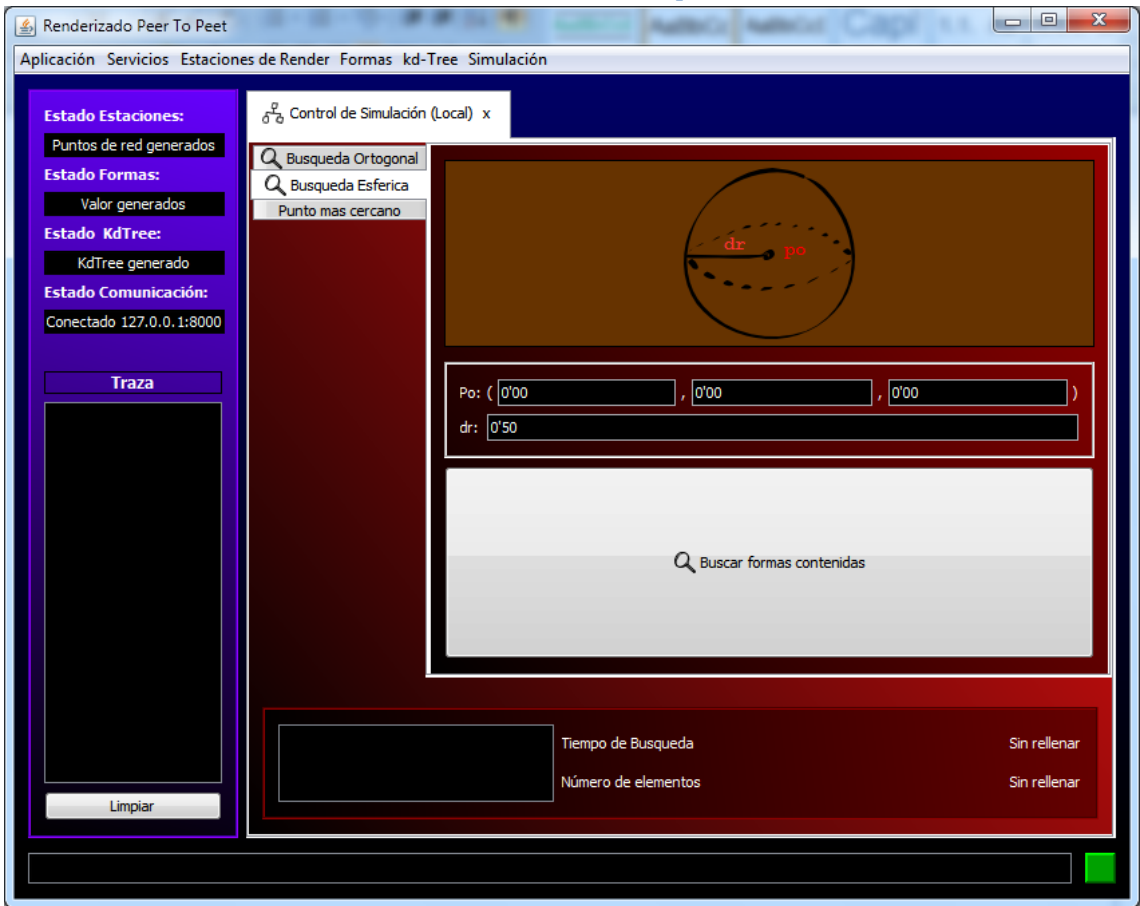
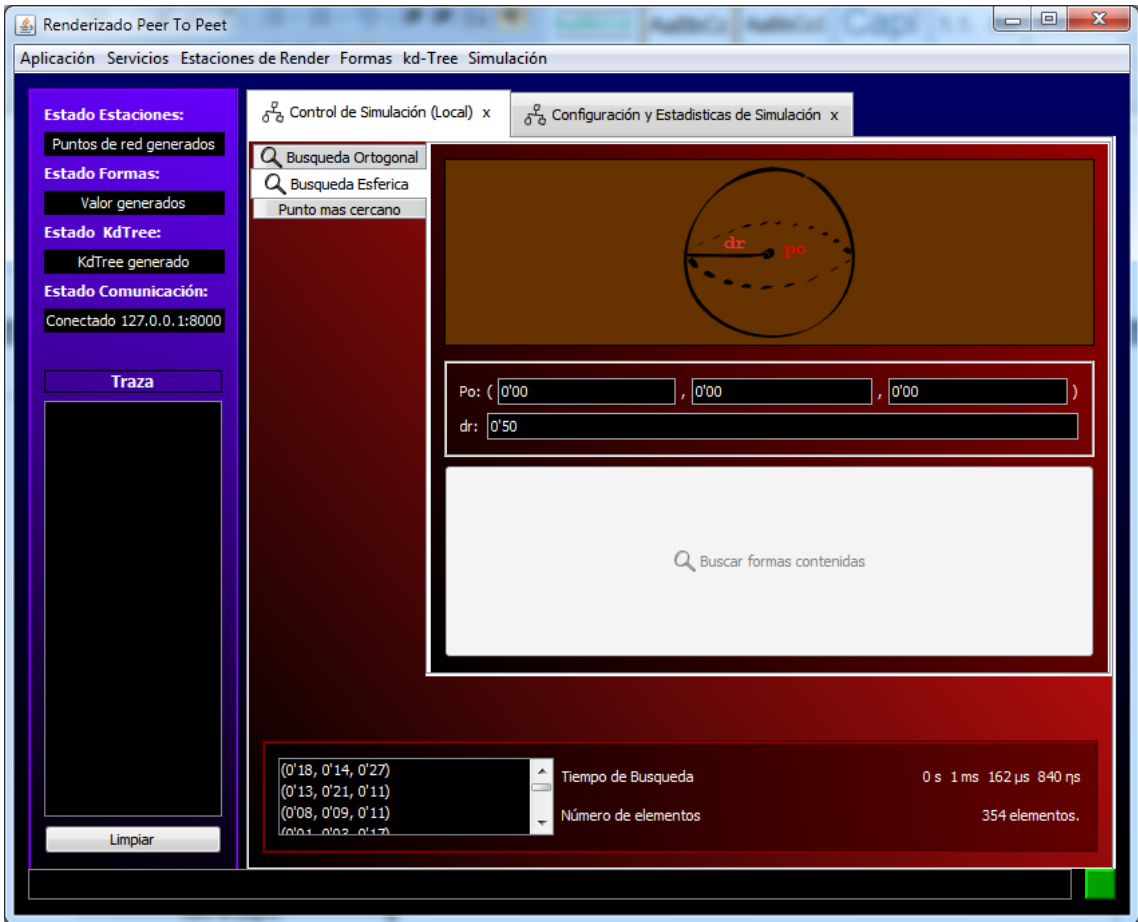


Ilustración 59 Abrimos en el menu del simulacion, la opcion de simulacion local

Paso 3: Configuramos los parametros de simulacion.

Si se han habilitado las repeticiones podemos visualizar el fichero de resultados en la carpeta del proyecto



5.5. Errores típicos

5.5.1. *Error de conexión de red*

Comprueba la correcta configuración de la conexión en tu sistema.

Comprueba los cables y distintos elementos de hardware implicados en tu red.

Comprueba que no tienes un firewall que este limitado el acceso de la aplicación.

Comprueba que el puerto de conexión no esté ya ocupado por otro servicio. De ser necesario cambia de puerto.

5.5.2. *No se pueden almacenar las formas*

Comprueba que no hay protección contra escritura

Comprueba que el disco no esté lleno o dañado

5.5.3. *No autorizados*

Comprueba los derechos de la operación correspondiente o consulta con tu administrador sobre los privilegios sobre el sistema que tienes asignados que puedan estar limitando tu uso de la aplicación.

Capítulo 6. LICENCIA DE SOFTWARE

TERMINOS Y CONDICIONES PARA COPIA, MODIFICACIÓN Y DISTRIBUCIÓN

0. Esta licencia aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta "GPL General Public License". El "Programa", utilizado en lo subsiguiente, se refiere a cualquier programa o trabajo original, y el "trabajo basado en el Programa" significa ya sea el Programa o cualquier trabajo derivado del mismo bajo la ley de derechos de autor: es decir, un trabajo que contenga el Programa o alguna porción de el, ya sea íntegra o con modificaciones y/o traducciones a otros idiomas. De aquí en adelante "traducción" estará incluida (pero no limitada a) en el término "modificación", y la persona a la que se aplique esta licencia será llamado "usted". Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance. El acto de ejecutar el programa no está restringido, y la salida de información del programa está cubierto sólo si su contenido constituye un trabajo basado en el Programa (es independiente de si fue resultado de ejecutar el programa). Si esto es cierto o no depende de la función del programa.

1. Usted puede copiar y distribuir copias fieles del código fuente del programa tal como lo recibió, en cualquier medio, siempre que proporcione de manera consciente y apropiada una nota de derechos de autor y una declaración de no garantía, además de mantener intactas todas las notas que se refieran a esta licencia y a la ausencia de garantía, y que le proporcione a las demás personas que reciban el programa una copia de esta licencia junto con el Programa. Usted puede aplicar un cargo por el acto físico de transferir una copia, y ofrecer protección de garantía por una cuota, lo cual no compromete a que el autor original del Programa responda por tal efecto.

2. Usted puede modificar su copia del Programa o de cualquier parte de él, formando así un trabajo basado en el Programa, y copiar y distribuir tales modificaciones o bien trabajar bajo los términos de la sección 1 arriba descrita, siempre que cumpla con las siguientes condiciones:

- Usted debe incluir en los archivos modificados notas declarando que modificó dichos archivos y la fecha de los cambios.
- Usted debe notificar que ese trabajo que distribuye contiene totalmente o en partes al Programa, y que debe ser licenciado como un conjunto sin cargo

alguno a cualquier otra persona que reciba sus modificaciones bajo los términos de esta Licencia.

- *Si el programa modificado lee normalmente comandos interactivamente cuando es ejecutado, usted debe presentar un aviso, cuando el programa inicie su ejecución en ese modo interactivo de la forma más ordinaria, que contenga una noticia de derechos de autor y un aviso de que no existe garantía alguna (o que sí existe si es que usted la proporciona) y que los usuarios pueden redistribuir el programa bajo esas condiciones, e informando al usuario como puede ver una copia de esta Licencia. (Excepción: si el programa en sí es interactivo pero normalmente no muestra notas, su trabajo basado en el Programa no tiene la obligación de mostrar tales notas)*

Estos requerimientos aplican al trabajo modificado como un todo. Si existen secciones identificables de tal trabajo que no son derivadas del Programa original, y pueden ser razonablemente consideradas trabajos separados e independientes como tal, entonces esta Licencia y sus términos no aplican a dichas secciones cuando usted las distribuye como trabajos separados. Pero cuando usted distribuye las mismas secciones como parte de un todo que es un trabajo basado en el Programa, la distribución del conjunto debe ser bajo los términos de esta Licencia, cuyos permisos para otras personas que obtengan el software se extienden para todo el software, así como para cada parte de él, independientemente de quién lo escribió

No es la intención de esta sección de reclamar derechos o pelear sus derechos sobre trabajos hechos enteramente por usted, en lugar de eso, la intención es ejercer el derecho de controlar la distribución de los trabajos derivados o colectivos basados en el Programa. Adicionalmente, el simple agregado de otro trabajo NO basado en el Programa al Programa en cuestión (o a un trabajo basado en el Programa) en algún medio de almacenamiento no pone el otro trabajo bajo el alcance de esta Licencia

3. *Usted puede copiar y distribuir el Programa (o un trabajo basado en él, bajo la Sección 2) en código objeto o en forma de ejecutable bajo los términos de las secciones 1 y 2 arriba descritas siempre que cumpla los siguientes requisitos:*

- *Acompañarlo con el correspondiente código fuente legible por la máquina, que debe ser distribuido bajo los términos de las secciones 1 y 2 y en un medio comúnmente utilizado para el intercambio de software, o*
- *Acompañarlo con una oferta escrita, válida por al menos 3 años y para cualquier persona, por un cargo no mayor al costo que conlleve la distribución física del código fuente correspondiente en un medio comúnmente utilizado para el intercambio de software, o*

- *Acompañarlo con la información que usted recibió sobre la oferta de distribución del código fuente correspondiente. (Esta alternativa está permitida sólo para distribución no-comercial y sólo si usted recibió el Programa en código objeto o en forma de ejecutable con tal oferta de acuerdo a la subsección b anterior*

El código fuente de un trabajo significa la forma preferida de hacer modificaciones al mismo. Para un trabajo ejecutable, un código fuente completo significa todo el código fuente de todos los módulos que contiene, mas cualquier archivo de definición de interfases, mas los programas utilizados para controlar la compilación y la instalación del ejecutable.

Sin embargo, como excepción especial, no se requiere que el código fuente distribuido incluya cualquier cosa que no sea normalmente distribuida con las componentes mayores (compilador, kernel, etc.) del sistema operativo en el cual el ejecutable corre, a menos de que una componente en particular acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace ofreciendo acceso a copiar desde un lugar designado, entonces el ofrecer acceso equivalente para copiar el código fuente desde el mismo lugar se considera distribución del código fuente, aunque las demás personas no copien el código fuente junto con el código objeto.

4. *Usted no puede copiar, modificar, sub-licenciar ni distribuir el Programa a menos que sea expresamente bajo esta Licencia, de otra forma cualquier intento de copiar, modificar, sub-licenciar o distribuir el programa es nulo, y automáticamente causará la pérdida de sus derechos bajo esta Licencia. Sin embargo, cualquier persona que haya recibido copias o derechos de usted bajo esta Licencia no verá terminadas sus Licencias ni sus derechos perdidos mientras ellas continúen cumpliendo los términos de esta Licencia.*

5. *Usted no está obligado a aceptar esta Licencia, dado que no la ha firmado. Sin embargo, nada le otorga el permiso de modificar o distribuir el Programa ni sus trabajos derivados. Estas acciones están prohibidas por la ley si usted no acepta esta Licencia. Sin embargo, modificando o distribuyendo el Programa (o cualquier trabajo basado en el Programa) indica su aceptación de esta Licencia y de todos sus términos y condiciones para copiar, distribuir o modificar el Programa y/o trabajos basados en él.*

6. *Cada vez que usted redistribuye el Programa (o cualquier trabajo basado en el Programa), la persona que lo recibe automáticamente recibe una licencia del autor original para copiar, distribuir o modificar el Programa sujeto a estos términos y condiciones. Usted no puede imponer ninguna restricción adicional a las personas que reciban el Programa sobre los derechos que en esta Licencia*

se les otorga. Usted no es responsable de forzar a terceras personas en el cumplimiento de esta Licencia.

7. Si como consecuencia de un veredicto de un juzgado o por el alegato de infringir una patente o por cualquier otra razón (no limitado solo a cuestiones de patentes) se imponen condiciones sobre usted que contradigan los términos y condiciones de esta Licencia, éstas no le excusan de los términos y condiciones aquí descritas. Si usted no puede distribuir el producto cumpliendo totalmente con las obligaciones concernientes a la resolución oficial y al mismo tiempo con las obligaciones que se describen en este contrato de Licencia, entonces no podrá distribuir más este producto. Por ejemplo, si una licencia de patente no permitirá la distribución del Programa de forma libre de regalías (sin pago de regalías) por parte de quienes lo reciban directa o indirectamente, entonces la única forma de cumplir con ambas obligaciones es renunciar a la distribución del mismo.

Si cualquier parte de esta sección resulta inválida, inaplicable o no obligatoria bajo cualquier circunstancia en particular, la tendencia de esta es a aplicarse, y la sección completa se aplicará bajo otras circunstancias.

La intención de esta sección no es la de inducirlo a infringir ninguna ley de patentes, ni tampoco infringir algún reclamo de derechos, ni discutir la validez de tales reclamos; esta sección tiene el único propósito de proteger la integridad del sistema de distribución del software libre, que está implementado por prácticas de licencia pública. Mucha gente ha hecho generosas contribuciones a la amplia gama de software distribuido bajo este sistema favoreciendo así la constante aplicación de este sistema de distribución; es decisión del autor/donador si su Programa será distribuido utilizando este u otro sistema de distribución, y la persona que recibe el software no puede obligarlo a hacer ninguna elección en particular

Esta sección pretende dejar muy claro, lo que se cree que será una consecuencia del resto de esta Licencia.

8. Si la distribución y/o el uso del Programa se restringe a algunos países ya sea por patentes, interfaces protegidas por derechos de autor, el propietario original de los derechos de autor que ubica su Programa bajo esta Licencia puede agregar una restricción geográfica de distribución explícita excluyendo los países que aplique, dando como resultado que su distribución sólo se permita en los países no excluidos. En tal caso, esta Licencia incorpora la limitación como si hubiera sido escrita en el cuerpo de esta misma Licencia.

9. La "FSF Free Software Foundation" puede publicar versiones nuevas o revisadas de la "GPL General Public License" de uno a otro momento. Estas nuevas versiones mantendrán el espíritu de la presente versión, pero pueden

diferir en la inclusión de nuevos problemas o en la manera de tocar los problemas o aspectos ya presentes.

Cada versión tendrá un número de versión que la distinga. Si el Programa especifica un número de versión para esta Licencia que aplique a él y "cualquier versión subsiguiente", usted tiene la opción de seguir los términos y condiciones de dicha versión o de cualquiera de las posteriores versiones publicadas por la "FSF". Si el programa no especifica una versión en especial de esta Licencia, usted puede elegir entre cualquiera de las versiones que han sido publicadas por la "FSF".

10. *Si usted desea incorporar partes del Programa en otros Programas de software libre cuyas condiciones de distribución sean distintas, deberá escribir al autor solicitando su autorización. Para programas de software protegidas por la "FSF Free Software Foundation", deberá escribir a la "FSF" solicitando autorización, en ocasiones hacemos excepciones. Nuestra decisión será guiada por dos metas principales:*

- *Mantener el estado de libertad de todos los derivados de nuestro software libre.*
- *Promover el uso comunitario y compartido del software en general.*

NO EXISTE GARANTÍA ALGUNA

11. *DEBIDO A QUE EL PROGRAMA SE OTORGA LIBRE DE CARGOS Y REGALIAS, NO EXISTE NINGUNA GARANTÍA PARA EL MISMO HASTA DONDE LO PERMITA LA LEY APLICABLE. A EXCEPCIÓN DE QUE SE INDIQUE OTRA COSA, LOS PROPIETARIOS DE LOS DERECHOS DE AUTOR PROPORCIONAN EL PROGRAMA "COMO ES" SIN NINGUNA GARANTÍA DE NINGÚN TIPO, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO NO LIMITADA A, LAS GARANTÍAS QUE IMPLICA EL MERCADEO Y EJERCICIO DE UN PROPÓSITO EN PARTICULAR. CUALQUIER RIESGO DEBIDO A LA CALIDAD Y DESEMPEÑO DEL PROGRAMA ES TOMADO COMPLETAMENTE POR USTED. SI EL SOFTWARE MUESTRA ALGÚN DEFECTO, USTED CUBRIRÁ LOS COSTOS DE CUALQUIER SERVICIO, REPARACIÓN O CORRECCIÓN DE SUS EQUIPOS Y/O SOFTWARE QUE REQUIERA.*

12. *EN NINGÚN CASO NI BAJO NINGUNA CIRCUNSTANCIA EXCEPTO BAJO SOLICITUD DE LA LEY O DE COMÚN ACUERDO POR ESCRITO, NINGUN PROPIETARIO DE LOS DERECHOS DE AUTOR NI TERCERAS PERSONAS QUE PUDIERAN MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA COMO SE PERMITE ARRIBA, SERAN RESPONSABLES DE*

LOS DAÑOS CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA, SIN IMPORTAR SI SON DAÑOS GENERALES, ESPECIALES, INCIDENTALES O CONSECUENTES CORRESPONDIENTES AL USO O IMPOSIBILIDAD DE USAR EL PROGRAMA (INCLUYENDO PERO NO LIMITADO A LA PÉRDIDA DE INFORMACIÓN O DETERIORO DE LA MISMA AFECTÁNDOLO A USTED, A TERCERAS PERSONAS QUE SEA POR FALLAS EN LA OPERACIÓN DEL PROGRAMA O SU INTERACCIÓN CON OTROS PROGRAMAS) INCLUSIVE SI TAL PROPIETARIO U OTRAS PERSONAS HAYAN SIDO NOTIFICADAS DE TALES FALLAS Y DE LA POSIBILIDAD DE TALES DAÑOS.

FIN DE TÉRMINOS Y CONDICIONES

TABLAS E ILUSTRACIONES

Índice de Tablas

Tabla 1 Modo de distribución	1
Tabla 2 Tamaño de la distribución	1
Tabla 3 Tamaño de la distribución sin comprimir.....	1
Tabla 4 Espacio mínimo en disco para la distribución	1
Tabla 5 Listado de archivos de la distribución.	8
Tabla 6 Secciones de la aplicación.....	12
Tabla 7 Barra de menú de aplicación	12
Tabla 8 Paneles de monitorización.....	12
Tabla 9 Consola de comandos de la aplicación	13
Tabla 10 Sección principal	13
Tabla 11 Barra de menús de la aplicación.....	14
Tabla 12 Submenú de aplicación	14
Tabla 13 Submenú de servicios.....	14
Tabla 14 Submenú de estaciones de render	14
Tabla 15 Submenú de formas.....	14
Tabla 16 Submenú de kdtree.....	14
Tabla 17 Submenú de simulación	14
Tabla 18 Acción salir.....	15
Tabla 19 Acción Iniciar.....	15
Tabla 20 Acción Parar.....	15
Tabla 21 Acción Configurar	15
Tabla 22 Acción monitorizar.....	16

Tabla 23 Acción guardar.....	16
Tabla 24 Acción cargar	16
Tabla 25 Acción exportar.....	16
Tabla 26 Acción importar	16
Tabla 27 Acción listar	16
Tabla 28 Acción generar.....	16
Tabla 29 Acción estadísticas y preferencias	17
Tabla 30 Acción del menú Formas [Sintetizar escena]	17
Tabla 31 Acción del menú Formas [Guardar]	17
Tabla 32 Acción del menú Formas [Cargar].....	17
Tabla 33 Acción del menú Formas [Exportar].....	17
Tabla 34 Acción del menú Formas [Importar]	17
Tabla 35 Acción del menú Formas [Listar].....	17
Tabla 36 Acción del menú Formas [Generar]	18
Tabla 37 Acción del menú Formas [Propagar]	18
Tabla 38 Acción del menú Formas [Estadísticas].....	18
Tabla 39 Acción del menú kd-tree [Representar]	18
Tabla 40 Acción del menú kd-tree [Guardar]	18
Tabla 41 Acción del menú kd-tree [Cargar].....	18
Tabla 42 Acción del menú kd-tree [Exportar]	18
Tabla 43 Acción del menú kd-tree [Importar].....	18
Tabla 44 Acción del menú kd-tree [Generar].....	19
Tabla 45 Acción del menú kd-tree [Estadísticas y preferencias].....	19
Tabla 46 Acción del menú Simulación [Sintetizar Simulación]	19
Tabla 47 Acción del menú Simulación [Simulación local]	19
Tabla 48 Acción del menú Simulación [Simulación distribuida]	19
Tabla 49 Acción del menú Simulación [Estadísticas y preferencias].....	19
Tabla 50 Monitor del sistema	20
Tabla 51 Monitor de Estado de las Estaciones de Render	20
Tabla 52 Monitor de Estado de las Formas	20
Tabla 53 Monitor de Estado del kd-tree	20
Tabla 54 Monitor de Estado de las Comunicaciones de red	20
Tabla 55 Monitor informativo general de la aplicación	21
Tabla 56 Acción de limpieza de monitor informativo general [Limpiar].....	21
Tabla 57 Consola de comandos	22
Tabla 58 Entrada de comandos.....	22
Tabla 59 Control de estado de los comandos.....	22
Tabla 60 Sección de configuración de servicios	23
Tabla 61 Datos de conexión de nuestra estación de render.....	23
Tabla 62 Estado de los servicios	23
Tabla 63 Automatizaciones de los servicios	23
Tabla 64 Sección de monitoreo de Estaciones de Render.....	24
Tabla 65 Barra de opciones del monitor.....	24
Tabla 66 Estado de los servicios	24
Tabla 67 Leyenda de interacción	24
Tabla 68 Leyenda de estados	24
Tabla 69 Sección de monitoreo de Estaciones de Render.....	25
Tabla 70 Listado de Estaciones de Render.....	25
Tabla 71 Acción de actualizar estaciones de render [Actualizar Estaciones]	25

Tabla 72 Acción de actualizar formas [Actualizar formas]	25
Tabla 73 Acción de realizar ping [Realizar Pings]	25
Tabla 74 Acción Insertar estación de render [Icono de añadir]	25
Tabla 75 Acción generar estaciones de render [Icono de generar]	26
Tabla 76 Acción de refrescar listado [Icono de refrescar]	26
Tabla 77 Acción de eliminar estaciones de render [Icono de eliminar]	26
Tabla 78 Sección de estadísticas de Estaciones de Render	27
Tabla 79 Estadísticas de las estaciones de render	27
Tabla 80 Sección de sintetizado de formas	28
Tabla 81 Controles del escenario	28
Tabla 82 Sección de listado de formas	29
Tabla 83 Listado de formas	29
Tabla 84 Acción de eliminar [Eliminar]	29
Tabla 85 Acción refrescar [Refrescar]	29
Tabla 86 Acción añadir punto [Icono añadir]	30
Tabla 87 Acción generar puntos [Icono generar]	30
Tabla 88 Acción añadir faceta [Icono añadir]	31
Tabla 89 Acción generar facetas [Icono generar]	31
Tabla 90 Acción añadir esfera [Icono añadir]	32
Tabla 91 Acción generar esferas [Icono generar]	32
Tabla 92 Acción añadir cubo [Icono añadir]	33
Tabla 93 Acción generar cubos [Icono generar]	33
Tabla 94 Sección de estadísticas de formas	34
Tabla 95 Estadísticas	34
Tabla 96 Sección de representación del árbol kd	35
Tabla 97 Barra de opciones de la representación	35
Tabla 98 Leyenda de niveles	35
Tabla 99 Leyenda de interacción	35
Tabla 100 Acción de selección de un nodo [Nodos dibujados]	36
Tabla 101 Visualización de una simulación	36
Tabla 102 Sección de estadísticas y configuración del árbol kd	37
Tabla 103 Estadísticas locales	37
Tabla 104 Configuración local	37
Tabla 105 Sección de sintetizado de simulación	38
Tabla 106 Descripción del modo de presentación	38
Tabla 107 Controles del escenario	38
Tabla 108 Sección de control de simulación local	39
Tabla 109 Resultado de elementos encontrados al realizar el test	39
Tabla 110 Estadísticas de la simulación	39
Tabla 111 Acción selección del test [Pestañas laterales]	39
Tabla 112 Datos de Simulación	40
Tabla 113 Acción inicio del test de búsqueda [Buscar formas contenidas]	40
Tabla 114 Datos de Simulación	41
Tabla 115 Acción inicio del test de búsqueda [Buscar formas contenidas]	41
Tabla 116 Datos de Simulación	42
Tabla 117 Acción inicio del test de búsqueda [Buscar formas contenidas]	42
Tabla 118 Sección de control de simulación distribuida	43
Tabla 119 Resultado de elementos encontrados al realizar el test	43
Tabla 120 Estadísticas de la simulación	43

Tabla 121 Acción selección del test [Pestañas laterales].....43

Tabla 122 Datos de Simulación44

Tabla 123 Acción inicio del test de búsqueda [Buscar formas contenidas]44

Tabla 124 Datos de Simulación45

Tabla 125 Acción inicio del test de búsqueda [Buscar formas contenidas]45

Tabla 126 Datos de Simulación46

Tabla 127 Acción inicio del test de búsqueda [Buscar formas contenidas]46

Tabla 128 Sección de configuración y estadísticas de la simulación47

Tabla 129 Estadísticas distribuidas.....47

Tabla 130 Modo de propagación del árbol [Icono de Propagar todo]47

Tabla 131 Modo de propagación del árbol [Icono de Propagar ramas].....47

Tabla 132 Modo de propagación del árbol [Icono de Propagar niveles].....48

Tabla 133 Configurar propagación.....48

Tabla 134 Configuración de repeticiones48

Índice de Ilustraciones

Ilustración 1 Estructura de directorios al ser extraída la distribución	2
Ilustración 2 Subdirectorio LIB.....	2
Ilustración 3 Página web de PeaZip.....	4
Ilustración 4 Página de descarga de MV Java	5
Ilustración 5 Página de descarga de Java3D	6
Ilustración 6 Selección de la plataforma Java3D	6
Ilustración 7 Explorador de Archivos, unidad de CD.....	7
Ilustración 8 Explorador de archivos, distribución.....	7
Ilustración 9 Explorador de archivos, dentro del directorio DISTRIBUCION.....	8
Ilustración 10 Operación de descompresión de la aplicación.....	8
Ilustración 11 Operación de descompresión, destino.....	9
Ilustración 12 Secciones de la aplicación.....	12
Ilustración 13 Barra de menú	14
Ilustración 14 Submenú de aplicación	15
Ilustración 15 Submenú de servicios.....	15
Ilustración 16 Submenú estaciones	16
Ilustración 17 Submenú formas.....	17
Ilustración 18 Submenú kd-tree.....	18
Ilustración 19 Submenú Simulación.....	19
Ilustración 20 Monitor del sistema	20
Ilustración 21 Consola de comandos	22
Ilustración 22 Sección de configuración de servicios	23
Ilustración 23 Sección de monitoreo de Estaciones de Render.....	24
Ilustración 24 Sección de monitoreo de Estaciones de Render.....	25
Ilustración 25 Sección de estadísticas de Estaciones de Render.....	27
Ilustración 26 Sección de sintetizado de formas.....	28
Ilustración 27 Sección de listado de formas	29
Ilustración 28 Añadir punto	30
Ilustración 29 Añadir faceta	31
Ilustración 30 Añadir esfera.....	32
Ilustración 31 Añadir cubo.....	33
Ilustración 32 Sección de estadísticas de formas.....	34
Ilustración 33 Sección de representación del árbol kd.....	35
Ilustración 34 Selección de un nodo	36
Ilustración 35 Sección de estadísticas y configuración del árbol kd	37
Ilustración 36 Sección de sintetizado de simulación.....	38
Ilustración 37 Sección de control de simulación local.....	39
Ilustración 38 Datos de Simulación Ortogonal	40
Ilustración 39 Datos de Simulación Esférica	41
Ilustración 40 Datos de Simulación más cercano	42
Ilustración 41 Sección de control de simulación distribuida	43
Ilustración 42 Datos de Simulación Ortogonal	44
Ilustración 43 Datos de Simulación Esférica	45
Ilustración 44 Datos de Simulación más cercano	46
Ilustración 45 Sección de configuración y estadísticas de la simulación	47
Ilustración 46 Iniciando la aplicación	49
Ilustración 47 Pantalla de inicio de la aplicación.....	49

<i>Ilustración 48 Menú de servicios</i>	<i>50</i>
<i>Ilustración 49 Sección de configuración de servicios</i>	<i>51</i>
<i>Ilustración 50 Menú de servicios</i>	<i>52</i>
<i>Ilustración 51 Sección de configuración de servicios</i>	<i>52</i>
<i>Ilustración 52 Iniciamos la aplicación.....</i>	<i>59</i>
<i>Ilustración 53 Abrimos en el menu de estaciones de render, la opcion de monitorizar</i>	<i>59</i>
<i>Ilustración 54 Iniciamos la aplicación.....</i>	<i>60</i>
<i>Ilustración 55 Abrimos en el menu de, la opcion de sintetizar escena</i>	<i>60</i>
<i>Ilustración 56 Iniciamos la aplicación.....</i>	<i>61</i>
<i>Ilustración 57 Abrimos en el menu del kdtree, la opcion de representar</i>	<i>61</i>
<i>Ilustración 58 Iniciamos la aplicación.....</i>	<i>62</i>
<i>Ilustración 59 Abrimos en el menu del simulacion, la opcion de simulacion local</i>	<i>62</i>

