



**Universidad**  
Zaragoza



Departamento de  
Informática e Ingeniería  
de Sistemas  
**Universidad Zaragoza**

Proyecto Final de Carrera

Ingeniería Informática

Curso 2010/2011

# **Memoria**

**ESTRUCTURA DE ACELERACIÓN PARA RENDER 3D**

**DISTRIBUIDA MEDIANTE COMUNICACIONES PEER TO PEER**

## **1/3**

**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*

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

Capítulo 1. Introducción .....	1
1.1. Preámbulo.....	1
1.2. Motivación.....	1
1.3. Objetivos .....	1
1.4. Estado del Arte.....	2
1.5. Estructura de la Memoria .....	3
Capítulo 2. Planificación .....	5
2.1. Preámbulo.....	5
2.2. Ciclo de vida.....	5
2.3. Análisis de Riesgos.....	6
2.3.1. Alta complejidad en la implantación y comprensión .....	6
2.3.2. Lapsos de inactividad del proyecto.....	6
2.3.3. Incertidumbre de no poder cumplir con la funcionalidad adecuada	6
2.4. Conclusiones.....	6
Capítulo 3. Contexto Tecnológico .....	7

3.1. Visión Global.....	7
3.2. Software y motores de Render.....	7
3.3. Limitaciones de los sistemas de render actuales.....	8
3.3.1. Granjas de render no especializadas: .....	8
3.3.2. Granjas de render profesionales:.....	8
3.4. Sistemas distribuidos de datos.....	8
3.4.1. Bases de datos Distribuidas .....	9
3.4.2. Tablas hash Distribuidas .....	10
3.4.3. Sistemas de índice centralizados .....	10
3.4.4. Sistemas personalizados.....	10
3.5. Estado de las técnica .....	10
3.5.1. Los algoritmos .....	10
3.5.2. Las redes de pares .....	11
Capítulo 4. Descripción del Sistema.....	13
4.1. Preámbulo.....	13
4.2. Diagrama del sistema .....	13
4.3. Estructura de una estación de render .....	14
4.3.1. Mantenimiento de la red .....	14
4.3.2. Modelo vista controlador.....	15
4.3.3. La vista .....	16
4.3.4. Características deseables para el interfaz de usuario .....	16
4.3.5. El controlador.....	17
4.3.6. Características deseables para los módulos del controlador.....	17
4.3.7. Funcionalidades del controlador.....	18
4.3.8. El modelo.....	23
Capítulo 5. Solución propuesta .....	27
5.1. Preámbulo.....	27
5.2. Sistema de comunicación .....	27
5.2.1. Análisis de los sistemas de comunicación mediante Servicios Web	27
5.2.2. Análisis de los sistemas de comunicación mediante Sockets .....	27
5.2.3. Análisis de los sistemas de comunicación mediante RPC.....	27
5.2.4. Análisis de los sistemas de comunicación mediante RMI.....	28
5.3. Interfaz de Usuario.....	28

5.3.1.	Línea de comandos .....	28
5.3.2.	Entorno Web.....	28
5.3.3.	Sistema empotrado.....	28
5.3.4.	Entorno de Escritorio .....	28
5.4.	Formato de almacenamiento de los datos .....	29
5.4.1.	Análisis del uso de una Base de datos .....	29
5.4.2.	Análisis del uso de XML .....	29
5.4.3.	Análisis del uso de Objetos con serialización .....	29
5.4.4.	Análisis del uso de Objetos con serialización personalizada .....	29
5.5.	Lenguajes de programación evaluados .....	30
5.6.	La gestión del proyecto .....	31
5.7.	Modelado del modelo.....	32
5.7.1.	Modelado del modelo de las estaciones de render.....	33
5.7.2.	Modelado del modelo del listado de formas .....	34
5.7.3.	Modelado del Árbol kd .....	35
5.8.	Modelado del Controlador .....	36
5.8.1.	Modelado del módulo del controlador de servicios.....	37
5.8.2.	Modelado del módulo del controlador del listado de estaciones de render	38
5.8.3.	Modelado del módulo del controlador de formas .....	38
5.8.4.	Modelado del módulo del controlador de árbol kd .....	39
5.8.5.	Modelado del módulo del controlador de simulación .....	39
5.8.6.	Modelado del módulo del controlador de monitorización.....	40
5.8.7.	Modelado del módulo del controlador de tareas .....	40
5.9.	Modelado de la Vista.....	41
Capítulo 6.	Estudio de los Resultados.....	43
6.1.	Preámbulo.....	43
6.2.	Objeto de estudio .....	43
6.3.	Tipos de experimentos llevados a cabo .....	43
6.4.	Procedimiento llevado en los experimentos .....	43
6.5.	Modo de generación de las muestras .....	44
6.6.	Entorno de pruebas.....	45
6.7.	Configuraciones de las pruebas.....	46

6.7.1. Estación A sin uso de red .....	46
6.7.2. Estación A con uso de protocolos de red en configuración local ..	46
6.7.3. Estación A con uso de protocolos de red en configuración local y la estación B con uso de protocolos de red en configuración remota.....	47
6.8. Resultados de los experimentos .....	48
6.8.1. Cálculo del tiempo base .....	48
Capítulo 7. Conclusiones .....	63
7.1. Preámbulo.....	63
7.2. Sobre la Planificación.....	63
7.3. Sobre la solución.....	63
7.4. Trabajo Futuro.....	65
7.5. Aplicación del PFC al mundo real .....	65
7.6. Valoración personal y conclusión final .....	65
Capítulo 8. Bibliografía .....	67
8.1. Lecturas .....	67
8.2. Otros enlaces de interés .....	68

# Capítulo 1. INTRODUCCIÓN

## 1.1. Preámbulo

En este capítulo se va a hacer una breve introducción de las motivaciones, los objetivos y el estado actual de las tecnologías implicadas en el proyecto. Además de dar una breve guía para navegar por la memoria del proyecto.

## 1.2. Motivación

Cuando inicie este proyecto las motivaciones eran bien claras. El usarlo como trampolín de aprendizaje de ciertos ámbitos tecnológicos que despertaban mi curiosidad. Sobre todo quería implementar mi propio sistema de compartición de datos y calculo al puro estilo P2P. Para este cometido, decidí emplear los algoritmos del intrincado mundo de de las imágenes sintéticas en 3D.

Según fue avanzando el proyecto, se despertó en mí un afán por optimizar los algoritmos, para ello tuve que explorar las posibilidades de las tecnologías del lenguaje elegido. Mis motivaciones fueron ampliándose en estos ámbitos.

## 1.3. Objetivos

El objetivo del proyecto, es la implementación de una estructura de aceleración de trazado de rayos distribuida en una red local.

Basada en indexación espacial mediante un knn-tree (kd-tree), el cual, indexa las facetas de la escena en el espacio.

La base de datos de facetas, estará distribuida en los nodos de la red, para permitir manejar escenas que superen la memoria individual de los mismos.

Cada nodo mantendrá un chache local dinámica de facetas, que se actualizara por medio de la red, consultando a los otros nodos en función de las necesidades locales.

El intercambio de facetas, se hará mediante un protocolo distribuido P2P, para evitar congestiones en la red.

Las dificultades más importantes son:

- Obtener un buen balance de carga de red.
- Evitar la saturación de la memoria en los nodos.

La implementación, se realizara en Java de Sun Microsystems, mediante un modelo MVC (Modelo-Vista-Controlador), que favorezca la modularidad del software basando las comunicaciones en RMI.

Los objetivos fundamentales son, la creación de la estructura de aceleración y la evaluación de prestaciones de la misma mediante una serie de experimentos.

*Como no podía ser de otro modo, mis objetivos fueron variando. Dando un peso cada vez mayor al uso de tecnologías de programación.*

*Como colofón final he ampliado el enfoque de los algoritmos estudiados, realizando un estudio de los mismos para el tratamiento de volúmenes, vislumbrando todo un mundo de aplicaciones.*

#### 1.4. Estado del Arte

El render no es hoy en día un misterio, existen soluciones reconocidas que dan solución a esta tarea. De hecho uno puede comprar una solución que implica software y hardware optimizados para dicho fin. El alquiler de estos sistemas también es posible, tan solo hay que realizar una búsqueda en Google para ver toda un oferta de servicios. También se puede recurrir a soluciones más caseras que hacen uso de una infraestructura no especializada para dicho cometido, repartiendo el trabajo en nuestros equipos de trabajo habituales cuando estos están inactivos.

En cuanto a la investigación de las técnicas de render, solo hay que investigar un poco para darse cuenta que no dejan de aparecer publicaciones que mejoran o realizan un efecto novedoso.

Aun así se puede consensuar que con el paso del tiempo unos algoritmos se han hecho más populares al observarse que tienen un mejor rendimiento o dan unos buenos resultados, en la mayoría de las situaciones. Estos algoritmos tienen en común que se realizan infinidad de accesos a los puntos del espacio a procesar. Los más populares de ellos hacen uso de estructuras llamadas árboles kd. Dichos árboles tienen el objetivo de dividir el espacio de puntos con el objetivo de hacer a estos accesibles para el procesado y con ello usar un tiempo mínimo en realizar los cálculos.

*Para más información ver el Capítulo 3 del presente documento.*

## 1.5. Estructura de la Memoria

El presente documento sigue una estructura de capítulos descompuesta de la siguiente manera:

Capítulo	Título	Descripción
<b>Capítulo 1</b>	Introducción	Pretende mostrar la motivación de este documento, así como resumir brevemente el estado actual de la computación grafica distribuida.
<b>Capítulo 2</b>	Planificación	Muestra los esfuerzos necesarios para la consecución de este proyecto.
<b>Capítulo 3</b>	Contexto tecnológico	Explica las tecnologías usadas, además de mostrar el proceso de aprendizaje seguido en ellas.
<b>Capítulo 4</b>	Descripción del sistema	Detalla los principales puntos a tener en cuenta del sistema, haciendo un análisis de las decisiones tomadas, los posibles riesgos y carencias.
<b>Capítulo 5</b>	Solución propuesta	Describe la implementación realizada.
<b>Capítulo 6</b>	Estudio de resultados	Un análisis de los resultados de la experimentación.
<b>Capítulo 7</b>	Conclusiones	Realiza un análisis de los resultados obtenidos en el ámbito global del proyecto, así como futuras ampliaciones.
<b>Capítulo 8</b>	Bibliografía	Presenta los distintos medios a los que se hace referencia en este documento de una manera detallada.

Tabla 1 Estructura de la Memoria





# Capítulo 2. PLANIFICACIÓN

## 2.1. Preámbulo

En este capítulo se detalla la planificación llevada en el presente proyecto, las decisiones tomadas en dicha planificación, las rectificaciones, los riesgos y desviaciones observadas, así como las conclusiones de la misma.

## 2.2. Ciclo de vida

En el desarrollo del proyecto se decidió el uso de un ciclo de vida en espiral. Para ello se planificó realizar tres ciclos completos con el fin de poder adaptarnos a los posibles contratiempos que pudiesen surgir.

Las fases de cada ciclo elegidas fueron las siguientes:

Fase	Descripción
Determinar objetivos	En esta primera fase plantearemos las metas a concluir en el ciclo.
Análisis de Riesgos	En esta segunda fase, evaluaremos los contratiempos que pueden obstaculizar las metas marcadas.
Desarrollar y probar	Llevaremos a cabo las tareas necesarias para realizar las metas planteadas.
Planificación	Evaluaremos el trabajo realizado en las fases de este ciclo.

Tabla 2 Fases elegidas en el ciclo de vida

Las primeras estimaciones nos dejaron las siguientes estimaciones:

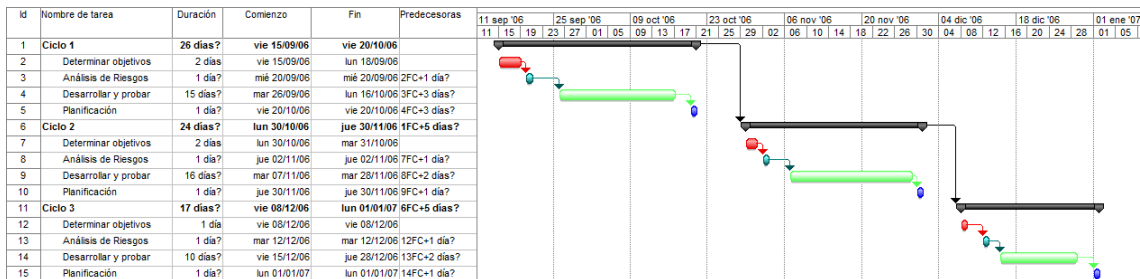


Ilustración 1 Estimación inicial del proyecto

Como no podría ser de otro modo los tiempos estimados en la planificación inicial se vieron alterados totalmente, sobrepasándolo los tiempos planificados inicialmente.

Debido a condiciones laborales y personales el tiempo inicialmente planificado para las distintas tareas se alargó más de lo estimado. A ello hay que añadir que el primer prototipo fue un completo desastre.

Finalmente según se fueron sucediendo los ciclos, estos fueron dando una proyección más ajustada a la realidad.

### 2.3. Análisis de Riesgos

En la realización del presente proyecto se pueden agrupar los riesgos evaluados durante las distintas etapas en estas tres temáticas

---

#### Temáticas de Riesgos

Alta complejidad en la implantación y comprensión.

Lapsos de inactividad del proyecto.

Incertidumbre de no poder cumplir con la funcionalidad adecuada.

---

Tabla 3 Temáticas de riesgos

#### 2.3.1. Alta complejidad en la implantación y comprensión

Ciertos algoritmos del sistema requieren un estudio altamente costoso en tiempo y complejidad. Además en la elaboración de los mismos se puede incurrir en fallos difíciles de detectar.

#### 2.3.2. Lapsos de inactividad del proyecto

Debido a condiciones laborales y personales la dedicación total se hace imposible. Existe una clara posibilidad de incurrir en grandes periodos sin actividad. El reanudar después de los periodos de inactividad puede ser complejo y lento.

#### 2.3.3. Incertidumbre de no poder cumplir con la funcionalidad adecuada

Hubo una gran incertidumbre de si se podría realizar una implementación adecuada y óptima.

### 2.4. Conclusiones

El ciclo de vida en espiral permitió realizar un mayor seguimiento del mismo permitiendo reevaluar el estado del proyecto dando una medida de progreso del mismo más real, ya que debido a las complejidades encontradas se hacía muy difícil una planificación más convencional.

Como una mejora a tener en cuenta en futuros proyectos se hace patente el tener más en cuenta los tiempos de inactividad del proyecto como un riesgo mayor del estimado en primer momento, dificultando la viabilidad del proyecto.

*Para más detalles ver los anexos – Planificación*

## Capítulo 3. CONTEXTO TECNOLÓGICO

### 3.1. Visión Global

A la hora de computar grandes cantidades de nubes de puntos, estos se reparten en una estructura de árbol kd. Esta estructura tiene como objetivo el ordenar los puntos del espacio para una óptima búsqueda en él.

Cuando nuestra nube de puntos tiene unas dimensiones aceptables estas pueden ser procesadas por un único computador. Aun así, la cantidad de datos a procesar hace de este proceso una tarea costosa.

Cuando nuestra nube de puntos alberga una ingente cantidad de elementos, la tarea de procesarlo e incluso almacenarlos no puede ser llevada por una sola máquina, debido a que los recursos de la misma son limitados. Con este hecho se hace patente la necesidad de compartir el procesado y el almacenamiento de dicha nube.

Lógicamente la distribución de los elementos nos resuelve esta limitación, pero a su vez nos implica una serie de retos a superar.

### 3.2. Software y motores de Render

Actualmente existen reconocidas aplicaciones de modelado 3D y render populares, tales como **3ds MAX** y **V-Ray**, que procesan en forma distribuida de manera sencilla y transparente para el usuario. Otro motor de render famoso es **Mental Ray**.

La última generación de motores de render, además de distribuir el proceso final en varios equipos o nodos de render, permite utilizar esta misma potencia de procesamiento para hacer el render en pantalla, aplicando cambios en tiempo real. **V-Ray RT** es precisamente uno de estos tipos de software. Estas modificaciones sobre luces, cámaras, y posiciones de objetos, evita la necesidad de realizar múltiples renderizados previos de estudio, lo redundando en un ahorro muy significativo de tiempo.

### 3.3. Limitaciones de los sistemas de render actuales

#### 3.3.1. Granjas de render no especializadas:

Principalmente hay dos limitaciones básicas:

Limitación	Consecuencia
En disponibilidad	Los ordenadores deben estar disponibles por tiempo indeterminado, es decir, lo necesario para completar el proceso global de render.
En heterogeneidad	La heterogeneidad entre los equipos, lo más usual, en un entorno de oficina, obliga a esperar a que concluya el render en el equipo más lento para disponer del render completo.

Tabla 4 limitaciones básicas de las granjas de render

#### 3.3.2. Granjas de render profesionales:

Requieren de un entorno controlado, acondicionamiento de instalaciones para un alto rendimiento. Sus instalaciones requieren de una compleja cantidad de elementos.

- Nodos de render
- Switch de conectividad
- Switch de monitoreo
- Servidor de archivos
- UPS
- Servidor de backup
- Y un largo etcétera de otras cosas

En conclusión requieren de una infraestructura específica y costosa. Muchas veces no asequibles por pequeñas organizaciones y sobre todo caras de mantener.

Como alternativa existe la posibilidad de alquilar estos servicios.

### 3.4. Sistemas distribuidos de datos

Actualmente existen diversas maneras de almacenar datos de manera distribuida.

#### Tipos de Sistemas de almacenamiento distribuidos

Bases de datos distribuidas.

Tablas hash distribuidas.

Sistemas de índice centralizados.

Sistemas personalizados.

Tabla 5 Tipos de Sistemas de almacenamiento distribuidos

### 3.4.1. Bases de datos Distribuidas

Las bases de datos distribuidas nos permiten almacenar los datos en varias máquinas y/o volúmenes.

#### *Ventajas de una base de datos distribuida:*

Ventaja	Descripción
Refleja una estructura organizacional	Los fragmentos de la base de datos se ubican en los departamentos a los que tienen relación.
Autonomía local	Un departamento puede controlar los datos que le pertenecen.
Disponibilidad	Un fallo en una parte del sistema solo afectará a un fragmento, en lugar de a toda la base de datos.
Rendimiento	Los datos generalmente se ubican cerca del sitio con mayor demanda, también los sistemas trabajan en paralelo, lo cual permite balancear la carga en los servidores.
Economía	Es más barato crear una red de muchas computadoras pequeñas, que tener una sola computadora muy poderosa.
Modularidad	Se pueden modificar, agregar o quitar sistemas de la base de datos distribuida sin afectar a los demás sistemas (módulos).

Tabla 6 Ventajas de una base de datos distribuida

#### *Desventajas de una base de datos distribuida*

Desventaja	Descripción
Complejidad	Se debe asegurar que la base de datos sea transparente, se debe lidiar con varios sistemas diferentes que pueden presentar dificultades únicas. El diseño de la base de datos se tiene que trabajar tomando en cuenta su naturaleza distribuida, por lo cual no podemos pensar en hacer joins que afecten varios sistemas.
Economía	La complejidad y la infraestructura necesaria implica que se necesitará una mayor mano de obra.
Seguridad	Se debe trabajar en la seguridad de la infraestructura así como cada uno de los sistemas.
Integridad	Se vuelve difícil mantener la integridad, aplicar las reglas de integridad a través de la red puede ser muy caro en términos de transmisión de datos.
Falta de experiencia	Las bases de datos distribuidas son un campo poco común por lo cual no existe mucho personal con experiencia o conocimientos adecuados.
Carencia de estándares	Aún no existen herramientas o metodologías que ayuden a los usuarios a convertir un DBMS centralizado en un DBMS distribuido.
Diseño de la base de datos se vuelve más complejo	Además de las dificultades que generalmente se encuentran al diseñar una base de datos, el diseño de una base de datos distribuida debe considerar la fragmentación, replicación y ubicación de los fragmentos en sitios específicos.
Diseño intrínseco	Las bases de datos en general no están preparadas para la consulta de una estructura de árbol. Aunque existen diversas instrucciones SQL no estándar para trabajar con árboles, estas funcionan en base a costosos cálculos las cuales no las hacen óptimas para el uso por sistemas de render.

Tabla 7 Desventajas de una base de datos distribuida

### 3.4.2. *Tablas hash Distribuidas*

Las tablas de hash distribuidas (DHTs), 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.

### 3.4.3. *Sistemas de índice centralizados*

Consiste en sistemas donde el índice de donde están los datos está en uno o varios equipos dedicados.

Implica una interacción muy férrea con el sistema encargado de indexar la información. Si estos sistemas de indexado caen, cae todo el sistema. Además existe un claro cuello de botella en los equipos indexadores.

### 3.4.4. *Sistemas personalizados*

Implica la implementación de algoritmos de relación y un algoritmo de mantenimiento. La mayoría de las veces incurren en costosos tiempos de desarrollo, pudiendo caer en no abordar todas las posibles situaciones.

## 3.5. Estado de las técnica

### 3.5.1. *Los algoritmos*

El campo de los algoritmos usados en los sistemas de render está muy estudiado. Hoy en día se sabe cuáles son los mejores algoritmos y optimizaciones para procesar una escena 3D.

Técnica	Fundamentos Físicos
Sombreado de Gouraud	Luz ambiente y difusa.
Sombreado de Phong	Luz ambiente, difusa y especular.
Raytracing	Simular luz como rayo, iluminación global.
Mapeado de fotones	Basado en Montecarlo, iluminación global.
Radiosidad	Equilibrio de energía.

Tabla 8 Algoritmos usados en los sistemas de render

En general el mundo de las imágenes sintéticas, cuando queremos sintetizar un fotograma de nuestra escena debemos mezclar distintos algoritmos con sus optimizaciones. Por ejemplo cuando se quiere simular una escena donde hay una niebla, se hacen los ajustes pertinentes para poder optimizar su procesamiento al máximo. Así es normal ver infinidad de publicaciones que mejoran la precisión y el realismo de un render bajo ciertas condiciones.

### 3.5.2. Las redes de pares

Alejándonos de las polémicas que salpican los medios de comunicación. Las redes de pares no se limitan a compartir archivos de dudosa legalidad. Estos sistemas se basan en la idea de compartir recursos directamente entre dos equipos de red.

De este paradigma se han aprovechado proyectos para hacer posible la realización de tareas faraónicas en esta era digital.

#### Ejemplo famosos de redes P2P

Quitando los sistemas de compartición de datos, aquí nombro algunos de los proyectos reseñables.

Proyecto	Finalidad
SETI@HOME	Búsqueda de inteligencia extraterrestre.
BOINC	Proyecto para la compartición de los recursos con diversos fines científicos.
Netsukuku	Es el intento por levantar una red completamente distribuida sin puntos únicos de fallo.

Tabla 9 Ejemplo famosos de redes P2P

#### Características deseables en una red P2P

Entre las características deseables en este tipo de redes destacamos:

Característica	Descripción
Escalabilidad	El sistema tiene que permitir escalar sin verse comprometida la eficiencia en general. Lo ideal es que cuanto mayor es el número de participantes más eficiente se vuelva el sistema.
Robustez	En estos sistemas se consigue la tolerancia a fallos mediante la redundancia de los recursos. Además en sistemas puros la necesidad de un sistema de índice queda solventada, no habiendo ningún punto singular de fallo.
Descentralización	Son sistemas descentralizados, todos los nodos son iguales. Por lo tanto no hay ningún nodo con una funcionalidad especial.
Distribución de costes	Los recursos individuales son compartidos por el sistema. Permitiendo su uso por cualquier nodo de la red.
Anonimato	Es una característica deseable en sistemas que sobrepasan el ámbito controlado de una organización.
Seguridad	Cuando el sistema escapa de los dominios de una organización. Se hace necesario el implementar en el sistema formas de evitar posibles nodos maliciosos, así como posibles espionajes o alteraciones de los datos de la red.

Tabla 10 Características deseables en una red P2P

### *Tipos de redes P2P*

Las redes P2P según el grado de centralización se pueden clasificar:

<b>Tipo</b>	<b>Descripción</b>
Centralizadas	Se rige bajo un único servidor, este servidor sirve de punto de enlace entre los distintos nodos. Todas las comunicaciones dependen del servidor.
Híbridas, semicentralizadas o mixtas	El servidor sirve de HUB y administrador de los recursos. Suelen permitir también trabajar en un funcionamiento de tipo puro.
Puras	Totalmente descentralizadas. Todos los nodos funcionan como cliente y servidor.

Tabla 11 Tipos de redes P2P



## Capítulo 4. DESCRIPCIÓN DEL SISTEMA

### 4.1. Preámbulo

Nuestro sistema se va a encargar de evaluar una serie de algoritmos típicos del render permitiendo la distribución de los cálculos y los datos de una manera descentralizada.

### 4.2. Diagrama del sistema

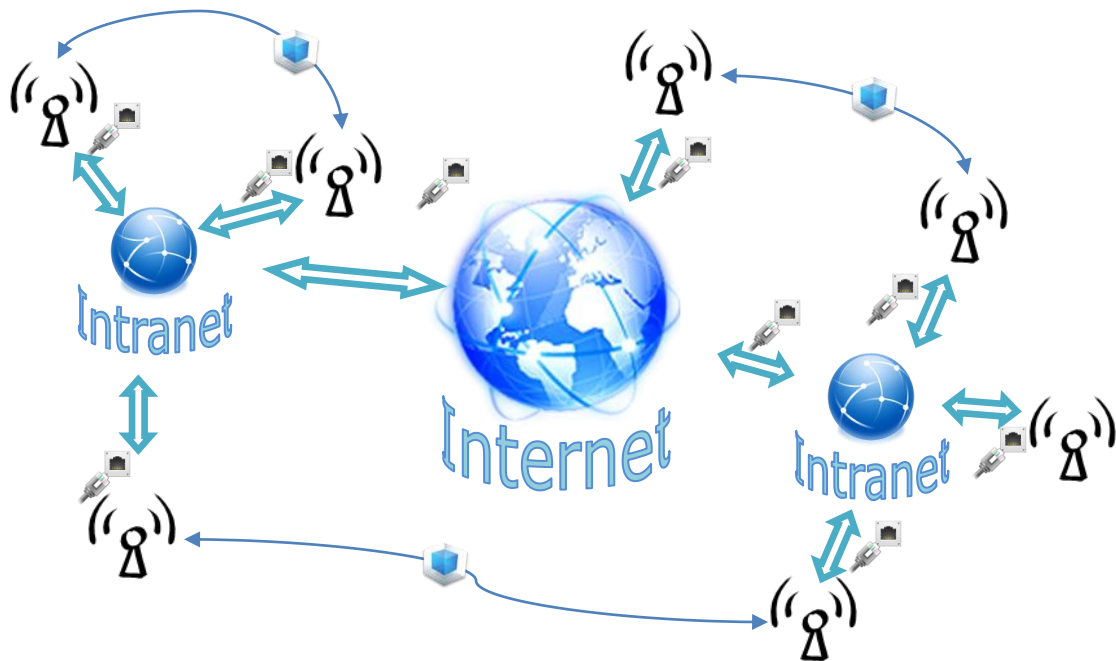


Ilustración 2 Diagrama del sistema

#### Descripción

El sistema a nivel global, estaría compuesto de varias estaciones de render. Cada estación haría las veces de cliente y servidor. Dichas estaciones se conectarían unas a otras mediante redes internas y/o internet.

A nivel lógico la conexión se haría punto a punto, es decir no habría intervención de terceros a la hora comunicarse, básicamente las estaciones trabajarían en el enrutamiento.

Tabla 12 Diagrama del sistema

### 4.3. Estructura de una estación de render

#### 4.3.1. *Mantenimiento de la red*

Cada estación de render conforma un nodo de la red de pares. Dicho nodo ejercerá funciones de cliente y servidor al más purista estilo **Peer to Peer**.

En el enrutamiento, se mantendrá una cache con los datos de conexión de los clientes, estaciones de render. También se dará soporte a una red tipo **overlay** usando dicha cache como tabla de enrutamiento.

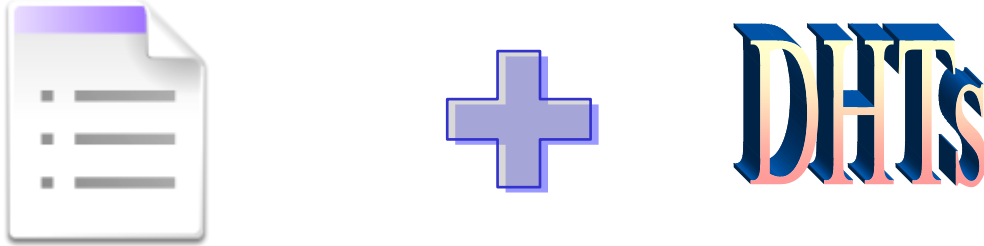


Ilustración 3 Mantenimiento de la red

---

#### Descripción

El sistema mantendría una pequeña cache de estaciones de render. La red tendrá como base una red tipo **overlay**.

Tabla 13 Mantenimiento de la red

#### 4.3.2. Modelo vista controlador

La estructura interna de las estaciones de render estará optimizadas para mejoras futuras. Para ello se hará uso del paradigma de programación MVC (modelo, vista, controlador). Además con el fin de aumentar las posibilidades futuras del controlador, este contendrá una estructura modular.



Ilustración 4 Modelo vista controlador

#### Descripción

Cada estación de render internamente estará separada en tres módulos independientes, los cuales deberán respetar una estructura básica, pudiendo esta ser ampliada. De esta manera se aporta a posibilidad de una implementación posterior de cualquiera de ellos sin perder compatibilidad. De esta manera se puede pensar en la implementación de una vista preparada para funcionar en la web, aumentar los medios de conectividad de la estación, o simplemente agregar una extensión al modelo.

Tabla 14 Modelo vista controlador

### 4.3.3. La vista

La aplicación está abierta a varios tipos de dispositivos. Los interfaces de cada uno de estos dispositivos pueden ser muy variados.



Ilustración 5 La vista

Las posibilidades de interfaz de un computador de escritorio no son las mismas que las de un teléfono móvil, ni las mismas que las de un dispositivo empotrado. La vista ha de servirnos como interfaz de entrada de los mandatos y proveernos de medios de realizar un seguimiento de la estación.

### 4.3.4. Características deseables para el interfaz de usuario

Vamos a procurar que el interfaz de usuario cumpla una serie de características que estimamos deseables.

Característica	Descripción
Procesado de la información	Los resultados de las distintas operaciones deben ser representados de una manera sencilla y lo suficiente detallada, gráficas, listados, cuadros de estadística.
Interfaz adaptada al dispositivo	El interfaz de usuario debe hacer uso de las posibilidades del dispositivo anfitrión, no todas las interfaces valen para todos los dispositivos.
Simplicidad de gestión	Los medios para operar el interfaz deberían ser lo más sencillos posibles.

Tabla 15 Características deseables para el interfaz de usuario

#### 4.3.5. El controlador

Nuestro controlador tendrá una estructura modular. De este modo siempre que implementemos una estructura básica se mantendrá la compatibilidad con el resto de los elementos de la aplicación.

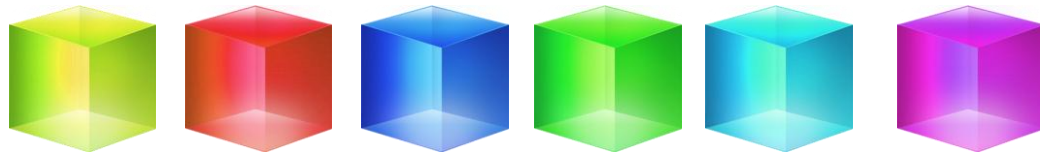


Ilustración 6 El controlador

#### Descripción

El controlador de la estación tendrá una estructura modular. Cada uno de los módulos implementará funcionalidades relacionadas, tales como funcionalidades relacionadas con los servicios, con las estaciones de render, con el árbol, etcétera...

Tabla 16 El controlador

#### 4.3.6. Características deseables para los módulos del controlador

Las características que deberían reunir los distintos módulos del servidor serían:

Característica	Descripción
Opacidad	Las interdependencias de los módulos perjudican el posible reemplazo de estos por parte de futuras versiones. Por ello su implementación ha de ser lo más opaca posible para el resto de los módulos.
Funcionalidades únicas	Los módulos no deberían duplicar funcionalidades.
Especialización	La especialización de un módulo nos permite el poder englobar todas las tareas comunes de la tarea, permitiendo la reutilización en el mismo.
Estructura	Los módulos tienen que definir una estructura básica que permita su adición en el sistema.
Reemplazo	Se ha de hacer posible el reemplazar la implementación de un módulo por otra conservando la funcionalidad.
Atomicidad en las funcionalidades	Las funcionalidades deben ser lo más atómicas posibles sin entrar en una elevada complejidad. Con esto se consigue reutilizar funcionalidades.

Tabla 17 Características deseables para los módulos del controlador

**4.3.7. Funcionalidades del controlador**

Las funcionalidades que implementa cada estación de render las agruparemos en las siguientes categorías:

Temática	Descripción
Sistema	Funcionalidades que afectan a nivel de sistema.
Servicios de red	Funcionalidades relacionadas con los servicios.
Estaciones de Render	Funcionalidades relacionadas con las estaciones de render.
Formas	Funcionalidades relacionadas con las formas.
Árbol kd	Funcionalidades relacionadas con el árbol kd.
Simulación	Funcionalidades relacionadas con la simulación.

Tabla 18 Funcionalidades del controlador

**Funcionalidades de Sistema**

Son funcionalidades relacionadas con la aplicación en general.

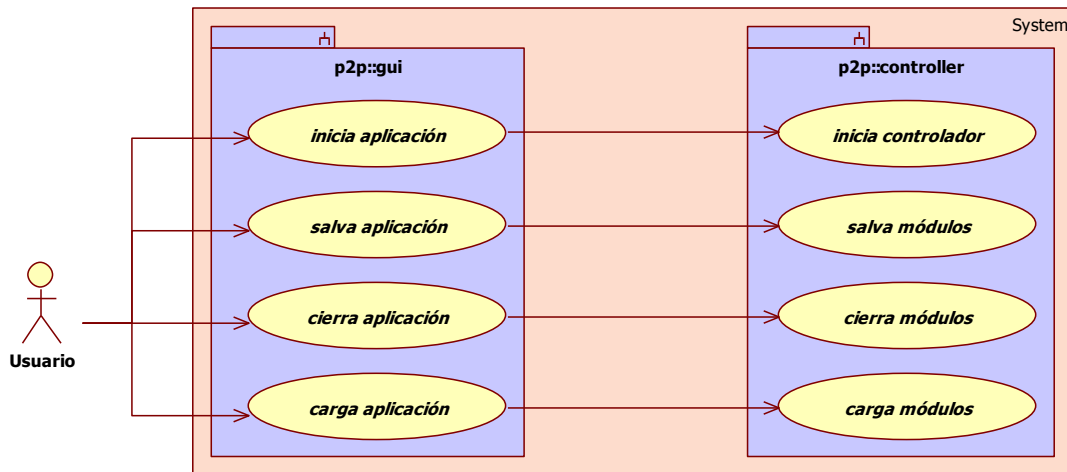
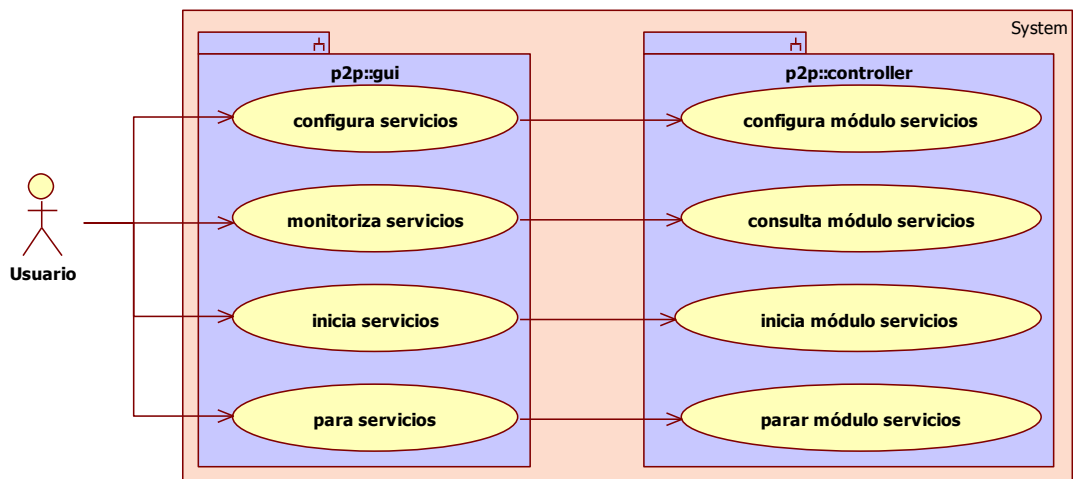


Ilustración 7 Funcionalidades de Sistema

Funcionalidad
Salvar el estado de la aplicación.
Cargar el estado de la aplicación.
Iniciar la aplicación.
Cerrar la aplicación.

Tabla 19 Funcionalidades de Sistema

*Funcionalidades de Servicios de red*



**Ilustración 8** Funcionalidades de Servicios de red

Funcionalidad
Iniciar los servicios de red.
Parar los servicios de red.
Configurar los servicios de red.
Monitorizar el estado de la red.

**Tabla 20** Funcionalidades de Servicios de red

*Funcionalidades relacionadas con las estaciones de render*

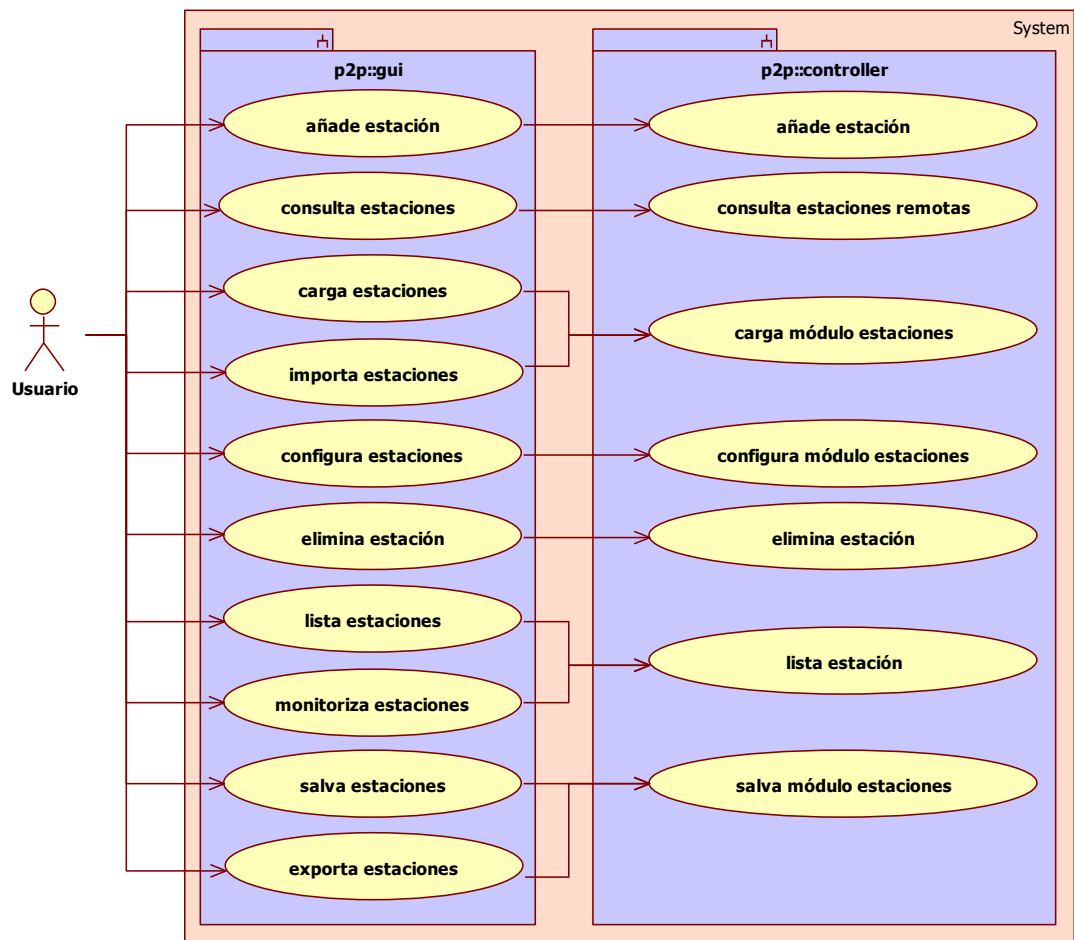


Ilustración 9 Funcionalidades relacionadas con las estaciones de render

Funcionalidad
Añadir estaciones.
Cargar estaciones.
Configurar la redistribución de las estaciones conocidas en la red.
Monitorizar el estado de las estaciones.
Guardar las estaciones.
Eliminar estaciones.
Listar estaciones.
Exportar estaciones.
Importar estaciones.
Propagar estaciones.
Consultar estaciones remotas.

Tabla 21 Funcionalidades relacionadas con las estaciones de render



*Funcionalidades relacionadas con las formas*

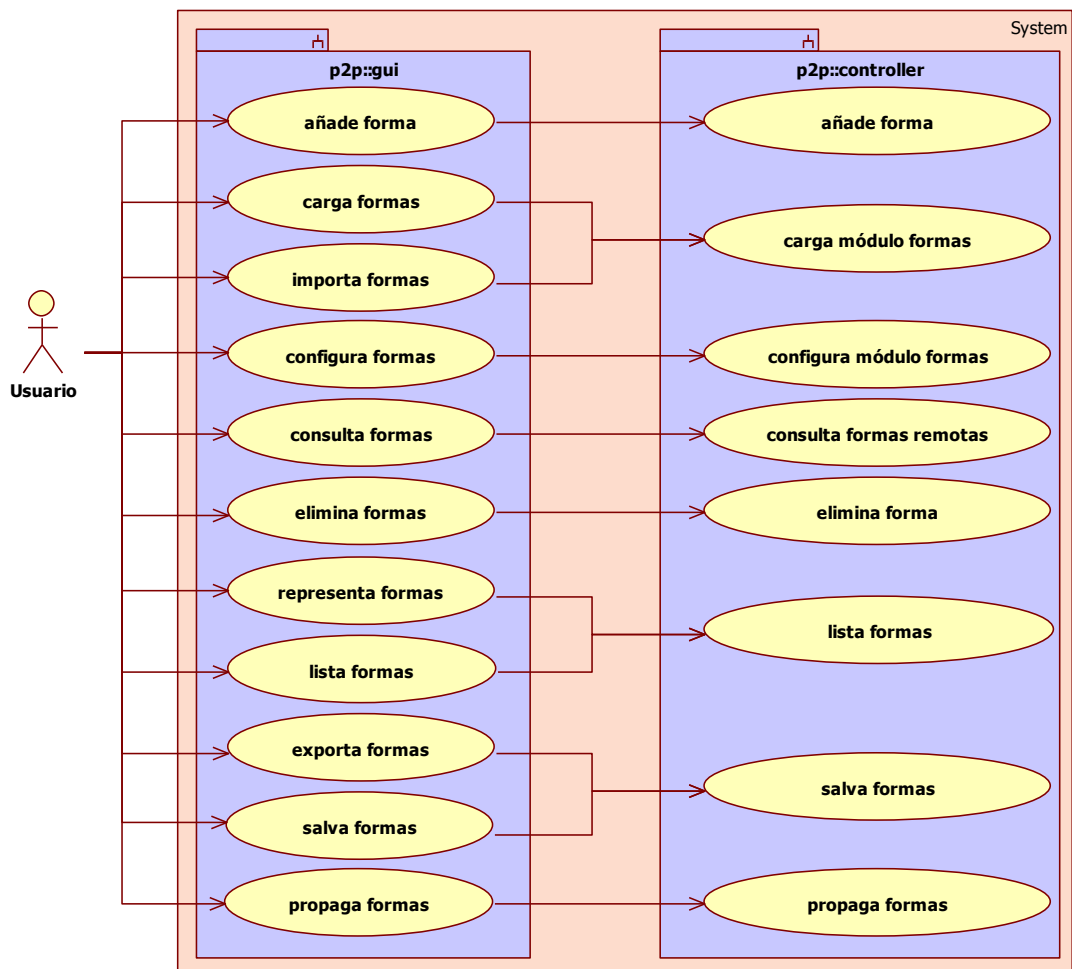


Ilustración 10 Funcionalidades relacionadas con las formas

Funcionalidad
Gestión de las formas
Configurar la redistribución de las formas en la red
Monitorizar las formas.
Estadísticas de las formas.
Añadir formas
Eliminar formas
Guardar formas
Cargar formas
Listar formas
Exportar formas
Importar formas
Propagar formas
Obtener formas remotas

Tabla 22 Funcionalidades relacionadas con las formas

*Funcionalidades relacionadas con el árbol kd*

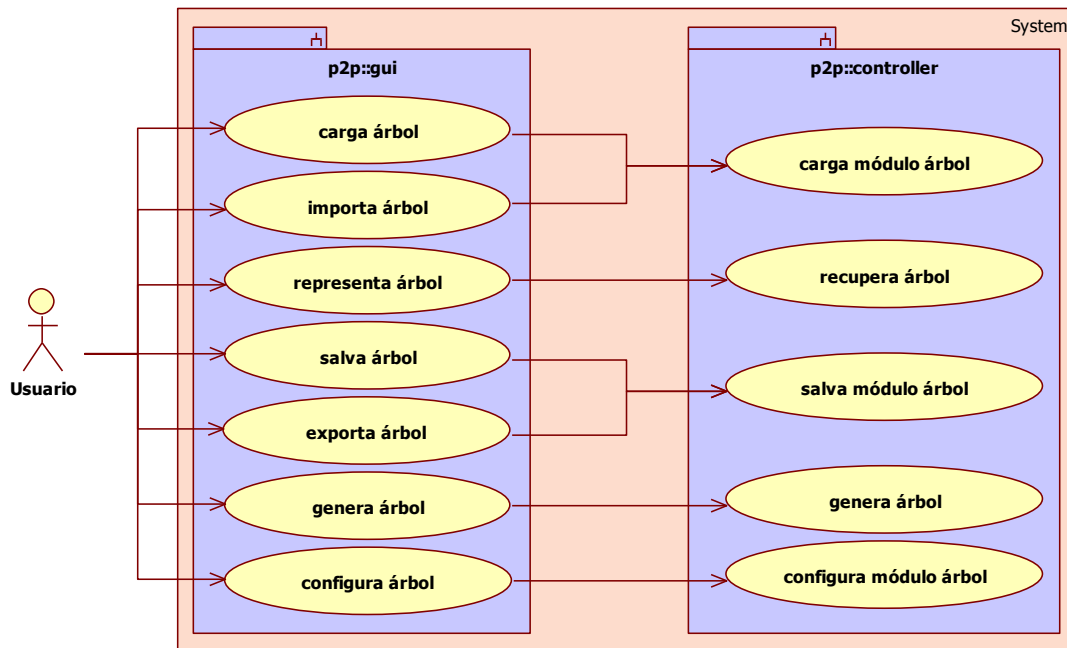
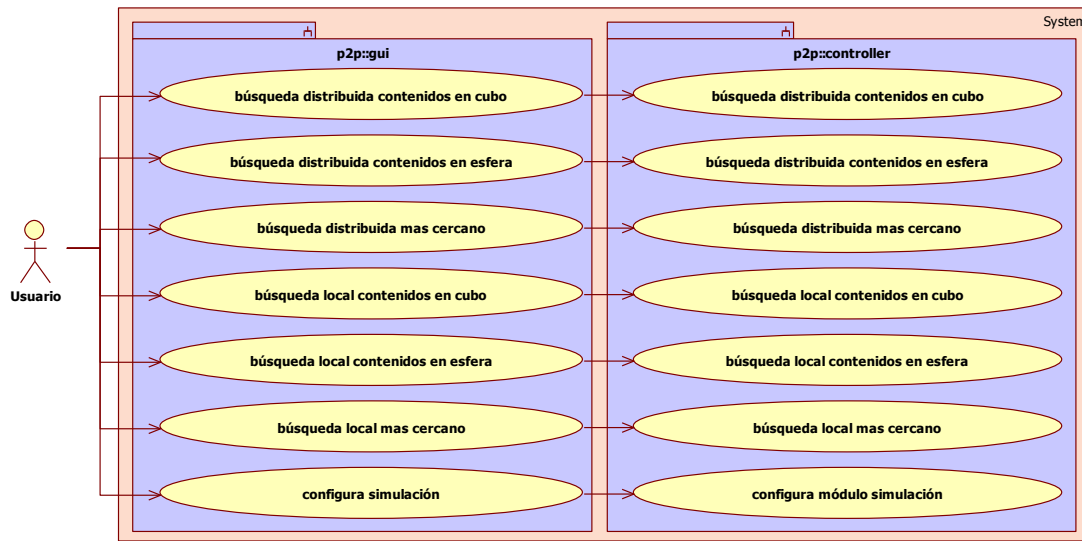


Ilustración 11 Funcionalidades relacionadas con el árbol kd

Funcionalidad
Gestión del árbol kd.
Generación del árbol kd.
Configuraciones relacionadas con el árbol.
Estadísticas del árbol.
Salva el árbol.
Carga el árbol.
Exporta el árbol.
Importa el árbol.
Propaga el árbol.
Representa el árbol.

Tabla 23 Funcionalidades relacionadas con el árbol kd

*Funcionalidades de simulación*



**Ilustración 12 Funcionalidades de simulación**

**Funcionalidad**

Búsqueda local del punto más cercano.

Búsqueda local contenidos en una esfera.

Búsqueda local contenidos en un cubo.

Búsqueda distribuida del punto más cercano.

Búsqueda distribuida contenidos en una esfera.

Búsqueda distribuida contenidos en un cubo.

Configuración de la simulación.

**Tabla 24 Funcionalidades de simulación**

**4.3.8. El modelo**

Los datos que deben ser almacenados por las estaciones de render son:

Datos	Descripción
De las estaciones de render	En cada estación se almacena una cache de estaciones de render.
Del listado de formas	En cada estación se almacena una parte del modelo de formas.
Del Árbol kd	En cada estación se almacena una parte del árbol kd.
De la configuración	El estado de las automatizaciones y configuraciones realizadas en la estación.

**Tabla 25 El modelo**

### *Cache de estaciones de render*

Cada estación de la red para realizar simulaciones debe comunicarse con el resto de la red. Con el fin de agilizar las comunicaciones se almacena una pequeña cache con los datos de enrutamiento de las mismas. Esta cache contendrá no necesariamente el listado de todas las estaciones de la red. En caso de que el tamaño de la red sobrepase unas dimensiones considerables se hará uso de un enrutamiento mediante una red **overlay**.

### *Listado de formas*

Nuestro modelo a simular generara un árbol kd mediante un listado con datos de las formas que están representadas en el espacio 3D. El tamaño de dichas formas puede ser lo suficientemente grande para no dar cabida en un solo computador. Por ello el almacenamiento de los datos ha de ser distribuido por las estaciones de render para su posterior procesado. Para dotar al sistema de mayor seguridad habrá redundancia de los datos.

### El árbol kd

Para la generación del árbol, podremos aplicar distintas técnicas de generación.

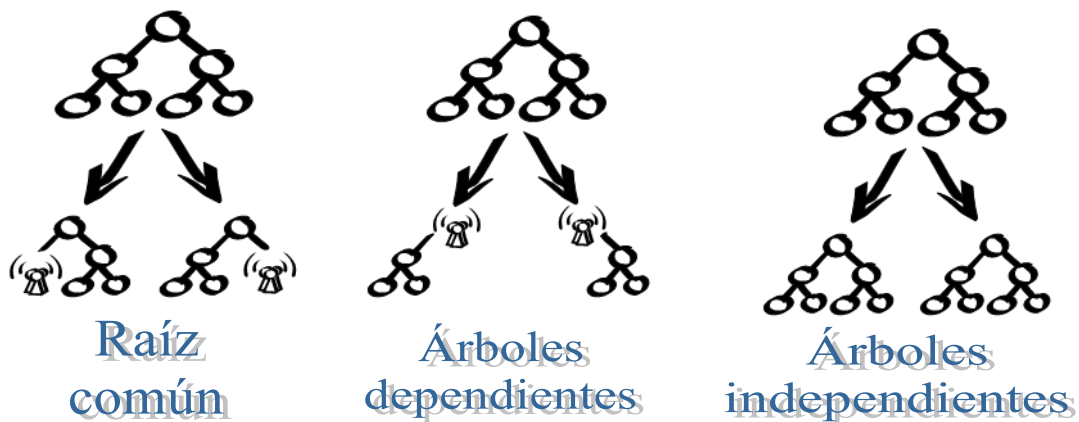


Ilustración 13 Técnicas de generación y reparto del árbol kdtree

Técnica	Descripción
Raíz común	Todas las estaciones de la red tienen una raíz del árbol común, dicha raíz contiene ramas del árbol que están almacenadas en otras estaciones de la red.
Árboles dependientes	El árbol está fragmentado en pequeños subárboles incompletos, dichos subárboles tienen nodos que están almacenados en otras estaciones de la red.
Árboles independientes	Cada estación genera su propio árbol con el listado de formas locales.

Tabla 26 Técnicas de generación y reparto del árbol kdtree

Según la situación a simular puede ser interesante optar por uno o por otro. Se podría implementar un reparto del árbol en el cual cada estación se ocupe de un área del espacio, por tener esta más interés en realizar búsquedas en su zona asignada. Por lo que sería interesante que el reparto no fuese aleatorio.

En nuestro caso nos centraremos en la búsqueda mediante árboles independientes, ya que nos interesa evaluar las prestaciones de ciertos algoritmos que no tiene cabida este tipo de reparto.



## Capítulo 5. SOLUCIÓN PROPUESTA

### 5.1. Preámbulo

Vamos a evaluar la solución propuesta. Para ello analizaremos las diferentes tecnologías existentes en el mercado y las decisiones que nos han llevado a adoptar unas frente a otras.

### 5.2. Sistema de comunicación

En las comunicaciones de red se quería un sistema que fuera lo más simple posible, obviándonos del duro trabajo de una implementación de cero, pero además que fuera lo suficientemente óptimo para que dichas comunicaciones no representasen un problema elevado de rendimiento.

Las opciones evaluadas fueron:

Sistema de comunicación
RMI
WEBSERVERS
SOCKETS
RPC

Tabla 27 Sistema de comunicación

#### 5.2.1. Análisis de los sistemas de comunicación mediante Servicios Web

El uso de Servicios Web no cumplen con la necesidad de realizar unas comunicaciones lo más óptimas posibles al sobredimensionar en exceso las llamadas.

#### 5.2.2. Análisis de los sistemas de comunicación mediante Sockets

En el caso de los Sockets nos implica una implementación demasiado a bajo nivel. Aunque es bastante óptima en cuanto a tamaño de datos.

#### 5.2.3. Análisis de los sistemas de comunicación mediante RPC

En el caso de RPC, los que utilizan el estándar IDL tienen una alta compatibilidad, pero su rendimiento depende en gran medida de la implementación realizada por cada uno de los actores en el juego.

#### 5.2.4. *Análisis de los sistemas de comunicación mediante RMI*

Teniendo como gran contra, el ser una comunicación cerrada al lenguaje Java, se optó por esta solución, al estar lo suficientemente madura y aportar una simplicidad a nuestras comunicaciones.

El uso de RMI posibilita el envío de clases serializadas entre los actores.

Como previsión se ha decidido el dar modularidad al núcleo de la aplicación, en previsión de poder en un futuro ampliar éste con la incorporación de distintos módulos que den soporte a otras formas de comunicación.

### 5.3. Interfaz de Usuario

Entre las alternativas para interactuar en la aplicación tenemos las siguientes:

---

#### **Tipo de interfaz**

Línea de comandos.

Entorno Web.

Entorno de escritorio.

Sistema empotrado.

---

Tabla 28 Tipos de interfaz

#### 5.3.1. *Línea de comandos*

El uso de un entorno de línea de comandos se hace demasiado complicado para un usuario estándar.

#### 5.3.2. *Entorno Web*

Un entorno Web es muy interesante, en un principio se inició su implantación. Pronto se hizo patente de que eleva la complejidad de nuestra aplicación a niveles demasiado elevados y entorpece su distribución al público en general, al tener que instalarse un servidor de aplicaciones Web.

#### 5.3.3. *Sistema empotrado*

Dados los medios materiales disponibles, se descartó esta posibilidad.

#### 5.3.4. *Entorno de Escritorio*

El uso de una solución de escritorio nos simplificó la implantación de la solución, permitiendo el uso de herramientas gráficas para presentar los datos al usuario con unos costes medios de tiempo, frente a su implementación en un entorno Web.

Debido a las posibilidades del desarrollo futuro de un entorno Web para la supervisión de la aplicación se ha decidido implementar la estructura de la aplicación siguiendo el patrón de diseño MVC.



## 5.4. Formato de almacenamiento de los datos

A la hora de utilizar una fuente de datos se nos presentan las siguientes alternativas:

---

### Formatos evaluados

Utilizar una base de datos.

Utilizar objetos con serialización estándar.

Utilizar XML.

Utilizar objetos con serialización personalizada.

---

Tabla 29 Formatos de almacenamiento de los datos

### 5.4.1. Análisis del uso de una Base de datos

El uso de una base de datos implicaría el uso de un gestor de BD, que aunque hay gestores incrustados en aplicaciones, no nos proporciona una estructura óptima para almacenar y acceder a un árbol kd.

### 5.4.2. Análisis del uso de XML

Utilizar XML nos sobredimensiona el tamaño de los datos, ocupando demasiado la estructura del formato frente al elemento a almacenar.

### 5.4.3. Análisis del uso de Objetos con serialización

El uso de una serialización estándar nos ocupa una memoria reducida pero esta puede ser mejorable.

### 5.4.4. Análisis del uso de Objetos con serialización personalizada

El uso de una serialización personalizada nos permitiría un uso óptimo de los recursos.

Finalmente se ha optado por una serialización personalizada con el fin de optimizar el uso de la memoria.

## 5.5. Lenguajes de programación evaluados

Los lenguajes evaluados han sido:

Lenguaje
Java
C/C++
PHP
Delphi
Visual Basic, C#
LISP, SMALLTALK,...

Tabla 30 Lenguajes de programación evaluados

Se ha optado por la implementación de una solución basada en java por tener los siguientes puntos a su favor:

### Puntos a favor de java

Ser esta una plataforma de código gratuita, lo cual redundará en un ahorro de gastos.

Ser multiplataforma, lo cual implica el poder llegar a un mayor número de entornos.

Tener una gran cantidad de APIs y documentación, esto es sumamente importante a la hora de poder desarrollar nuestra solución con una complejidad lo menor posible.

Facilidad de aprendizaje e implementación. Otros lenguajes exóticos o bien carecen de la simplicidad de este lenguaje o no tienen una comunidad de la que nutrirnos de soluciones.

Es un lenguaje ya utilizado con lo que se tienen la base en el desarrollo del mismo.

Tabla 31 Puntos a favor de java

Como puntos en contra, frente a una plataforma nativa:

### Puntos en contra de java

Al ser un lenguaje que corre en una máquina virtual por lo que tiene un rendimiento menor que códigos nativos. Existen diversos estudios demuestran que en rendimiento están más o menos empatados, siendo más importante las implementaciones. Aun así el rendimiento de no es un factor determinante para nuestra elección ya que el banco de pruebas a ejecutar no va a ser multilinguaje. Y la comparación se realizara con un mismo lenguaje como banco de pruebas.

Necesidad de instalación de máquina virtual.

Tabla 32 Puntos en contra de java

## 5.6. La gestión del proyecto

Se ha decidido modelar la solución dando un soporte modular a nuestra implementación. Para ello se ha decidido dividir la solución en tres proyectos al más puro estilo MVC.

Primero tendremos la implementación del modelo. Tiene el objetivo de proveer de un modelo de datos al sistema. Este modelo modelara un interfaz común para alimentar al sistema permitiendo la integración de nuevas implementaciones que permitan el aumentar las posibilidades del mismo manteniendo la compatibilidad con versiones anteriores.

En cuanto al controlador tendrá una estructura modular. Esta tiene el fin de poder escalar las comunicaciones permitiendo proveer de nuevos medios de integración. Además de permitir la modificación del sistema de almacenamiento o computación.

El separar la vista del controlador nos permitirá poder adaptar la solución a otros medios como puede ser una página web o una consola de comandos.



Ilustración 14 Tres proyectos al más puro estilo MVC

### 5.7. Modelado 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 generar 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 33 Modelado de los datos del modelo

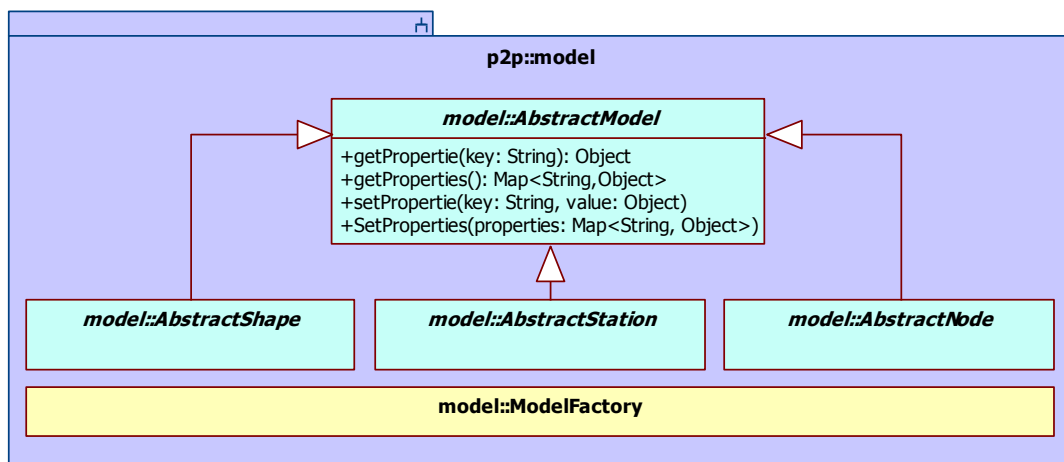


Ilustración 15 Modelado del modelo

#### 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 34 Modelado del modelo

### 5.7.1. Modelado del modelo de las estaciones de render

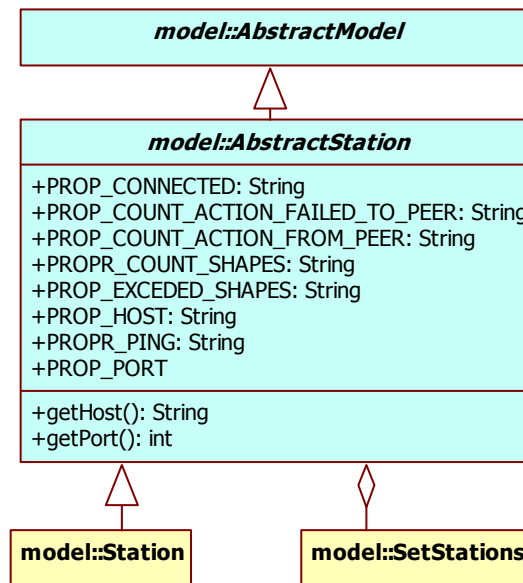


Ilustración 16 Modelado del modelo de las estaciones de render

#### 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 heredaran esta estructura, asegurando la compatibilidad.

Tabla 35 Modelado del modelo de las estaciones de render

5.7.2. Modelado del modelo del listado de formas

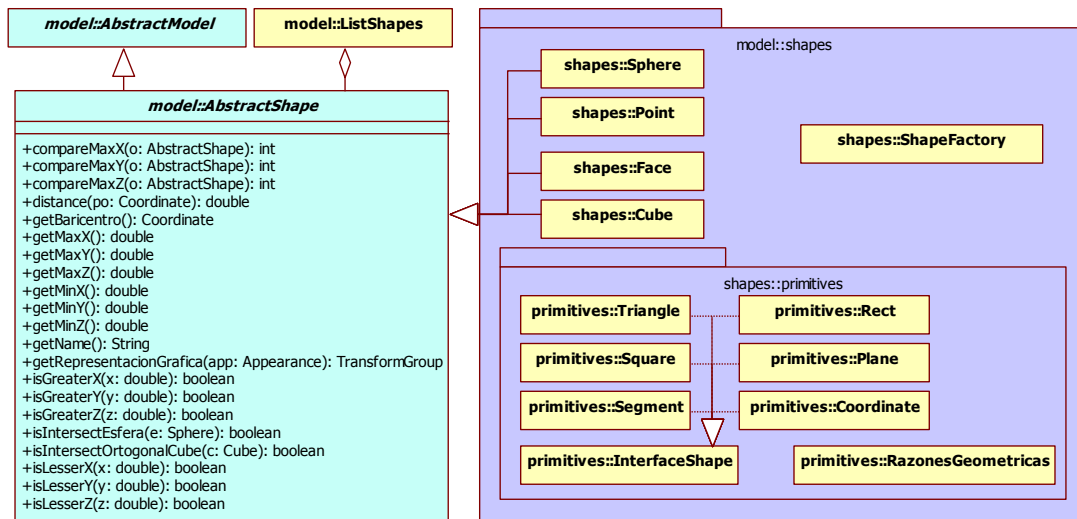


Ilustración 17 Modelado del modelo del listado de formas

**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 heredaran 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 36 Modelado del modelo del listado de formas

5.7.3. Modelado del Árbol kd

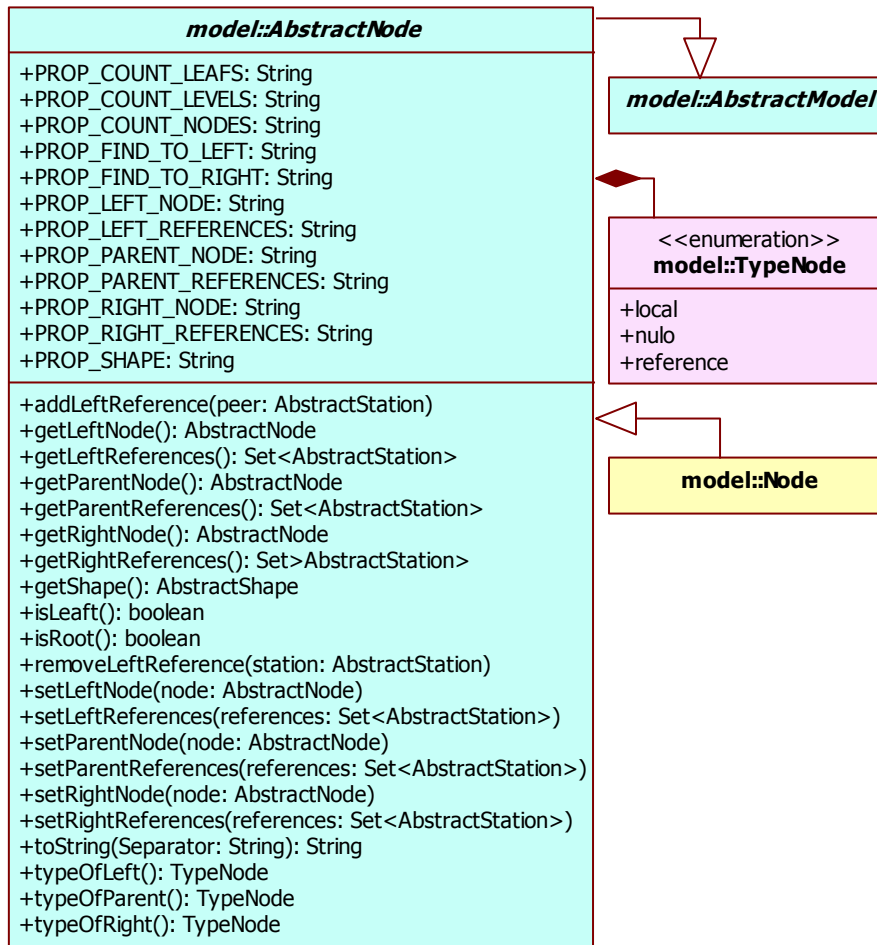


Ilustración 18 Modelado del Árbol kd

**Descripción**

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

Tabla 37 Modelado del Árbol kd

*Para más detalles ver los anexos – Análisis*

### 5.8. Modelado del Controlador

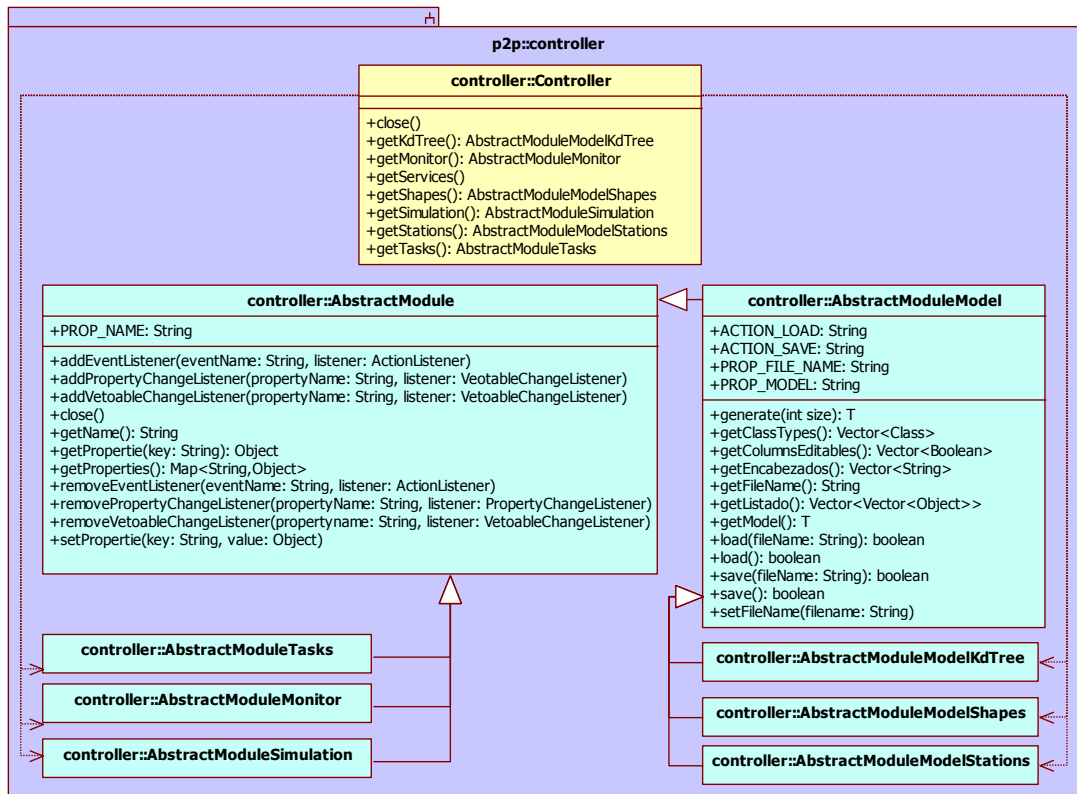


Ilustración 19 Modelado del Controlador

#### 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 está descrita por módulos abstractos, con el fin de poder ampliar dicha funcionalidad, manteniendo la compatibilidad de las distintas versiones.

Tabla 38 Modelado del Controlador



5.8.1. Modelado del módulo del controlador de servicios

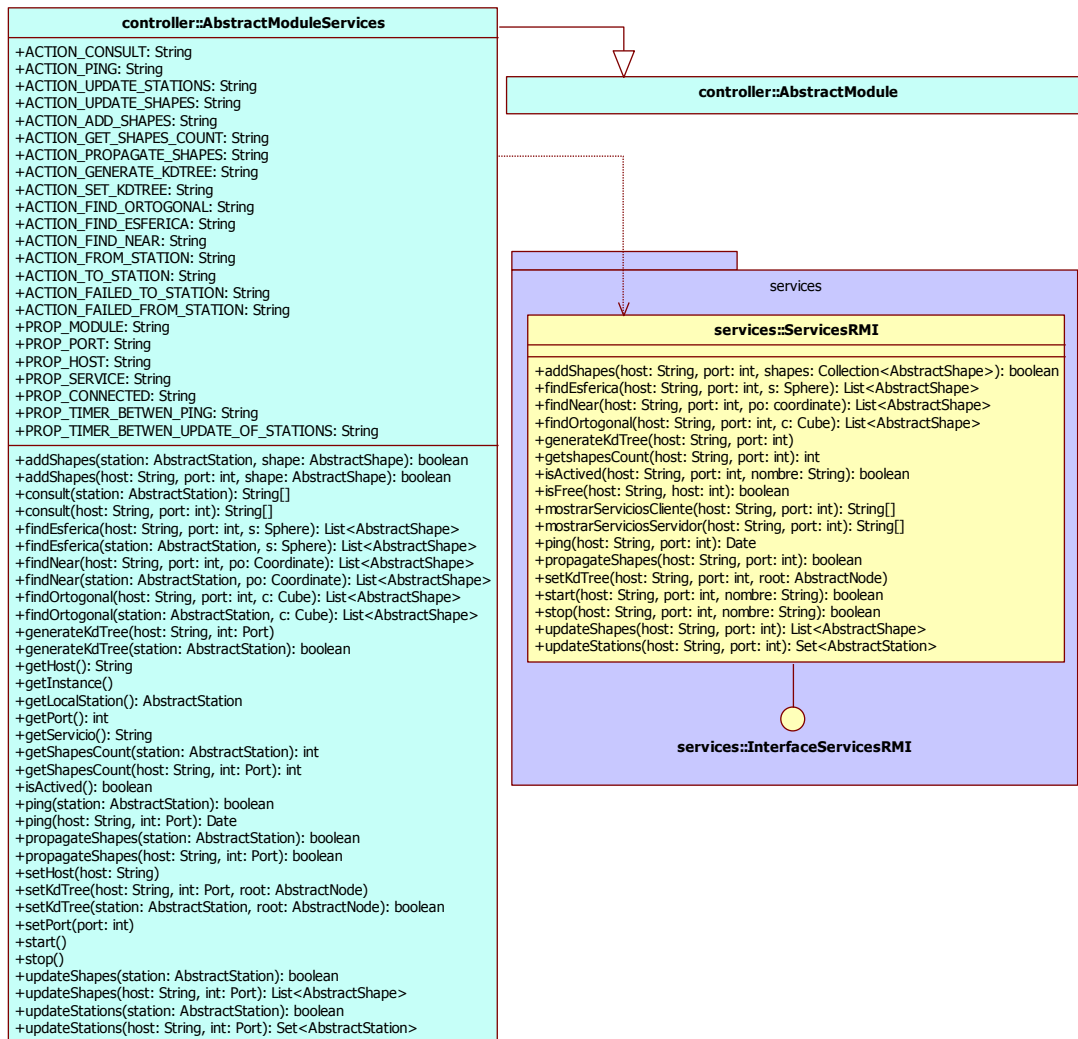


Ilustración 20 Modelado del módulo del controlador de servicios

**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 39 Modelado del módulo del controlador de servicios

5.8.2. Modelado del módulo del controlador del listado de estaciones de render

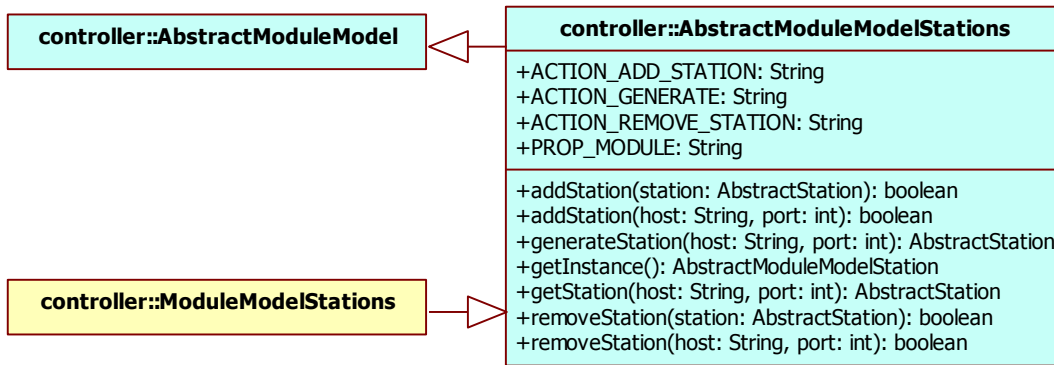


Ilustración 21 Modelado del módulo del controlador del listado de estaciones de render

**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 40 Modelado del módulo del controlador del listado de estaciones de render

5.8.3. Modelado del módulo del controlador de formas

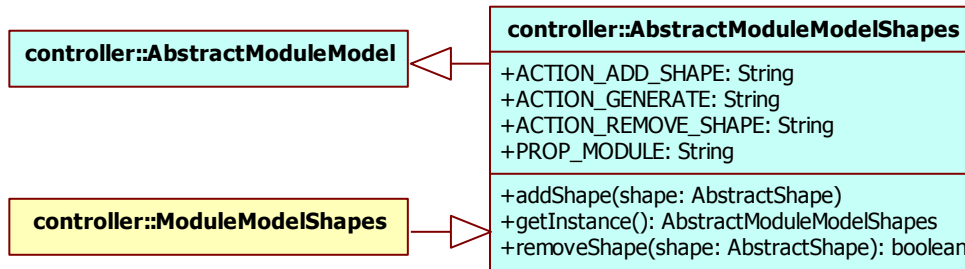


Ilustración 22 Modelado del módulo del controlador de formas

**Descripción**

El modulo del listado de formas es el encargado de gestionar localmente las formas que posteriormente se almacenarán en el árbol.

Tabla 41 Modelado del módulo del controlador de formas

5.8.4. Modelado del módulo del controlador de árbol kd

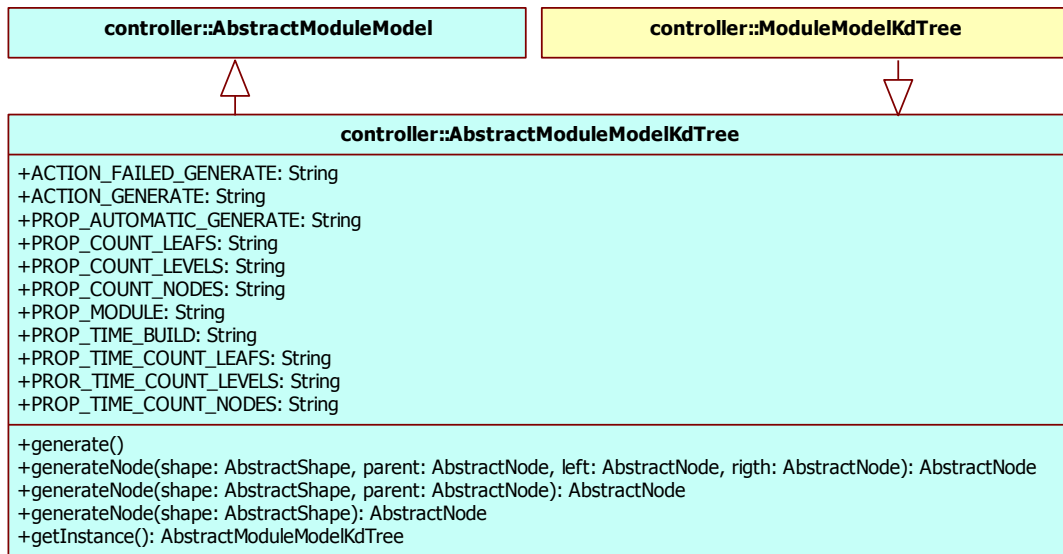


Ilustración 23 Modelado del módulo del controlador de árbol kd

**Descripción**

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

Tabla 42 Modelado del módulo del controlador de árbol kd

5.8.5. Modelado del módulo del controlador de simulación

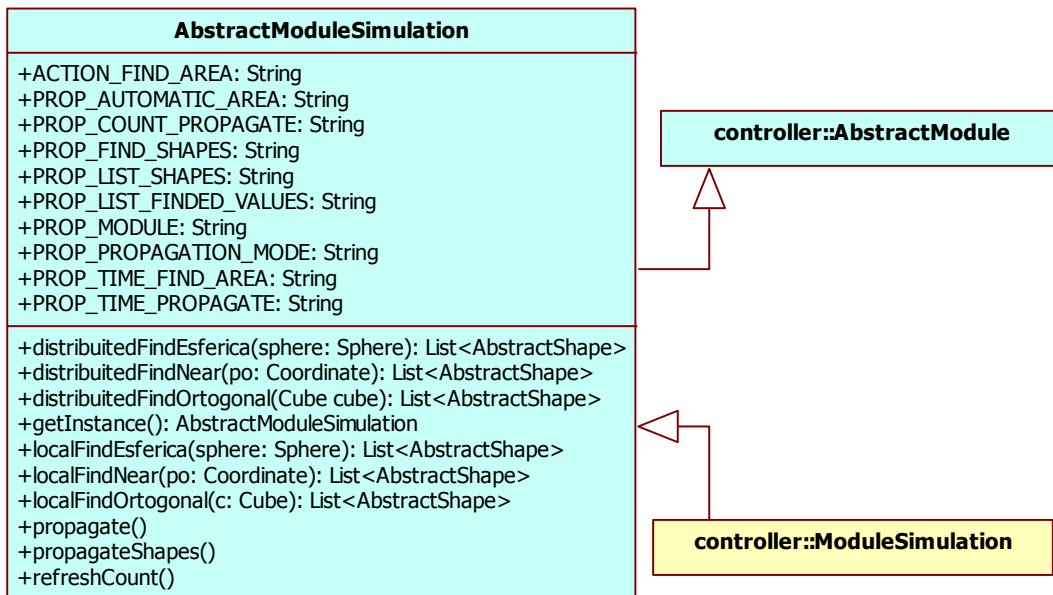


Ilustración 24 Modelado del módulo del controlador de simulación

**Descripción**

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

Tabla 43 Modelado del módulo del controlador de simulación

5.8.6. Modelado del módulo del controlador de monitorización

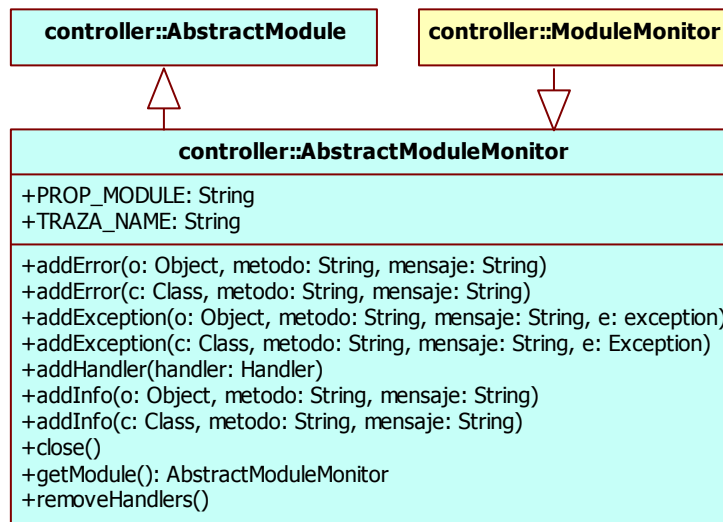


Ilustración 25 Modelado del módulo del controlador de monitorización

**Descripción**

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

Tabla 44 Modelado del módulo del controlador de monitorización

5.8.7. Modelado del módulo del controlador de tareas

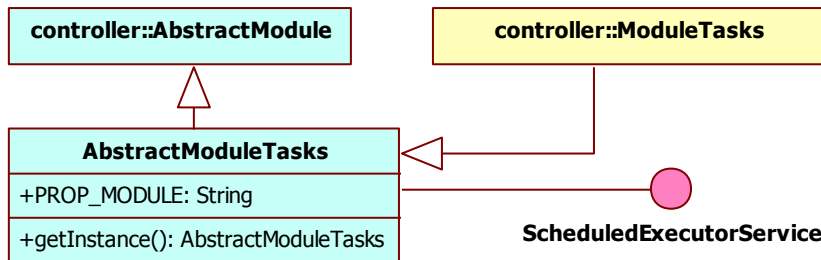


Ilustración 26 Modelado del módulo del controlador de tareas

**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 45 Modelado del módulo del controlador de tareas

Para más detalles ver los anexos – **Análisis**

### 5.9. Modelado de la Vista

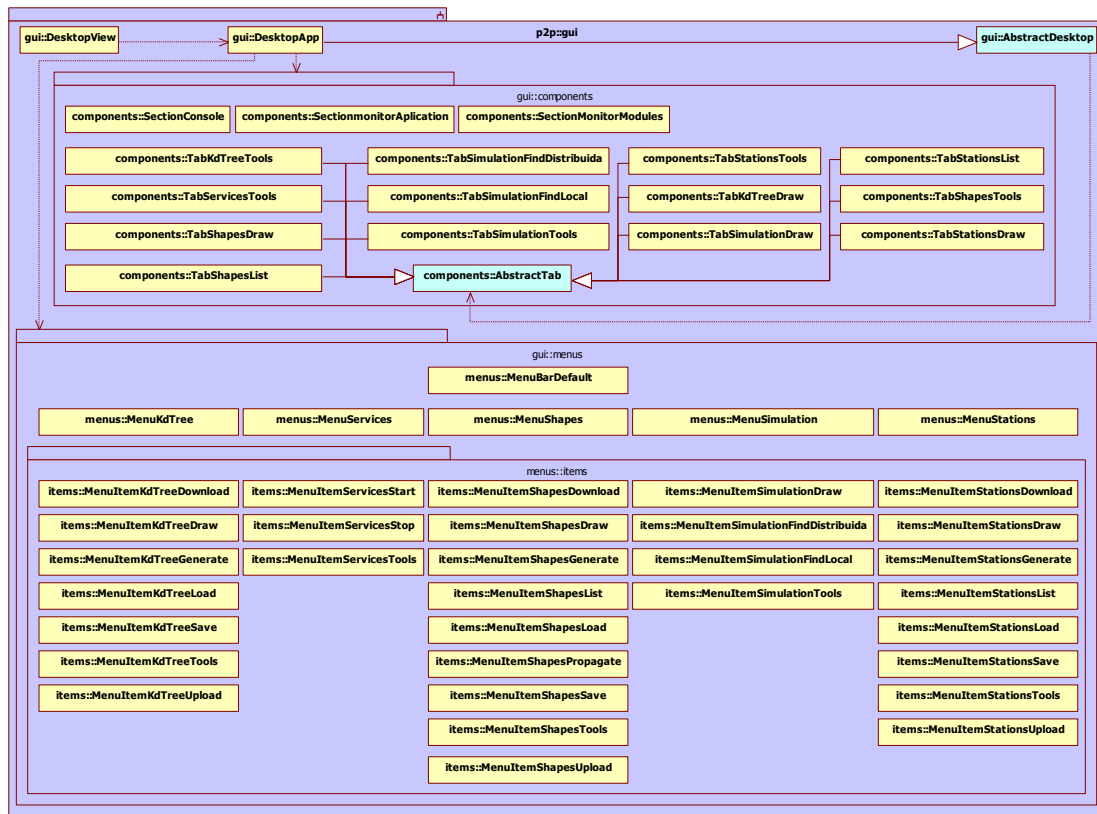


Ilustración 27 Modelado de la Vista

#### 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 46 Modelado de la Vista

*Para más detalles ver los anexos – Análisis*



## Capítulo 6. ESTUDIO DE LOS RESULTADOS

### 6.1. Preámbulo

Para evaluar el sistema, habremos de calcular el sistema en diferentes situaciones. Aquí se resume el resultado de llevar a cabo dichos estudios.

### 6.2. Objeto de estudio

El objeto de estudio es el rendimiento del sistema en un ámbito distribuido, para ello se realizaran los experimentos en un sistema sin distribuir para posteriormente realizar las mismas en un sistema distribuido.

### 6.3. Tipos de experimentos llevados a cabo

Los experimentos llevados a cabo son:

#### Tipos de Experimentos

La búsqueda de elementos contenidos en un cubo ortogonal.

La búsqueda de los elementos contenidos en una esfera.

La búsqueda del elemento más cercano a una coordenada.

Tabla 47 Tipos de Experimentos

### 6.4. Procedimiento llevado en los experimentos

Con el fin de analizar el rendimiento medio de los experimentos, se procederá de la siguiente forma.

Por cada experimento se evaluarán distintos tamaños de muestra.

0	500	1000	1500	2000	2500	3000
3500	4000	4500	5000	5500	6000	6500
7000	7500	8000	8500	9000	9500	10000

Tabla 48 tamaños de muestra

El experimento se repetirá un total de 100 veces con muestras generadas aleatoriamente por cada una de las repeticiones.

Así se realizarán 100 experimentos de búsqueda en una esfera con un tamaño de muestra de 500 elementos y cada uno de los experimentos se regenerará el espacio de muestras.

Con los resultados de la experimentación con estos distintos tamaños de muestras obtendremos el comportamiento de los algoritmos.

El uso de un tamaño de muestra nulo, nos permitirá calcular el tiempo base de las distintas configuraciones, permitiendo tener una apreciación de los tiempos perdidos en operaciones distintas del cálculo de los algoritmos de búsqueda.

### 6.5. Modo de generación de las muestras

De los tres modos de generación después de unas pruebas iniciales calculando los tiempos base de las tres posibilidades de generación:

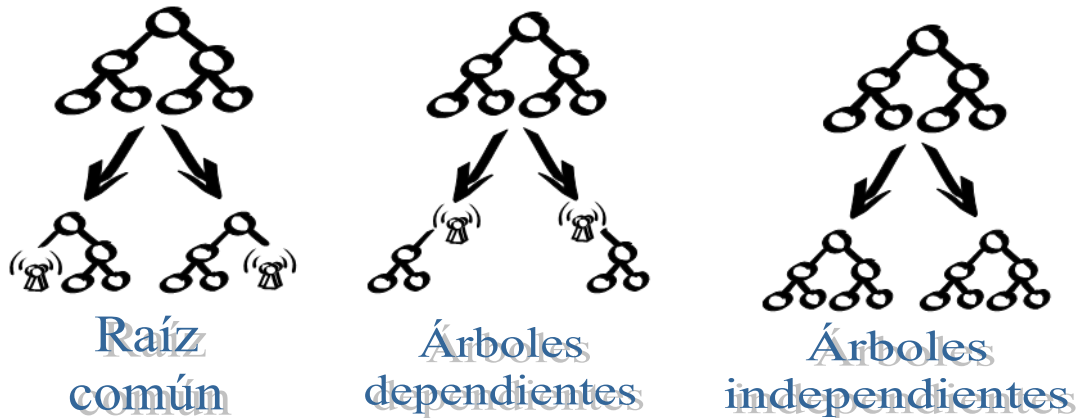


Ilustración 28 Opciones de generación

El modo de generación elegido ha sido el de generar arboles independientes, esto es debido a que los tiempos de red son costosos. Si eligiéramos cualquiera de las otras dos opciones las operaciones de red se llevarían a niveles que no justificarían la elección.

El uso de las otras opciones tiene sentido si quisiéramos que cada nodo analizara un espacio concreto del espacio. Por ejemplo imaginemos un mundo virtual, cada ser de ese mundo es consciente solo de su alrededor de su mundo. A la hora de visualizar esa parte del mundo con sus datos podría generar la imagen de su mundo. Como mucho si tuviera que ver a la distancia pediría datos a un nodo de dicha área, que con probabilidad contendría los datos que necesitaría. Se podría decir que el árbol está repartido a las estaciones según su geolocalización dentro de ese mundo.

Dado que lo que queremos evaluar es su rendimiento sobre ciertos algoritmos estos otros modos de reparto del árbol no tienen cabida.



## 6.6. Entorno de pruebas

En nuestro entorno de pruebas vamos a realizar las pruebas sobre dos estaciones en red, como distribuidor un ROUTER WIFI.



Ilustración 29 Entorno de pruebas

### ROUTER

Cable Modem XDSL Wireless 802.11 b/g/Turbo G.

Tabla 49 Router

### Estación A (LOCAL)

Sistema operativo	Windows Vista Home Premiun sp2 32 bits
Memoria RAM	4 GB
Procesador	Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz 2.32GHz
Adaptador de red	D-Link DWL-G122 Adapter Wifi USB 54

Tabla 50 Maquina A (Local)

### Estación B (REMOTA)

Sistema operativo	Windows 7 32 bits
Memoria RAM	4 GB
Procesador	AMD Turion (tm) 64 X2 Mobile Technology TL-64 2.20 GHz
Adaptador de red	Broadcom 4321AG 802.11a/b/g/draft-n Wi-Fi Adapter

Tabla 51 Maquina B (Remota)

La estación A será la que ejecute las pruebas, con lo que en las llamadas el enrutamiento para las pruebas distribuidas será a nivel de la propia maquina. La estación B sufrirá los retrasos de enrutamiento por el ROUTER.

## 6.7. Configuraciones de las pruebas

### 6.7.1. Estación A sin uso de red

En esta configuración, las pruebas no hacen uso alguno de protocolos de red. Es una simulación en la cual una estación procesa las muestras teniendo en local el árbol.



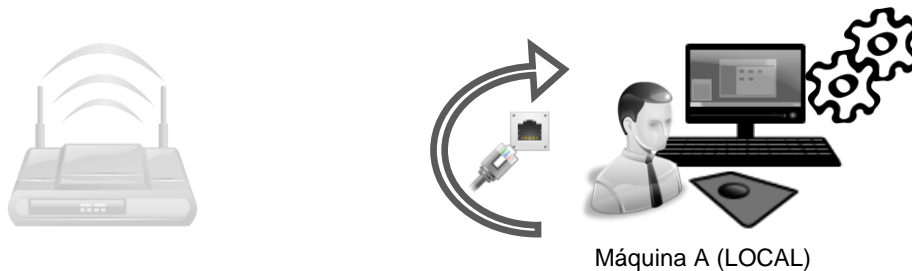
Estación A (LOCAL)

Ilustración 30 Estación local

Esta medida nos da el tiempo perdido en cálculo por dicha estación, sin el perjuicio de las transacciones de red.

### 6.7.2. Estación A con uso de protocolos de red en configuración local

En esta configuración, las pruebas hacen uso de protocolos de red, pero en su enrutamiento no pasara por el ROUTER, ya que el sistema identifica la dirección de la estación como local y la direcciona a sí mismo. Es una simulación en la cual una estación procesa las muestras teniendo en local el árbol.



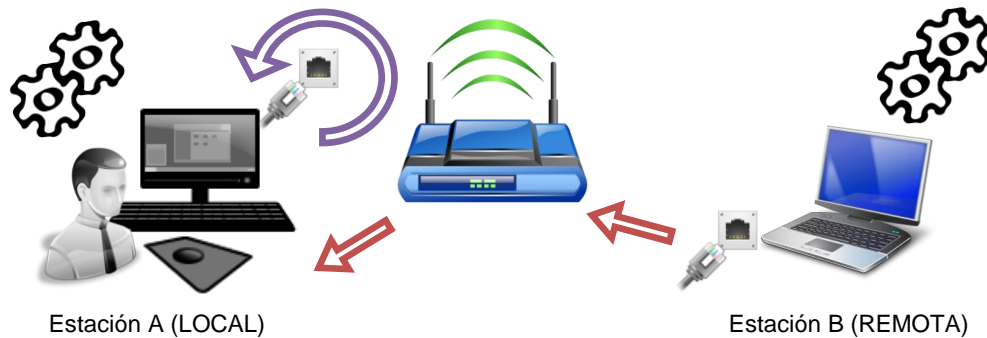
Máquina A (LOCAL)

Ilustración 31 Estación A con uso de protocolos de red en configuración local

Esta medida nos da el tiempo perdido en cálculo en dicha estación más el tiempo perdido en protocolos de red. Con los resultados de la configuración anterior podremos sacar el tiempo que se pierde en protocolos de red.

### 6.7.3. Estación A con uso de protocolos de red en configuración local y la estación B con uso de protocolos de red en configuración remota.

En esta configuración el árbol está repartido en dos estaciones. Ambas estaciones harán uso de los protocolos de red. La estación A estará en una configuración local, por lo que en su enrutamiento no hará uso del ROUTER. Por otro lado, la estación B estará en una configuración remota, por lo que si hará uso en el enrutamiento del RUTER.



**Ilustración 32 Estación A con uso de protocolos de red en configuración local y la estación B con uso de protocolos de red en configuración remota**

Esta medida nos dará el tiempo que se tarda en la simulación en total, así como el tiempo que tarda tanto la maquina A como la maquina B en sus enrutamientos. Con estos datos y los de las configuraciones anteriores podremos calcular el tiempo que se pierde en el dispositivo enrutador.

## 6.8. Resultados de los experimentos

### 6.8.1. 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

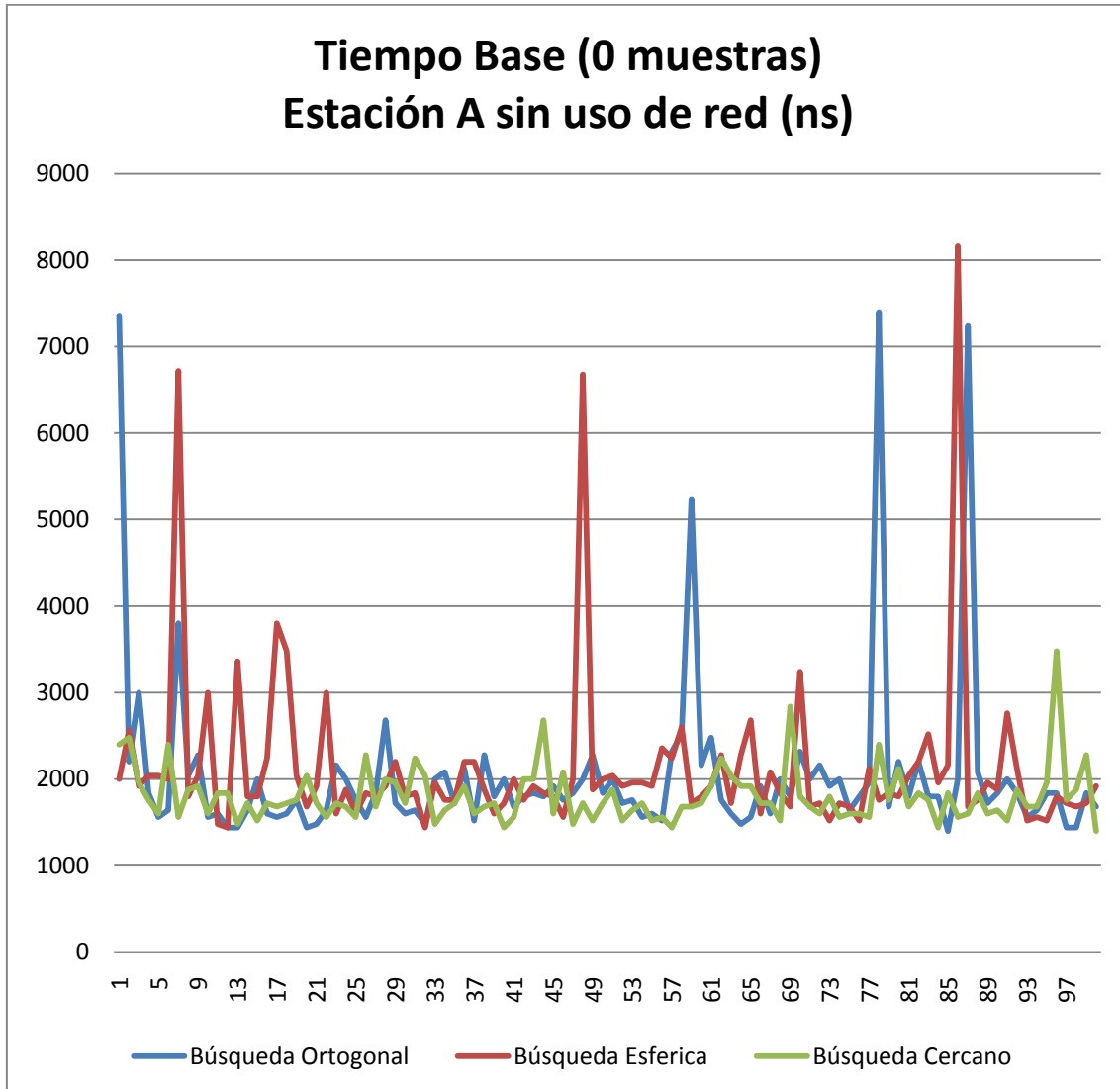


Ilustración 33 Tiempo Base Estación A sin uso de red

#### 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 52 Tiempo Base Estación A sin uso de red

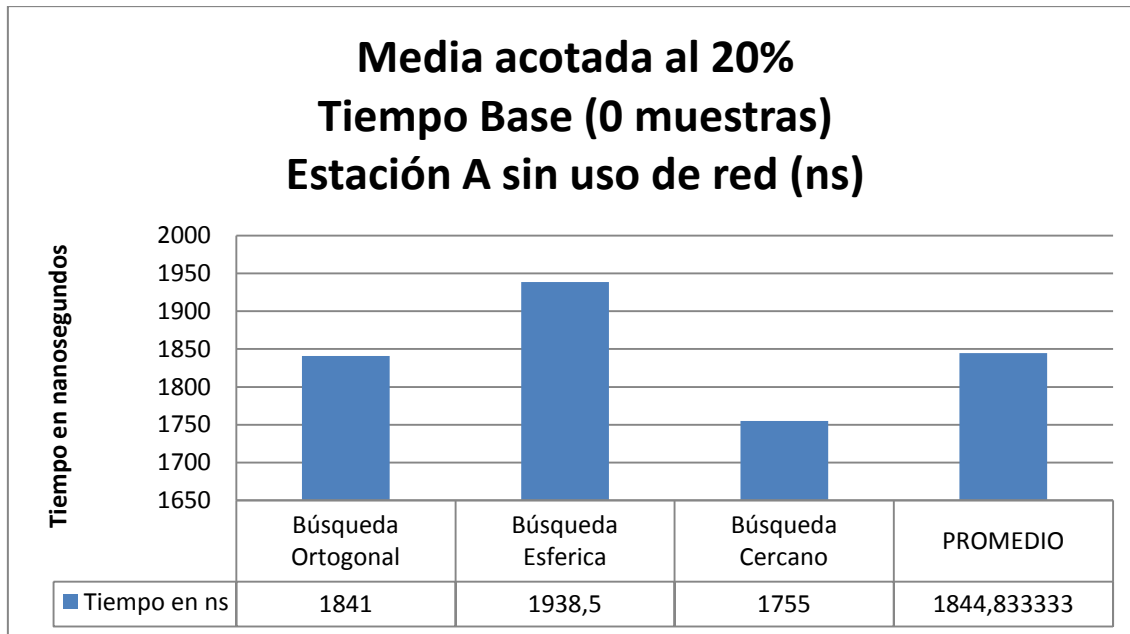
*Análisis del cálculo del tiempo base de la estación A sin uso de red*

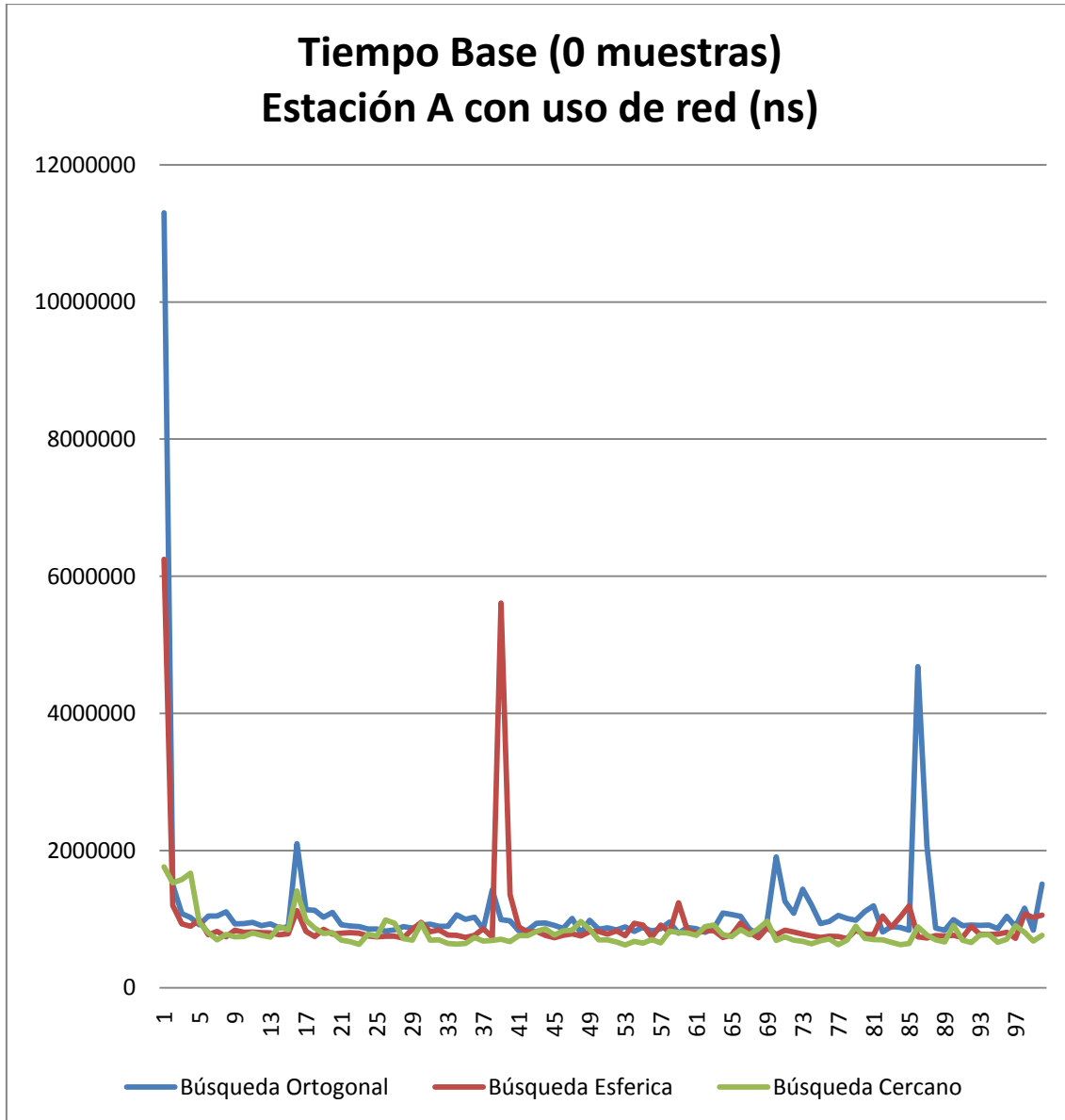
Ilustración 34 Análisis del cálculo del tiempo base de la estación A sin uso de red

**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 53 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*



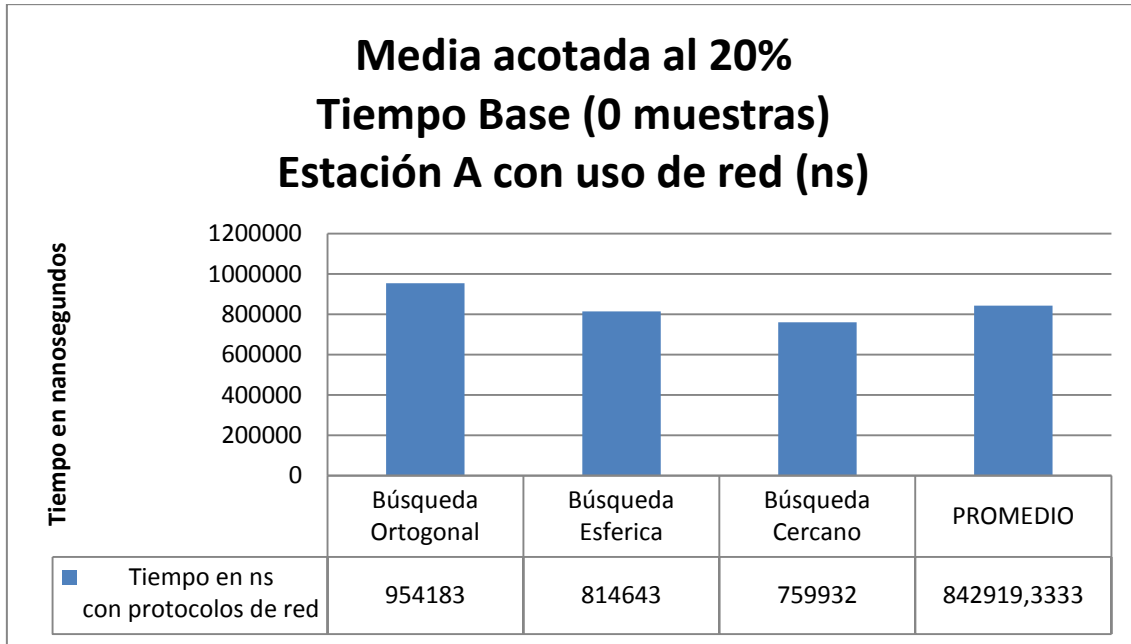
**Ilustración 35** Tiempo base de la estación A con uso de red en local

**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 a de configurar los protocolos de red, en un entorno de uso intensivo es de esperar que se llegue a un comportamiento estable de la red.

**Tabla 54** 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*



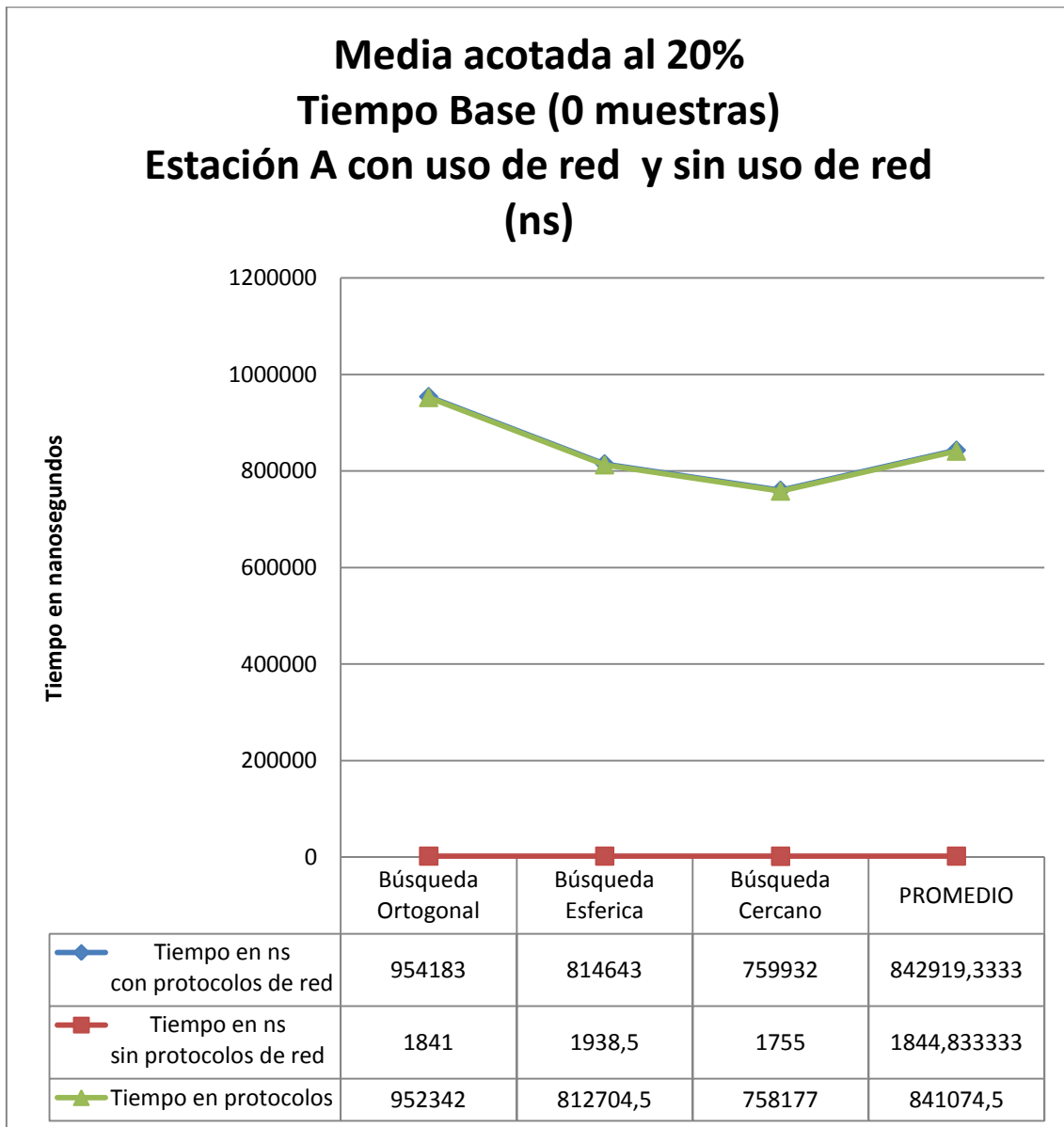
**Ilustración 36** Tiempo base de la estación A con uso de red en local

**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 55** 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 37** 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.

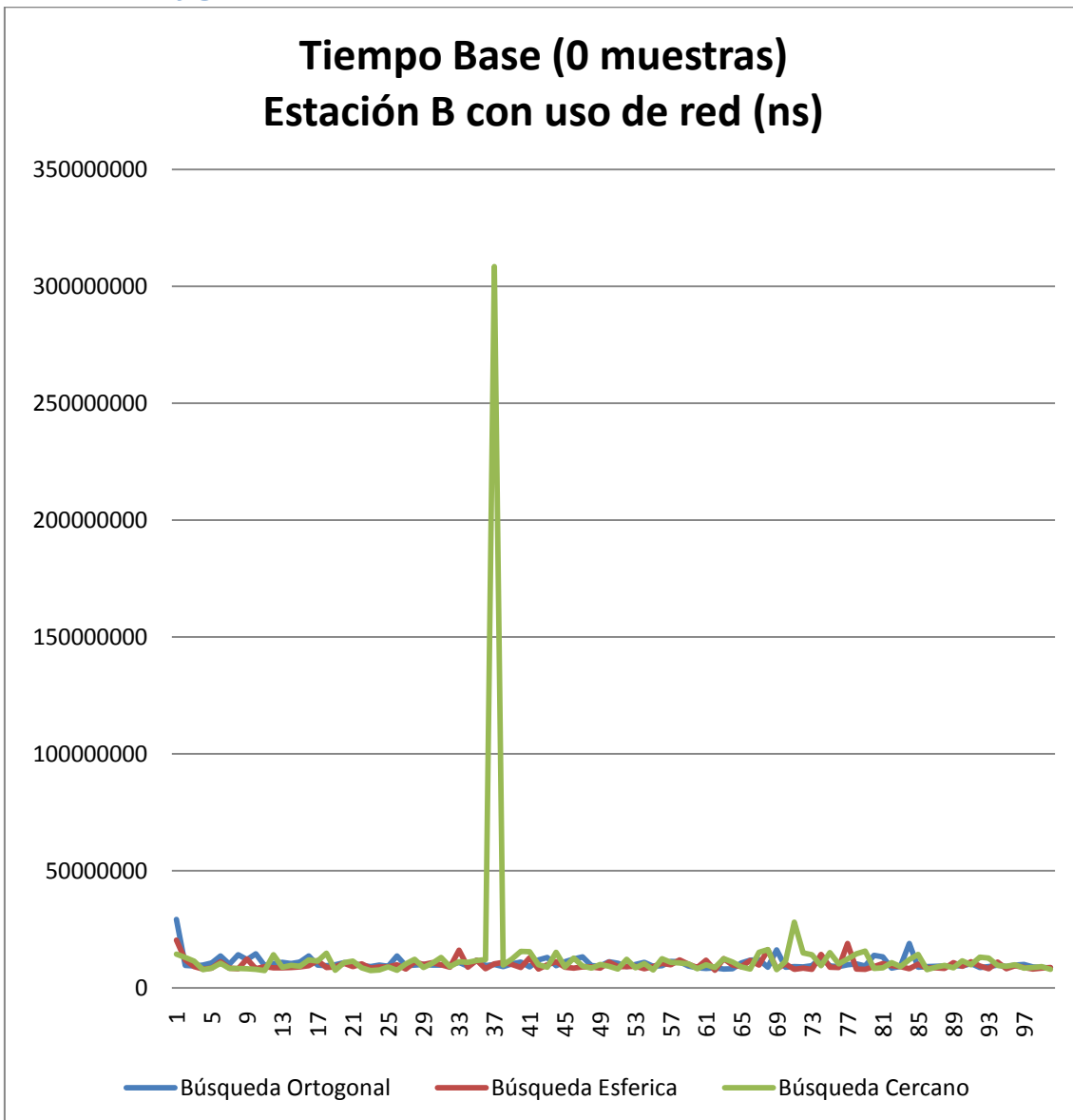
$$\begin{array}{r} 842919,3333 \text{ ns} \\ - 1844,8333 \text{ ns} \\ \hline 841074,499967 \text{ ns} \end{array}$$

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 56** 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*



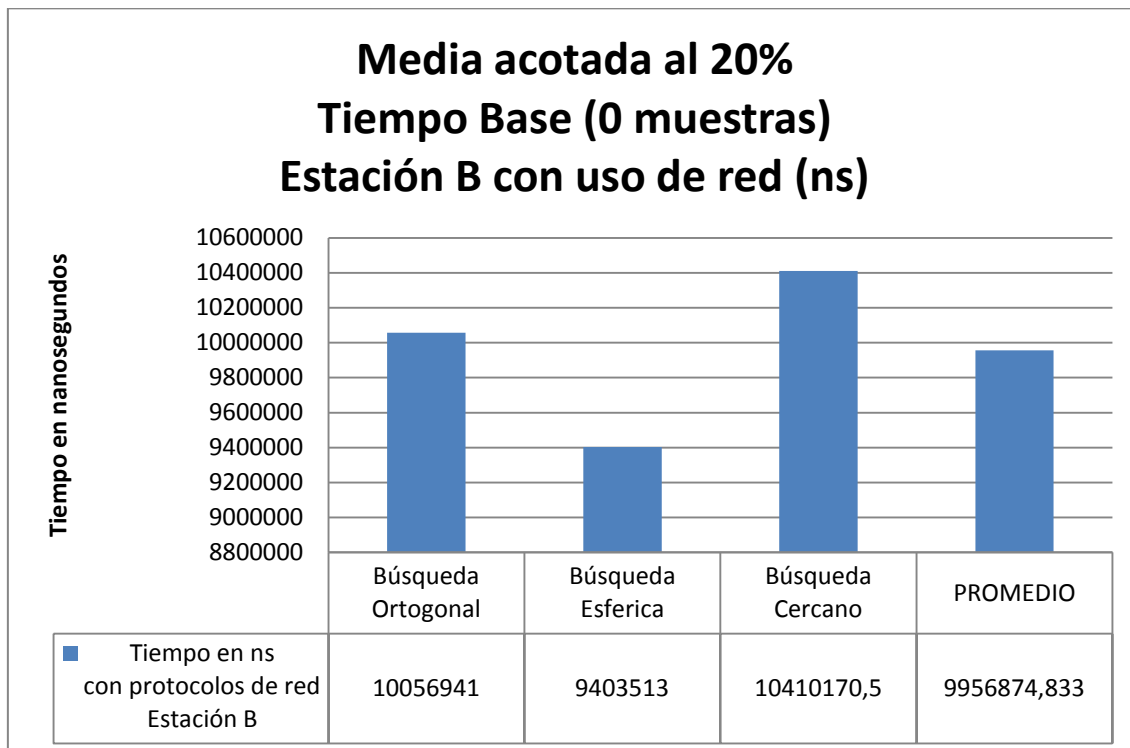
**Ilustración 38** Tiempo base de la estación B con uso de protocolos de red en configuración remota

**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 57** 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*



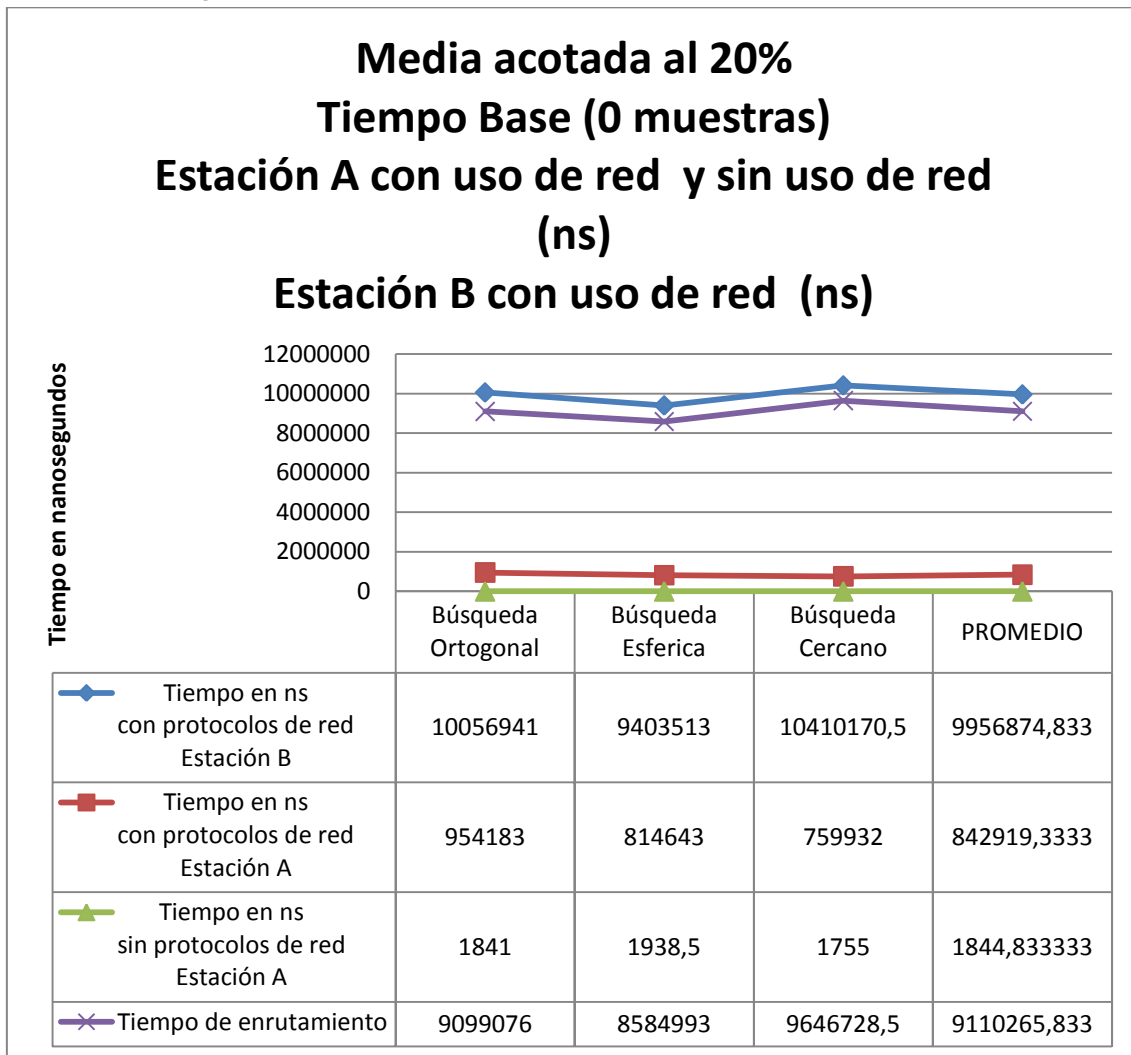
**Ilustración 39** Tiempo base de la estación B con uso de protocolos de red en configuración remota

### 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 58** 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 40** 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**

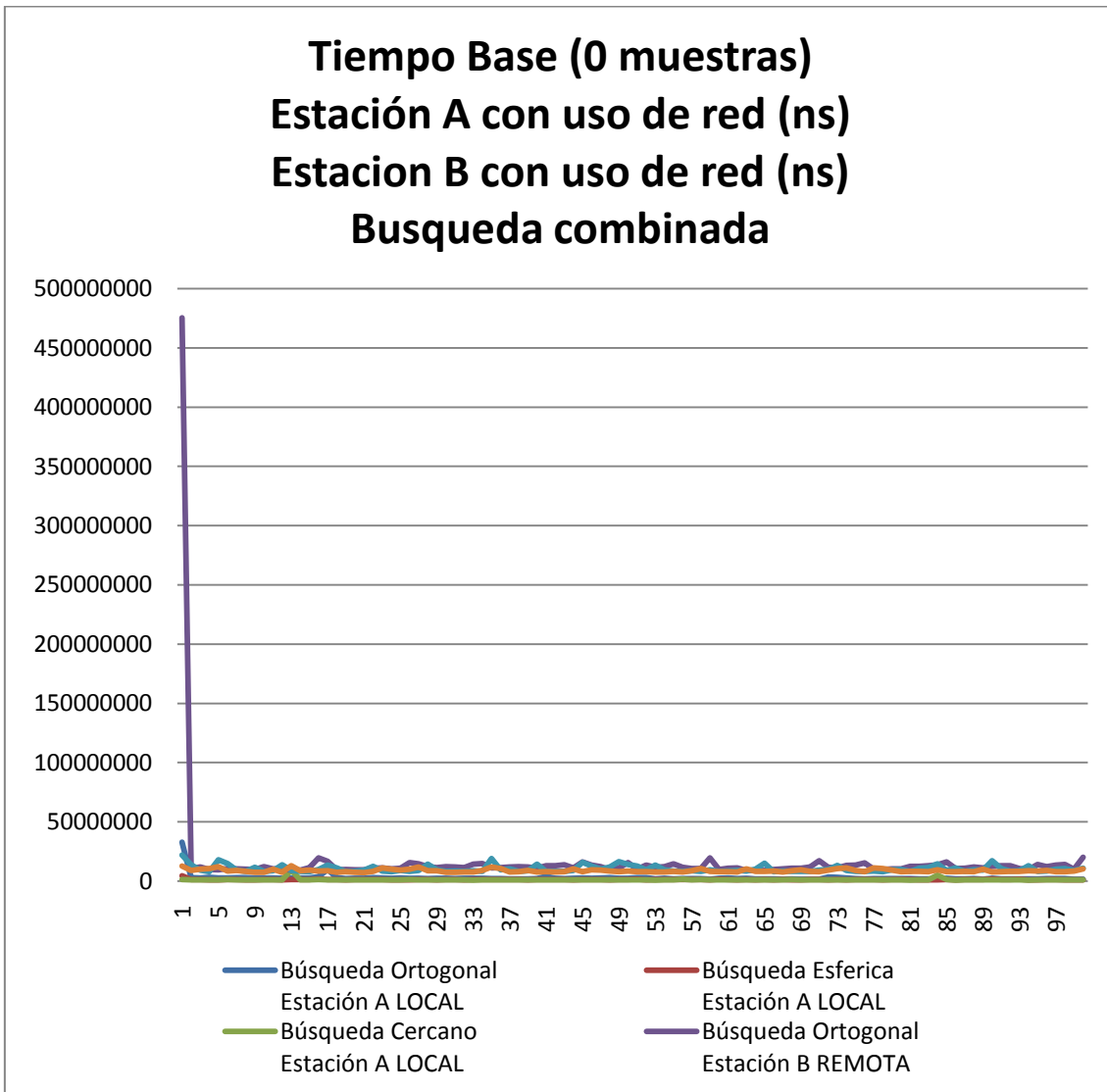
Si comparamos los tiempos anteriores descubriremos que el tiempo perdido en protocolos de red es bastante elevado.

$$\begin{array}{r}
 9956874,833 \text{ ns} \\
 - 842919,3333 \text{ ns} \\
 + \quad \quad \quad \\
 -1844,833333 \text{ ns} \\
 -1844,833333 \text{ ns} \\
 \hline
 9110265,833 \text{ ns}
 \end{array}$$

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

**Tabla 59** 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*



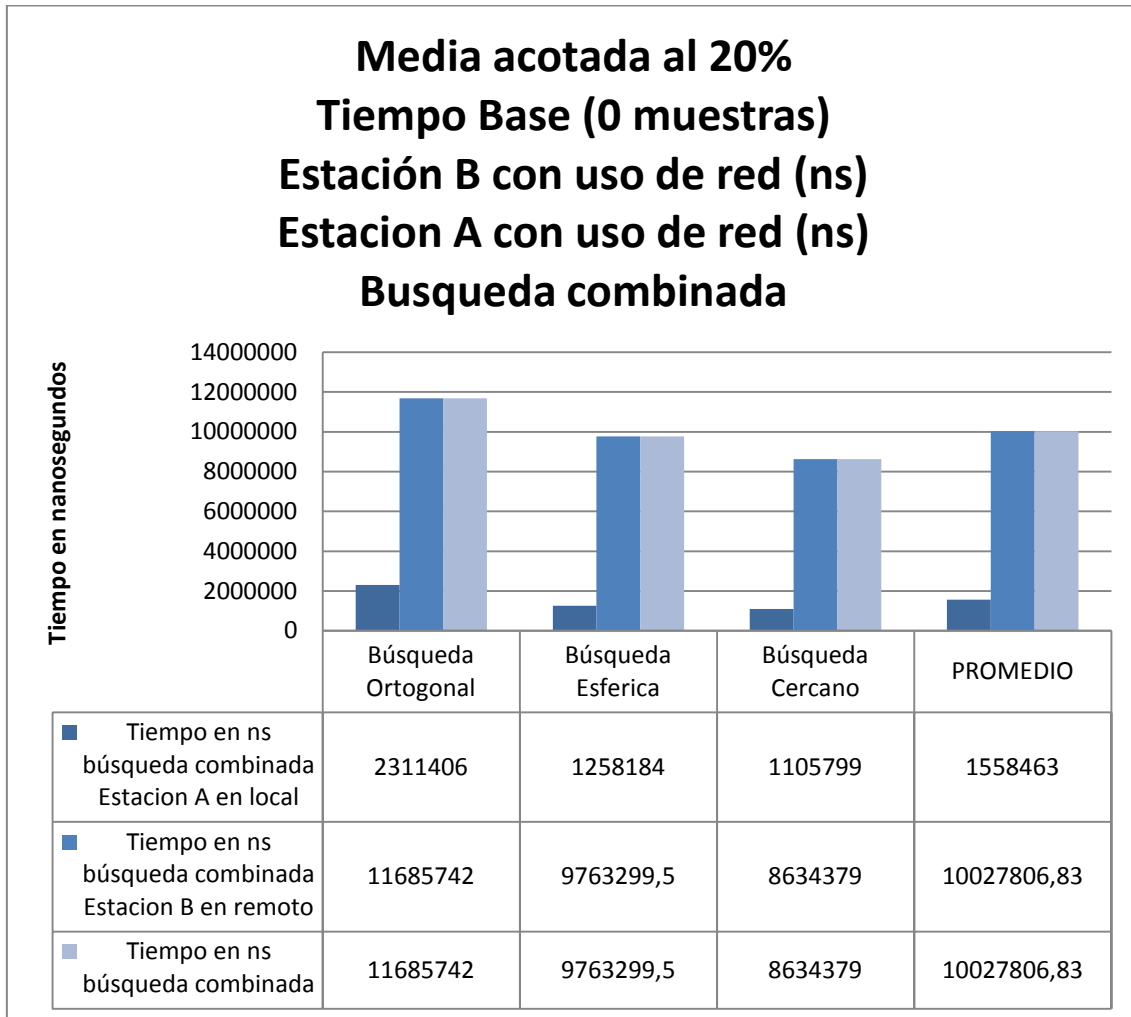
**Ilustración 41** Tiempo base de la estación B con uso de protocolos de red en configuración local y estación B con uso de protocolos de red en configuración remota en búsqueda combinada

**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 60** 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 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

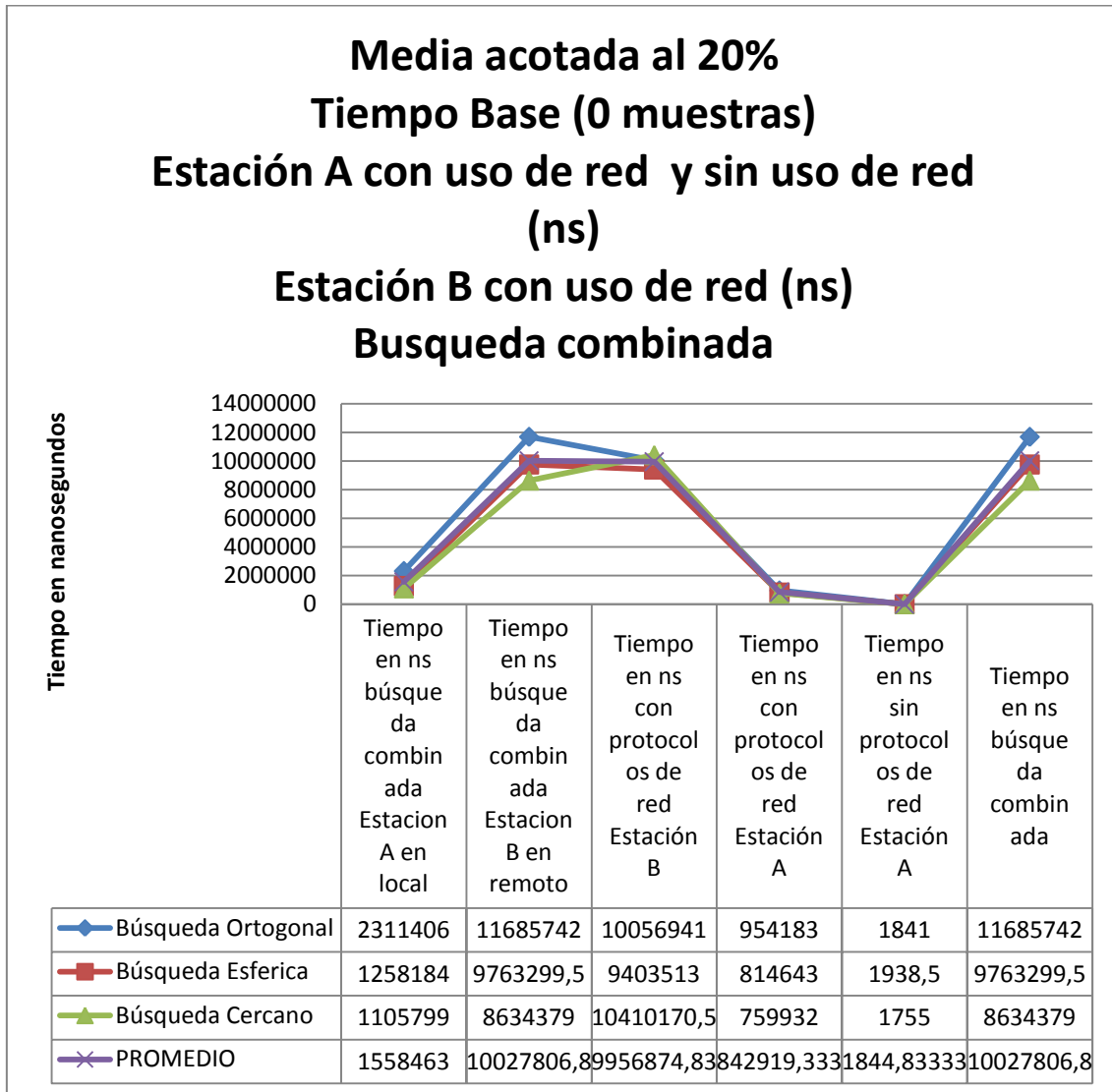
**Descripción**

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.

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

**Tabla 61** 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 43** 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 62** 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*

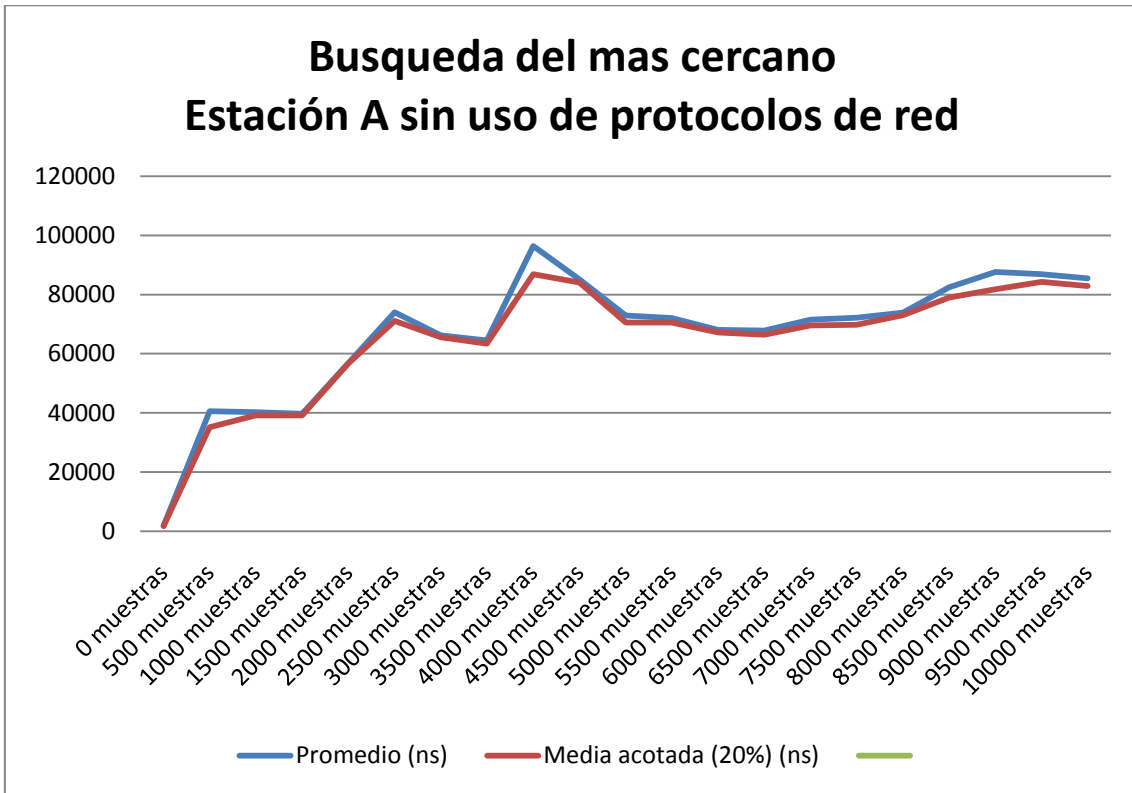


Ilustración 44 Análisis del algoritmo del más cercano

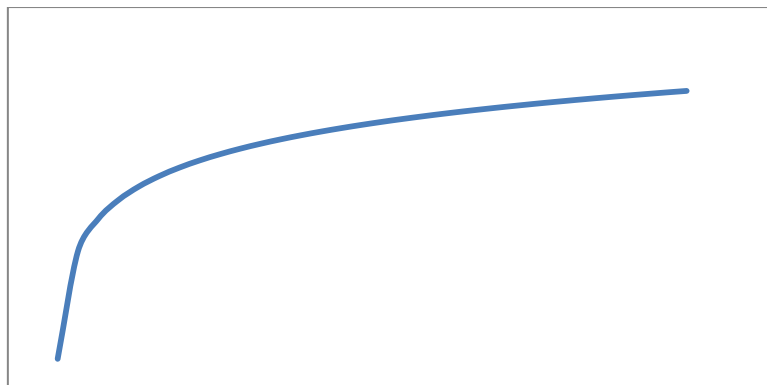


Ilustración Tendencia de la simulación O(Log N)

**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 63 Simulación local de la búsqueda del más cercano Máquina 1

*Análisis del algoritmo de búsqueda ortogonal*

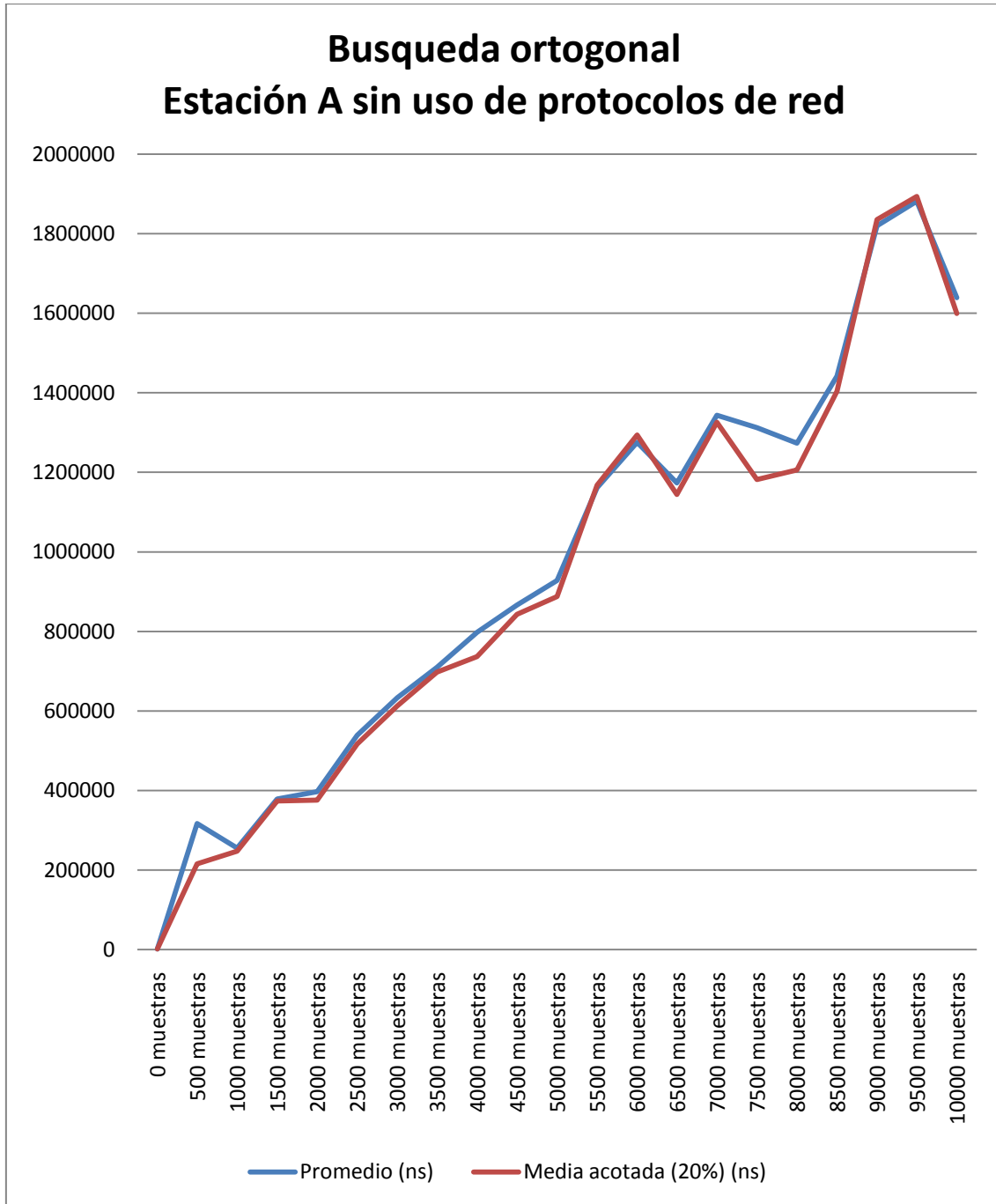


Ilustración 45 Análisis del algoritmo de búsqueda ortogonal

**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 64 Análisis del algoritmo de búsqueda ortogonal



Análisis del algoritmo de búsqueda esférica

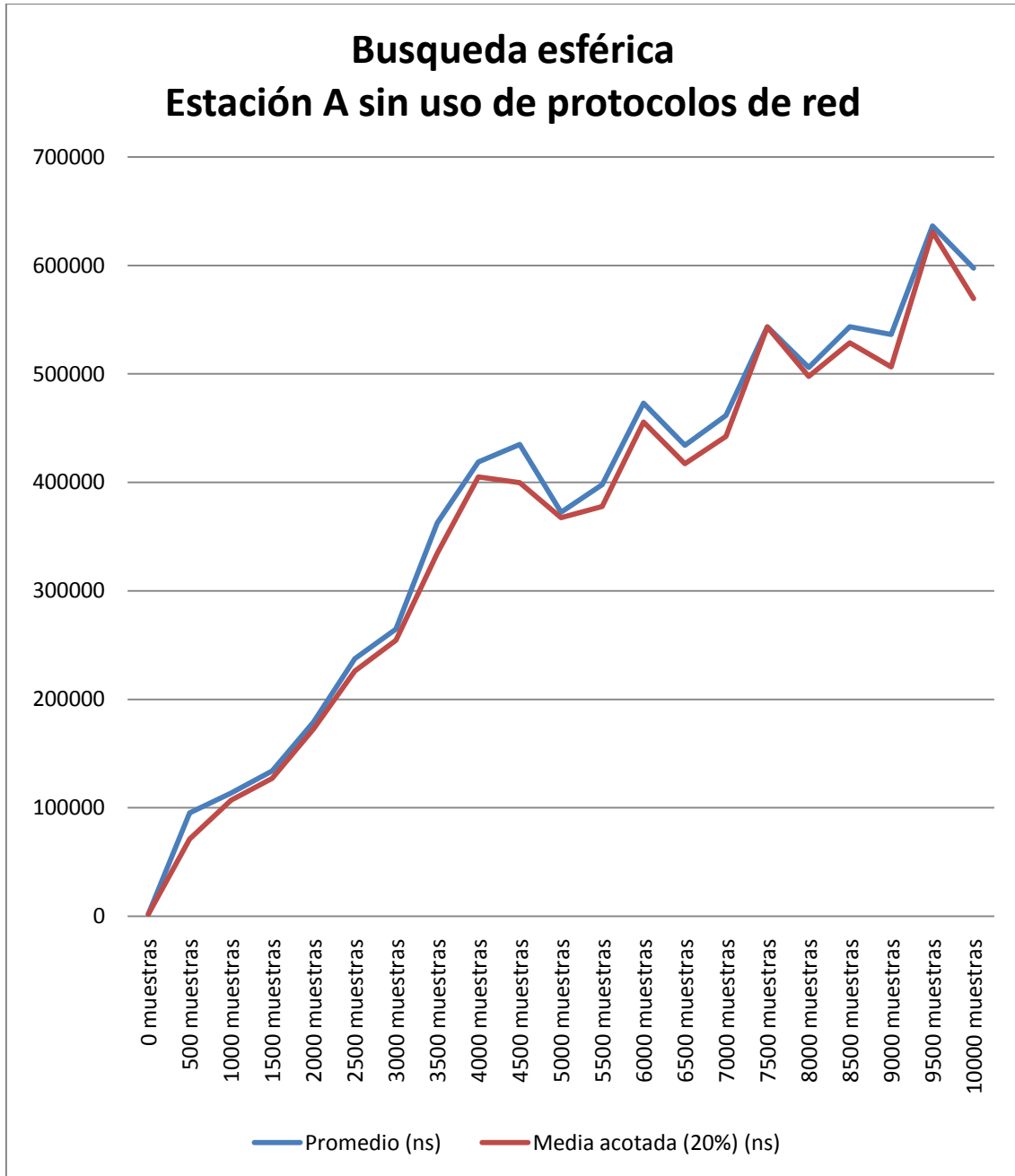


Ilustración 46 Análisis del algoritmo de búsqueda esférica

**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 65 Análisis del algoritmo de búsqueda esférica



## Capítulo 7. CONCLUSIONES

### 7.1. Preámbulo

En este capítulo se hará un análisis y resumen de las conclusiones del presente proyecto.

### 7.2. Sobre la Planificación

La planificación inicial fue un completo fracaso, surgieron una gran cantidad de imprevistos que desbarajustaron los tiempos y las tareas planificadas inicialmente.

Afortunadamente el modelo de planificación del tipo cíclico de vida en espiral permitió ir adaptándonos a los imprevistos surgidos.

### 7.3. Sobre la solución

Se ha comprobado la viabilidad real de una solución de render distribuida y descentralizada.

Se ha conseguido el distribuir en el espacio la base de datos de las facetas, la cual está contenida entre las estaciones de red. De esta manera, la solución permite escalar tanto como se quiera. El límite del tamaño de la escena está limitado el número de estaciones de render y su almacenamiento individual. Esto significa que contra mas estaciones se tengan, la escena podrá contener un mayor número de muestras. En la práctica como nuestras potenciales estaciones de render son cualquier dispositivo capaz de ejecutar la aplicación significa que no tenemos limitación de espacio de almacenamiento.

La replicación de trozos de la escena repartidos entre las estaciones de render, nos permite tener una gran tolerancia de fallos. Se puede desconectar un nodo en mitad de la simulación y los resultados con una gran probabilidad no serán afectados, pudiendo concluirse el proceso sin mayores problemas.

El intercambio de las facetas está restringido a mantener la duplicidad de datos y devolver los resultados.

Se ha observado que las comunicaciones de red consumen una gran cantidad del tiempo de las experimentaciones.

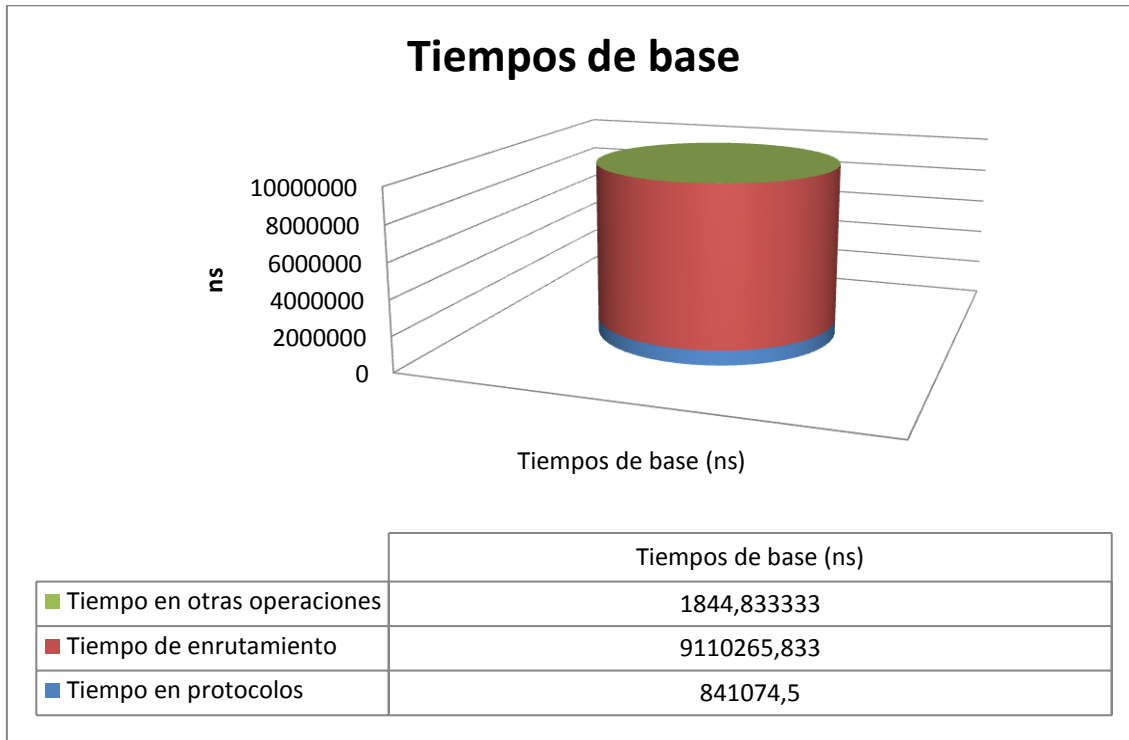


Tabla 66 Tiempos base

En una solución profesional estas transacciones de red se ven incrementadas frente a una solución distribuida del árbol. Esto es debido a que en una solución profesional existen las transacciones de red para la obtención de los datos que alimentan las operaciones, los cuales son obtenidos de un servidor central. Al tener almacenados los datos en cada nodo estas transacciones en una solución distribuida no existen. En cuanto a las transacciones de mantenimiento de la red rápidamente alcanzan un punto estable que consume pocos recursos.

La carga de red es mínima al no haber puntos críticos en la red, en una solución profesional los puntos críticos son sobre todo el servidor de datos y el servidor que se encarga de distribuir los mandatos de cálculo, pero en la solución estas responsabilidades están repartidas entre las estaciones. Según se configuren pueden derivar tareas de gestión entre otras estaciones de render.

La saturación de la memoria por parte de las facetas esta acotada. Se han encontrado problemas para en una misma estación de render procesar grandes cantidades de datos, esto es debido a tener que implementar gran cantidad de observadores y otras operaciones de seguimiento de los cálculos. Los cálculos, son necesarios con el fin de poder realizar un seguimiento exhaustivo de los tiempos de proceso afín de reunir información para su posterior análisis.

## 7.4. Trabajo Futuro

Como trabajo futuro queda:

---

### Trabajo Futuro

La implementación de un sistema de gestión y monitoreo Web.

La implantación de un sistema de cálculo de colisiones utilizando la estructura estudiada.

La implementación de un modulo que adapte el sistema a un cálculo distribuido en los navegadores Web.

El reparto mediante geolocalización del árbol de render a un dispositivo móvil.

---

Tabla 67 Trabajo Futuro

## 7.5. Aplicación del PFC al mundo real

Las aplicaciones del proyecto al mundo real son bien claras. Cualquier estudio famoso podría utilizar su estructura interna de ordenadores sin colapsar las comunicaciones, al ser estas descentralizadas, evitando el cuello de botella de una red centralizada. Además al estar la información descentralizada con redundancia nos evitamos problemas de pérdidas de datos. Así mismo no hace falta el proveer al servidor central de una gran capacidad de red y almacenamiento, al no existir este.

## 7.6. Valoración personal y conclusión final

He comprendido que el cálculo distribuido tiene una gran potencial. Una solución de cálculo mediante una red P2P tiene grandes posibilidades. El reparto del árbol entre distintas estaciones nos permitiría indexar escenas de un increíble tamaño. La realización del proyecto me ha enseñado tecnologías que desconocía como el mundo de Java 2D. También me ha permitido profundizar en la geometría espacial más de lo que me hubiera imaginado.



## Capítulo 8. BIBLIOGRAFÍA

### 8.1. Lecturas

- Redes entre Pares  
Alejandro Alons - Dpto. Ing. de Sistemas Telemático  
Universidad Politécnica de Madrid
- Distribución de Información en Algoritmos  
Evolutivos P2P  
M.G. Arenas, P.A. Castillo, G. Romero, J.J. Merelo
- Comparación entre métodos de acceso espaciales y  
de puntos multidimensionales  
Isabel Besembel y Stuart Roberts®  
.Grupo de Ingeniería de Datos y Conocimiento (GIDyC)  
Departamento de Computación  
Escuela de Ingeniería de Sistemas  
Universidad de Los Andes. Mérida-Venezuela  
School of Computer Studies  
University of Leeds. Leeds-U.K.
- Optimización del Cálculo de Colisiones para Mallas  
Deformables mediante Voxelización de Primitivas  
J. Gascón, M.A. Otaduy, J.M. Espadero, A. Rodríguez  
Universidad Rey Juan Carlos (URJC)  
Universidad Politécnica de Madrid (UPM)
- Geometría métrica y descriptiva:  
Ejercicios resueltos y comentados en el sistema de planos acotados  
Juan Auñón López, José A. Ferri Aranda  
Ed. Univ. Politéc. Valencia, 2002
- Geometría  
Ana Helvia Quintero  
Editorial de la Universidad de Puerto Rico 1994
- Geometría en el plano y en el espacio  
Ana Berenice Guerrero G.
- Geometría Euclídea. Volumen I y II  
San José, Costa Rica 1997
- Compendio de Matemática  
Escrito por Placencia Valero, Job
- Fórmulas y propiedades geométricas:  
Geometría espacial  
Fernando Izquierdo Asensio

## 8.2. Otros enlaces de interés



Ilustración 47 Web de Wikipedia



Ilustración 48 Web de Worldingo



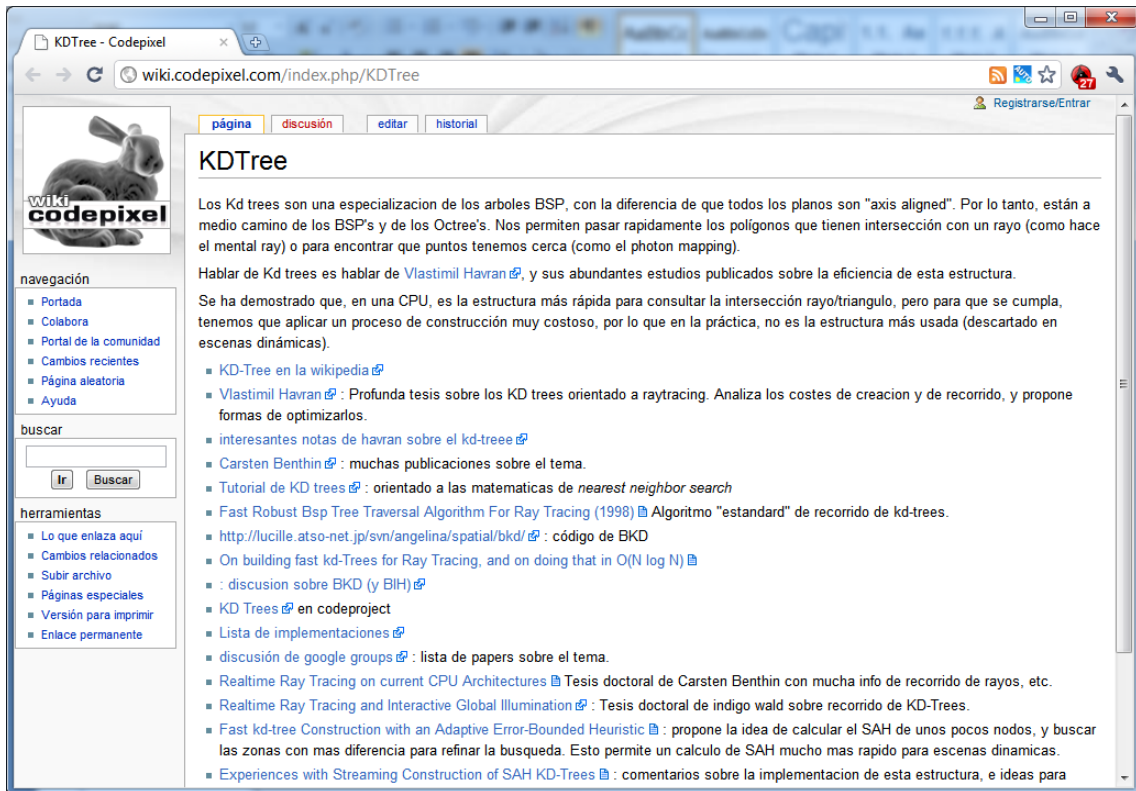


Ilustración 49 Web de CodePixel

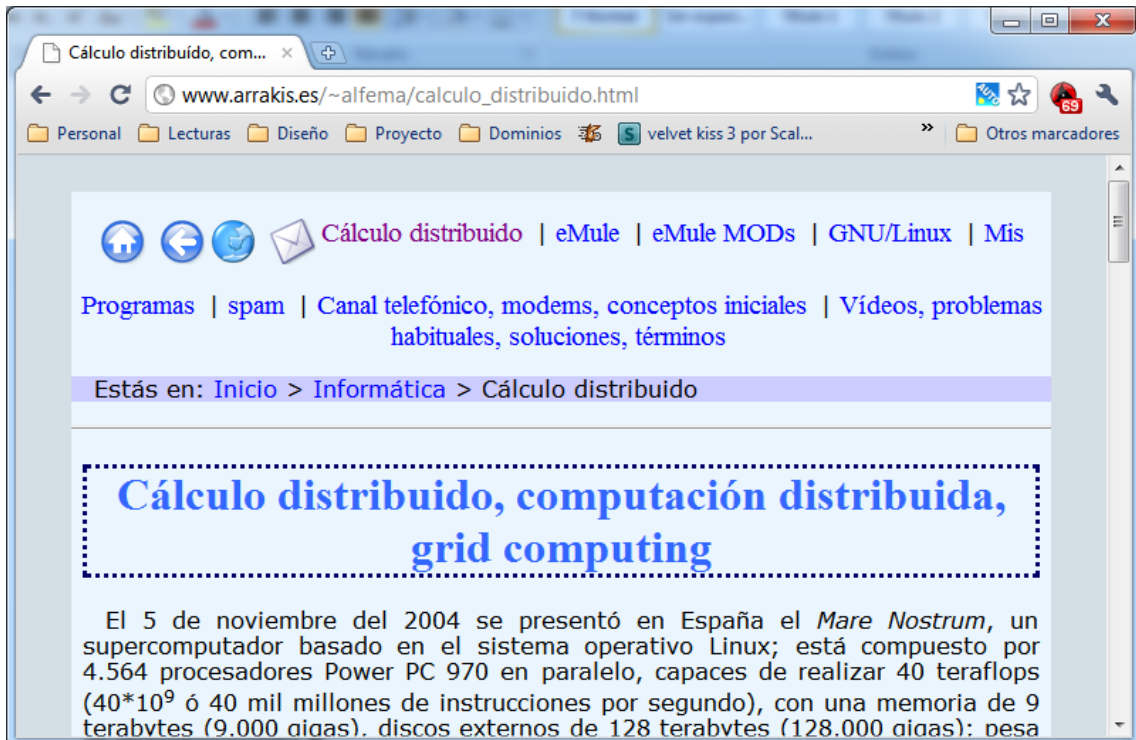


Ilustración 50 Web de Arrakis



Ilustración 51 Web de Sargue

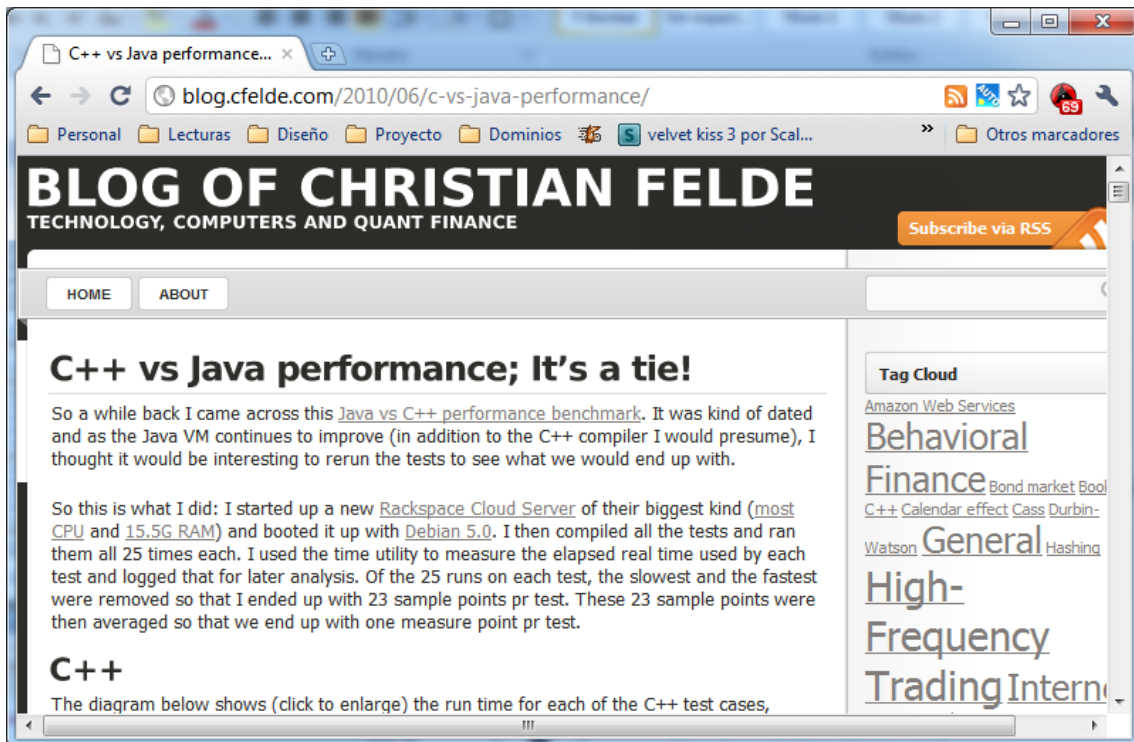


Ilustración 52 Web de cfelde



Ilustración 53 Web de cadstock



Ilustración 54 Web de CodePixel



Ilustración 55 Web de corej2eepatterns

# TABLAS E ILUSTRACIONES

## Índice de Tablas

<i>Tabla 1 Estructura de la Memoria</i> .....	3
<i>Tabla 2 Fases elegidas en el ciclo de vida</i> .....	5
<i>Tabla 3 Temáticas de riesgos</i> .....	6
<i>Tabla 4 limitaciones básicas de las granjas de render</i> .....	8
<i>Tabla 5 Tipos de Sistemas de almacenamiento distribuidos</i> .....	8
<i>Tabla 6 Ventajas de una base de datos distribuida</i> .....	9
<i>Tabla 7 Desventajas de una base de datos distribuida</i> .....	9
<i>Tabla 8 Algoritmos usados en los sistemas de render</i> .....	10
<i>Tabla 9 Ejemplo famosos de redes P2P</i> .....	11
<i>Tabla 10 Características deseables en una red P2P</i> .....	11
<i>Tabla 11 Tipos de redes P2P</i> .....	12
<i>Tabla 12 Diagrama del sistema</i> .....	13
<i>Tabla 13 Mantenimiento de la red</i> .....	14
<i>Tabla 14 Modelo vista controlador</i> .....	15
<i>Tabla 15 Características deseables para el interfaz de usuario</i> .....	16
<i>Tabla 16 El controlador</i> .....	17
<i>Tabla 17 Características deseables para los módulos del controlador</i> .....	17
<i>Tabla 18 Funcionalidades del controlador</i> .....	18
<i>Tabla 19 Funcionalidades de Sistema</i> .....	18
<i>Tabla 20 Funcionalidades de Servicios de red</i> .....	19
<i>Tabla 21 Funcionalidades relacionadas con las estaciones de render</i> .....	20

Tabla 22 Funcionalidades relacionadas con las formas .....	21
Tabla 23 Funcionalidades relacionadas con el árbol kd .....	22
Tabla 24 Funcionalidades de simulación .....	23
Tabla 25 El modelo .....	23
Tabla 26 Técnicas de generación y reparto del árbol kdtree.....	25
Tabla 27 Sistema de comunicación .....	27
Tabla 28 Tipos de interfaz .....	28
Tabla 29 Formatos de almacenamiento de los datos.....	29
Tabla 30 Lenguajes de programación evaluados .....	30
Tabla 31 Puntos a favor de java .....	30
Tabla 32 Puntos en contra de java .....	30
Tabla 33 Modelado de los datos del modelo.....	32
Tabla 34 Modelado del modelo.....	32
Tabla 35 Modelado del modelo de las estaciones de render.....	33
Tabla 36 Modelado del modelo del listado de formas .....	34
Tabla 37 Modelado del Árbol kd.....	35
Tabla 38 Modelado del Controlador.....	36
Tabla 39 Modelado del módulo del controlador de servicios.....	37
Tabla 40 Modelado del módulo del controlador del listado de estaciones de render.....	38
Tabla 41 Modelado del módulo del controlador de formas .....	38
Tabla 42 Modelado del módulo del controlador de árbol kd .....	39
Tabla 43 Modelado del módulo del controlador de simulación .....	39
Tabla 44 Modelado del módulo del controlador de monitorización.....	40
Tabla 45 Modelado del módulo del controlador de tareas .....	40
Tabla 46 Modelado de la Vista.....	41
Tabla 47 Tipos de Experimentos .....	43
Tabla 48 tamaños de muestra.....	43
Tabla 49 Router .....	45
Tabla 50 Maquina A (Local).....	45
Tabla 51 Maquina B (Remota).....	45
Tabla 52 Tiempo Base Estación A sin uso de red .....	48
Tabla 53 Análisis del cálculo del tiempo base de la estación A sin uso de red .....	49
Tabla 54 Tiempo base de la estación A con uso de red en local.....	50
Tabla 55 Tiempo base de la estación A con uso de red en local.....	51
Tabla 56 Tiempo base de la estación A con uso de red en local y sin uso de red .....	52
Tabla 57 Tiempo base de la estación B con uso de protocolos de red en configuración remota .....	53
Tabla 58 Tiempo base de la estación B con uso de protocolos de red en configuración remota .....	54
Tabla 59 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.....	55
Tabla 60 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 .....	56
Tabla 61 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 .....	57
Tabla 62 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 .....	58
Tabla 63 Simulación local de la búsqueda del más cercano Máquina 1.....	59
Tabla 64 Análisis del algoritmo de búsqueda ortogonal .....	60
Tabla 65 Análisis del algoritmo de búsqueda esférica.....	61
Tabla 66 Tiempos base .....	64
Tabla 67 Trabajo Futuro .....	65

## Índice de Ilustraciones

<i>Ilustración 1 Estimación inicial del proyecto</i> .....	5
<i>Ilustración 2 Diagrama del sistema</i> .....	13
<i>Ilustración 3 Mantenimiento de la red</i> .....	14
<i>Ilustración 4 Modelo vista controlador</i> .....	15
<i>Ilustración 5 La vista</i> .....	16
<i>Ilustración 6 El controlador</i> .....	17
<i>Ilustración 7 Funcionalidades de Sistema</i> .....	18
<i>Ilustración 8 Funcionalidades de Servicios de red</i> .....	19
<i>Ilustración 9 Funcionalidades relacionadas con las estaciones de render</i> .....	20
<i>Ilustración 10 Funcionalidades relacionadas con las formas</i> .....	21
<i>Ilustración 11 Funcionalidades relacionadas con el árbol kd</i> .....	22
<i>Ilustración 12 Funcionalidades de simulación</i> .....	23
<i>Ilustración 13 Técnicas de generación y reparto del árbol kdtree</i> .....	25
<i>Ilustración 14 Tres proyectos al más puro estilo MVC</i> .....	31
<i>Ilustración 15 Modelado del modelo</i> .....	32
<i>Ilustración 16 Modelado del modelo de las estaciones de render</i> .....	33
<i>Ilustración 17 Modelado del modelo del listado de formas</i> .....	34
<i>Ilustración 18 Modelado del Árbol kd</i> .....	35
<i>Ilustración 19 Modelado del Controlador</i> .....	36
<i>Ilustración 20 Modelado del módulo del controlador de servicios</i> .....	37
<i>Ilustración 21 Modelado del módulo del controlador del listado de estaciones de render</i> .....	38
<i>Ilustración 22 Modelado del módulo del controlador de formas</i> .....	38
<i>Ilustración 23 Modelado del módulo del controlador de árbol kd</i> .....	39
<i>Ilustración 24 Modelado del módulo del controlador de simulación</i> .....	39
<i>Ilustración 25 Modelado del módulo del controlador de monitorización</i> .....	40
<i>Ilustración 26 Modelado del módulo del controlador de tareas</i> .....	40
<i>Ilustración 27 Modelado de la Vista</i> .....	41
<i>Ilustración 28 Opciones de generación</i> .....	44
<i>Ilustración 29 Entorno de pruebas</i> .....	45
<i>Ilustración 30 Estación local</i> .....	46
<i>Ilustración 31 Estación A con uso de protocolos de red en configuración local</i> .....	46
<i>Ilustración 32 Estación A con uso de protocolos de red en configuración local y la estación B con uso de protocolos de red en configuración remota</i> .....	47
<i>Ilustración 33 Tiempo Base Estación A sin uso de red</i> .....	48
<i>Ilustración 34 Análisis del cálculo del tiempo base de la estación A sin uso de red</i> .....	49
<i>Ilustración 35 Tiempo base de la estación A con uso de red en local</i> .....	50
<i>Ilustración 36 Tiempo base de la estación A con uso de red en local</i> .....	51
<i>Ilustración 37 Tiempo base de la estación A con uso de red en local y sin uso de red</i> .....	52
<i>Ilustración 38 Tiempo base de la estación B con uso de protocolos de red en configuración remota</i> .....	53
<i>Ilustración 39 Tiempo base de la estación B con uso de protocolos de red en configuración remota</i> .....	54
<i>Ilustración 40 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</i> .....	55
<i>Ilustración 41 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</i> .....	56
<i>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</i> .....	57

<i>Ilustración 43</i>	<i>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>58</i>
<i>Ilustración 44</i>	<i>Análisis del algoritmo del más cercano .....</i>	<i>59</i>
<i>Ilustración 45</i>	<i>Análisis del algoritmo de búsqueda ortogonal .....</i>	<i>60</i>
<i>Ilustración 46</i>	<i>Análisis del algoritmo de búsqueda esférica.....</i>	<i>61</i>
<i>Ilustración 47</i>	<i>Web de Wikipedia.....</i>	<i>68</i>
<i>Ilustración 48</i>	<i>Web de Worldingo.....</i>	<i>68</i>
<i>Ilustración 49</i>	<i>Web de CodePixel .....</i>	<i>69</i>
<i>Ilustración 50</i>	<i>Web de Arrakis.....</i>	<i>69</i>
<i>Ilustración 51</i>	<i>Web de Sargue.....</i>	<i>70</i>
<i>Ilustración 52</i>	<i>Web de cfelde .....</i>	<i>70</i>
<i>Ilustración 53</i>	<i>Web de cadstock.....</i>	<i>71</i>
<i>Ilustración 54</i>	<i>Web de CodePixel .....</i>	<i>71</i>
<i>Ilustración 55</i>	<i>Web de corej2eepatterns.....</i>	<i>72</i>





