



Universidad
Carlos III de Madrid

Departamento de Ingeniería Informática

INGENIERÍA INFORMÁTICA

PROYECTO FIN DE CARRERA

DISEÑO DE UN SISTEMA DE GESTIÓN DE CACHÉS

Autor: Jose Antonio Navarro Fuentes

Tutor:María Jesús Poza Lara

Leganés, marzo de 2017

Título: DISEÑO DE UN SISTEMA DE GESTIÓN DE CACHÉS

Autor: Jose Antonio Navarro Fuentes

Director: María Jesús Poza Lara

EL TRIBUNAL

Presidente: ALEJANDRO CALDERÓN MATEOS

Vocal: ISRAEL GONZÁLEZ CARRASCO

Secretario: JOSÉ LUIS LÓPEZ CUADRADO

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 15 de marzo de 2017 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco a María Jesús Poza su apoyo y tiempo dedicado, para conseguir una titulación que empecé hace años.

Resumen

Este proyecto tiene como objetivo realizar un estudio de las ventajas que aporta a las aplicaciones informáticas el almacenamiento en memoria (almacenamiento en caché) de información obtenida de recursos externos. Se realizará un análisis de la situación actual y se propondrá el diseño de un sistema de gestión de cachés que normalice el uso del almacenamiento en memoria por parte de cualquier aplicación.

El sistema diseñado permitiría gestionar el almacenamiento en memoria de la información necesaria en tiempo de ejecución de las aplicaciones.

El resultado del proyecto será el diseño de un sistema que ofrezca soporte a cachés de nivel 1 (en la propia instancia de la aplicación) y nivel 2 (compartida por diversas instancias de aplicaciones), de forma que se permita el uso de diversas implementaciones de caché de forma transparente a la aplicación usuaria.

Palabras clave: IMDG, caché, datagrid, JCache, JSR-107, Infinispan, Coherence, EhCache, Hazelcast.

Abstract

This project aims to study the advantages of the memory storage in computer applications (caching) of the information obtained from external resources. An analysis of the current situation will be carried out and a design of a system of Cache management that normalizes the use of storage in memory by any application will be proposed.

The system designed will manage memory storage of the necessary information at runtime applications.

The result of the project will be the design of a system that offers support caches level 1 (in the actual instance of the application) and level 2 (shared by multiple instances of applications). It is allowed the use of various implementations of cache in a transparent way to the application.

Keywords: IMDG, cache, datagrid, JCache, JSR-107, Infinispan, Coherence, EhCache, Hazelcast.

Índice general

1. CAPÍTULO 1.....	17
1. Introducción y objetivos.....	17
1.1 Introducción	17
1.2 Motivación del proyecto	18
1.2.1 Mejora del rendimiento	18
1.2.2 Mejora de la fiabilidad	20
1.2.3 Mejora de la escalabilidad.....	20
1.2.4 Mejoras obtenidas	20
1.3 Objetivos.....	21
1.4 Fases del desarrollo	23
1.4.1 Fase RU	24
1.4.2 Fase RS/DA.....	25
1.4.2.1 Obtención de los casos de uso.....	27
1.4.2.2 Obtención de los requisitos software	28
1.4.3 Fase DD.....	29
1.5 Medios empleados	29
1.6 Contenido de la memoria.....	29
2. CAPÍTULO 2.....	31
2. Estado del arte	31
2.1 Situación actual.....	31
2.1.1 Java Specification Request (JSR) número 107	31
2.1.1.1 Ejemplo de uso de JCache	32
2.1.1.2 Anotaciones.....	33
2.1.2 Spring-cache.....	35
2.1.3 Proveedores.....	36
2.1.3.1 Infinispan	36
2.1.3.2 Coherence.....	37
2.1.3.3 EhCache	38
2.1.3.4 Hazelcast	38
2.2 Carencias cubiertas por el PFC	38

3. CAPÍTULO 3.....	41
3. Requisitos de usuario	41
3.1 Introducción.....	41
3.1.1 Propósito	41
3.1.1.1 Definiciones	41
3.2 Descripción general	42
3.2.1 Perspectiva del producto	42
3.2.2 Funcionalidades generales	42
3.3 Requisitos específicos.....	43
3.3.1 Requisitos de capacidad	43
3.3.2 Requisitos de restricción	47
4. CAPÍTULO 4.....	49
4. Requisitos de software y diseño arquitectónico	49
4.1 Introducción.....	49
4.1.1 Propósito	49
4.2 Descripción del modelo.....	50
4.2.1 Casos de uso	50
4.2.1.1 Caso de uso “Uso de caché”.....	50
4.2.1.2 Caso de uso “Administrar caché”	54
4.2.1.3 Caso de uso “Configurar caché”	62
4.3 Requisitos de software.....	64
4.3.1 Requisitos funcionales	64
4.3.2 Requisitos de Interfaz.....	65
4.3.3 Requisitos de Operación	66
4.3.4 Requisitos de Recursos	66
4.3.5 Requisitos de Verificación	66
4.3.6 Requisitos de Documentación.....	67
4.3.7 Requisitos de Portabilidad.....	67
4.3.8 Requisitos de Fiabilidad.....	67
4.3.9 Requisitos de Mantenimiento.....	68
4.3.10 Requisitos de Seguridad.....	68
4.4 Diseño arquitectónico	68
5. CAPÍTULO 5.....	71
5. Diseño detallado.....	71
5.1 Diseño detallado de la consola de administración	71
5.1.1 Pantalla principal de la aplicación.....	71
5.1.2 Pantalla de consulta de las cachés de una aplicación.....	72
5.1.3 Pantalla de consulta la información de la caché de Nivel 2.....	74
5.1.4 Pantalla de consulta la información de la caché de Nivel 1.....	75
5.1.5 Diagrama de clases de la consola de administración	76
5.1.6 Diagrama de secuencia de la consola de administración	78
5.2 Diseño detallado del modelo de datos.....	80
5.2.1 Diagrama de clases del modelo de datos.....	80
5.2.2 Diagrama de secuencia de uso de caché.....	83
6. CAPÍTULO 6.....	85
6. Conclusiones y trabajos futuros	85
6.1 Conclusiones.....	85
6.2 Líneas de trabajo futuro.....	87
7. CAPÍTULO 7.....	89
7. Planificación y presupuesto.....	89
7.1 Introducción.....	89

7.1.1	Planificación.....	90
7.1.2	Presupuesto	91
7.1.2.1	Costes de personal.....	91
7.1.2.2	Costes de material	92
7.1.2.3	Costes indirectos	92
7.1.2.4	Resumen del presupuesto.....	92
7.1.2.5	Detalle del presupuesto	93
8.	CAPÍTULO 8.....	95
8.	Bibliografía y referencias	95
8.1	<i>Estándares de caché</i>	95
8.1.1	JSR 107: JCache.....	95
8.1.2	Spring-cache.....	96
8.1.3	Caché.....	97
8.2	<i>Proveedores de caché</i>	97
8.2.1	Coherence.....	97
8.2.2	Infinispan	97
8.2.3	Ehcache	97
8.2.4	Hazelcast	97
8.2.5	ChronicalMap.....	97
8.3	<i>Ingeniería del software</i>	97
9.	CAPÍTULO 9.....	99
9.	Glosario	99

Índice de figuras

Figura 1. Evolución precio-tamaño de la memoria (fuente Distributed computing with Hazelcast [c.21])	19
Figura 2. Aplicación con acceso a recursos externos.....	21
Figura 3 Aplicación con acceso a recursos externos usando caché de N1 y N2	23
Figura 4. Ejemplo de uso de JCache	32
Figura 5. @CacheDefaults valores por defecto de caché.....	33
Figura 6. @CacheResult obtención de un valor de la caché	34
Figura 7. @CachePut almacenamiento de un valor en caché	34
Figura 8. @CacheRemove borrado de un valor en caché	34
Figura 9. @CacheRemoveAll, borrado de todos los valores de la caché	35
Figura 10. Dependencias Maven para usar spring-cache.....	36
Figura 11. Dependencias Maven para usar Infinispan como proveedor	37
Figura 12. Dependencias Maven para usar Coherence como proveedor	37
Figura 13. Dependencias Maven para usar EhCache como proveedor.....	38
Figura 14. Dependencias Maven para usar Hazelcast como proveedor.....	38
Figura 15. Casos de uso de uso de caché	51
Figura 16. Casos de uso de administrar caché	55
Figura 17. Casos de uso de configurar caché	62
Figura 18. Capa de caché	68
Figura 19. Arquitectura del SGC	69
Figura 20. Arquitectura de la consola de administración.....	70
Figura 21. Pantalla principal de la consola de administración.....	71
Figura 22. Pantalla de consulta de las cachés de una aplicación	72
Figura 23. Pantalla indicando las cachés de una aplicación.....	73
Figura 24. Información de la caché de Nivel 1	73
Figura 25. Información de la caché de Nivel 2	73
Figura 26. Pantalla indicando una aplicación sin datos almacenados.....	74
Figura 27. Consulta de la caché de Nivel 2.....	74
Figura 28. Información de la caché de Nivel 2	75

Figura 29. Consulta de la caché de Nivel 1 de un nodo	75
Figura 30. Información de la caché de Nivel 1	75
Figura 31. Pantalla indicando una aplicación sin datos almacenados en Nivel 1	76
Figura 32. Diagrama de clases de los comandos.....	77
Figura 33. Diagrama de clases de las respuestas de los comandos	78
Figura 34. Diagrama de secuencia de la consola de administración.....	79
Figura 35. Diagrama de secuencia de eliminar dato	79
Figura 36. Diagrama de secuencia de modificar dato	80
Figura 37. Diagrama de clases de las cachés.....	81
Figura 38. Diagrama de clases del gestor de cachés	82
Figura 39. Diagrama de secuencia de uso de cache	83
Figura 40. Planificación del proyecto.....	90
Figura 41. Presupuesto del proyecto	93

Índice de tablas

Tabla 1. Tiempos de acceso distintos recursos (fuente Distributed computing with Hazelcast [c.21])	19
Tabla 2. Plantilla de requisito.....	25
Tabla 3. Plantilla de caso de uso	28
Tabla 4. Relación anotaciones Spring-cache JCache	35
Tabla 5. Requisito 001.RU.C-Almacenar dato sin caducidad	43
Tabla 6. Requisito 002.RU.C-Almacenar dato con caducidad	43
Tabla 7. Requisito 003.RU.C-Obtener dato	43
Tabla 8. Requisito 004.RU.C-Actualizar dato	44
Tabla 9. Requisito 005.RU.C-Actualizar caducidad dato	44
Tabla 10. Requisito 006.RU.C-Eliminar dato	44
Tabla 11. Requisito 007.RU.C-Establecer nivel de caché	44
Tabla 12. Requisito 008.RU.C-Consultar nivel de caché	44
Tabla 13. Requisito 009.RU.C-Establecer proveedor caché N1	45
Tabla 14. Requisito 010.RU.C-Establecer proveedor caché N2	45
Tabla 15. Requisito 011.RU.C-Obtener proveedor de caché N1	45
Tabla 16. Requisito 012.RU.C- Obtener proveedor de caché N2	45
Tabla 17. Requisito 013.RU.C-Configurar clúster de caché N1	45
Tabla 18. Requisito 014.RU.C-Añadir nodo a clúster de caché N1	46
Tabla 19. Requisito 015.RU.C-Eliminar nodo de clúster de caché N1.....	46
Tabla 20. Requisito 016.RU.C-Obtener todas las claves almacenadas.....	46
Tabla 21. Requisito 017.RU.C-Acceso múltiple	46
Tabla 22. Requisito 018.RU.C-Administración en tiempo de ejecución	46
Tabla 23. Requisito 001.RU.R-Facilidad de configuración	47
Tabla 24. Requisito 002.RU.R-Lenguaje de programación.....	47
Tabla 25. Requisito 003.RU.R-Acceso concurrente	47
Tabla 26. Requisito 004.RU.R-Acceso a caché de N2	47
Tabla 27. Requisito 005.RU.R-Soporte JCache.....	47
Tabla 28. Requisito 006.RU.R-Librerías y APIs estándar	48

Tabla 29. Requisito 007.RU.R-Componentes de código abierto	48
Tabla 30. Caso de uso 001.CU - Obtener dato.....	52
Tabla 31. Caso de uso 002.CU - Acceso caché Nivel 1	52
Tabla 32. Caso de uso 003.CU - Almacenar dato caché Nivel 1	52
Tabla 33. Caso de uso 004.CU - Obtener dato caché Nivel 1	53
Tabla 34. Caso de uso 005.CU - Acceso caché Nivel 2	53
Tabla 35. Caso de uso 006.CU - Almacenar dato caché Nivel 2	53
Tabla 36. Caso de uso 007.CU - Obtener dato caché Nivel 2	54
Tabla 37. Caso de uso 008.CU - Obtener dato recurso externo	54
Tabla 38. Caso de uso 009.CU - Administrar caché	56
Tabla 39. Caso de uso 010.CU - Administrar caché Nivel 2	56
Tabla 40. Caso de uso 011.CU - Consultar dato caché Nivel 2	57
Tabla 41. Caso de uso 012.CU - Modificar dato caché Nivel 2.....	58
Tabla 42. Caso de uso 013.CU - Eliminar dato caché Nivel 2.....	59
Tabla 43. Caso de uso 014.CU - Administrar caché Nivel 1	59
Tabla 44. Caso de uso 015.CU - Consultar dato caché Nivel 1	60
Tabla 45. Caso de uso 016.CU - Modificar dato caché Nivel 1.....	61
Tabla 46. Caso de uso 017.CU - Eliminar dato caché Nivel 1.....	62
Tabla 47. Caso de uso 018.CU - Configurar aplicación.....	63
Tabla 48. Caso de uso 019.CU - Instalar y configurar aplicación.....	64
Tabla 49. Requisito 001.RS.F-Almacenar dato.....	64
Tabla 50. Requisito 002.RS.F-Recuperar dato.....	64
Tabla 51. Requisito 003.RS.F-Recuperar dato.....	64
Tabla 52. Requisito 004.RS.F-Consulta de información almacenada.....	65
Tabla 53. Requisito 005.RS.F-Borrado de información.....	65
Tabla 54. Requisito 006.RS.F-Configurar el sistema.....	65
Tabla 55. Requisito 007.RS.F-Caducidad por defecto.....	65
Tabla 56. Requisito 001.RS.I-Consola AngularJS	65
Tabla 57. Requisito 002.RS.I-Sistema portable	66
Tabla 58. Requisito 001.RS.O-Almacenar dat	66
Tabla 59. Requisito 001.RS.R-Ordenadores en red	66
Tabla 60. Requisito 002.RS.R-Máquina virtual Java	66
Tabla 61. Requisito 003.RS.R-Proveedores soportados.....	66
Tabla 62. Requisito 001.RS.V-Acceso simultáneo	67
Tabla 63. Requisito 001.RS.V-El sistema será portable	67
Tabla 64. Requisito 002.RS.V-El sistema será portable	67
Tabla 65. Requisito 003.RS.V-La caché de N2 será portable.....	67
Tabla 66. Requisito 001.RS.iF-El sistema funcionará de forma óptima.....	67
Tabla 67. Requisito 001.RS.M-La caché de N2 será portable	68
Tabla 68. Requisito 001.RS.S-Acceso securizado	68
Tabla 69. Requisito 002.RS.S-Confirmación de operaciones.....	68
Tabla 70. Comparación sin caché vs con caché	86
Tabla 71. Comparación SGC vs JCache	87
Tabla 72. Actividades del proyecto y duración.....	91
Tabla 73. Roles del proyecto y duración.....	91
Tabla 74. Gastos de personal imputables al proyecto	92
Tabla 75. Gastos de material imputables al proyecto.....	92
Tabla 76. Resumen del presupuesto.....	92

CAPÍTULO 1

1. Introducción y objetivos

En este capítulo se muestra el contexto en el que se enmarca el presente Proyecto Fin de Carrera, explicando en qué consiste y las ventajas que se obtienen en las aplicaciones con el uso de almacenamiento de la información en caché, así como los objetivos que se pretende alcanzar, las fases en que se va a desarrollar y la estructura de esta memoria.

1.1 Introducción

Los consumidores de hoy que utilizan tecnología son cada vez más exigentes, debido a que a medida que la misma se ha convertido en más fácil de usar y más generalizada, las expectativas de estos han aumentado. Actualmente, los usuarios esperan entre otras cosas lo siguiente:

- Acceso en cualquier momento.
- Sitios web o aplicaciones **rápidas** y siempre disponibles.
- Capacidad de respuesta en **tiempo real**.
- **Acceso inmediato** a nuevos productos y servicios en cuanto estén disponibles.

A pesar de que no siempre es fácil cumplir las expectativas tecnológicas de los usuarios, es necesario que los sistemas cubran dichas expectativas para dar un servicio acorde con la demanda de los mismos. Si un sitio web o aplicación es lento, los usuarios aunque hasta ese momento hayan sido fieles no dudan en pasar a un competidor.

La tecnología que más se está popularizando es la basada principalmente en dispositivos móviles siempre conectados que requiere el manejo de grandes volúmenes de información procedentes de cada vez mayor número de fuentes que proporcionan datos muy variados. Además está impulsando la demanda de sistemas informáticos más rápidos y más escalables, por lo que las empresas deben utilizar sistemas informáticos que soporten las siguientes características:

- **Alta disponibilidad:** la infraestructura debe estar siempre disponible y lista para recibir y procesar grandes cantidades de datos.

- **Alto rendimiento:** Las comunicaciones deben ser rápidas y eficientes, los retrasos y cuellos de botella se amplifican y los clientes o usuarios no los van a tolerar.
- **Facilidad de escalabilidad:** La infraestructura debe ser compatible con una gran variabilidad en la cantidad de usuarios conectados y ser capaz en cada momento de dar soporte en tiempo real, y optimizando el uso de los recursos.

A medida que crece la cantidad de la información manejada y aumenta la demanda de un acceso instantáneo, la tecnología debe evolucionar para mantenerse al día. Las empresas que respondan a estos desafíos obtendrán mayores beneficios que las empresas que no lo consiguen que serán menos competitivas.

1.2 Motivación del proyecto

Con estas cuestiones en consideración una de las preguntas que surge cuando se afronta el desarrollo de una aplicación es, ¿cómo se puede mejorar la experiencia del usuario?.

Como hemos visto las nuevas tendencias tecnológicas implican que las aplicaciones se enfrentan a un gran número de usuarios distribuidos y concurrentes que acceden al sistema y utilizan repetidamente datos similares, en estos accesos no se admiten tiempos lentos o fallos.

Para optimizar la experiencia de uso de estas aplicaciones es necesario mejorar en tres aspectos:

- El **rendimiento**, reduciendo el tiempo de espera para obtener resultados.
- La **fiabilidad**, reduciendo el impacto de los fallos técnicos.
- La **escalabilidad**, aumentando la capacidad de procesamiento para soportar el crecimiento esperado y proteger contra picos de uso imprevistos.

Una vez detectados los puntos de mejora, se plantea la pregunta de qué mecanismos existen para mejorarlos. El mecanismo tratado en el presente Proyecto Fin de Carrera es el almacenamiento en una caché en memoria a dos niveles tanto interno en la aplicación o caché de nivel uno, como en un “data grid”¹ externo a la aplicación o caché de nivel 2. Como veremos a lo largo de la presente memoria esto permite mejorar en los aspectos de rendimiento, fiabilidad y escalabilidad. En el estudio realizado por Gartner en 2013, se indica “aunque el mercado de los “data grid” residentes en memoria, un segmento IMC² clave, no es muy grande, su crecimiento es cada vez más rápido y se espera que alcance los 1000 millones de dólares en 2016” [d.1]. Por lo que dicho mecanismo va a ganar cada vez más protagonismo en la infraestructura necesaria para optimizar las aplicaciones.

1.2.1 Mejora del rendimiento

Las aplicaciones obtienen los datos de diversas formas:

- Accediendo a repositorios externos.
- Invocando a servicios de terceros.
- Realizando transacciones con mainframes

¹ [Data grid](#): Red de datos almacenados en memoria, Ver Glosario

² [IMC](#): In Memory Computing, procesamiento en memoria, ver Glosario

Introducción

- Etc.

Estas operaciones son en muchas ocasiones costosas y lentas, por lo que se hace necesario reducir tanto el coste de operación, como el tiempo de respuesta ya que son criterios básicos para el éxito de la aplicación. Si la aplicación necesita acceder a estos recursos externos para cada operación su rendimiento será lento y supondrá un alto coste. Para acceso a recursos costosos, se puede decir que los mejores accesos son los que no tienen que llevarse a cabo.

Recursos	Tiempo (ns)	Tiempo (ms)
Caché Nivel 1	0,5	
Caché Nivel 2	7	
4k leídos de SSD	150.000	0,15
Búsqueda en disco	10.000.000	10,00

Tabla 1. Tiempos de acceso distintos recursos (fuente Distributed computing with Hazelcast [[c.21](#)])

En este tipo de aplicaciones, se puede almacenar en una caché en memoria los datos una vez obtenidos. Esto permite a la aplicación recuperar información entre diferentes solicitudes y usuarios, gestionando la propia caché el ciclo de vida de los datos. Utilizando la tecnología de almacenamiento en caché en puntos estratégicos se reduce el número de accesos a recursos lentos y costosos. Aunque los repositorios de caché requieren recursos de memoria y CPU, estos son mucho más baratos y rápidos, por lo que el uso de cachés ofrece mejoras generales de rendimiento.

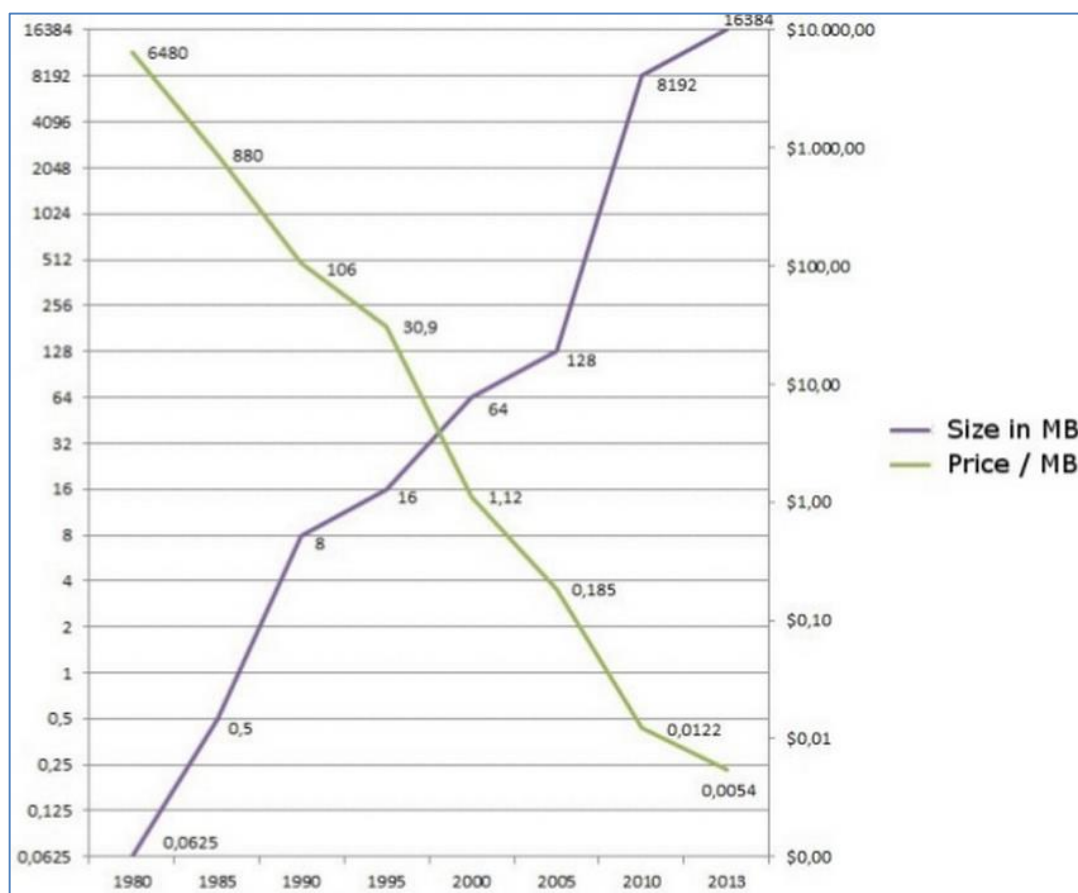


Figura 1. Evolución precio-tamaño de la memoria (fuente Distributed computing with Hazelcast [[c.21](#)])

Al hacer uso del almacenamiento en caché se almacena o mantiene en memoria la información, de tal forma que se puede mejorar en gran medida la velocidad de la aplicación, almacenando temporalmente información a la que se accede frecuentemente en un lugar que es rápido y fácilmente accesible. Esto permite reducir el acceso a recursos más lentos y más costosos. En un “data grid” se puede almacenar cualquier objeto de negocio, de cualquier complejidad, generado como resultado de numerosas transacciones. Con este almacenamiento se reducen accesos a sistemas externos, se descargan los servidores y se permite que las bases de datos precarguen datos en la caché. [\[d.2\]](#)

1.2.2 Mejora de la fiabilidad

Un sistema de caché instalado en un “data grid” es un clúster distribuido de servidores que ejecutan middleware³ para almacenar en memoria datos, esto permite a la aplicación acceder a ellos de forma más rápida y fiable protegiendo al sistema de fallos de algún elemento del clúster, al estar la información distribuida.

El sistema es tolerante a fallos porque los datos almacenados son replicados automáticamente. No existen puntos únicos de fallo. Los servidores se monitorizan entre sí, se reparten la responsabilidad. La información se balancea automáticamente mediante una comunicación punto a punto (máximo un salto). Existe una vista local en cada nodo del clúster por lo que no hacen falta búsquedas en caso de fallo de una instancia porque los restantes miembros se reparten sus datos. Lo mismo ocurre durante el proceso de añadir o parar una instancia.

1.2.3 Mejora de la escalabilidad

Los “data grids” de cachés son middleware que gestionan datos entre servidores distribuidos en una arquitectura escalada horizontalmente, permitiendo un sistema que puede escalar de forma dinámica por lo que siempre puede satisfacer las demandas actuales, mediante la adición de recursos al clúster. No es necesario escalar servidores de base de datos, mainframes o servicios asociados. El sistema es escalable horizontalmente.

Al añadir o eliminar instancias, se produce una repartición de datos automática, absorbiendo en tiempo real aumentos o descensos de carga en el sistema, con esto se consigue una escalabilidad extrema al ser delegada en el propio clúster de la caché, que es compuesto por nodos tecnológicamente sencillos e internos a la organización.

1.2.4 Mejoras obtenidas

Almacenando la información en una caché en memoria las aplicaciones ofrecen mejores prestaciones y permiten favorecer la experiencia y satisfacción del cliente y mejorar la retención de los mismos. Las cachés en memoria proporcionan un enfoque sencillo para superar el manejo de datos, de manera que se puede:

- **Manejar grandes volúmenes** de datos.
- Responder a **requisitos de elevado tiempo** de actividad.

³ [Middleware](#): Software para intercambio de información entre aplicaciones, ver Glosario

- Proporcionar **acceso rápido** a información precisa en tiempo real.
- **Optimizar** la interacción con capas de datos complejas y rígidas.

Una caché en memoria permite almacenar información en memoria. Según Gartner las cachés permiten tratar datos con las “tres V's”: velocidad, variabilidad y volumen, pudiendo soportar cientos de miles de accesos por segundo, y pudiendo ser montado en clústers que permiten ser escalados de forma que soportan grandes cantidades de datos.

Las principales aportaciones de las cachés son:

- **Mejora del rendimiento** porque los datos son accedidos directamente en memoria.
- Gran **capacidad de escalado** porque los clústers montados son fácilmente adaptados a las necesidades de las aplicaciones.
- El almacenamiento en forma de clave/valor ofrece una gran **flexibilidad** a las aplicaciones al permitir el almacenamiento de cualquier tipo de dato.

1.3 Objetivos

Este proyecto tiene como objetivo, diseñar un sistema de gestión de cachés que normalice el uso de almacenamiento en memoria por parte de cualquier aplicación. El sistema diseñado permitirá gestionar el almacenamiento en memoria de la información necesaria en tiempo de ejecución de las aplicaciones. Permitirá almacenar información tanto en una caché de nivel 1 (en la propia instancia de la aplicación) y nivel 2 (compartida por diversas instancias de aplicaciones), de tal forma que se permita el uso de diversas implementaciones de caché de forma transparente a la aplicación usuaria. Además debe de ofrecer mecanismos de mantenimiento de la caché para permitir que un dato sea eliminado o actualizado de la caché.

El Sistema de Gestión de Cachés permitirá que una aplicación que acceda a recursos externos tenga la posibilidad de hacer uso de almacenamiento en memoria.

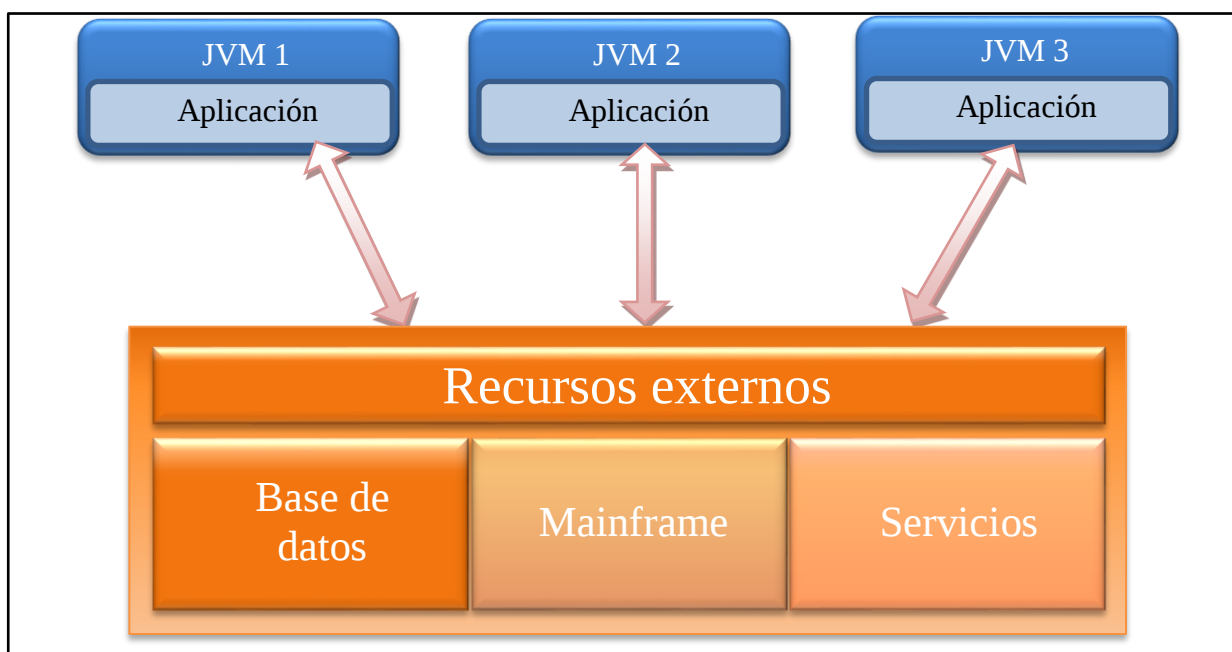


Figura 2. Aplicación con acceso a recursos externos

El sistema propuesto debe:

1. **Permitir cambiar de librería de caché por configuración**, esto puede ser útil por decisión de utilizar otro proveedor de caché por el motivo que sea, decisión estratégica de la empresa, detección de algún error en la librería utilizada.
2. **Permitir la gestión de la propia caché**, pudiendo forzar la eliminación o caducidad de datos almacenados, esto puede ser necesario porque se deba proceder a una modificación de la información, y sea necesario forzar una recarga de las cachés, por ejemplo en el caso de que se tenga almacena la información de tablas maestras.
3. **Permitir cambiar los niveles de caché** durante la ejecución de la aplicación pudiendo pasar a utilizar cualquier combinación de los niveles (L1, L1, L1-L2, o sin caché), esto puede ser debido a que se ha detectado algún problema en alguna infraestructura y es necesario desactivar el almacenamiento en la caché de la información, o bien que se tiene la caché disponible e inhabilitada en las aplicaciones y bajo determinadas circunstancias se decide activar el uso de la caché para la descarga de la infraestructura.

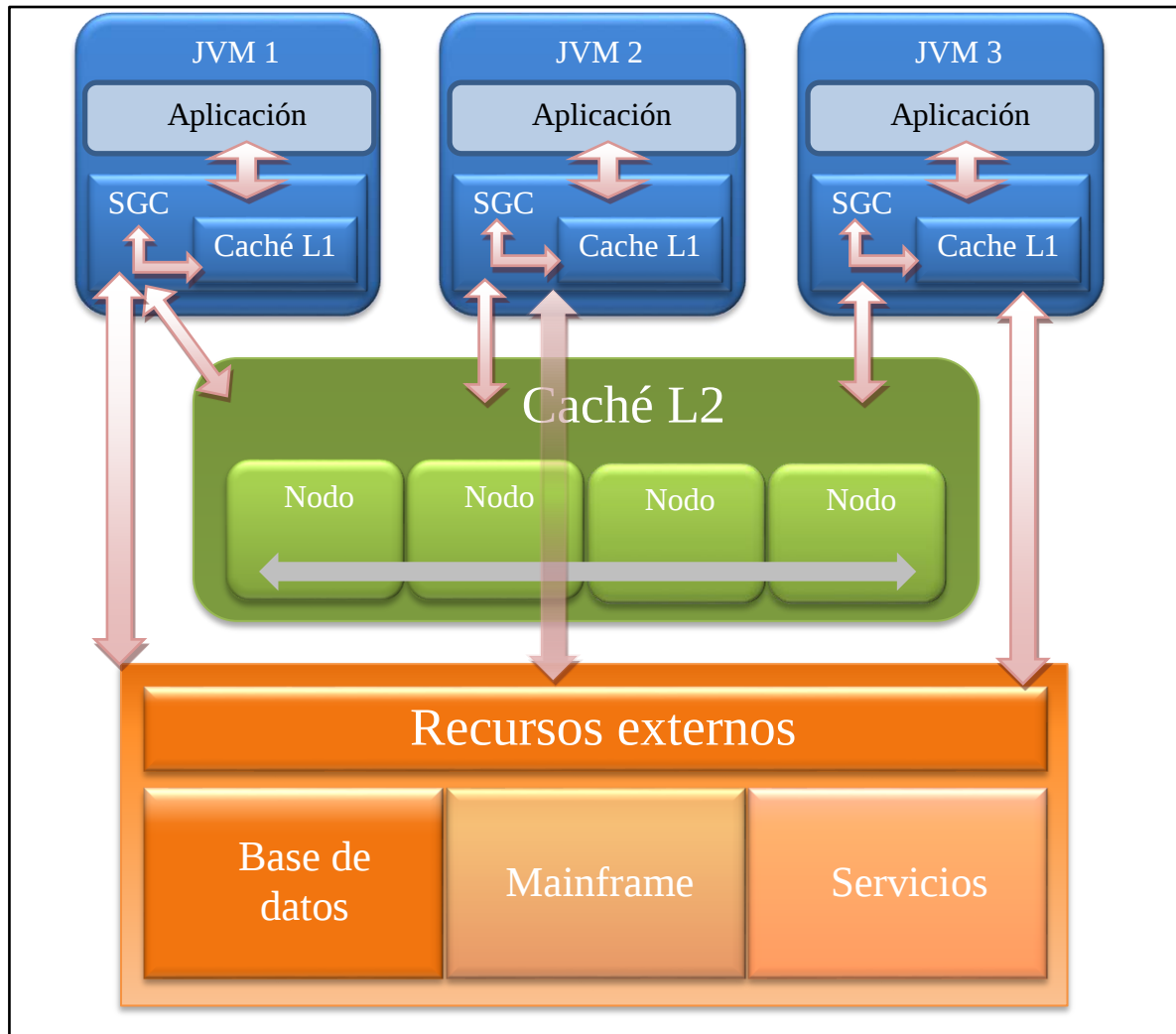


Figura 3 Aplicación con acceso a recursos externos usando caché de N1 y N2

1.4 Fases del desarrollo

En el presente proyecto se utiliza como guía para el ciclo de vida del proyecto una versión del Estándar de Ingeniería de Software ESA PSS-05-0 (European Space Agency, Agencia Espacial Europea) [e.5], la versión PSS-05 lite [e.5.5], [e.5.6]. El estándar PSS-05-0, está orientado a proyectos de una gran envergadura, para proyectos en los que el esfuerzo es menor, se utiliza la citada guía denominada PSS-05 lite y que es una versión compacta de PSS-05-0.

En el estándar PSS-05-0 [e.5] se definen las siguientes fases

- Fase RU - Definición de los requisitos de usuario
- Fase RS - Definición de los requisitos de software
- Fase DA - Definición del diseño arquitectónico
- Fase DD - Diseño detallado y producción del código
- Fase TR - Transferencia de software a las operaciones
- Fase OM - Operaciones y mantenimiento

La versión PSS-05 lite [e.5.5], [e.5.6], propone, entre otros cambios combinar las fases RS y DA en una fase única RS/DA. Quedando las siguientes fases

- Fase RU - Definición de los requisitos de usuario
- Fase RS/DA - Definición de los requisitos de software y del diseño arquitectónico
- Fase DD - Diseño detallado y producción del código
- Fase TR - Transferencia de software a las operaciones
- Fase OM - Operaciones y mantenimiento

En el presente Proyecto Fin de Carrera dado que su alcance es el de diseño de un Sistema de Gestión de Cachés, se realizarán las siguientes fases:

- Fase RU - Definición de los requisitos de usuario
- Fase RS/DA - Definición de los requisitos de software y del diseño arquitectónico
- Fase DD - Diseño detallado, no se abordará la producción de código

Para poder llevar a cabo un proyecto software, se deben de seguir las especificaciones marcadas en la metodología elegida. En el caso del presente proyecto fin de carrera, siguiendo la guía PSS-05 lite [\[e.5.5\]](#), [\[e.5.6\]](#), y teniendo como alcance del propio proyecto fin de carrera llegar al diseño del Sistema de Gestión de Cachés, se establecen dentro del ciclo de vida del presente proyecto tres fases.

1.4.1 Fase RU

La fase de definición de los requisitos de usuario [\[e.5.1\]](#), [\[e.5.5\]](#), tiene como objetivo identificar de manera completa y formal las funcionalidades y las características que debe tener el sistema que se va a construir. Según la metodología seguida durante esta fase se deben abordar las tareas indicadas a continuación.

En esta etapa, también conocida como fase de definición del problema, es necesaria la participación del usuario ya que son los responsables de definir las necesidades del sistema. Como resultado de esta fase además de la propia captura de requisitos se debe de obtener una especificación de los requisitos que debe de cumplir el sistema, dichos requisitos son la base para determinar si el sistema construido se adecua a las necesidades del cliente. Esto se realiza en un proyecto real mediante entrevistas ó encuestas, en el presente Proyecto, al tratarse de un contexto universitario, se han definido a partir de la propia experiencia del proyectista.

Los requisitos serán clasificados en dos tipos de capacidad y de restricción, los requisitos de capacidad son aquellos que indican que es lo que necesita el usuario del sistema. Los requisitos de restricción imponen limitaciones a cómo se debe llevar a cabo el sistema, por ejemplo si se debe de cumplir algún estándar, utilizar un determinado sistema operativo o determinado protocolo.

Los requisitos de usuario deben de tener los siguientes atributos:

- I. Identificador: todo requisito ha de tener un identificador unívoco que permita su trazabilidad en fases posteriores.
- II. Necesidad: mide el grado de importancia de la inclusión de un requisito de usuario en el sistema. Los grados son esencial, deseable y opcional.
- III. Prioridad: cada requisito debe tener una prioridad establecida que indique el orden de implementación al desarrollador en caso de que exista un desarrollo evolutivo.

Introducción

- IV. Estabilidad: indica el grado en el cual puede encontrarse un requisito de usuario a lo largo de todo el ciclo de vida del proyecto.
- Estable, indica que no va a presentar variaciones, mientras que
 - No Estable indica que puede estar sujeto a otros factores y que por tanto puede estar sujeto a cambios.
- V. Fuente: sirve para determinar quién fue la persona concreta que introdujo el requisito.
- VI. Claridad: indica la falta de ambigüedad del requisito, es decir, las posibles interpretaciones diferentes del requisito. Se mide en una escala de Alta, Media y Baja.
- VII. Verificabilidad: atributo que expresa el grado en el cual se puede comprobar que un requisito ha sido incorporado al sistema. Se mide en una escala de Alta, Media y Baja.
- VIII. Descripción: describe de manera textual el requisito de usuario atendiendo a la mayor claridad y detalle posible.

En este proyecto se utilizará la plantilla recomendada en [\[e.1.1\]](#).

Identificador: XXX.RU.Y-Título			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción			

Tabla 2. Plantilla de requisito

El identificador tendrá el siguiente formato:

- Identificador: XX-RUY-Título
- XX-RUY: es el identificador del requisito.
 - La XX es un número de dos cifras que sigue el orden en el que se encuentra.
 - La Y puede tomar los siguientes valores:
 - C: para especificar que es un requisito de capacidad.
 - R: para especificar que es un requisito de restricción.
- Título: es el título del requisito que indica a que hace referencia.

El resto de campos de la tabla tienen el significado indicado anteriormente.

1.4.2 Fase RS/DA

La fase de definición de los requisitos de software y del diseño arquitectónico [\[e.5.2\]](#), [\[e.5.3\]](#), [\[e.5.5\]](#), es la combinación de las fases RS y DA que se recogen en el estándar PSS-05-0 [\[e.5\]](#).

La fase de requisitos de software, también es denominada fase de análisis del problema, el objetivo de esta fase es analizar los requisitos de usuario y generar los de requisitos software de una forma completa y consistente, esto se verá reflejado en los propios requisitos software, los modelos realizados, los casos de uso, y el modelo conceptual.

El estándar ESA define los siguientes tipos de requisitos software:

- Funcionales: especifican una función que el sistema ó componente del sistema debe realizar.
- De rendimiento: especifican valores numéricos para variables medibles.
- De interfaz: especifican elementos con los que el software debe interactuar.
- De operación: especifican cómo se comunicará el sistema con operadores humanos.
- De recursos: especifican los límites máximos de recursos físicos como la capacidad de procesamiento.
- De verificación: especifican cómo se va a verificar el producto, condicionando el diseño del mismo.
- De aceptación de test: condicionan el diseño del producto; son un tipo de requisito de Verificación.
- De documentación: especifican requisitos para documentación, además de los establecidos en el propio estándar de la ESA.
- Garantía: especifican requisitos para hacer el sistema frente a amenazas a la seguridad, confidencialidad e integridad.
- De portabilidad: especifican cómo de sencillo es llevarse el software de un entorno a otro.
- De calidad: especifican los atributos del software que lo hacen apropiado para su propósito.
- De fiabilidad: especifican el tiempo medio entre fallos aceptable.
- De mantenimiento: especifican la facilidad con la que el software se puede modificar para corregir fallos o hacer mejoras.
- De seguridad: especifican los requisitos para reducir la posibilidad de daños que pueden derivarse de fallos del software.

En el presente Proyecto Fin de Carrera los requisitos de software vendrán identificados por el prefijo “RS”, seguido de una letra indicativa del tipo de requisito, y a continuación un número.

Los prefijos para cada tipo de requisito serán:

- F : Funcionales.
- Re : de Rendimiento.
- I : de Interfaz.
- O : de Operación.
- R : de Recursos.
- V : de Verificación.
- A : de Aceptación de test.
- D : de Documentación.
- G : de Garantía.
- P : de Portabilidad.
- Q : de Calidad.
- Fi : de Fiabilidad.
- M : de Mantenimiento.
- S : de Seguridad.

En la fase del diseño arquitectónico se definen los componentes software e interfaces mediante las que se comunican dichos componentes con el fin de elaborar un marco en el que se pueda desarrollar el sistema.

Durante la fase RS/DA se realiza una combinación de las tareas correspondientes a las fases RS y DA, debiendo realizarse la siguiente tarea:

- Construir un modelo de implementación lo más independiente posible de lo que necesite el usuario, cubriendo las peticiones del mismo. Este modelo generado se utilizará para el análisis de los requisitos de software y aplicarlo de manera adecuada. Un buen modelo lógico debería estar formado por funciones con propósitos diferenciados, con un nivel de abstracción adecuado, con unos atributos de rendimiento bien definidos, que minimice las interfaces para evitar posibles acoplamientos y que evite información técnica relativa a la implementación.

1.4.2.1 Obtención de los casos de uso

Los siguientes requerimientos se deben aplicar de forma universal a todos los casos de uso descritos en este documento, excepto si explícitamente se especifica lo contrario:

- La terminología es expresada siguiendo el estándar de la IETF (Internet Engineering Task Force) RFC2119.
- La definición de los Casos de Uso sigue la actual especificación de la superestructura de UML (Unified Modeling Language) de la OMG (Object Management Group) en su versión 2.4.1.
- Los Casos de Uso especifican datos o parámetros que el actor obligatoriamente debe suministrar de forma apropiada.
- Cada Caso de Uso debe devolver como parte del resultado de la operación si se ha ejecutado correctamente o no (ACK o NACK). En el caso de devolver un error se tendrá que realizar una gestión apropiada de errores que se analizarán durante la fase de desarrollo del proyecto.
- Cada definición de Caso de Uso trata, tanto como sea posible, de describir flujos alternativos para escenarios de error. Sin embargo, puede haber errores que no hayan sido considerados y que deben ser manejados apropiadamente durante las fases de desarrollo del proyecto.
- En todos los Casos de Uso se debe validar que se cumplen las pre-condiciones y en caso de no cumplirse se informará del error mediante la gestión de errores que se defina en la fase de desarrollo.

En este proyecto se utilizará la siguiente plantilla para documentar los casos de uso, la información presente en cada caso de uso se indica a continuación:

- Identificador: lo identifica de forma única. Tiene el formato XXX -CU, donde las X corresponden a un número de tres cifras único en cada caso.
- Nombre: breve descripción del caso de uso.
- Actores: tipo de usuario que inicia el caso de uso.
- Objetivo: finalidad del caso de uso
- Precondiciones: estado en el que se debe encontrar para poder realizar una operación.
- Postcondiciones: estado en el que queda el sistema tras realizar una operación.
- Escenario básico: especifica cómo interactúa el usuario con el sistema, y la respuesta que se obtiene.
- Escenario alternativo: condiciones excepcionales que afectan al escenario y las respuestas del sistema ante esas situaciones.

XXX.CU-Título	
ID	XXX-CU
Nombre	Título
Descripción	
Actor principal	
Pre-condiciones	
Post-condiciones	
Flujo principal	
Flujo alternativo	

Tabla 3. Plantilla de caso de uso

1.4.2.2 Obtención de los requisitos software

Se deben extraer los requisitos software del modelo lógico. El conjunto de requisitos recogidos debe evitar cualquier tipo de inconsistencia y errores que puedan propagarse en posteriores fases de desarrollo.

Los requisitos software se clasificarán según la categoría a la que pertenezcan: funcionales, rendimiento, interfaz, operativos, recursos, verificación, test de aceptación, documentación, seguridad, portabilidad, calidad, fiabilidad y mantenimiento.

Al igual que ocurre con los requisitos de usuario, los requisitos software deben ser etiquetados con un conjunto de atributos que los describan. En el presente proyecto se utilizarán los mismos atributos y plantilla que en los requisitos de usuario ver [Tabla 2. Plantilla de requisito](#).

- Construcción del modelo físico. El modelo físico es una descripción del diseño del sistema, empleando terminología asociada a la implementación. La transformación del modelo lógico al físico se hace distribuyendo las funciones identificadas entre los distintos componentes y definiendo las entradas y salidas de dichas funciones. Por otra parte es recomendable realizar la descomposición de los componentes de forma jerárquica, permitiendo así una correcta encapsulación de la información.
- Especificación del diseño arquitectónico. Está considerado como el modelo físico, pero documentado de manera completa. Para ello es preferible

realizar la documentación por medio de diagramas que representen la interacción entre los componentes de los distintos niveles de la arquitectura. El resultado de este proceso es un conjunto de componentes que tienen perfectamente definidos sus datos de entrada y salida y las funciones que proporcionan.

1.4.3 Fase DD

Tal como se ha comentado anteriormente durante la fase de Diseño Detallado, se debería realizar la implementación del sistema, lo cual queda fuera del alcance del presente proyecto. Por lo que el principal objetivo dentro del diseño detallado [\[e.5.4\]](#), [\[e.5.5\]](#), es detallar el diseño planteado durante la etapa anterior RS/DA, para posteriormente poder codificarlo, documentarlo y probarlo.

En esta fase se va a realizar la descomposición de los componentes del diseño arquitectónico hasta poder expresarlos como módulos o unidades independientes de programación.

1.5 Medios empleados

Los medios empleados para el desarrollo de este proyecto fin de carrera son los siguientes:

Hardware

- Portátil HP ProBook 650 G1

Software

- Java 1.8.0_101
- Eclipse Neon

1.6 Contenido de la memoria

La estructura de este documento se ha dividido en varias partes:

Capítulo 1: Introducción

En este capítulo se presenta una introducción general al proyecto, se explica cuál es la motivación del mismo y se enumeran los objetivos. También se explican las fases de que se seguirán en el proyecto y finalmente se indican los medios empleados y se presenta en este apartado la estructura en la que se ha organizado la memoria.

Capítulo 2: Estado del arte

El capítulo comienza con una exposición de la situación actual, y se finaliza explicando los puntos que se encuentran sin cubrir en el momento actual, y que son a los que pretende dar cobertura el presente Proyecto Fin de Carrera.

Capítulo 3: Fase de requisitos de usuario

En este capítulo se realiza una enumeración de los requisitos de usuario que deben de estar cubiertos por el Sistema de Gestión de Cachés.

Capítulo 4: Fase de requisitos software y diseño arquitectónico

En este capítulo se lleva a cabo la definición de los requisitos de software, mostrándose los diferentes casos de uso. En el capítulo también se realiza el diseño arquitectónico del SGC.

Capítulo 5: Fase de diseño detallado

En este capítulo se realiza el diseño detallado del sistema, se divide en dos apartados, en el primero se realiza el diseño detallado de la consola de administración, definiendo las pantallas necesarias así como el modelo de clases y los diagramas de secuencia. En el segundo apartado se realiza el diseño detallado del propio Sistema de Gestión de Cachés, definiendo tanto el diagrama del clases del modelo de datos, como los diagramas de secuencia necesarios.

Capítulo 6: Conclusiones y trabajos futuros

El capítulo se divide en dos partes, en la primera se muestran las conclusiones obtenidas a lo largo del trabajo, en la segunda, se presentan algunos de los caminos a seguir como líneas de trabajo futuro.

Capítulo 7: Planificación y presupuesto

En este capítulo se muestra la planificación y el presupuesto del proyecto fin de carrera. Tanto en la planificación como en el presupuesto se detalla cada una de las tareas que se deben de realizar.

Capítulo 8: Bibliografía y referencias

En este capítulo se detallan las fuentes bibliográficas y webgráficas que se han utilizado para llevar a cabo el presente Proyecto Fin de Carrera.

Capítulo 9: Glosario

En este capítulo se enumeran los términos utilizados en el presente proyecto fin de carrera.

CAPÍTULO 2

2. Estado del arte

En este apartado se realiza una investigación sobre el estado en el que se encuentran actualmente las tecnologías existentes, lo que ayuda a la comprensión del proyecto.

2.1 Situación actual

Existen diversos proveedores de librerías que permiten la gestión de cachés en memoria, dichas librerías, aun teniendo API's semejantes, no seguían un estándar, por lo que si una aplicación hacía uso de una librería de caché suministrada por un determinado proveedor, el cambio a otro proveedor implicaba cambios dentro del código de la aplicación.

Ante este problema surgió la iniciativa de definir un estándar que fue recogido dentro del JSR-107 [\[a.1.1\]](#), una vez aprobada la especificación y liberada una versión estable, los principales proveedores realizaron implementaciones de la especificación [\[a.1.2\]](#) dándose un gran paso en la normalización y estandarización del uso de las cachés.

A lo largo del presente capítulo se explicará la situación actual, para en el siguiente capítulo explicar la necesidad de un sistema de gestión de cachés.

2.1.1 Java Specification Request (JSR) número 107

JCache JSR 107 especifica un API y su semántica para el almacenamiento temporal de objetos Java en memoria, incluyendo la creación de objetos, acceso compartido, purgado y mantenimiento de la consistencia entre máquinas virtuales. Esta especificación ofrece un interface similar a HashMap [\[a.1.1\]](#).

JCache fue un esfuerzo de normalización que surgió debido a que existían múltiples soluciones como EhCache, Hazelcast, Infinispan, Coherence que permitían almacenar la información en memoria. Las librerías más simples son los mapas de objetos en memoria con reglas de mantenimiento muy simples, mientras que las librerías más avanzadas

tienen características para el uso más eficiente del almacenamiento y opciones que permiten la configuración de cómo se va a almacenar la información.

La mayoría de las soluciones existentes de almacenamiento en caché se basan en un mapa como estructura de datos y JCache lo estandariza al ser el caso más común.

2.1.1.1 Ejemplo de uso de JCache

Aunque el API de JCache API no permite disponer de todas las características específicas de la implementación concreta utilizada, el API es bastante versátil y permite adaptarse a los casos de uso más comunes. En el siguiente fragmento de código, se puede obtener una visión general de cómo crear y configurar una memoria caché y usarlo para almacenar en caché un costoso servicio de llamada al método.

```
import javax.cache.Cache;
import javax.cache.CacheManager;
import javax.cache.Caching;
import javax.cache.configuration.MutableConfiguration;
import javax.cache.spi.CachingProvider;

.....
/*
Obtiene la implementacion por defecto de CachingProvider del classpath de la aplicación.
Este método funcionará si y solo si hay una única implementación de JCache en el
classpath.. Si hay mas de un proveedor se debe de dar el nombre cualificado del proveedor
p.ej. org.ehcache.jsr107.EhcacheCachingProvider
*/
CachingProvider cachingProvider = Caching.getCachingProvider();

//Obtiene la instancia por defecto de CacheManager utilizando el proveedor
CacheManager cacheManager = cachingProvider.getCacheManager();

//Crea un configurador de cache utilizando MutableConfiguration
MutableConfiguration<String, String> config = new MutableConfiguration<String,
String>().setStoreByValue(true).setTypes(String.class, String.class);

// Usando el cache manager, se crea una cache llamada "cacheSGC", con la configuración
creada
Cache<String, String> cache = cacheManager.createCache("cacheSGC", config);

// poner un par clave, valor en la cache
cache.put("Clave", "Valor");

//obtener una clave e imprimirla en la salida
System.out.println("Valor en la cache es " + cache.get("Clave"));
```

Figura 4. Ejemplo de uso de JCache

El trozo de código mostrado crea y utiliza una caché usando el api definido por JCache, para cualquier proveedor de caché que soporte el api de JCache el código

anterior será válido únicamente incorporando las librerías necesarias y realizando los ajustes de configuración propuestos por el proveedor.

2.1.1.2 Anotaciones

JCache define una integración con CDI⁴, por lo que se pueden decorar los servicios con interceptores que permiten la gestión de la caché utilizando anotaciones. Se pueden indicar las necesidades de almacenamiento poniendo las anotaciones de JCache en clases o métodos.

Las anotaciones existentes son:

- `javax.cache.annotation.CacheDefaults`

Se utiliza a nivel de clase. Todos los métodos dentro de la clase utilizarán los valores indicados por defecto.

```
@CacheDefaults(cacheName = "ejemploCache")
public class ClaseEjemplo
{
    ///
}
```

Figura 5. `@CacheDefaults` valores por defecto de caché

Declara que el nombre de la caché por defecto será `ejemploCache`. Todos los métodos de la clase heredarán esta opción si no se indica un nombre en los mismos.

- `javax.cache.annotation.CacheKey`

Anotación usada en un atributo de un método mediante la que se indica que dicho atributo será la clave de la información almacenada.

- `javax.cache.annotation.CacheResult`

Permite obtener un valor almacenado a partir de su clave. Si el valor indicado en la clave existe en la caché será devuelto sin entrar en el cuerpo del método. Si no existe el flujo de ejecución entra en el cuerpo del método y almacena el objeto devuelto en la caché.

⁴ [CDI](#): Estándar de Java para la inyección de dependencias. Ver Glosario

```
@CacheDefaults(cacheName = "ejemploCache")
public class CacheEjemplo
{
    @CacheResult
    public ClaseEjemplo2 get(@CacheKey Long id)
    {
        ClaseEjemplo2 ejemplo = //
        return ejemplo;
    }
}
```

Figura 6. @CacheResult obtención de un valor de la caché

- javax.cache.annotation.CacheValue

Anotación usada en un atributo de un método mediante la que se indica que dicho atributo será el valor de la información almacenada.

- javax.cache.annotation.CachePut

Almacena un valor en la caché identificado por una clave. Se puede indicar en los parámetros cual será la clave con la anotación @CacheKey y cuál será el valor con la anotación @CacheValue.

```
@CacheDefaults(cacheName = "ejemploCache")
public class CacheEjemplo
{
    @CachePut
    public void put(@CacheKey Long id, @CacheValue ClaseEjemplo2 valor)
    {
        //
    }
}
```

Figura 7. @CachePut almacenamiento de un valor en caché

- javax.cache.annotation.CacheRemove

Borra un valor almacenado en la caché he identificado por una clave.

```
@CacheDefaults(cacheName = " ejemploCache ")
public class CacheEjemplo
{
    @CacheRemove
    public void borrar(@CacheKey Long id)
    {
        //
    }
}
```

Figura 8. @CacheRemoce borrado de un valor en caché

- javax.cache.annotation.CacheRemoveAll

```
@CacheDefaults(cacheName = " ejemploCache ")
public class CacheEjemplo
{
    @CacheRemoveAll
    public void borraTodo()
    {
        //
    }
}
```

Figura 9. @CacheRemoveAll, borrado de todos los valores de la caché

2.1.2 Spring-cache

Existe otro intento de normalización de uso de almacenamiento en memoria que se inició como un Proyecto independiente pero finalmente también ha convergido a JCache. Spring-cache es parte de Spring, estando disponible desde la versión 3.1.

Spring es un marco de trabajo que ofrece diferentes módulos cuya finalidad es facilitar el desarrollo de aplicaciones Java.

Desde la versión 4.1 de Spring, Spring-cache ofrece una serie de anotaciones que se detallan a continuación con su equivalente en **JCache**.

Spring-cache	JCache
@CacheConfig	@CacheDefaults
@Cacheable	@CacheResult
@CachePut	@CachePut
@CacheEvict	@CacheRemove
@CacheEvict(allEntries=true)	@CacheRemoveAll

Tabla 4. Relación anotaciones Spring-cache JCache

Para habilitar el soporte de las anotaciones del almacenamiento en caché de Spring, es necesario utilizar la anotación `@EnableCaching` en la clase de configuración de la aplicación, o bien usar el elemento `<cache:annotation-driven/>` dentro del fichero de configuración del contexto de ejecución de spring. Además es necesario, en el caso de utilizar Maven como herramienta de construcción del sistema, tener las siguientes dependencias.

```
//Dependencias Maven

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>javax.cache</groupId>
  <artifactId>cache-api</artifactId>
  <version>1.0.0</version>
</dependency>
```

Figura 10. Dependencias Maven para usar spring-cache

2.1.3 Proveedores

A continuación se pasa a realizar un análisis de la situación de los proveedores de caché más extendidos en la actualidad y su situación en relación al estándar JCache.

2.1.3.1 Infinispan

Desde la versión 7.0.0, Infinispan ofrece una implementación del API JCache. Para poder utilizar la implementación de JCache de Infinispan es necesario tener en el classpath de la aplicación el jar `infinispan-jcache`, o bien utilizar la siguiente dependencia de Maven.

```
//Dependencias Maven

<dependency>
  <groupId>org.infinispan</groupId>
  <artifactId>infinispan-jcache</artifactId>
  <version>${infinispan.version}</version>
</dependency>
```

Figura 11. Dependencias Maven para usar Infinispan como proveedor

Para hacer uso de la caché de Infinispan se debe de seguir la especificación de JCache mostrado en la [Figura 3. Ejemplo de uso de JCache](#) y tener disponible la implementación de Infinispan dentro de la aplicación.

2.1.3.2 Coherence

Coherence incluye una implementación de JCache, que se basa en las capacidades existentes de Coherence y ofrece un recubrimiento al API de NamedCache. Esto permite reutilizar y exponer muchas de sus mejores capacidades utilizando la interfaz JCache.

Los archivos clave para el soporte a JCache son cache-api.jar y coherence-jcache.jar. El archivo de cache-api.jar contiene las librerías de JCache. El archivo coherence-jcache.jar contiene la implementación de Coherence que se construye sobre JCache. Ambos archivos deben aparecer en la ruta de clase de la aplicación.

El uso de la caché de Coherence se realiza siguiendo la especificación de JCache ver [Figura 3. Ejemplo de uso de JCache](#).

```
//Dependencias Maven

<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>coherence</artifactId>
  <version>${coherence.version}</version>
</dependency>
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>coherence-jcache</artifactId>
  <version>${coherence.version}</version>
</dependency>
<dependency>
  <groupId>javax.cache</groupId>
  <artifactId>cache-api</artifactId>
  <version>1.0.0</version>
</dependency>
```

Figura 12. Dependencias Maven para usar Coherence como proveedor

2.1.3.3 EhCache

Desde la versión 3.x EhCache es proveedor de JCache. Miembros de EhCache han participado en el grupo de expertos que ha elaborado la JSR que especifica JCache.

```
//Dependencias Maven

<dependency>
  <groupId>org.ehcache</groupId>
  <artifactId>ehcache</artifactId>
  <version>${ehcache.version}</version>
</dependency>
<dependency>
  <groupId>javax.cache</groupId>
  <artifactId>cache-api</artifactId>
  <version>1.0.0</version>
</dependency>
```

Figura 13. Dependencias Maven para usar EhCache como proveedor

El uso de la caché de EhCache se realiza siguiendo la especificación de JCache ver [Figura 3. Ejemplo de uso de JCache](#).

2.1.3.4 Hazelcast

Desde la versión 3.3.1, Hazelcast ofrece una implementación de JCache. Al igual que el resto de los proveedores lo único necesario es que esté disponible la implementación de Hazelcast en el classpath de la aplicación.

```
//Dependencias Maven

<dependency>
  <groupId>com.hazelcast</groupId>
  <artifactId>hazelcast</artifactId>
  <version>${hazelcast.version}</version>
</dependency>
<dependency>
  <groupId>javax.cache</groupId>
  <artifactId>cache-api</artifactId>
  <version>1.0.0</version>
</dependency>
```

Figura 14. Dependencias Maven para usar Hazelcast como proveedor

2.2 Carencias cubiertas por el PFC

Tal como hemos visto actualmente existe una especificación, la **JSR-107**, que define JCache como abstracción de uso de almacenamiento de memoria por parte de las aplicaciones. Para poder utilizar un determinado proveedor de caché lo único necesario

por parte de la aplicación es tener accesible en el classpath las librerías indicadas por el proveedor.

La mejora obtenida por esta abstracción es indudable debido a que independiza a la aplicación de la implementación de la caché elegida, pero existen una serie de **problemas no cubiertos** por esta abstracción.

- JCache funciona tomando del classpath la implementación suministrada por un determinado proveedor, por lo que esta **implementación** debe de ser **única** y **no es posible cambiarla en tiempo de ejecución**, ni ser indicada por configuración.
- El uso de la caché viene indicado por los ficheros de configuración del proveedor de caché elegido, por lo que **la caché no podrá ser de Nivel 1 y/o Nivel 2, ni podrá ser elegida una u otra por configuración**.
- **La caché no es administrable**, por lo que si se desea invalidar, borrar o actualizar un dato almacenado (por ejemplo por un cambio en el comportamiento del servicio que lo genera, o por una modificación de la tabla que lo contiene) no es posible realizarlo en tiempo de ejecución de la aplicación.

En conclusión JCache permite estandarizar el uso de las cachés de Nivel 1, permitiendo a las aplicaciones almacenar información en memoria en el propio nodo de la aplicación. Y como se ha visto existen una serie de características que deben ser abordadas y resueltas para que el uso de caché sea lo más eficaz y versátil posible por parte de las aplicaciones. El presente PFC cubrirá las características que no están dentro del alcance de la definición de JCache:

- **Cambiar el proveedor** de caché utilizado **por configuración**, lo cual permitirá que para una aplicación que está dando servicio en un entorno real, el cambio de proveedor por cualquier motivo, sea únicamente un cambio de configuración sin necesidad de una nueva instalación ni de parada ni arranque de la misma.
- Poder hacer **uso de una caché Nivel 2**, esto permite que para aplicaciones que se ejecutan en multitud de instancias, solo sea necesario obtener un dato del recurso externo una vez, el cual quedará almacenado en la caché de Nivel 2, pasando a estar disponible para las cachés de Nivel 1 desde ese momento.
- **Administración y mantenimiento** de la información contenida en las cachés, lo que permite que la información almacenada en caché se pueda mantener en tiempo de ejecución.

CAPÍTULO 3

3. Requisitos de usuario

Este capítulo corresponde a la fase de requisitos de usuario que especifica el Estándar de Ingeniería del Software de la ESA.

La fase RU es la “fase de definición del problema” en el ciclo de vida del proyecto. El propósito de esta fase es obtener una idea acerca de la tarea a ser realizada, para llegar a una definición de los que se espera del sistema.

3.1 Introducción

El objetivo de la fase de requisitos de usuario del Proyecto Fin de Carrera, es especificar de forma detallada las necesidades que debe de cubrir el sistema que se está llevando a cabo, quedando definidos todos los aspectos referidos al mismo y buscando la eliminación de ambigüedades en los requerimientos de usuario.

3.1.1 Propósito

En esta fase se busca una definición de las características del Sistema de Gestión de Cachés. Se analizan las capacidades que debe tener la aplicación, identificando los requerimientos del usuario. Es el primer paso para el posterior desarrollo de la fase de requisitos de software.

Los requisitos de usuario se dividen en dos categorías:

- Capacidades del sistema.
- Restricciones impuestas al sistema.

3.1.1.1 Definiciones

En el apartado **1.4.1 Fase RU** se describen las propiedades que define el estándar que se sigue en este proyecto, que deben de tener los requisitos para estar perfectamente identificados y descritos, y serán los atributos que se completarán. En el capítulo citado

se indica la plantilla que se utilizará para completar los requisitos ver **Tabla 2. Plantilla de requisito**.

3.2 Descripción general

En este apartado se va a describir el producto, las características que deben cumplir los usuarios y las restricciones generales del sistema entre otros puntos.

3.2.1 Perspectiva del producto

El sistema que se va a desarrollar consiste en una pieza middleware que podrá dar servicio a otras aplicaciones, por lo cual se puede considerar un elemento de la arquitectura de la organización en la que se integre, debido a que será un sistema que de forma horizontal dará servicio a cualquiera de las aplicaciones que se desarrollen.

La aplicación será independiente del sistema operativo sobre el que se ejecute, lo cual vendrá facilitado por el hecho de que el lenguaje Java en el que está desarrollado el sistema es multiplataforma. Asimismo, las librerías que precise la aplicación obedecerán a APIs estándar, de forma que sean fácilmente reemplazables por otras en un momento dado.

3.2.2 Funcionalidades generales

La función principal del sistema es proporcionar un sistema de gestión del uso de caché en memoria, lo cual implica permitir almacenar y recuperar datos de la caché, poder configurar cómo se va a utilizar la caché.

Sobre la función principal del Sistema de Gestión de Cachés se añadirán otras funcionalidades auxiliares como:

- Acceso desde una consola de administración a las cachés configuradas.
- Posibilidad de poder modificar la configuración de las cachés desde dicha consola.
- Posibilidad de borrar, o caducar la información almacenada en la caché desde la consola.

3.3 Requisitos específicos

3.3.1 Requisitos de capacidad

Tal como se ha comentado anteriormente los requisitos de capacidad describen qué quieren hacer los usuarios con el sistema. A continuación se presentan los Requisitos de Capacidad que se han identificado.

Identificador: 001.RU.C-Almacenar dato sin caducidad			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Almacenar un dato en la caché, un dato será un par <Clave, Valor>, se almacena el dato sin indicar su caducidad. Si está definida la caché de Nivel 1 se almacena en dicha caché. Si está definida la caché de Nivel 2 se almacena en dicha caché		

Tabla 5. Requisito 001.RU.C-Almacenar dato sin caducidad

Identificador: 002.RU.C-Almacenar dato con caducidad			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Almacenar un dato en la caché, un dato será un par <Clave, Valor>, se almacena el dato indicando la caducidad del mismo. Cuando se cumpla el tiempo de caducidad el dato será borrado de la caché. Si está definida la caché de Nivel 1 se almacena en dicha caché. Si está definida la caché de Nivel 2 se almacena en dicha caché		

Tabla 6. Requisito 002.RU.C-Almacenar dato con caducidad

Identificador: 003.RU.C-Obtener dato			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Obtener un dato. Si está definida la caché de Nivel 1 se consulta si existe en dicha caché y se devuelve, si no existe y está configurada la caché de nivel 2, se consulta si existe el dato en la misma y se devuelve, si no existe se devuelve un objeto de una clase que permita identificar que no existe el dato. Si solo está definida la caché de Nivel 2, se consulta si existe y se devuelve. Si no existe se ejecuta la lógica definida para obtener el dato del recurso externo.		

Tabla 7. Requisito 003.RU.C-Obtener dato

Identificador: 004.RU.C-Actualizar dato			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		

Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Modificar un dato almacenado, identificado por su clave, y con un nuevo valor. Si está definida la caché de Nivel 1 se actualizará en dicho nivel. Si está definida la caché de Nivel 2 se actualizará en dicho nivel.		

Tabla 8. Requisito 004.RU.C-Actualizar dato

Identificador: 005.RU.C-Actualizar caducidad dato			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Modificar la caducidad de un dato, utilizando su clave y la nueva caducidad, la caducidad se modificará aún en el caso de no existir caducidad anterior. Si está definida la caché de Nivel 1 se actualizará en dicho nivel. Si está definida la caché de Nivel 2 se actualizará en dicho nivel.		

Tabla 9. Requisito 005.RU.C-Actualizar caducidad dato

Identificador: 006.RU.C-Eliminar dato			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Eliminar un dato de la caché, identificado por su clave, la eliminación se realizará aun en el caso de que exista caducidad y no se haya cumplido la misma. Si está definida la caché de Nivel 1 se borrará de dicho nivel. Si está definida la caché de Nivel 2 se borrará de dicho nivel.		

Tabla 10. Requisito 006.RU.C-Eliminar dato

Identificador: 007.RU.C-Establecer nivel de caché			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Establecer el nivel de caché que se quiere en el sistema, que puede ser, Sin caché, Nivel 1, Nivel 2, Nivel 1 y Nivel 2.		

Tabla 11. Requisito 007.RU.C-Establecer nivel de caché

Identificador: 008.RU.C-Consultar nivel de caché			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Obtener el nivel de la caché definida en ese momento.		

Tabla 12. Requisito 008.RU.C-Consultar nivel de caché

Identificador: 009.RU.C-Establecer proveedor caché N1			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Establecer cuál es el proveedor de caché de Nivel 1, se deben de ofrecer proveedores suficientes: EhCache, HazelCast, ChronicalMap, Coherence, DataGrid-Infinispan.		

Tabla 13. Requisito 009.RU.C-Establecer proveedor caché N1

Identificador: 010.RU.C-Establecer proveedor caché N2			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Establecer cuál es el proveedor de caché de Nivel 2, se deben de ofrecer proveedores suficientes: EhCache, HazelCast, ChronicalMap, Coherence, DataGrid-Infinispan.		

Tabla 14. Requisito 010.RU.C-Establecer proveedor caché N2

Identificador: 011.RU.C-Obtener proveedor de caché N1			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Obtener el proveedor de la caché de Nivel 1.		

Tabla 15. Requisito 011.RU.C-Obtener proveedor de caché N1

Identificador: 012.RU.C-Obtener proveedor de caché N2			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Obtener el proveedor de la caché de Nivel 2.		

Tabla 16. Requisito 012.RU.C- Obtener proveedor de caché N2

Identificador: 013.RUC-Configurar clúster de caché N1			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Configurar el clúster de la caché de Nivel 1, nombre de clúster, elementos permitidos.		

Tabla 17. Requisito 013.RU.C-Configurar clúster de caché N1

Identificador: 014.RU.C-Añadir nodo a clúster de caché N1			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Añadir un nodo al clúster de cachés de Nivel 1.		

Tabla 18. Requisito 014.RU.C-Añadir nodo a clúster de caché N1

Identificador: 015.RU.C-Eliminar nodo de clúster de caché N1			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Eliminar un nodo del clúster de caché de Nivel 1		

Tabla 19. Requisito 015.RU.C-Eliminar nodo de clúster de caché N1

Identificador: 016.RU.C-Obtener todas las claves almacenadas			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Obtener todas las claves almacenadas.		

Tabla 20. Requisito 016.RU.C-Obtener todas las claves almacenadas

Identificador: 017.RU.C-Acceso múltiple			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Soportar accesos múltiples a contenidos en Caché (Múltiples hilos de ejecución).		

Tabla 21. Requisito 017.RU.C-Acceso múltiple

Identificador: 018.RU.C-Administración en tiempo de ejecución			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Administrar la caché en tiempo de ejecución. Se podrán realizar las siguientes acciones: •Limpieza contenido de caché •Modificación de tiempos de caducidad de caché •Posibilidad de cacheo o no de información •Visualizar contenidos cacheados en tiempo de ejecución •Visualizar estructura de contenidos con formato de árbol.		

Tabla 22. Requisito 018.RU.C-Administración en tiempo de ejecución

3.3.2 Requisitos de restricción

Los requisitos de restricción indican requisitos sobre cómo se alcanzan los requisitos de usuario. Se pueden poner restricciones relacionadas con interfaces, calidad, recursos y tiempos.

Identificador: 001.RU.R-Facilidad de configuración			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	La configuración del Sistema debe de ser sencilla, basada en ficheros xml, json y/o yaml.		

Tabla 23. Requisito 001.RU.R-Facilidad de configuración

Identificador: 002.RU.R-Lenguaje de programación			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☐ Esencial ☒ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El lenguaje de programación utilizado será Java.		

Tabla 24. Requisito 002.RU.R-Lenguaje de programación

Identificador: 003.RU.R-Acceso concurrente			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	La caché de N2 debe de permitir el acceso concurrente del sistema de gestión de cachés		

Tabla 25. Requisito 003.RU.R-Acceso concurrente

Identificador: 004.RU.R-Acceso a caché de N2			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	La caché de N2 será accesible desde cualquier punto en el que se instalen aplicaciones que hagan uso del sistema de gestión de cachés.		

Tabla 26. Requisito 004.RU.R-Acceso a caché de N2

Identificador: 005.RU.R-Soporte JCache			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El sistema de gestión de cachés cumplirá la especificación JSR-107.		

Tabla 27. Requisito 005.RU.R-Soporte JCache

Identificador: 006.RU.R-Librerías y APIs estándar				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema hará uso exclusivamente de librerías que implementen APIs que obedezcan especificaciones estándares, de forma que sea portable a cualquier sistema operativo, y que los componentes sobre los que funciona sean fácilmente reemplazables por cualquier otro que adopte dichos estándares.			

Tabla 28. Requisito 006.RU.R-Librerías y APIs estándar

Identificador: 007.RU.R-Componentes libre distribución				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	Proyectista	
Necesidad	☐ Esencial ☒ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema hará uso de componentes de código abierto.			

Tabla 29. Requisito 007.RU.R-Componentes de código abierto

CAPÍTULO 4

4. Requisitos de software y diseño arquitectónico

Este capítulo corresponde a la fase de requisitos de software y diseño arquitectónico que especifica el Estándar de Ingeniería del Software de la ESA, en su versión lite.

La fase RS/DA es la “fase de análisis del problema” en el ciclo de vida del proyecto. El propósito de esta fase es analizar los requisitos de usuario y transformarlos en un conjunto de requisitos software de la manera más completa y consistente posible.

4.1 Introducción

En este capítulo se va a realizar una descripción de la funcionalidad que va abarcar el sistema de gestión de cachés. Esto quedará reflejado en los requisitos software y los modelos realizados, tanto los casos de uso, como el modelo conceptual, que permitirán tener una visión del sistema a desarrollar.

4.1.1 Propósito

- Identificación del producto a generar
 - “SGC”
- Objetivos del Sistema
 - El uso de caché en memoria para almacenar información.
 - Acceso a caché de Nivel 1 y de Nivel 2
 - Posibilidad de administrar las cachés existentes y la información almacenada

Este documento de requerimientos de software es la base del desarrollo de software del proyecto.

4.2 Descripción del modelo

A lo largo de la sección se van a definir y desarrollar los casos de uso y los requisitos específicos del sistema de gestión de cachés.

4.2.1 Casos de uso

En esta sección del proyecto se describe cada uno de los casos de uso dentro del alcance del desarrollo del proyecto.

4.2.1.1 Caso de uso “Uso de caché”

Las acciones que se realizan en este caso de uso son las siguientes:

- **Obtener dato:** una aplicación que utiliza el Sistema de Gestión de Cachés, solicita un dato.
- **Acceso caché Nivel 1:** el Sistema de Gestión de Cachés comprueba la existencia del dato en la caché de Nivel 1, y lo devuelve o pasa a comprobar la caché de Nivel 2.
- **Acceso caché Nivel 2:** el Sistema de Gestión de Cachés comprueba si existe el dato en la caché de Nivel 2, devolviéndolo y almacenándolo en la cache de Nivel 1 o pasando a solicitarlo al recurso externo correspondiente.
- **Obtener dato recurso externo:** si no existe el dato en ninguna caché se obtiene del recurso correspondiente y se almacena en las cachés correspondientes.

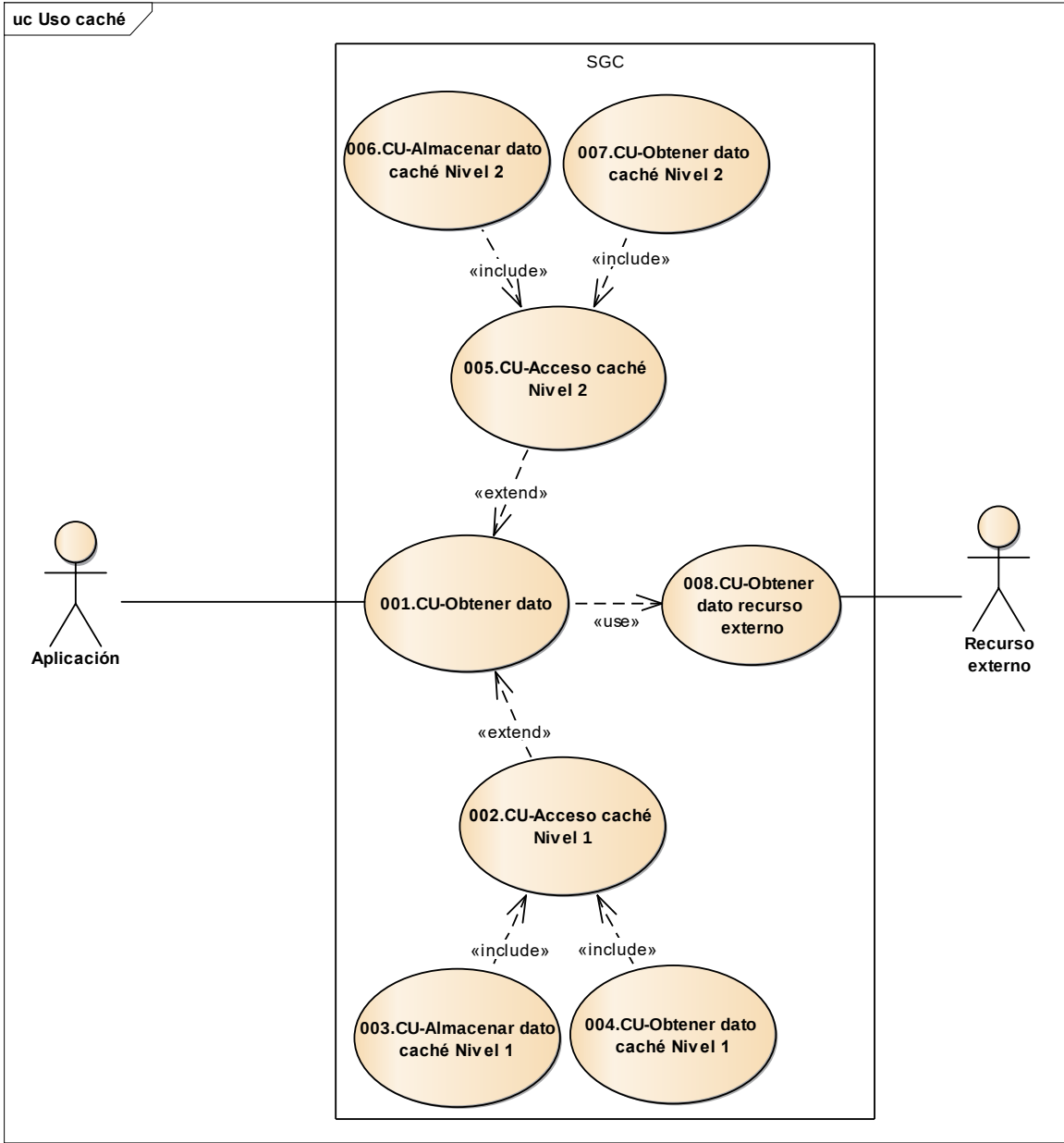


Figura 15. Casos de uso de uso de caché

001.CU-Obtener dato	
ID	001.CU
Nombre	Obtener dato
Descripción	Una aplicación que utiliza el SGC solicita un dato.
Actor principal	Aplicación
Pre-condiciones	Es necesario que este configurado uno o dos de los niveles de caché.
Post-condiciones	El dato es devuelto a la aplicación.
Flujo principal	1. Este caso de uso se inicia cuando se solicita un dato gestionado por el SGC. 2. El sistema deberá consultar que nivel de caché está establecido. 3. Si está establecido el Nivel 1 de caché. a. El SGC accede a la cache de Nivel 1 002.CU. 4. Si está establecido el Nivel 2 de caché y no se ha obtenido el dato previamente.

	a. El sistema accede a la caché de Nivel 2 005.CU. b. Si se obtiene el dato i. Se almacena en caché de Nivel 1 si está configurada 002.CU. ii. Se devuelve el valor. 5. Si no se ha obtenido el dato, se obtiene del recurso externo 008.CU. a. Si está establecido el Nivel 2, se almacena en caché de Nivel 2 005.CU. b. Se almacena en caché de Nivel 1 si está configurada 002.CU. 6. Se devuelve el dato. 7. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 30. Caso de uso 001.CU - Obtener dato

002.CU-Acceso caché Nivel 1	
ID	002.CU
Nombre	Acceso caché Nivel 1
Descripción	Se encarga de ofrecer al acceso a la caché de Nivel 1.
Actor principal	001.CU
Pre-condiciones	Es necesario que esté configurado la caché de Nivel 1.
Post-condiciones	Se ha accedido a la caché de Nivel 1.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la cache de Nivel 1. 2. Se obtiene el proveedor del Nivel 1 de caché. 3. Si se intenta obtener un dato, se recupera de la caché 004.CU. 4. Si se quiere almacenar un dato, se inserta en la caché 003.CU. 5. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 31. Caso de uso 002.CU - Acceso caché Nivel 1

003.CU-Almacenar dato caché Nivel 1	
ID	003.CU
Nombre	Almacenar dato caché Nivel 1
Descripción	Se almacena un dato en la caché.
Actor principal	002.CU
Pre-condiciones	Es necesario que esté configurada la caché de Nivel 1.
Post-condiciones	Se almacenará el dato en la caché.
Flujo principal	1. Este caso de uso se inicia cuando se almacena un dato en la caché. 2. El sistema accede a la caché y almacena el dato con su clave. 3. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 32. Caso de uso 003.CU - Almacenar dato caché Nivel 1

004.CU-Obtener dato caché Nivel 1	
ID	004.CU
Nombre	Obtener dato caché Nivel 1

Descripción	Se recupera un dato de la caché a partir de su clave.
Actor principal	002.CU
Pre-condiciones	Es necesario que esté configurada la caché de Nivel 1.
Post-condiciones	Se recuperará el dato de la caché de Nivel 1.
Flujo principal	1. Este caso de uso se inicia cuando se recupera un dato de la caché de Nivel 1. 2. El sistema accede a la caché y recupera el dato a partir de su clave de la caché. 3. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 33. Caso de uso 004.CU - Obtener dato caché Nivel 1

005.CU-Acceso caché Nivel 2	
ID	005.CU
Nombre	Acceso caché Nivel 2
Descripción	Se encarga de ofrecer acceso a la cache de Nivel 2.
Actor principal	001.CU
Pre-condiciones	Es necesario que esté configurado el acceso a la caché de Nivel 2.
Post-condiciones	Se ha accedido a la caché de Nivel 1.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la cache de Nivel 2. 2. Se obtiene el proveedor del Nivel 2 de caché. 3. Si se intenta obtener un dato, se recupera de la caché 007.CU. 4. Si se quiere almacenar un dato, se inserta en la caché 006.CU. 5. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 34. Caso de uso 005.CU - Acceso caché Nivel 2

006.CU-Almacenar dato caché Nivel 2	
ID	006.CU
Nombre	Almacenar dato caché Nivel 2
Descripción	Se almacena un dato en la caché de Nivel 2.
Actor principal	005.CU
Pre-condiciones	Es necesario que este configurado uno o dos de los niveles de caché.
Post-condiciones	Se recuperará del dato de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se almacena un dato en la caché de Nivel 2. 2. El sistema accede a la caché y almacena el dato con su clave en la misma. 3. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 35. Caso de uso 006.CU - Almacenar dato caché Nivel 2

007.CU- Obtener dato caché Nivel 2	
ID	007.CU
Nombre	Obtener dato caché Nivel 2
Descripción	Se recupera un dato de la caché a partir de su clave.
Actor principal	005.CU
Pre-condiciones	Es necesario que esté configurada la caché de Nivel 2.

Post-condiciones	Se recuperará del dato de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se recupera un dato de la caché de Nivel 2. 2. El sistema accede a la caché y recupera el dato a partir de su clave de la caché. 3. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 36. Caso de uso 007.CU - Obtener dato caché Nivel 2

008.CU-Obtener dato recurso externo	
ID	008.CU
Nombre	Obtener dato recurso externo
Descripción	Se recupera un dato de un recurso externo.
Actor principal	001.CU
Pre-condiciones	Es necesario que esté definida la lógica para el acceso al recurso externo.
Post-condiciones	Se obtendrá el dato del recurso externo.
Flujo principal	1. Este caso de uso se inicia cuando se debe de acceder a un recurso externo para obtener un dato. 2. El sistema cede el control a la lógica definida para el acceso al recurso externo.. i. Se devuelve el dato. 3. Fin del caso de uso.
Flujo alternativo	No aplica

Tabla 37. Caso de uso 008.CU - Obtener dato recurso externo

4.2.1.2 Caso de uso “Administrar caché”

Las acciones que se realizan en este caso de uso son las siguientes:

- **Administrar caché:** un administrador accede a la consola de administración para administrar la caché.
- **Administrar caché Nivel 1:** la consola de administración accede al Sistema de Gestión de Cachés para realizar las acciones solicitadas por el administrador en la caché de Nivel 1.
- **Administrar caché Nivel 2:** la consola de administración accede al Sistema de Gestión de Cachés para realizar las acciones solicitadas por el administrador en la caché de Nivel 2.

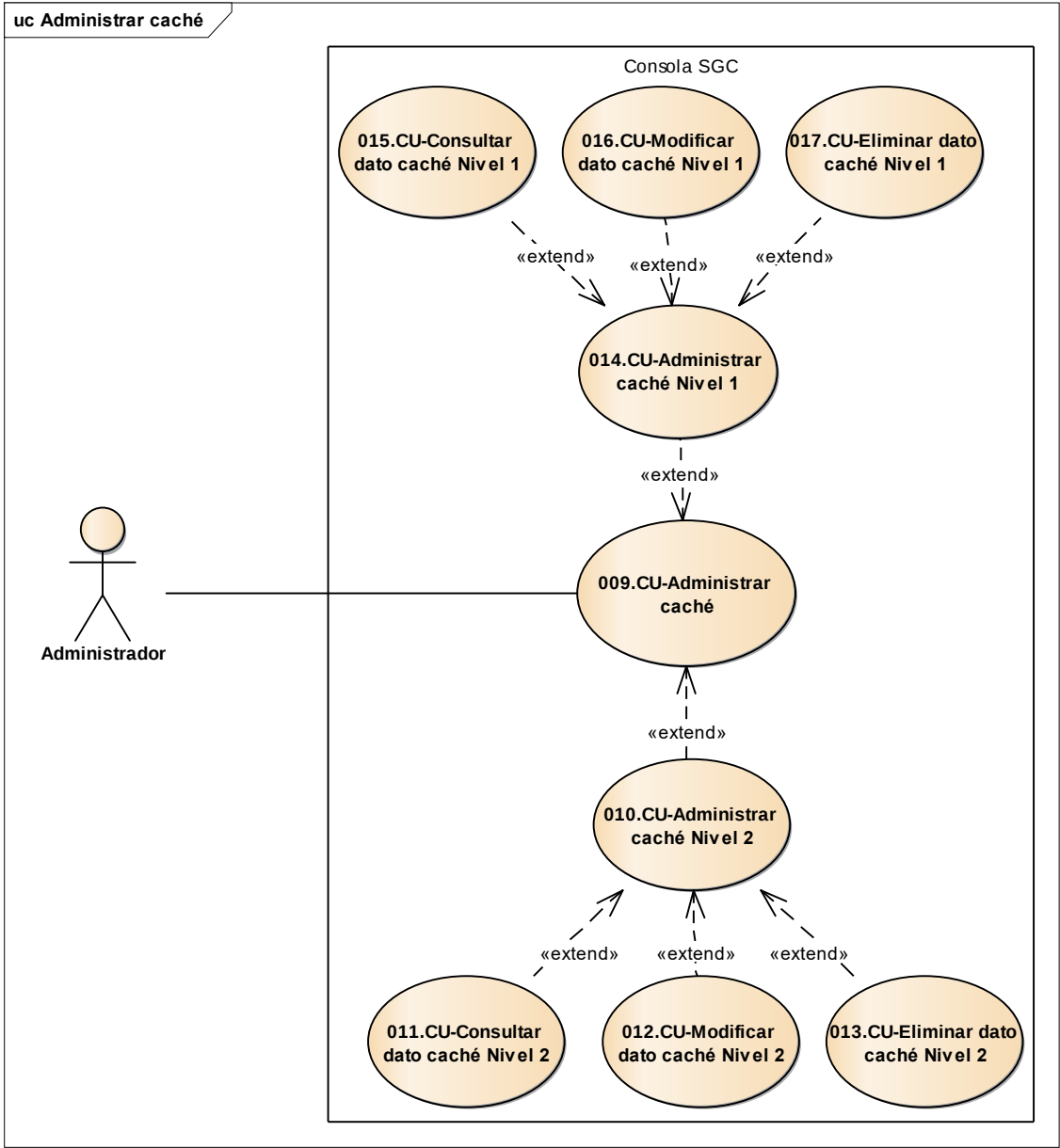


Figura 16. Casos de uso de administrar caché

009.CU-Administrar caché	
ID	009.CU
Nombre	Administrar caché
Descripción	Se ofrece una consola de administración que permite consultar la información almacenada.
Actor principal	Administrador
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, o alguna aplicación con caché de Nivel 1.
Post-condiciones	Se consultará la información de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 2. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones.

	- Consultar las claves almacenadas en una caché de Nivel 1. - Consultar las claves almacenadas en todas las cachés de Nivel 1. - Consultar las claves almacenadas en la caché de Nivel 2. - Consultar todas las claves almacenadas. - Si se tratan las claves de una caché de Nivel 1, se administra la caché de Nivel 1 010.CU. - Si se tratan las claves de una caché de Nivel 2, se administra la caché de Nivel 2 014.CU. - Fin del caso de uso
Flujo alternativo	No aplica

Tabla 38. Caso de uso 009.CU - Administrar caché

010.CU-Administrar caché Nivel 2	
ID	010.CU
Nombre	Administrar caché Nivel 2
Descripción	Se permite administrar la información almacenada en la caché de Nivel 2.
Actor principal	009.CU
Pre-condiciones	Es necesario que esté configurada y en funcionamiento la caché de Nivel 2.
Post-condiciones	Se administrará la información de la caché.
Flujo principal	- Este caso de uso se inicia cuando se administra la información de la caché de Nivel 2. - El sistema muestra las cachés existentes en la caché de Nivel 2 organizadas por aplicación. - Se permite consultar la información de la caché de Nivel 2 011.CU. - Se permite modificar la información de la caché de Nivel 2 012.CU. - Se permite eliminar la información de la caché de Nivel 2 013.CU. - Fin del caso de uso
Flujo alternativo	No aplica

Tabla 39. Caso de uso 010.CU - Administrar caché Nivel 2

011.CU-Consultar dato caché Nivel 2	
ID	011.CU
Nombre	Consultar dato caché Nivel 2
Descripción	Se permite consultar la información almacenada en la caché de Nivel 2.
Actor principal	010.CU
Pre-condiciones	Es necesario que esté configurada y en funcionamiento la caché de Nivel 2.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	- Este caso de uso se inicia cuando se accede a la consola de administración de la caché. - El sistema recolectará la información disponible de todas las

	cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones. - Consultar las claves almacenadas en una caché de Nivel 1. - Consultar las claves almacenadas en todas las cachés de Nivel 1. - Consultar las claves almacenadas en la caché de Nivel 2. - Consultar todas las claves almacenadas. 4. Si se consultan las claves de una caché de Nivel 1. - Se recupera la información del Nivel 1 de la caché del nodo indicado. - Se muestra en la consola. 5. Si se consultan las claves de todas las cachés de Nivel 1. - Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. - Se muestra en la consola. 6. Si se consultan las claves de la caché de Nivel 2. - Se recupera la información del Nivel 2 de la caché. - Se muestra en la consola. 7. Si se consultan todas las claves almacenadas. - Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. - Se muestra en la consola. 8. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 40. Caso de uso 011.CU - Consultar dato caché Nivel 2

012.CU- Modificar dato caché Nivel 2	
ID	012.CU
Nombre	Modificar dato caché Nivel 2
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	010.CU
Pre-condiciones	Es necesario que esté configurada y en funcionamiento la caché de Nivel 2.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	9. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 10. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 11. La consola ofrecerá las siguientes acciones. - Consultar las claves almacenadas en una caché de Nivel 1. - Consultar las claves almacenadas en todas las cachés de Nivel 1. - Consultar las claves almacenadas en la caché de Nivel 2. - Consultar todas las claves almacenadas. 12. Si se consultan las claves de una caché de Nivel 1. - Se recupera la información del Nivel 1 de la caché del

	nodo indicado. b. Se muestra en la consola. 13. Si se consultan las claves de todas las cachés de Nivel 1. a. Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. b. Se muestra en la consola. 14. Si se consultan las claves de la caché de Nivel 2. a. Se recupera la información del Nivel 2 de la caché. b. Se muestra en la consola. 15. Si se consultan todas las claves almacenadas. a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola. 16. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 41. Caso de uso 012.CU - Modificar dato caché Nivel 2

013.CU-Eliminar dato caché Nivel 2	
ID	013.CU
Nombre	Eliminar dato caché Nivel 2
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	010.CU
Pre-condiciones	Es necesario que esté configurada y en funcionamiento la caché de Nivel 2.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	- Este caso de uso se inicia cuando se accede a la consola de administración de la caché. - El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. - La consola ofrecerá las siguientes acciones. - Consultar las claves almacenadas en una caché de Nivel 1. - Consultar las claves almacenadas en todas las cachés de Nivel 1. - Consultar las claves almacenadas en la caché de Nivel 2. - Consultar todas las claves almacenadas. - Si se consultan las claves de una caché de Nivel 1.. - Se recupera la información del Nivel 1 de la caché del nodo indicado. - Se muestra en la consola. - Si se consultan las claves de todas las cachés de Nivel 1. - Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. - Se muestra en la consola. - Si se consultan las claves de la caché de Nivel 2. - Se recupera la información del Nivel 2 de la caché. - Se muestra en la consola. - Si se consultan todas las claves almacenadas.

	a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola.
Flujo alternativo	No aplica

Tabla 42. Caso de uso 013.CU - Eliminar dato caché Nivel 2

014.CU-Administrar caché Nivel 1	
ID	014.CU
Nombre	Administrar caché Nivel 1
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	009.CU
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, o alguna aplicación con caché de Nivel 1.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 2. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones. a. Consultar las claves almacenadas en una caché de Nivel 1. b. Consultar las claves almacenadas en todas las cachés de Nivel 1. c. Consultar las claves almacenadas en la caché de Nivel 2. d. Consultar todas las claves almacenadas. 4. Si se consultan las claves de una caché de Nivel 1. a. Se recupera la información del Nivel 1 de la caché del nodo indicado. b. Se muestra en la consola. 5. Si se consultan las claves de todas las cachés de Nivel 1. a. Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. b. Se muestra en la consola. 6. Si se consultan las claves de la caché de Nivel 2. a. Se recupera la información del Nivel 2 de la caché. b. Se muestra en la consola. 7. Si se consultan todas las claves almacenadas. a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola. 8. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 43. Caso de uso 014.CU - Administrar caché Nivel 1

015.CU-Consultar dato caché Nivel 1	
ID	015.CU
Nombre	Consultar dato caché Nivel 1
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	014.CU
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, o alguna aplicación con caché de Nivel 1.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 2. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones. a. Consultar las claves almacenadas en una caché de Nivel 1. b. Consultar las claves almacenadas en todas las cachés de Nivel 1. c. Consultar las claves almacenadas en la caché de Nivel 2. d. Consultar todas las claves almacenadas. 4. Si se consultan las claves de una caché de Nivel 1. a. Se recupera la información del Nivel 1 de la caché del nodo indicado. b. Se muestra en la consola. 5. Si se consultan las claves de todas las cachés de Nivel 1. a. Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. b. Se muestra en la consola. 6. Si se consultan las claves de la caché de Nivel 2. a. Se recupera la información del Nivel 2 de la caché. b. Se muestra en la consola. 7. Si se consultan todas las claves almacenadas. a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola. 8. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 44. Caso de uso 015.CU - Consultar dato caché Nivel 1

016.CU- Modificar dato caché Nivel 1	
ID	016.CU
Nombre	Modificar dato caché Nivel 1
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	014.CU
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, o alguna aplicación con caché de Nivel 1.

Post-condiciones	Se modificará la información de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 2. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones. a. Consultar las claves almacenadas en una caché de Nivel 1. b. Consultar las claves almacenadas en todas las cachés de Nivel 1. c. Consultar las claves almacenadas en la caché de Nivel 2. d. Consultar todas las claves almacenadas. 4. Si se consultan las claves de una caché de Nivel 1. a. Se recupera la información del Nivel 1 de la caché del nodo indicado. b. Se muestra en la consola. 5. Si se consultan las claves de todas las cachés de Nivel 1. a. Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. b. Se muestra en la consola. 6. Si se consultan las claves de la caché de Nivel 2. a. Se recupera la información del Nivel 2 de la caché. b. Se muestra en la consola. 7. Si se consultan todas las claves almacenadas. a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola. 8. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 45. Caso de uso 016.CU - Modificar dato caché Nivel 1

017.CU-Eliminar dato caché Nivel 1	
ID	017.CU
Nombre	Eliminar dato caché Nivel 1
Descripción	Se ofrece una consola de administración que permite actuar sobre la información almacenada.
Actor principal	014.CU
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, o alguna aplicación con caché de Nivel 1.
Post-condiciones	Se modificará la información de la caché.
Flujo principal	1. Este caso de uso se inicia cuando se accede a la consola de administración de la caché. 2. El sistema recolectará la información disponible de todas las cachés existentes y la mostrará gráficamente. 3. La consola ofrecerá las siguientes acciones. a. Consultar las claves almacenadas en una caché de Nivel 1. b. Consultar las claves almacenadas en todas las cachés de

	Nivel 1. c. Consultar las claves almacenadas en la caché de Nivel 2. d. Consultar todas las claves almacenadas. 4. Si se consultan las claves de una caché de Nivel 1.. a. Se recupera la información del Nivel 1 de la caché del nodo indicado. b. Se muestra en la consola. 5. Si se consultan las claves de todas las cachés de Nivel 1. a. Se recupera la información del Nivel 1 de todas las cachés de los nodos administrados. b. Se muestra en la consola. 6. Si se consultan las claves de la caché de Nivel 2. a. Se recupera la información del Nivel 2 de la caché. b. Se muestra en la consola. 7. Si se consultan todas las claves almacenadas. a. Se recupera la información del Nivel 1 de la caché de todos los nodos, así como la información de la caché de Nivel 2. b. Se muestra en la consola. 8. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 46. Caso de uso 017.CU - Eliminar dato caché Nivel 1

4.2.1.3 Caso de uso “Configurar caché”

Las acciones que se realizan en este caso de uso son las siguientes:

- **Configurar aplicación:** un administrador ofrece los recursos necesarios a la aplicación que va a utilizar el Sistema de Gestión de Cachés.
- **Instalar y configurar caché de Nivel 2:** un administrador instala y configura correctamente para su uso una caché de Nivel 2.

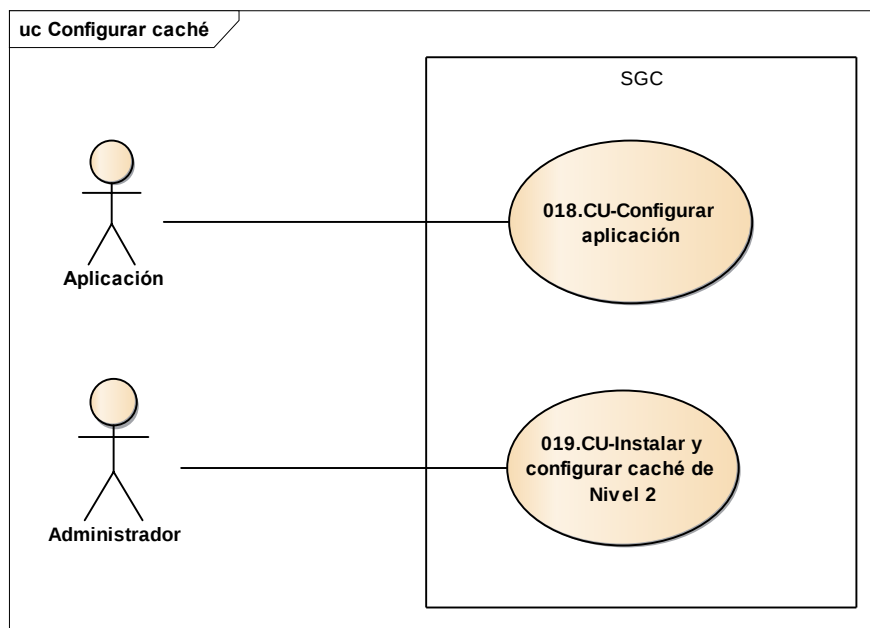


Figura 17. Casos de uso de configurar caché

018.CU-Configurar aplicación	
ID	018.CU
Nombre	Configurar aplicación
Descripción	Se ofrece un fichero de configuración que deberán de adaptar las aplicaciones que utilicen el sistema de gestión de cachés. En dicho fichero deberá indicarse el nivel de caché utilizado (Nivel 1, Nivel 2 o ambos), los proveedores de caché utilizados en cada uno de los niveles, si la caché será administrada lo cual indicará que admitirá los comandos mandados por la consola indicada en la configuración.
Actor principal	Aplicación
Pre-condiciones	Es necesario que esté configurada la caché. Y que esté en funcionamiento la caché de Nivel 2, y la consola de administración si se configura el uso de las mismas.
Post-condiciones	La aplicación podrá hacer uso de las cachés configuradas y aparecerá en la consola de administración si se ha configurado su uso.
Flujo principal	1. Este caso de uso se inicia cuando una aplicación va a hacer uso del sistema de gestión de cachés. 2. La aplicación debe de configurar el uso que va a realizar del sistema de gestión de cachés. 3. Se deben de configurar los siguientes elementos. a. Nivel. b. Si las cachés de la aplicación van a ser administradas. 4. Si se configura uso de caché de Nivel 1 a. Se debe de indicar cuál es el proveedor de Nivel 1. 5. Si se configura uso de caché de Nivel 2. a. Se debe indicar cuál es el proveedor de Nivel 2. b. Hay que configurar los datos de conexión a la caché de Nivel 2. 6. Si la caché es administrada. a. Se debe de configurar los datos de conexión para ser accesible por la consola de administración. 7. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 47. Caso de uso 018.CU - Configurar aplicación

019.CU-Instalar y configurar caché de Nivel 2	
ID	019-CU
Nombre	Configurar caché de Nivel 2
Descripción	Es necesario tener correctamente instalada y configurada una caché de Nivel 2.
Actor principal	Administrador
Pre-condiciones	Es necesario disponer de un conjunto de máquinas conectadas entre sí y accesibles desde las aplicaciones que hagan uso del sistema de gestión de cachés.
Post-condiciones	Estará correctamente instalada y configurada la caché de Nivel 2..
Flujo principal	1. Este caso de uso se inicia cuando se decide la instalación de una caché de Nivel 2. 2. Se debe de seleccionar el proveedor que se desee para que de servicio como caché de Nivel 2..

	3. Se debe de obtener el software necesario, y las guías de instalación y configuración. 4. Se debe proceder a la instalación y configuración. 5. Una vez finalizado el punto anterior es necesario validar el correcto funcionamiento de la caché de Nivel 2. 6. Fin del caso de uso
Flujo alternativo	No aplica

Tabla 48. Caso de uso 019.CU - Instalar y configurar aplicación

4.3 Requisitos de software

En este apartado se describen los requisitos que deben de ser satisfechos por el sistema, han sido extraídos del análisis de los requisitos de usuario.

4.3.1 Requisitos funcionales

Identificador: 001.RS.F-Almacenar dato				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema permitirá almacenar datos en caché en memoria.			
	La información a almacenar tendrá el formato <Clave, Valor>. Siendo la Clave el identificador para poder acceder al dato, y el Valor el propio dato.			
	Ambos deben de ser serializables.			

Tabla 49. Requisito 001.RS.F-Almacenar dato

Identificador: 002.RS.F-Recuperar dato				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema permitirá recuperar datos de la caché en memoria. La información a recuperar se realizará a partir de la Clave y se devolverá el valor almacenado.			

Tabla 50. Requisito 002.RS.F-Recuperar dato

Identificador: 003.RS.F-Acceso concurrente				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema permitirá el acceso concurrente tanto a la caché de Nivel 1 como a la caché de Nivel 2.			

Tabla 51. Requisito 003.RS.F-Recuperar dato

Identificador: 004.RS.F-Consulta de información almacenada				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		

Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El sistema permitirá la consulta de la información almacenada tanto en la caché de Nivel 1 como en la caché de Nivel 2.		

Tabla 52. Requisito 004.RS.F-Consulta de información almacenada

Identificador: 005.RS.F-Borrado de información			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Un usuario con perfil administrador podrá borrar la información almacenada en la caché.		

Tabla 53. Requisito 005.RS.F-Borrado de información

Identificador: 006.RS.F-Configurar el sistema			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Un usuario con perfil administrador podrá configurar en tiempo de ejecución el sistema de cachés.		

Tabla 54. Requisito 006.RS.F-Configurar el sistema

Identificador: 007.RS.F-Caducidad por defecto			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Se establecerá una caducidad por defecto en la caché de Nivel 1 de 21600 segundos, y en la caché de Nivel 2 de 86400 segundos.		

Tabla 55. Requisito 007.RS.F-Caducidad por defecto

4.3.2 Requisitos de Interfaz

Identificador: 001.RS.I-Consola AngularJS			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	La consola de administración se realizará en AngularJS.		

Tabla 56. Requisito 001.RS.I-Consola AngularJS

Identificador: 002.RS.I-Sistema portable			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		

Descripción	El sistema se realizará en Java para garantizar su portabilidad.
-------------	--

Tabla 57. Requisito 002.RS.I-Sistema portable

4.3.3 Requisitos de Operación

Identificador: 001. RS.O-Almacenar dato				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema permitirá almacenar datos en caché en memoria. La información a almacenar tendrá el formato <Clave, Valor>. Siendo la Clave el identificador para poder acceder al dato, y el Valor el propio dato. Ambos deben de ser serializables.			

Tabla 58. Requisito 001.RS.O-Almacenar dat

4.3.4 Requisitos de Recursos

Identificador: 001.RS.R-Ordenadores en red				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema se ejecutará en ordenadores conectados en red.			

Tabla 59. Requisito 001.RS.R-Ordenadores en red

Identificador: 002.RS.R-Máquina virtual Java				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema se ejecutará en ordenadores que tengan correctamente instalado el JRE 8.			

Tabla 60. Requisito 002.RS.R-Máquina virtual Java

Identificador: 003.RS.R-Proveedores soportados				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional			
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja	
Estabilidad	☒ Alta ☐ Media ☐ Baja			
Descripción	El sistema admitirá como mínimo los siguientes proveedores Infinispan/DataGrid, ehCache, Hazelcast, Coherence.			

Tabla 61. Requisito 003.RS.R-Proveedores soportados

4.3.5 Requisitos de Verificación

Identificador: 001.RS.V-Acceso simultáneo				
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente		

Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Se gestiona correctamente el caso del acceso a un datp simultáneamente por más de un nodo.		

Tabla 62. Requisito 001.RS.V-Acceso simultáneo

4.3.6 Requisitos de Documentación

Identificador: 001.RS.D-Documentación completa			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Se realizará toda la documentación en español.		

Tabla 63. Requisito 001.RS.V-El sistema será portable

4.3.7 Requisitos de Portabilidad

Identificador: 001.RS.P-El sistema será portable			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El sistema será portable a cualquier ordenador que cumpla los requerimientos de recursos.		

Tabla 64. Requisito 002.RS.V-El sistema será portable

Identificador: 002.RS.P-La caché de N2 será portable			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	La caché de N2 será portable a cualquier ordenador que cumpla las especificaciones indicadas por el proveedor de dicha caché.		

Tabla 65. Requisito 003.RS.V-La caché de N2 será portable

4.3.8 Requisitos de Fiabilidad

Identificador: 001.RS.Fi-El sistema funcionará de forma óptima			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Se garantiza un correcto funcionamiento del sistema siempre que se cumplan los requisitos mínimos de hardware, de infraestructura y este correctamente configurado.		

Tabla 66. Requisito 001.RS.iF-El sistema funcionará de forma óptima

4.3.9 Requisitos de Mantenimiento

Identificador: 001.RS.M-Mantenimiento del sistema sencillo			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El sistema deberá tener una codificación que permita un mantenimiento no complejo, estando el código correctamente documentado.		

Tabla 67. Requisito 001.RS.M-La caché de N2 será portable

4.3.10 Requisitos de Seguridad

Identificador: 001.RS.S-Acceso securizado			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	El acceso a la consola de administración estará controlado por usuario y contraseña.		

Tabla 68. Requisito 001.RS.S-Acceso securizado

Identificador: 002.RS.S-Confirmación de operaciones			
Prioridad	☒ Alta ☐ Media ☐ Baja	Fuente	
Necesidad	☒ Esencial ☐ Deseable ☐ Opcional		
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Estabilidad	☒ Alta ☐ Media ☐ Baja		
Descripción	Para las acciones de modificación y borrado se mostrará una ventana de confirmación		

Tabla 69. Requisito 002.RS.S-Confirmación de operaciones

4.4 Diseño arquitectónico

El Sistema de Gestión de Caches ofrece una capa adicional entre la aplicación y los servicios externos, de tal forma que la aplicación que utilice el Sistema de Gestión de Cachés no accede directamente a los recursos externos si estos ya se encuentran almacenados en la caché definida.



Figura 18. Capa de caché

La siguiente ilustración muestra una descripción gráfica de la arquitectura del Sistema de Gestión de Cachés, en el que se ve la composición de las cachés de Nivel 1 y de Nivel 2.

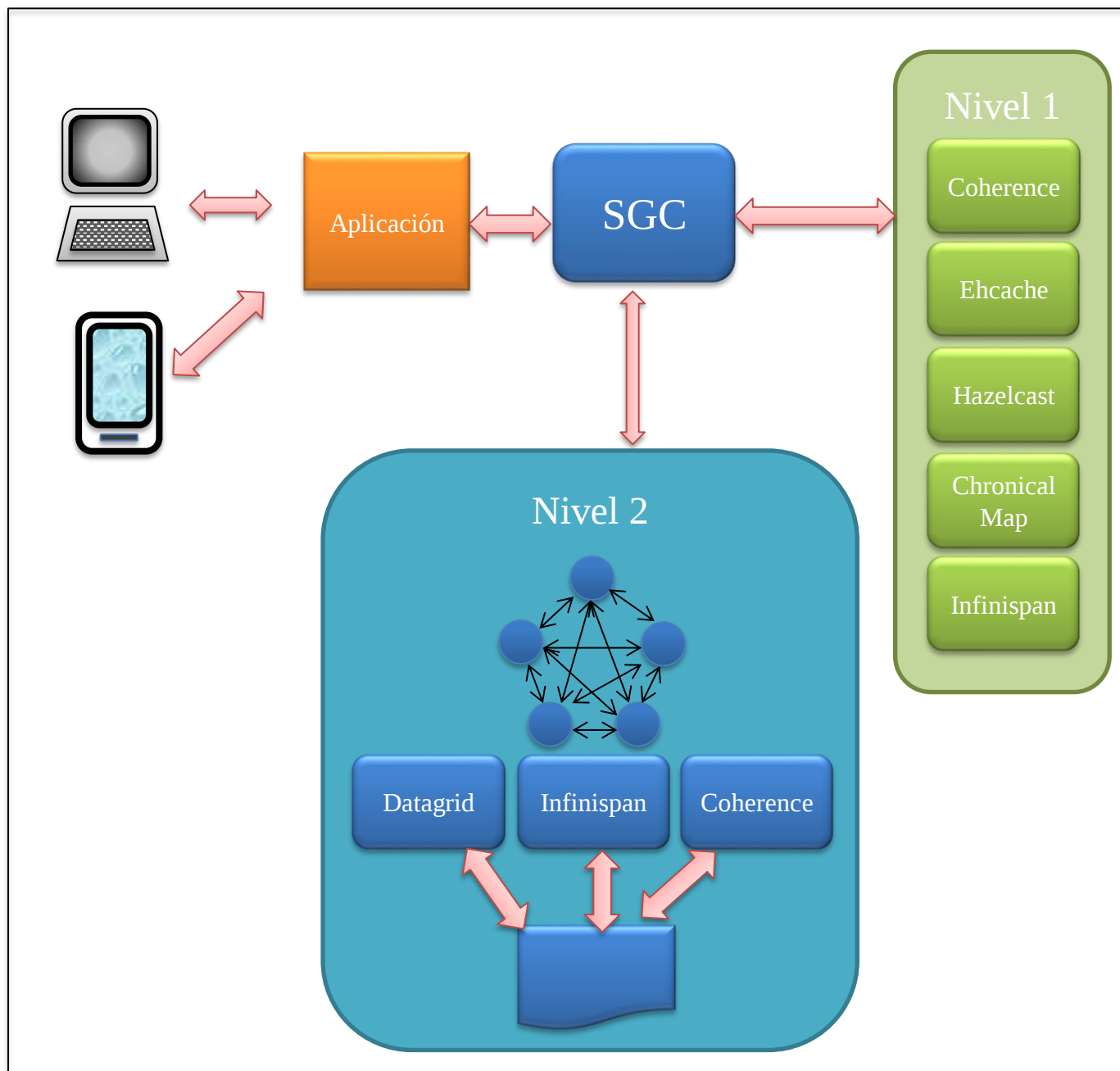


Figura 19. Arquitectura del SGC

En la siguiente figura se ve el diseño correspondiente a la consola de administración del Sistema de Gestión de Cachés.

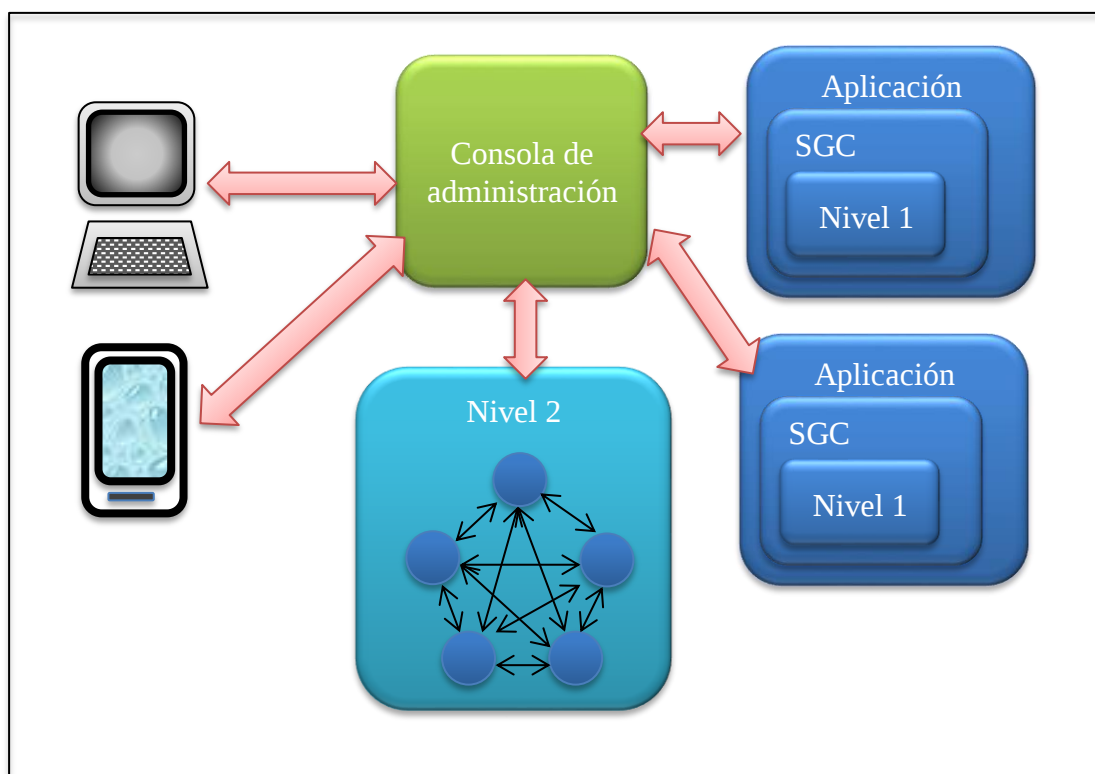


Figura 20. Arquitectura de la consola de administración

CAPÍTULO 5

5. Diseño detallado

En este apartado se debería realizar la implementación del sistema, lo cual queda fuera del alcance del presente proyecto. Por lo que el principal objetivo dentro del diseño detallado es detallar el diseño planteado durante la etapa anterior RS/DA, para posteriormente poder codificarlo, documentarlo y probarlo.

5.1 *Diseño detallado de la consola de administración*

Se describen a continuación las principales pantallas que conforman la aplicación, así como una breve descripción de cada una de ellas, y la funcionalidad que deben de contener.

5.1.1 Pantalla principal de la aplicación

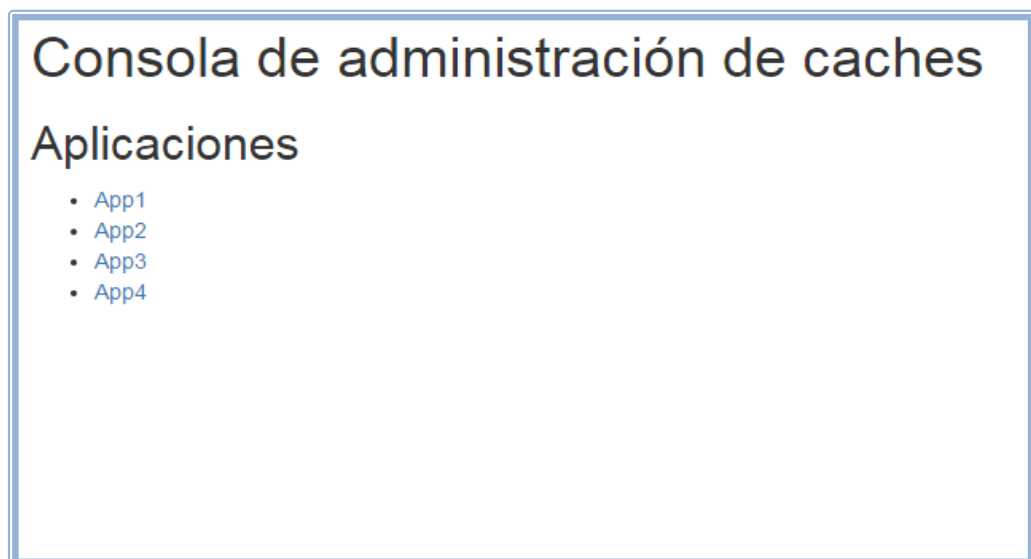


Figura 21. Pantalla principal de la consola de administración

En la pantalla principal se deberán mostrar todas las aplicaciones administradas. Las aplicaciones administradas deberán contener en su configuración la indicación de que van a ser administradas para que admitan los comandos que lleguen desde la interfaz, y deben de tener acceso al backend de la consola para poder construir un clúster con la propia consola.

5.1.2 Pantalla de consulta de las cachés de una aplicación

Al pulsar sobre una de las aplicaciones se debe de mostrar la información de todas las cachés utilizadas por la aplicación seleccionada y para cada una de las instancias de dicha aplicación. La pantalla debe de contener al menos la información mostrada a continuación, adaptando el aspecto tal como se considere oportuno.

Consola de administración de caches

Aplicaciones

- App1
- App2
- App3
- App4

Administracion de caches App1

Existen 2 caches administradas

Borrar cache

Informacion de la cache traficoBD

Cache de nivel 2

Borrar cache de nivel 2

Ver claves de nivel 2

Nodos que utilizan cache de nivel 2

NODE1-16274

NODE1-10332

Cache de nivel 1

Borrar cache del nodo

Ver claves del nodo

NODE1-16274 (1)

Borrar cache del nodo

Ver claves del nodo

NODE1-10332 (1)

Borrar cache

Informacion de la cache traficoServ

Cache de nivel 2

Borrar cache de nivel 2

Ver claves de nivel 2

Nodos que utilizan cache de nivel 2

NODE1-16274

NODE1-10332

Cache de nivel 1

Borrar cache del nodo

Ver claves del nodo

NODE1-16274 (0)

Borrar cache del nodo

Ver claves del nodo

NODE1-10332 (0)

Figura 22. Pantalla de consulta de las cachés de una aplicación

En el ejemplo mostrado y tal como se ve en la siguiente figura se indica que la aplicación utiliza dos cachés traficoBD y traficoServ. En cada una de las cachés se indica los nodos existentes para la caché de Nivel 1 y la caché de Nivel 2.

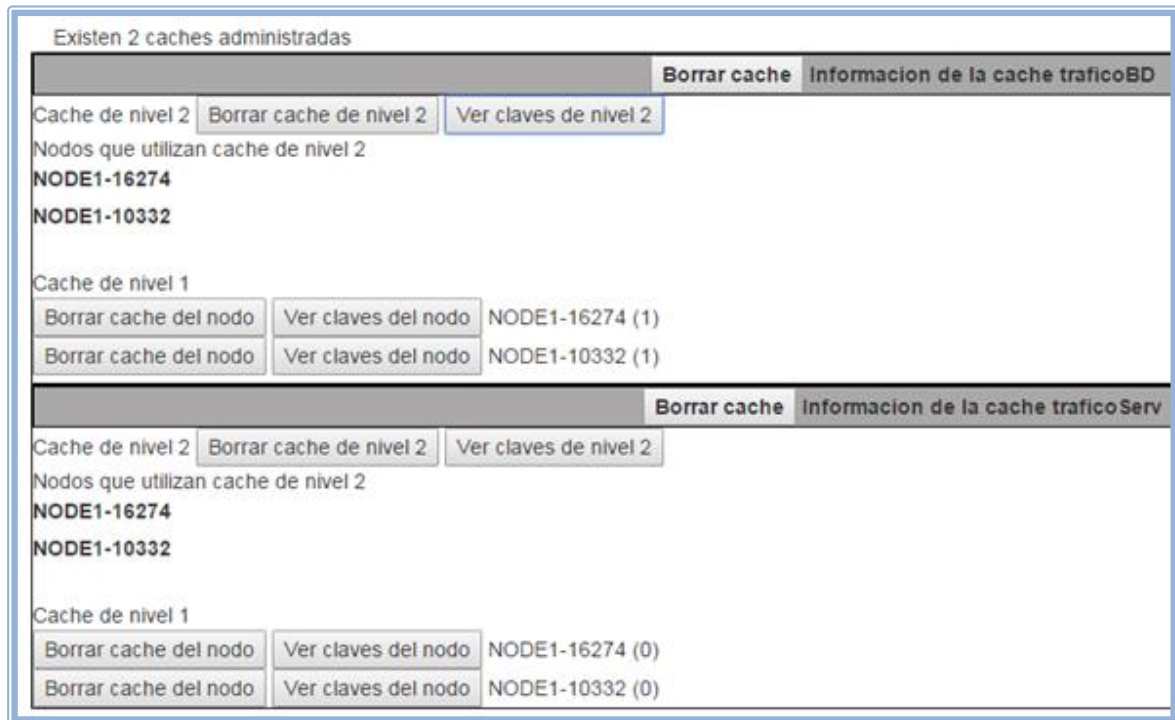


Figura 23. Pantalla indicando las cachés de una aplicación

En la pantalla anterior se ve una sección para cada una de las cachés utilizadas. Dentro de cada una de las secciones se muestra la información de todas las cachés de Nivel 1 utilizadas, si se ha configurado su uso por parte de la aplicación. En el ejemplo siguiente se indica que la caché de Nivel 1 está presente en dos nodos y en cada uno de los nodos existe un dato almacenado.

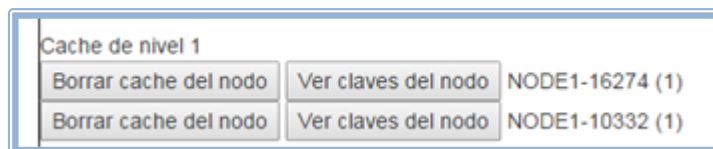


Figura 24. Información de la caché de Nivel 1

También se muestra la información de la caché de Nivel 2 si se está utilizando, para esta aplicación, en el ejemplo mostrado se indica que existen dos nodos que están almacenando datos en la caché de Nivel 2.

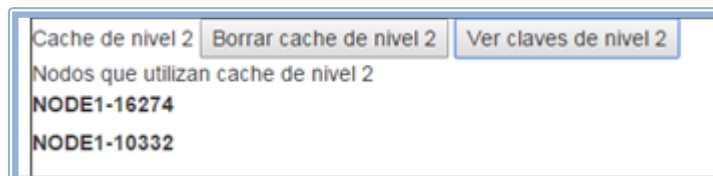


Figura 25. Información de la caché de Nivel 2

Desde la ventana principal pulsando el botón **Borrar cache** se debe de borrar la información presente en la caché en los niveles configurados, en este ejemplo si se pulsa el botón en la caché traficoBD, se borrará la información, mostrándose la información actualizada tal como se indica a continuación donde se observa que no hay datos almacenados.

Consola de administración de caches

Aplicaciones

- App1
- App2
- App3
- App4

Administracion de caches App1

Existen 2 caches administradas

		Borrar cache	Informacion de la cache traficoBD
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2	
Nodos que utilizan cache de nivel 2			
NODE1-16274			
NODE1-10332			
Cache de nivel 1			
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (0)	
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (0)	
		Borrar cache	Informacion de la cache traficoServ
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2	
Nodos que utilizan cache de nivel 2			
NODE1-16274			
NODE1-10332			
Cache de nivel 1			
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (0)	
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (0)	

Figura 26. Pantalla indicando una aplicación sin datos almacenados

5.1.3 Pantalla de consulta la información de la caché de Nivel 2

Desde la pantalla principal de la consola de administración de cachés se puede pulsar el botón **Ver claves de nivel 2** que lanzará un comando que una vez ejecutado mostrará la información de la caché de Nivel 2, tal como se ve en la figura siguiente.

Consola de administración de caches

Aplicaciones

- App1
- App2
- App3
- App4

Administracion de caches App1

Existen 2 caches administradas

		Borrar cache	Informacion de la cache traficoBD
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2	
Nodos que utilizan cache de nivel 2			
NODE1-16274			
NODE1-10332			
Cache de nivel 1			
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (1)	
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (1)	
Contenido de la cache traficoBD del nodo LEVEL2			
Key: traficoBDbdidiomas		Modificar dato	
		Borrar cache	Informacion de la cache traficoServ
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2	
Nodos que utilizan cache de nivel 2			
NODE1-16274			
NODE1-10332			
Cache de nivel 1			
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (0)	
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (0)	

Figura 27. Consulta de la caché de Nivel 2

En esta pantalla se observa que hay un dato almacenado con la clave traficoBDbdidiomas, para cada uno de los datos almacenados se puede modificar

pulsando el botón **Modificar dato** que permitirá introducir un nuevo valor que será enviado a la caché de Nivel 2.

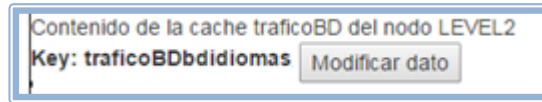


Figura 28. Información de la caché de Nivel 2

Desde la pantalla principal de la consola, se puede realizar un borrado de la información de la caché de Nivel 2 pulsando el botón **Borrar cache de nivel 2** que realizará el borrado de la información de la caché de Nivel 2.

5.1.4 Pantalla de consulta la información de la caché de Nivel 1

De forma semejante a como se ha actuado con la caché de Nivel 2, desde la pantalla principal de la consola de administración de cachés se puede pulsar el botón **Ver claves del nodo**, para un nodo determinado de la aplicación de a que se quiere consultar la información almacenada, que lanzará un comando que una vez ejecutado mostrará la información de la caché de Nivel 1 de ese nodo, tal como se ve en la figura siguiente.



Figura 29. Consulta de la caché de Nivel 1 de un nodo

En esta pantalla se observa que hay un dato almacenado con la clave traficoBDdbdiomas, para cada uno de los datos almacenados se puede modificar pulsando el botón **Modificar dato** que permitirá introducir un nuevo valor que será enviado a la caché de Nivel 1. Como podemos ver el dato almacenado en este nodo es el mismo que el almacenado en la caché de Nivel 2.

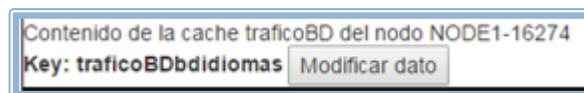


Figura 30. Información de la caché de Nivel 1

Desde la pantalla principal de la consola, se puede realizar un borrado de la información de la caché de Nivel 1 de uno nodo pulsando el botón **Borrar cache del nodo** que realizará el borrado de la información de la caché de Nivel 1 de dicho nodo, mostrando la información actualizada en la pantalla.

Consola de administración de caches

Aplicaciones

- App1
- App2
- App3
- App4

Administracion de caches App1

Existen 2 caches administradas

	Borrar cache	Informacion de la cache traficoBD
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2
Nodos que utilizan cache de nivel 2		
NODE1-16274		
NODE1-10332		
Cache de nivel 1		
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (0)
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (0)

	Borrar cache	Informacion de la cache traficoServ
Cache de nivel 2	Borrar cache de nivel 2	Ver claves de nivel 2
Nodos que utilizan cache de nivel 2		
NODE1-16274		
NODE1-10332		
Cache de nivel 1		
Borrar cache del nodo	Ver claves del nodo	NODE1-16274 (0)
Borrar cache del nodo	Ver claves del nodo	NODE1-10332 (0)

Figura 31. Pantalla indicando una aplicación sin datos almacenados en Nivel 1

5.1.5 Diagrama de clases de la consola de administración

A continuación se muestran los diagramas de clases del backend de la consola de administración, se muestran los diagramas correspondientes a la gestión de los comandos que la consola enviará a las aplicaciones, en el diagrama se muestran las interfaces y las clases necesarias para definir el gestor de comandos, en las clases si indican los métodos que deberían estar definidos, para proporcionar la funcionalidad necesaria al gestor de comandos de la consola.

Adicionalmente tanto la consola como las aplicaciones hacen uso de las clases que suministra el SGC, que son descritas en **5.2.1 Diagrama de clases del modelo de datos**.

Diseño detallado

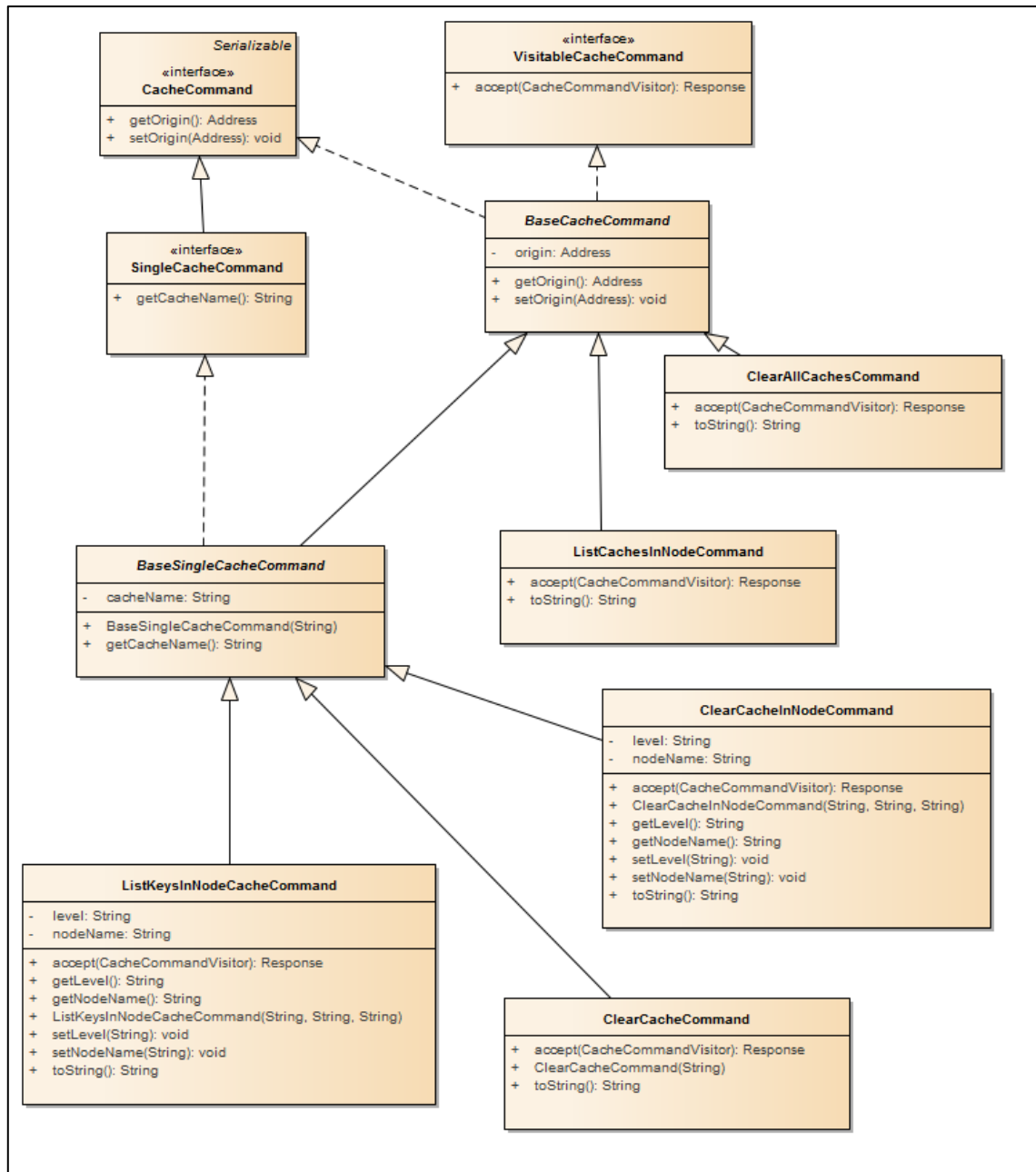


Figura 32. Diagrama de clases de los comandos

A continuación se muestra el diagrama de clases de las posibles respuestas dadas por los comandos invocados, en este diagrama se detallan las interfaces y clases necesarias, indicando los métodos que se deben definir e implementar para dar la funcionalidad necesaria a las respuestas dadas por los comandos.

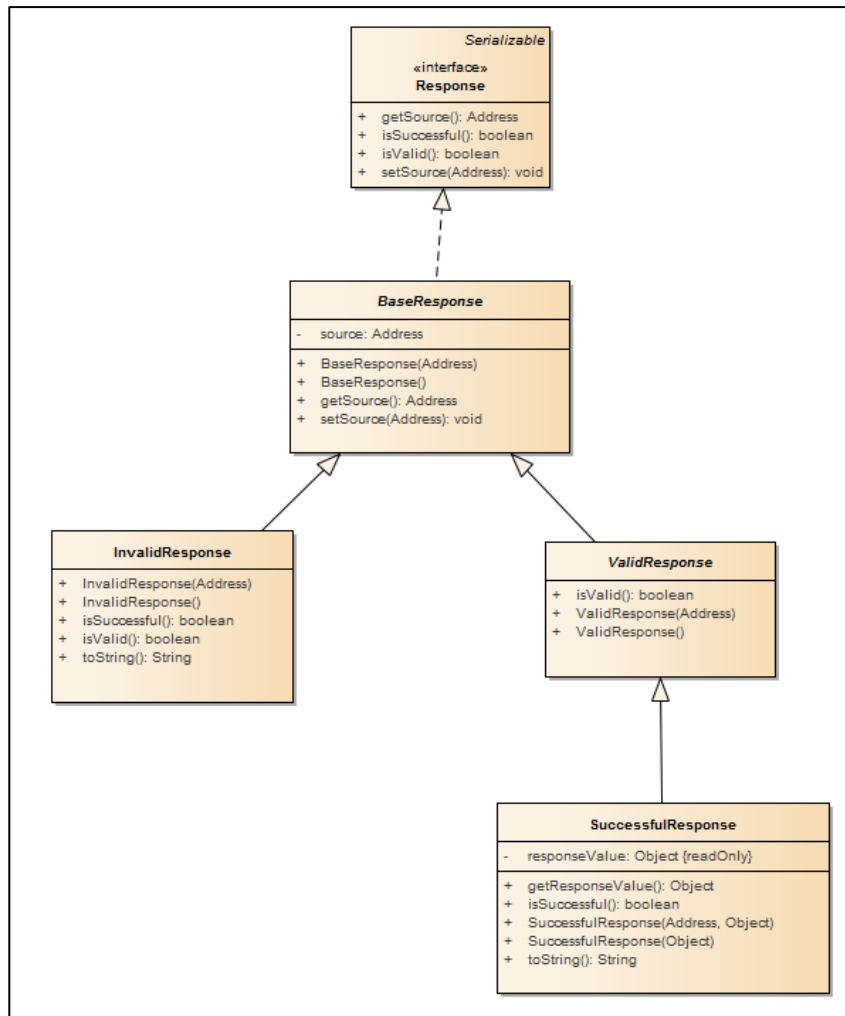


Figura 33. Diagrama de clases de las respuestas de los comandos

5.1.6 Diagrama de secuencia de la consola de administración

La siguiente figura muestra el diagrama de secuencia cuando el usuario consulta las caches en la consola de administración. La consola como resultado de la acción mostrará al usuario una ventana con la información contenida en las caches utilizadas por la aplicación.

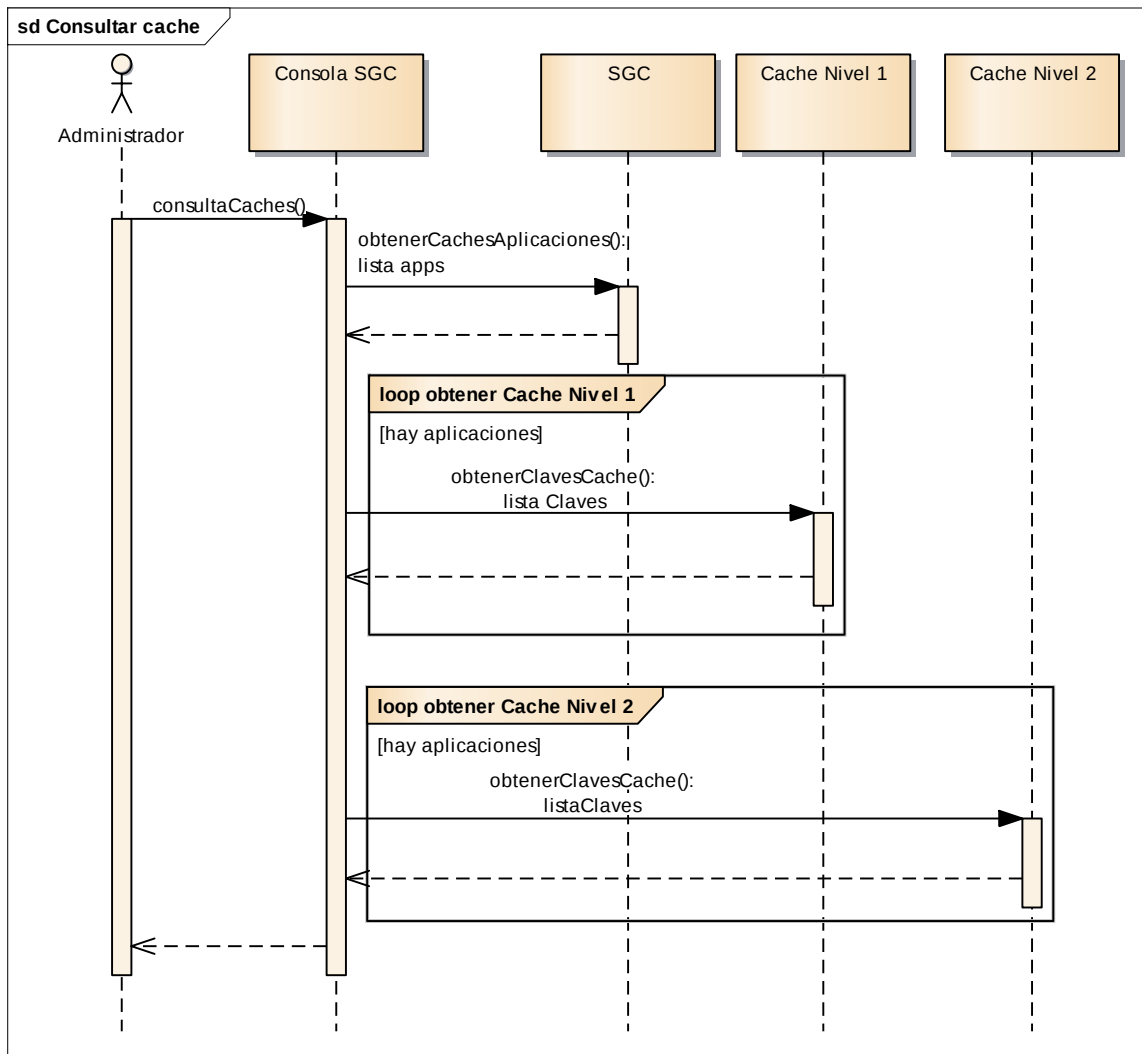


Figura 34. Diagrama de secuencia de la consola de administración

La siguiente figura muestra el diagrama de secuencia cuando el usuario elimina información de la cache de nivel 2 de una aplicación, previamente a pulsar la opción de eliminar dato se ha realizado el diagrama de secuencia de consultar caché.

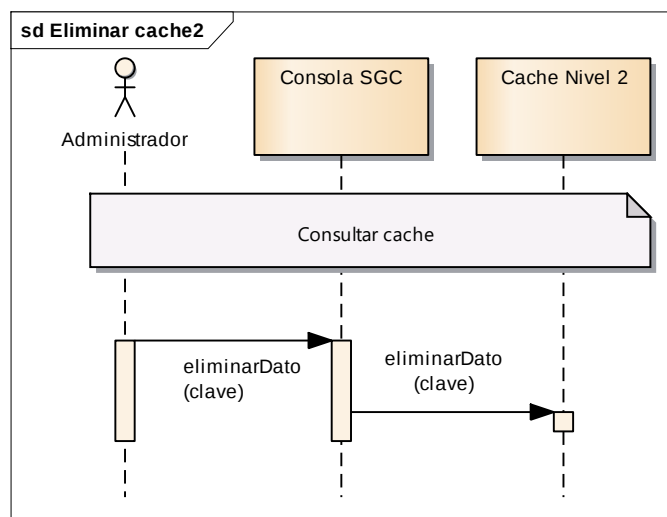


Figura 35. Diagrama de secuencia de eliminar dato

La siguiente figura muestra el diagrama de secuencia cuando el usuario modifica un dato de la cache de nivel 2 de una aplicación, previamente a pulsar la opción de eliminar dato se ha realizado el diagrama de secuencia de consultar caché.

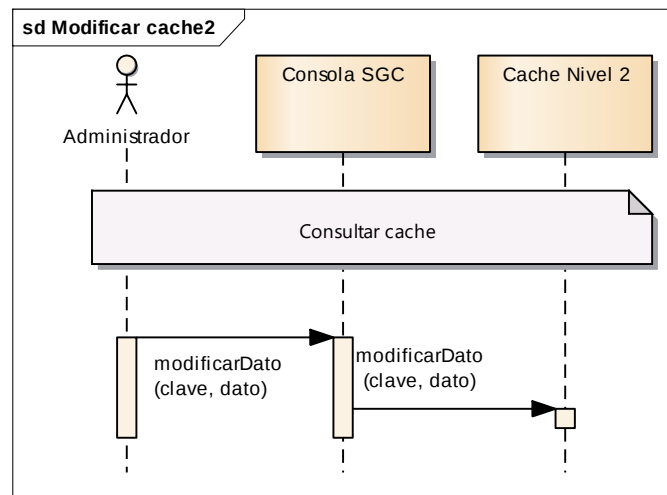


Figura 36. Diagrama de secuencia de modificar dato

Los diagramas de secuencia correspondientes eliminar dato y modificar dato de la caché de Nivel 1 son semejantes a los realizados para la caché de nivel 2.

5.2 Diseño detallado del modelo de datos

Se describen a continuación los diagramas de clases del modelo de datos del Sistema de Gestión de Cachés, también se indican los diagramas de secuencia.

5.2.1 Diagrama de clases del modelo de datos

El siguiente diagrama corresponde al modelo de datos de las cachés, en él se muestran las interfaces y clases necesarias para definir la funcionalidad necesaria para la gestión de las cachés de Nivel 1 y de Nivel 2. La clase ProviderCache indica una clase específica para un proveedor, debiendo existir una clase por proveedor soportado en el Nivel 1 por ejemplo (EhCacheCache, InfinispanCache, HazelCastCache).

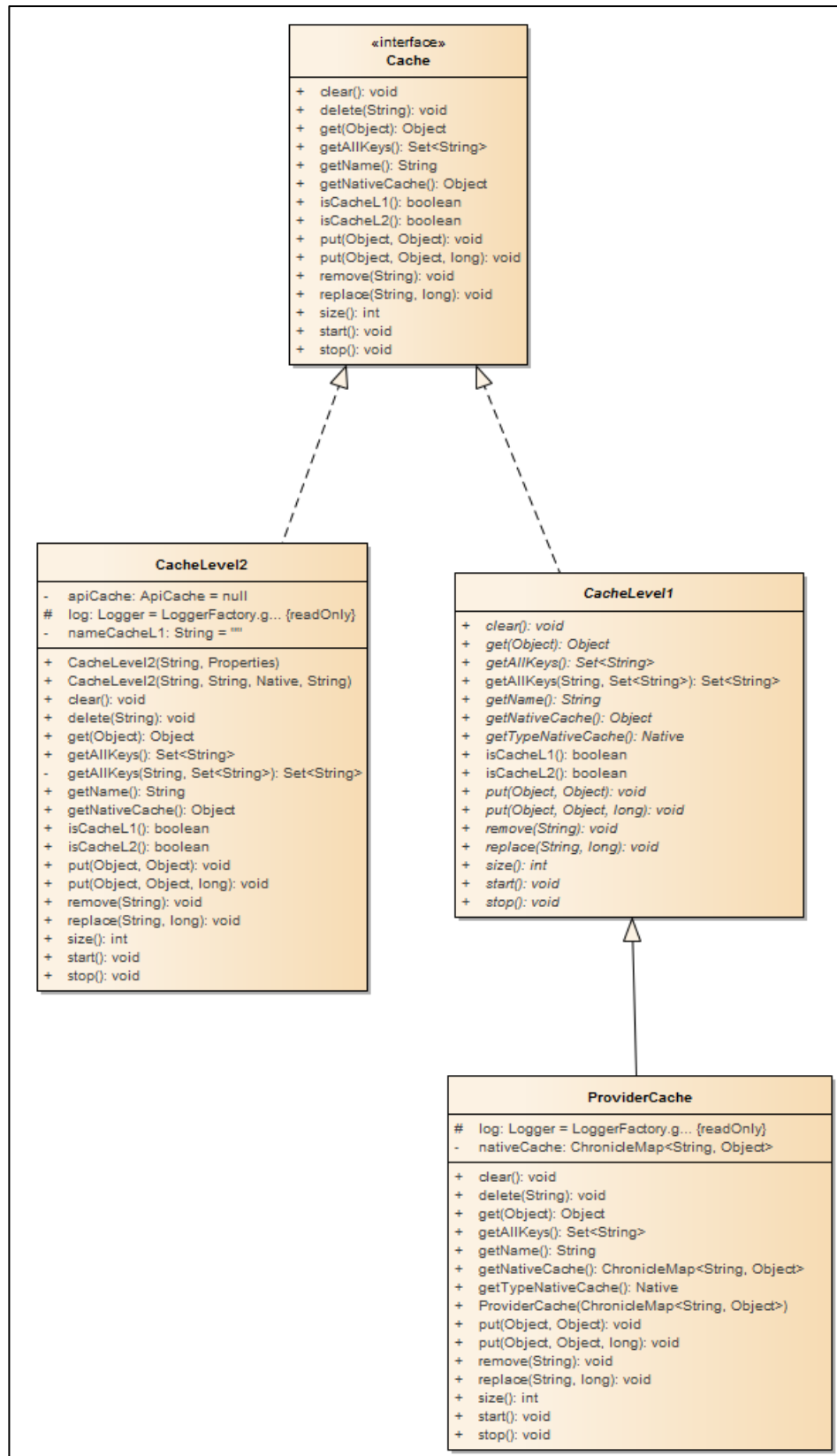


Figura 37. Diagrama de clases de las cachés

El siguiente diagrama muestra las interfaces y clases, junto con los métodos necesarios que se deben definir para el gestor de cachés, estas clases serán las encargadas de gestionar las caches definidas, y de interceptar la llamada realizada por la aplicación

para hacer las comprobaciones necesarias en las cachés definidas para verificar si el dato ya ha sido solicitado anteriormente y almacenado en la caché.

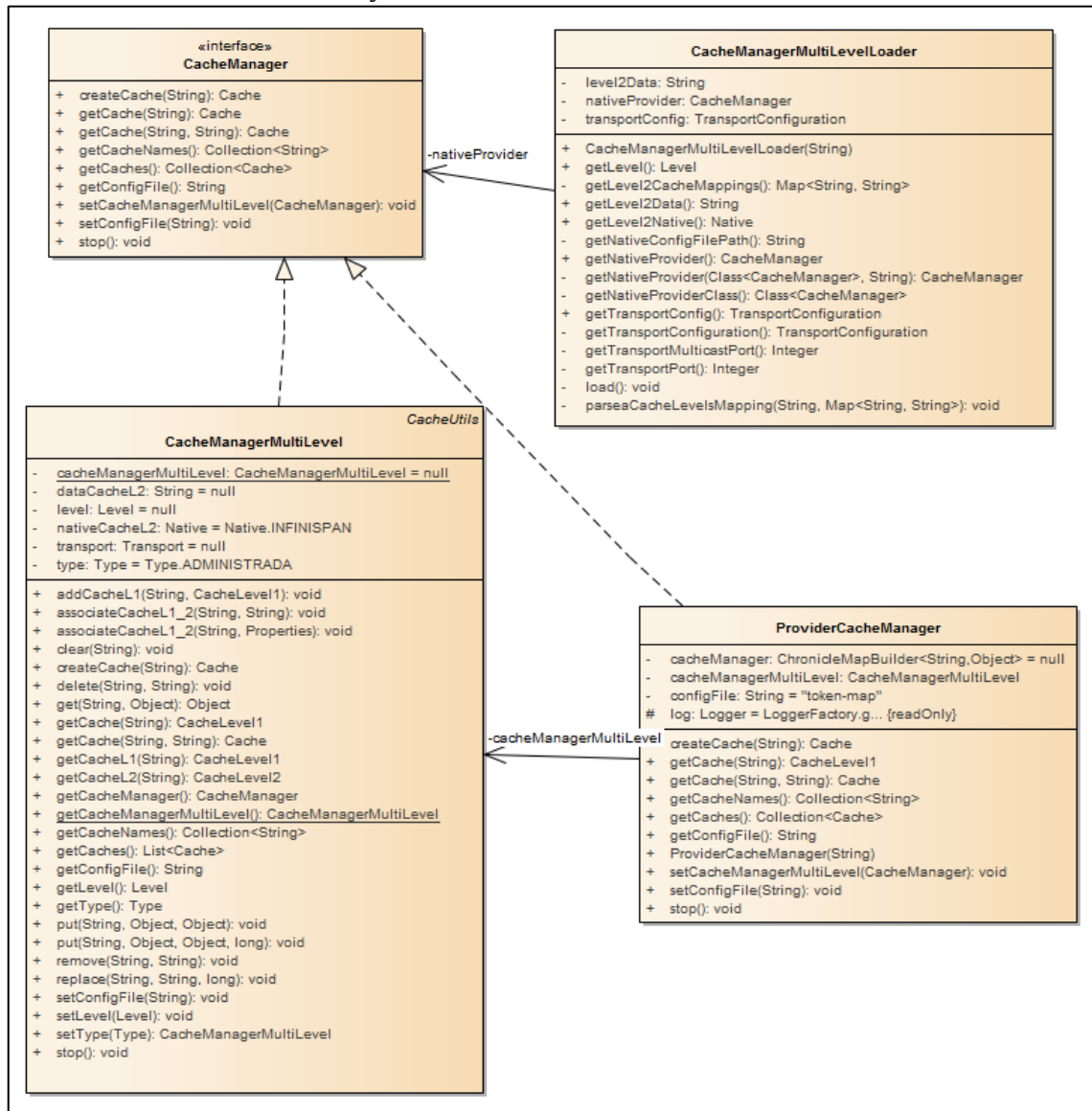


Figura 38. Diagrama de clases del gestor de cachés

El siguiente diagrama de secuencia corresponde a la acción de obtener un dato por parte de una aplicación, verificando la existencia del mismo en las cachés definidas, e invocando al recurso externo en caso de ser necesario.

5.2.2 Diagrama de secuencia de uso de caché

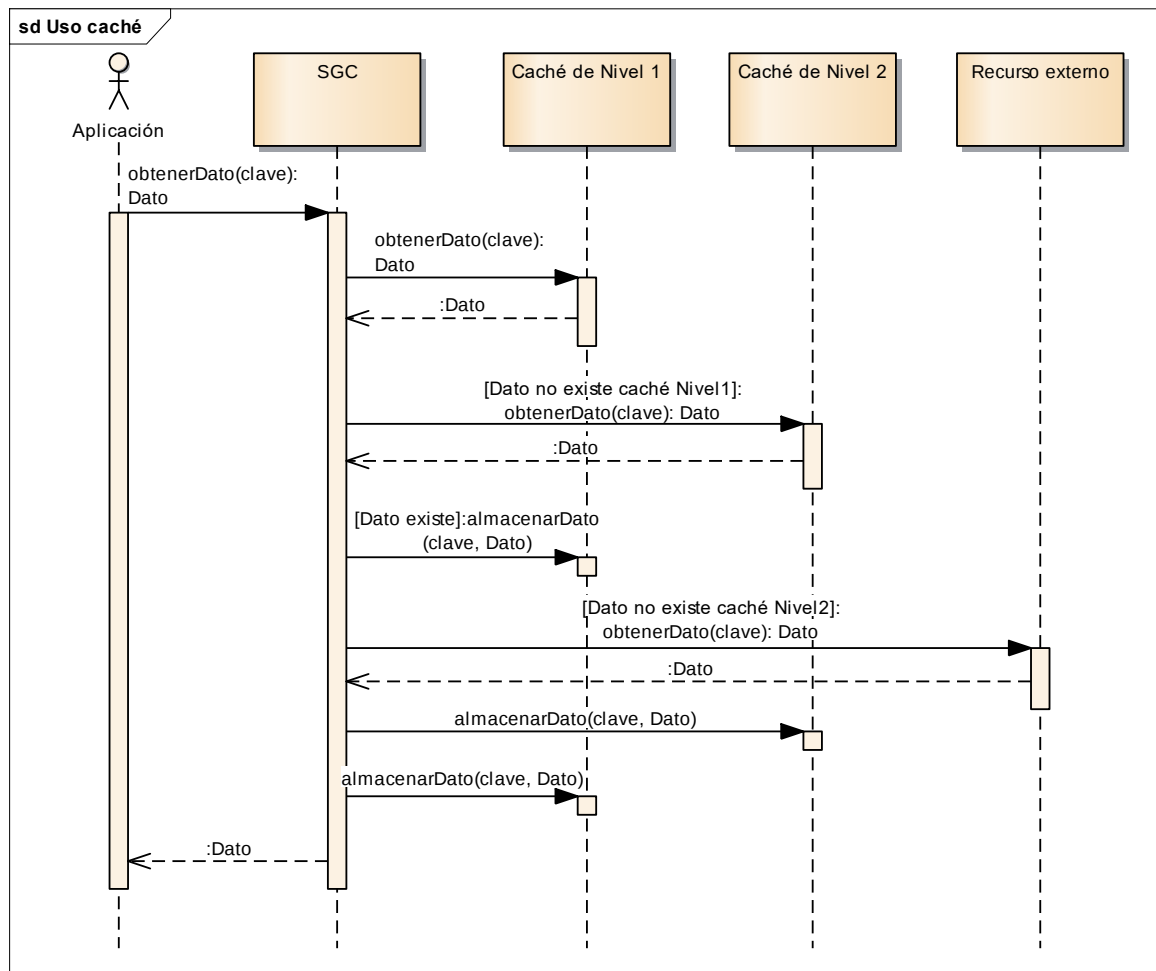


Figura 39. Diagrama de secuencia de uso de cache

CAPÍTULO 6

6. Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones obtenidas al finalizar el diseño de un sistema de gestión de cachés centralizado y se muestran algunas líneas de trabajo futuro que se podrían llevar a cabo como continuidad del trabajo realizado.

6.1 Conclusiones

Como se ha visto en este proyecto, las aplicaciones deben dar solución a una serie de problemas entre los que se encuentran dar una respuesta en un tiempo reducido, atender a un variable número de usuarios y ser fiables ante diferentes tipos de problemas técnicos. Dar solución a estos problemas mejora la percepción que tiene el usuario y mejora su retención.

Para dar solución a estos problemas existe la posibilidad de almacenar datos en memoria (caché), de tal forma que una vez obtenidos de un recurso externo, cuyo acceso lento y costoso, se pueden almacenar en memoria para posteriormente ser recuperados a partir de una clave que los identifique. Dicho almacenamiento se puede realizar en dos niveles distintos: Nivel 1 o empujado en la propia aplicación y Nivel 2, en un “data grid” externo, que puede dar servicio a varias cachés de Nivel 1.

Este almacenamiento permite reducir el tiempo de respuesta dado que el acceso a la memoria es mucho más rápido y eficaz que el acceso a un recurso externo que puede también suponer la ejecución de una serie de procesos costosos tanto en tiempo como en dinero.

El almacenamiento en memoria también permite adaptarse a una gran variabilidad en el número de usuarios, ya que ante el aumento o disminución de los mismos, no es necesario modificar el dimensionamiento del recurso externo, basta con adaptar el número de nodos de la caché, lo cual supone únicamente el arranque o parada de un

proceso, de tal forma que se adapta la capacidad de dar respuesta de una forma muy eficaz.

El almacenamiento en memoria, por último, aumenta la fiabilidad de la aplicación, ya que la caída de un nodo de la caché no implica pérdida de la información, ni pérdida de servicio, debido a que los datos se encuentran distribuidos en el “data grid”, y automáticamente se vuelven a redistribuir para garantizar su disponibilidad. En el caso de no existir dicho almacenamiento, la caída del recurso externo implicaría que no se pudiera acceder a los datos del mismo.

	Acceso directo a los recursos	Con almacenamiento en caché
Tiempo de respuesta	Dependiente del recurso al que se accede.	Si el dato está en caché Nivel 1, acceso directo a memoria. Posibilidad de precargar la caché para eliminar el retardo del primer acceso.
Coste	Depende del recurso al que se accede por el número de accesos.	Un único acceso al recurso externo, posteriormente el dato está almacenado.
Escalabilidad	Dependiente del recurso.	Escalabilidad horizontal sin límite, e instantánea añadiendo nodos a la cache de Nivel 2.
Fiabilidad	Necesidad de la existencia del recurso en el momento de utilizarlo (cuello de botella)	Sin cuello de botella, al residir el dato en un “data grid” que garantiza su disponibilidad.

Tabla 70. Comparación sin caché vs con caché

Existen múltiples soluciones de almacenamiento en caché, por lo que las aplicaciones pueden optar por cualquier proveedor para dar soporte al almacenamiento en memoria. Cada uno de los proveedores al tener una infraestructura distinta ofrece un API propio, por lo que la aplicación queda directamente vinculado al proveedor elegido.

En los últimos años se ha realizado un esfuerzo de normalización en el uso de almacenamiento en memoria, formándose un equipo de trabajo que ha finalizado su tarea con la liberación de la JSR-107 que define JCache. Esta especificación define un interfaz único de acceso al almacenamiento en memoria por lo que todos aquellos proveedores que implementen dicha norma podrán ser utilizados indistintamente sin tener que modificar el sistema. En los últimos años los principales proveedores de caché, han ofrecido una implementación de sus API's dentro de la norma de JCache.

JCache ha supuesto una gran paso adelante al facilitar la toma de decisión a la hora de utilizar el almacenamiento en memoria, dado que la decisión se ha dividido en dos fases:

1. El uso del propio almacenamiento, que una vez, decidido se realiza mediante JCache,
2. La elección del proveedor, decisión que es fácilmente modificable entre aquellos proveedores que implementan JCache.

Conclusiones y trabajos futuros

En este Proyecto Fin de Carrera, se ha explicado las ventajas que ofrece el almacenamiento en memoria, y la gran ventaja que ha supuesto JCache, pero se han identificado una serie de **carencias** para las que se ha ofrecido una solución. Dichas carencias son:

- JCache define un almacenamiento a **un único nivel** de caché.
- Desde JCache **no se puede cambiar de proveedor** por configuración, ya que su funcionamiento se basa en utilizar la implementación existente en el classpath de la aplicación, debiendo ser dicha implementación única.
- La **información en la caché no es administrable**, por lo que puede suponer un problema si es necesario actuar sobre un dato almacenado debido a que ha cambiado el propio dato en el recurso externo.

En este Proyecto Fin de Carrera se ha dado una **solución** a cada uno de los problemas detectados, permitiendo a la aplicación que utilice el Sistema de Gestión de Cachés:

- **Definir** el almacenamiento a **Nivel 1 y/o Nivel 2**.
- Definir el **proveedor** a utilizar en cada nivel de caché **por configuración**, el proveedor de Nivel 1 al ser empotrado en la propia aplicación no implica ninguna acción adicional. El proveedor de Nivel 2 supone que el “data grid” seleccionado debe de estar instalado y en funcionamiento.
- Proveer una **consola de administración** que debe de permitir el acceso a toda la información almacenada en las cachés y aplicaciones manejadas, y actuar sobre dicha información.

	SGC	JCache
Nivel	Nivel 1 y Nivel 2	Nivel 1
Proveedor	Configurable	No configurable
Administración	Administrable	No administrable

Tabla 71. Comparación SGC vs JCache

6.2 Líneas de trabajo futuro

Tomando este Proyecto fin de Carrera se abren una serie de líneas de trabajo, que continúen el estudio realizado y terminen ofreciendo un sistema de gestión de cachés que permita el uso de almacenamiento en memoria de la forma más completa posible, entre las posibles líneas de trabajo se encuentran las siguientes:

Continuar con el propio proyecto y realizar la implementación del sistema de gestión de cachés de tal forma que se cumplan los requisitos identificados, y se contemplen todos los casos de uso.

Realizar una aplicación que utilice el sistema de gestión de cachés, y sobre la que se puedan realizar demostraciones de uso en diferentes escenarios, sin caché y con diversas combinaciones de caché a diferentes niveles, para poder mostrar el impacto que tiene en cada una de las soluciones que ofrece el almacenamiento en caché, es decir, reducción en el tiempo de respuesta, variabilidad en el número de usuarios y fiabilidad ante problemas técnicos.

CAPÍTULO 7

7. Planificación y presupuesto

7.1 Introducción

En este capítulo se expone el coste que tendría el presente proyecto en el mercado real. Por un lado se especifica la planificación necesaria, dicha planificación corresponde a todas las fases necesarias para completar el Sistema de Gestión de Cachés, pero tal como se ha comentado el alcance del presente proyecto no incluye a producción de código.

También se presenta un presupuesto para la elaboración del proyecto con un desglose de costes de personal, costes del material y costes totales.

7.1.1 Planificación

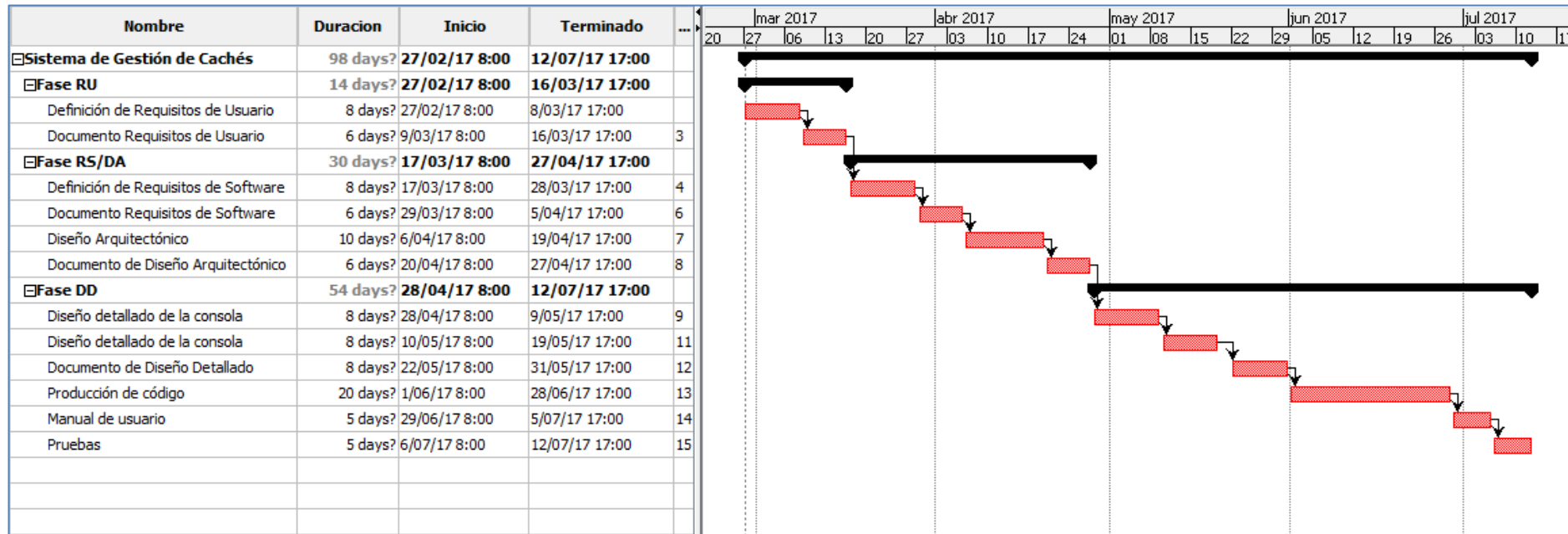


Figura 40. Planificación del proyecto

7.1.2 Presupuesto

Para la realización del presupuesto se ha tenido en cuenta la duración del proyecto y desglose de actividades resultado de la planificación, se ha calculado por un lado los costes directos, asociados con los elementos que directamente afectan al desarrollo y puesta en funcionamiento del proyecto, como son los costes de personal, medios técnicos empleados, material fungible, etc, y por otro lado los costes indirectos correspondientes al consumo energético, el coste de la conexión a internet, etc.

El tiempo total que ha requerido la realización del proyecto ha sido de un total de 98 días. El trabajo ha sido realizado a razón de 8 horas diarias durante los días especificados. El total de horas de trabajo será 784 (98 días * 8 horas/día = 784 horas).

En la siguiente tabla se detalla el tiempo empleado en cada actividad del proyecto:

Actividad	Horas
Requisitos de Usuario	112 h.
Requisitos de Software y Diseño Arquitectónico	240 h.
Diseño Detallado	432 h.
Total	784 h.

Tabla 72. Actividades del proyecto y duración

En la elaboración del proyecto se requiere personal informático cualificado, el cual deberá adoptar distintos roles en cada una de las actividades que forman el proyecto. La siguiente tabla muestra el coste de cada rol utilizado en la elaboración del proyecto.

Categoría	Horas
Analista	224 h.
Responsable diseño	320 h.
Responsable implementación	200 h.
Responsable de pruebas	40 h.
Total	784 h.

Tabla 73. Roles del proyecto y duración

7.1.2.1 Costes de personal

Teniendo en cuenta la categoría de los participantes del proyecto y el coste de cada hora dedicada se calcula los costes directos relacionados con los gastos de personal.

La siguiente tabla muestra el coste total de cada categoría. El importe total es de 20840 €.

Categoría	Horas	Coste hora	Total
Analista	224 h.	35,00 €	7840 €
Responsable diseño	320 h.	35,00 €	11200 €
Responsable	200 h.	30,00 €	6000 €

implementación			
Responsable de pruebas	40 h.	30,00 €	1200 €
Total gastos de personal			20840 €

Tabla 74. Gastos de personal imputables al proyecto

7.1.2.2 Costes de material

En esta sección se detalla el material hardware necesario para el desarrollo del proyecto y el importe asociado al material fungible, el software utilizado será de libre distribución. La siguiente tabla muestra la descripción del material, el coste, el periodo de depreciación, el porcentaje de uso dedicado, la dedicación en meses y el coste imputable.

Descripción	Coste	% Uso	Dedicación (meses)	Periodo de depreciación	Coste imputable
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
Impresora LaserJet Pro CP1025NW	150,00	100	4	60	10,00
Total gastos de personal					170 €

Tabla 75. Gastos de material imputables al proyecto

7.1.2.3 Costes indirectos

El coste indirecto se realiza sobre la base de un 20% del total de los costes del proyecto ascendiendo a $(20.840 + 170) \cdot 20/100 = 4202$. El importe total de costes indirectos es de 4202 €.

7.1.2.4 Resumen del presupuesto

La siguiente tabla muestra un resumen del presupuesto económico del proyecto:

Descripción	Importe
Costes de personal	20840 €
Costes de material	170 €
Costes indirectos	4202 €
Total presupuesto	25212 €

Tabla 76. Resumen del presupuesto (Impuestos no incluidos)

7.1.2.5 Detalle del presupuesto



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO (Impuestos aparte)

1.- Autor:
José Antonio Navarro Fuentes

2.- Departamento:

3.- Descripción del Proyecto:

- Título: **Sistema de Gestión de Cashés**

- Duración (meses): **4**

Tasa de costes indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):
25.212,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación horas	Coste hombre hora	Coste (Euro)	Firma de conformidad
		Analista	224	35,00	7.840,00	
		Responsable diseño	320	35,00	11.200,00	
		Responsable implementación	20	30,00	600,00	
		Responsable de pruebas	40	30,00	1.200,00	
					0,00	
Horas: 804				Total	20.840,00	

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ⁽¹⁾
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
HP ProBook 650 G1	600,00	100	4	60	40,00
Impresora LaserJet Pro CP1025NW	150,00	100	4	60	10,00
					0,00
Total					170,00

⁽¹⁾ Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = período de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	20.840
Amortización	170
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	4.202
Total	25.212

Figura 41. Presupuesto del proyecto

“El presupuesto total de este proyecto asciende a la cantidad de veinticinco mil doscientos doce Euros, Impuestos (IVA) no incluidos.

Leganés a 20 de Febrero de 2017

El ingeniero proyectista

Fdo. Jose Antonio Navarro Fuentes”

CAPÍTULO 8

8. Bibliografía y referencias

La bibliografía se organiza en apartados que engloban estándares de caché, proveedores de caché y estándares de ingeniería de software.

8.1 *Estándares de caché*

8.1.1 JSR 107: JCache

a.1.1 JSR 107: JCACHE - Java Temporary Caching API

<https://jcp.org/en/jsr/detail?id=107>

a.1.2 Lista de implementaciones compatibles con JCache

<https://jcp.org/aboutJava/communityprocess/implementations/jsr107/index.html>

a.1.5 Coherence working with JCache

<https://docs.oracle.com/middleware/1213/coherence/tutorial/jcache.htm>

a.1.6 Using the Coherence JCache Implementation

http://docs.oracle.com/middleware/1213/coherence/develop-applications/jcache_part.htm

a.1.7 Introduction to Coherence JCache

<https://docs.oracle.com/middleware/1221/coherence/develop-applications/GUID-BD999810-73B1-46CC-BE53-06D00611D4A4.htm#COHDG5780>

a.1.10 Infinispan using a cache with the JCache API

<http://infinispan.org/tutorials/simple/jcache/>

a.1.15 Ehcache JSR-107 and Spring 4.2

http://www.ehcache.org/blog/2016/05/17/ehcache3_jsr107_spring.html

a.1.16 The Ehcache 3.x JSR-107 Provider

<http://www.ehcache.org/documentation/3.0/107.html>

a.1.20 Hazelcast

<https://hazelcast.com/use-cases/caching/jcache-provider/>

a.1.25 JCP

<https://jcp.org/en/home/index>

a.1.26 JSR

<https://www.jcp.org/en/jsr/overview>

8.1.2 Spring-cache**a.2.1 Cache Abstraction**

<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/cache.html>

a.2.2 Cache Abstraction: JCache (JSR-107) Annotations Support

<https://spring.io/blog/2014/04/14/cache-abstraction-jcache-jsr-107-annotations-support>

a.2.5 Spring Book – Chapter 10 – Caching

<http://www.javacodebook.com/2013/07/20/spring-book-chapter-10-caching/10/>

a.2.6 Spring Integration with Coherence Cache

<http://vjydhulipala.blogspot.com.es/>

a.2.7 Coherence Spring Integration

<https://java.net/projects/cohspr>

a.2.15 Ehcache JSR-107 and Spring

http://www.ehcache.org/blog/2016/05/17/ehcache3_jsr107_spring.html

a.2.16 Spring Caching and Ehcache

<http://www.mkyong.com/spring/spring-caching-and-ehcache-example/>

a.2.20 Hazelcast

<https://hazelcast.com/use-cases/caching/spring-cache/>

a.2.21 Spring Integration

<http://docs.hazelcast.org/docs/3.3/manual/html/springintegration.html>

8.1.3 Caché

d.1 Gartner Says In-Memory Computing Is Racing Towards Mainstream Adoption

<http://www.gartner.com/newsroom/id/2405315>

d.2 Navarro Fuentes, Jose Antonio ¿Cómo lograr aplicaciones más rápidas, eficientes y con tolerancia a fallos?

<https://www.youtube.com/watch?v=tj9FOhMUbUw>

8.2 Proveedores de caché

8.2.1 Coherence

c.1 Coherence

<http://www.oracle.com/lad/products/middleware/cloud-app-foundation/coherence/overview/index.html>

c.2 Oracle Fusion Middleware 12c Oracle Coherence

<http://docs.oracle.com/middleware/1221/coherence/index.html>

8.2.2 Infinispan

c.5 Infinispan

<http://infinispan.org/docs/8.2.x/index.html>

8.2.3 Ehcache

c.10 Ehcache

<http://www.ehcache.org/>

8.2.4 Hazelcast

c.20 Hazelcast

<https://hazelcast.com/products/>

c.21 Distributed computing with Hazelcast

<http://es.slideshare.net/ChristophEngelbert/distributed-computing-with-hazelcast>

8.2.5 ChronicalMap

c.25 ChronicalMap

<http://chronicle.software/products/chronicle-map/>

8.3 Ingeniería del software

e.1 Ingeniera del Software Universidad Carlos III de Madrid

<http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii>

e.1.1 MC-F-007. **Análisis de los sistemas de información. Paradigmas y metodologías.** http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_07_ASI.pdf/view

e.1.2 MC-F-008. **Diseño de los sistemas de información**
http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_08_DSI.pdf/view

e.5 **ESA software engineering standards Issue 2. ESA PSS-05-0 Issue 2. ESA Publications Division, ESTEC, Noordwijk, The Netherlands. ISSN 0379-4059. 1994 by European Space Agency**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS050.pdf>

e.5.1 **Guide to the user requirements definition phase. ESA PSS-05-02 Issue 1 Revision 1 (March 1995). ESA Publications Division, ESTEC, Noordwijk, The Netherlands. ISSN 0379-4059. 1995 by European Space Agency**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0502.pdf>

e.5.2 **Guide to the software requirements definition phase. ESA PSS-05-03 Issue 1 Revision 1 (March 1995). ESA Publications Division, ESTEC, Noordwijk, The Netherlands. ISSN 0379-4059. 1995 by European Space Agency**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0503.pdf>

e.5.3 **Guide to the software architectural design phase. ESA PSS-05-03 Issue 1 Revision 1 (March 1995). ESA Publications Division, ESTEC, Noordwijk, The Netherlands. ISSN 0379-4059. 1995 by European Space Agency**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0504.pdf>

e.5.4 **Guide to the software detailed design and production phase. ESA PSS-05-03 Issue 1 Revision 1 (March 1995). ESA Publications Division, ESTEC, Noordwijk, The Netherlands. ISSN 0379-4059. 1995 by European Space Agency**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0505.pdf>

e.5.5 **ESA Lite Guide to applying the ESA Software Engineering Standards to small software projects, BSSC(96)2**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/Bssc962.pdf>

e.5.6 **ESA Lite Guía para la aplicación de Estándares de Ingeniería de Software ESA Versión en castellano**
<http://www.ie.inf.uc3m.es/grupo/docencia/reglada/Is1y2/ESA/BSSC962-ES.PDF>

e.5.7 **Guide to applying the ESA Software Engineering Standards to Projects using Object-Oriented Methods, BSSC(98)1**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/bssc981h.pdf>

e.5.8 **Java coding standard BSSC(2005)2**
<ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/Java-Coding-Standards-20050303-releaseA.pdf>

CAPÍTULO 9

9. Glosario

API	Application Programming Interface, es el contrato ofrecido por una biblioteca software para el acceso a su funcionalidad. Este API garantiza la comunicación entre distintos componentes software.
Clave/Valor	Forma de almacenar la información en la caché la clave es el identificador con el que se almacena la información en la caché, valor es el propio dato a almacenar
Caché Nivel 1	Caché embebido en la propia aplicación, por lo que se ejecuta en la misma JVM, el acceso es inmediato.
Caché Nivel 2	Caché disponible en un Datagrid externo, permite ser compartido por varias aplicaciones, por lo que una vez establecido un valor se evita el acceso al recurso correspondiente.
CDI	Contexto e inyección de dependencias o Contexts and Dependency Injection es un estándar definido en JSR 299, conocido como Web Beans. Existen una serie de especificaciones relacionadas que CDI utiliza entre las que se encuentran JSR 330, o inyección de dependencias para Java y JSR 316 que es la especificación de los Managed Beans. CDI define el mecanismo para resolver dependencias entre servicios dentro del estándar JEE.
Clúster	Conjunto de elementos conectados entre sí.
Data grid	Red de datos, en este PFC base de datos distribuida en memoria, de gran escalabilidad, que ha sido diseñada para dar acceso rápido a grandes volúmenes de datos. La cuadrícula de datos suele proporcionar una capa complementaria a la base de

datos relacional y la aplicación.

IMC

In Memory Computing, tecnología para el tratamiento de datos almacenados en memoria. Los sistemas generalmente están basados en almacenamiento en disco y/o bases de datos relacionales, esto aumenta los tiempos de proceso debido al retardo en la recuperación de la información. El almacenamiento en memoria, permite el tratamiento de grandes volúmenes de datos en tiempo real.

JCP

Java Community Process es la comunidad de desarrollo de especificaciones, establecida en 1998, establece el proceso por el que se define y mantiene el estándar de la plataforma Java, este proceso permite a las partes interesadas involucrarse en la definición de las futuras versiones y características, garantizando compatibilidad y portabilidad. El proceso JCP conlleva el uso de Java Specification Request (JSR) ver JSR. **[a.1.25]**

JSR

Las Java Specification Request definen las especificaciones y tecnologías de la plataforma Java. Las revisiones públicas de los JSRs son controladas antes de que los JSR se conviertan en final y sean votados por el Comité Ejecutivo JCP. Un JSR final suministra una especificación que es un documento que describe la tecnología. Una implementación de referencia la cual da una implementación libre de la tecnología y un Kit de Compatibilidad que son un conjunto de pruebas que permiten verificar si una implementación cumple o no con la especificación. **[a.1.26]**

K/V

Ver [Clave/Valor](#).

Key/Value

Ver [Clave/Valor](#).

L1

Ver [caché Nivel 1](#).

L2

Ver [caché Nivel 2](#).

Middleware

Es un software que permite a una aplicación comunicarse con otras aplicaciones, redes, hardware o sistemas operativos. Éste simplifica el trabajo dado que el propio middleware gestiona las conexiones y sincronizaciones necesarias al ser un sistema distribuido. Funciona como una capa de abstracción, que se sitúa entre las capas de aplicaciones y las capas inferiores.