Autor: fernando de la torre Cueva UO245182

|

Automatización de la generación de universos semánticos basados en el análisis de consultas SQL

Tabla de contenido

[Capitulo 1 Que es este trabajo? 3](#_Toc126004039)

[Resumen 3](#_Toc126004040)

[Capítulo 2 planificación del Sistema de información. 4](#_Toc126004041)

[PSI 1 Inicio de la planificación de sistemas de la información 4](#_Toc126004042)

[PSI 1.1 Análisis de la necesidad del PSI 4](#_Toc126004043)

[PSI 1.2 Identificación del alcance del PSI 4](#_Toc126004044)

[PSI 1.3 Identificación de responsables 5](#_Toc126004045)

[PSI 3 Estudio de la información relevante 5](#_Toc126004046)

[PSI 3.1 Selección y análisis de Antecedentes 5](#_Toc126004047)

[Planificación temporal 10](#_Toc126004048)

[Resumen General 10](#_Toc126004049)

[Fase de análisis del Estado del Arte 10](#_Toc126004050)

[Fase de análisis del Sistema 10](#_Toc126004051)

[Fase de implementación de la POC 10](#_Toc126004052)

[Capitulo 3 PSI 7 Definición de la Arquitectura tecnológica para la POC 10](#_Toc126004053)

[PSI 7.1 IDENTIFICACIÓN DE LAS NECESIDADES DE INFRAESTRUCTURA TECNOLÓGICA 10](#_Toc126004054)

[Capitulo 4 Estudio de viabilidad del sistema. 11](#_Toc126004055)

[EVS 4, 5 y 6: Estudio y Valoración de Alternativas de Solución y Selección de Alternativa Final 11](#_Toc126004056)

[Capítulo 5 Toda la movida de DPPI 11](#_Toc126004057)

[Capítulo 6 Análisis del sistema 11](#_Toc126004058)

[6.1 Elección del Sistema para la POC 11](#_Toc126004059)

[6.1.1 Selección del sistema de estructuración de datos 11](#_Toc126004060)

[6.1.2 Selección del sistema de análisis de consultas 11](#_Toc126004061)

[6.2 Estudio de la estructuración de datos 12](#_Toc126004062)

[6.2.1 Selección del mapeo SGDB/consultas 12](#_Toc126004063)

[6.3 análisis intermedio de la estructura semántica 14](#_Toc126004064)

[6.4 Resumen de elecciones tomadas 15](#_Toc126004065)

[6.5 Obtención de salidas 15](#_Toc126004066)

[6.5.1 Elección de formato de salida 16](#_Toc126004067)

[6.5.2 Regeneración del sistema de estructuración de datos 16](#_Toc126004068)

[6.6 análisis de resultados 16](#_Toc126004069)

[6.6.1 Mediciones 16](#_Toc126004070)

[6.7.2 Comparativas 16](#_Toc126004071)

[Capitulo 7 Conclusiones 16](#_Toc126004072)

[Capítulo 8 Integración con Sistemas Distribuidos 17](#_Toc126004073)

[Capítulo 9 Anexos 17](#_Toc126004074)

[Anexo. Definiciones y abreviaturas 17](#_Toc126004075)

[Definiciones 17](#_Toc126004076)

[Abreviaturas 17](#_Toc126004077)

[Anexo. Analisis de riesgos 18](#_Toc126004078)

[Anexo. Presupuestos 19](#_Toc126004079)

# Capitulo 1 Que es este trabajo?

## Resumen

Este trabajo de investigación estudia la posibilidad de generar universos o capas semánticos en entornos de estructuración de datos tabulares.

La investigación trata de encontrar una forma de automatizar este proceso y para ello se basa en la forma de uso de un sistema de estructuración de información tabular, es decir en el análisis de las consultas a dicho sistema.

Se analizarán las formas de estructurar la información obtenida del análisis de las consultas, así como la forma de buscar agregaciones semánticas basadas en dicho análisis.

Finalmente se ejecutará el proceso de forma automatizada, obteniendo, en distintos niveles de agregación, distintas representaciones semánticas denormalizadas.

Actualmente este es un campo de estudio y puesta en producción de la empresa Accenture

**Palabras Clave**: Accenture, Consultas, Universos semánticos, Denormalizacion.

# Capítulo 2 planificación del Sistema de información.

## PSI 1 Inicio de la planificación de sistemas de la información

En esta sección se analizarán las necesidades de investigación.

### PSI 1.1 Análisis de la necesidad del PSI

Vivimos en un entorno en el empleo los datos son una parte neurálgica de las empresas.

Hoy en día la memoria ha dejado de ser un problema ya que existen sistemas distribuidos conoces permiten amplificar y ampliar la memoria de forma dinámica añadiendo recursos tanto en la capacidad de almacenamiento de cada nodo del sistema cómo en el número de nuevas del sistema.

Sin embargo y por contra se han incrementado los tiempos de ejecución debido a que el gran aumento del volumen de datos manejados ha provocado que requiramos de un gran nivel de distribución y paralelización.

Por lo tanto, el problema hoy en día es la performance de acceso a los datos y de escritura en base de datos.

Para intentar aliviar este problema se han empezado a implementar y utilizar capas semánticas sobre nuestros datos siendo capaces de extraer información semántica de los datos, generar informes, incluso inferir información nueva.

Este proceso hoy en día se realiza basándose en cómo un experto analiza un sistema de datos, pero, no existe una forma estandarizada de realizar este proceso de obtención de información semántica de forma automatizada y basada en el uso de una aplicación.

En este punto es donde nos enfocaremos en nuestro proyecto de investigación intentando ser capaces bongo entrar una forma de realizar los siguientes procesos:

* Analizar el uso de una aplicación.
* Analizar como afectan al sistema de estructuración de datos
* Re modelizar el SGBD para optimizar la perfomance de lectura basándonos en el uso de la aplicación

### PSI 1.2 Identificación del alcance del PSI

Debido a que se trata de un proyecto de investigación no se establecerá un alcance estricto sí la puse efectuará un análisis de necesidades previamente realizado, se procederá a analizar el estado del arte, se implementará una prueba de concepto basándose en la investigación realizada y se tratará de obtener resultados analíticos basados en la ejecución de esta prueba de concepto.

Sin embargo, podemos realizar el PBS de la parte de investigación, siendo el siguiente:  
Diagrama

Descripción generada automáticamente

### PSI 1.3 Identificación de responsables

En esta parte veremos aquellas personas o stakeholders interesados de la documentación o información que se intercambia durante la ejecución del proyecto.

* **Jefe de proyecto**: Responsable de los documentos oficiales que se intercambian con el cliente.
* **Responsable por parte del cliente:** Responsable de recibir los documentos que afecten al proyecto en la parte del cliente
* **Responsable por parte de la universidad**: Responsables de solicitar cambios en el código de carácter crítico y de mantener internamente una comunicación fluida.

## PSI 3 Estudio de la información relevante

### PSI 3.1 Selección y análisis de Antecedentes

#### PSI 3.1.1 Estado del arte. Tecnologías

En el entorno de mercado actual donde los datos son parte neurálgica de las empresas es necesario poseer información semántica de los mismos y ser capaces de extraer la información que se necesita en cada momento de bancos de datos muy grandes.

Sin embargo, en el entorno del software tanto de pago como open source no existen tecnologías que automaticen el proceso de performance en el acceso a datos si no que existen herramientas que proveen de una capa de businness inteligence a los datos.

Existen numerosas opciones como las mencionada a continuación:

* Power Bi

Power BI no solo es una herramienta de Business Intelligence. Se trata de un servicio de análisis empresarial de Microsoft cuyo objetivo es proporcionar visualizaciones interactivas y capacidades de inteligencia empresarial con una interfaz lo suficientemente simple como para que los usuarios finales creen sus propios informes y paneles.

* Tableau

Tableau es nuestra segunda herramienta de Business Intelligence, un software de análisis e inteligencia de negocios.

Su principal servicio es el desarrollo de productos de visualización de datos interactivos​ que se enfocan en inteligencia empresarial.

* QlikView

QlikView es otra herramienta imprescindible de Inteligencia Empresarial de software que combina todos los datos, ayuda a crear informes y a obtener conocimientos empresariales de forma rápida.  
Además, se puede exportar la información en formato excel. Incluye integración de datos, inteligencia de negocios impulsada por el usuario y análisis de conversación.

* SAP BI

SAP BI es un software de Business Intelligence, una tecnología de inteligencia de negocios con la que se puede compartir información estratégica y tomar mejores decisiones con la suite SAP Business Objects Business Intelligence (BI).

Su plataforma analítica puede dar soporte a sus usuarios y de una única herramienta a múltiples herramientas e interfaces. La herramienta tiene implementación on-premise,  
Business Intelligence en tiempo real y proporciona una mayor autonomía del usuario.

* Pentaho

Pentaho es un conjunto de programas libres para generar inteligencia empresarial.  
Incluye herramientas integradas para generar informes, minería de datos, ETL, etc.

* Zoho Analytics

Zoho Analytics es un software de inteligencia empresarial, generación de informes y análisis de datos que permite analizarlos visualmente y descubrir información en cuestión de minutos.

Con esta herramienta se pueden transformar enormes cantidades de datos en bruto en informes. Además se puede realizar un seguimiento de las métricas empresariales clave, revisar tendencias antiguas, identificar valores atípicos y revelar información oculta.

* Oracle BI

Oracle BI es una plataforma de herramientas de Business Intelligence que permite simplificar las estrategias de analítica a través de una plataforma moderna e integrada.

Esta herramienta sirve a las personas de toda la organización para tomar decisiones comerciales más rápidas, más informadas y habilitadas para dispositivos móviles. Además, la plataforma ha impulsado el lanzamiento actual de análisis de nube líder en la industria, Oracle Analytics.

* Sisense

Sisense es una herramienta de analítica permite preparar, analizar y explorar fácilmente datos en crecimiento de múltiples fuentes.

Las principales funciones de esta herramienta de BI ayudan a simplificar el análisis de datos y desbloquear datos de la nube para que todos puedan analizar los datos y generar mejores resultados. Además, es perfecto para construir experiencias personalizadas y automatizar acciones de varios pasos para acelerar los flujos de trabajo.

#### PSI 3.1.2 Estado del Arte. Modelo Conceptual – Conceptual/Teórico – Conceptos teóricos relaccionados con la representación del conocimiento relaccionados con la estructuracion de datos

##### PSI 3.1.2.1 Estado del Arte. 5FN

La quinta forma normal (5FN), también conocida como forma normal de proyección-unión (PJ/NF), es un nivel de normalización de bases de datos diseñado para reducir redundancia en las bases de datos relacionales que guardan hechos multi-valores aislando semánticamente relaciones múltiples relacionadas. Una tabla se dice que está en 5NF si y solo si está en 4NF y cada dependencia de unión (join) en ella es implicada por las claves candidatas.

La quinta forma normal fue definida por Ronald Fagin en su contribución al congreso «Normal forms and relational database operators» de 1979

Solamente en contadas ocasiones una tabla 4NF no se corresponde con una 5NF. Estas son situaciones en las cuales una restricción compleja del mundo real, que limita las combinaciones válidas de los valores de atributos en la tabla 4NF, no está implícita en la estructura de esa tabla. Si esa tabla no se normaliza a 5NF, la tarea de mantener la consistencia lógica de los datos dentro de la tabla debe ser llevada en parte por la aplicación responsable de inserciones, borrados, y actualizaciones a ella; y hay un riesgo elevado de que los datos dentro de la tabla se vuelvan inconsistentes. Por el contrario, el diseño 5NF excluye la posibilidad de tales inconsistencias.

###### PSI 3.1.2.1.1 5FN vs RDF

* Tripleta Semántica

Una tripleta o terna semántica es la entidad atómica de datos en el modelo de datos Resource Description Framework (RDF)​ Como su nombre indica, una terna es un conjunto de tres entidades que codifica una declaración sobre datos semánticos en forma de expresiones sujeto–predicado–objeto (por ejemplo, «Antonio tiene 35 años» o «Antonio conoce a Berta»).Este formato permite la representación del conocimiento de una manera legible por máquinas. En particular, cada una de las partes de una terna RDF se puede direccionar de forma unívoca mediante un URI único. Por ejemplo, la declaración «Antonio conoce a Berta» se podría representar en RDF como http://www.ejemplo.com/AntonioCastillo12 http://xmlns.com/foaf/0.1/knows http://www.ejemplo.com/BertaDiaz34. Dada esta representación precisa, es posible hacer consultas semánticas y razonamientos sobre ellos sin ambigüedad.

##### Los componentes de una terna (tal como la declaración «el cielo tiene el color azul») son el sujeto («el cielo»), el predicado («tiene el color») y el objeto («azul»). Esto es similar a la notación clásica de un modelo entidad-atributo-valor en el diseño orientado a objetos, en la cual este mismo ejemplo quedaría expresado como una entidad («cielo»), un atributo («color») y un valor («azul»). A partir de esta estructura básica, las ternas se pueden componer unas con otras para formar modelos más complejos, por ejemplo: Miguel → dijo → (las ternas → pueden ser → objetos).

##### Dada la consistencia de su estructura, las colecciones de tripletas se suelen almacenar en bases de datos especialmente diseñadas para ello, conocidas como almacenes de tripletas (en inglés, triplestores).

En esta explicación se ve una similitud con la quinta forma normal (5FN) donde don conceptos/celdas están relacionados a través de sus respectivas columnas.

En este caso tendríamos una relación del tipo:

* CELDA 🡪 columna 🡪 CELDA

##### PSI 3.1.2.2 Estado del Arte. Denormalizacion

La desnormalización es el proceso de procurar optimizar el funcionamiento de una base de datos por medio de agregar datos redundantes​ A veces es necesario porque los actuales sistemas de gestión de bases de datos (SGBD) implementan el modelo relacional de manera limitada. Un SGBD relacional que cumpla las recomendaciones ISO debe permitir una base de datos completamente normalizada a nivel lógico, soportado por un almacenamiento físico de los datos afinado para los requisitos necesarios del sistema.

Un diseño normalizado a menudo almacenará diferentes, pero relacionadas, piezas de información en tablas lógicas separadas (denominada relaciones). Si estas relaciones están almacenadas físicamente en dos archivos distintos en disco, su recuperación para realizar una consulta a la base de datos que se base en información de varias relaciones (una operación JOIN) puede ser muy lenta, llegando a no satisfacer determinados requisitos no funcionales del sistema. Hay dos estrategias para tratar de solventar esta situación:

El método preferido es mantener normalizado el diseño lógico, pero indicar al SGBD que almacene en el disco información redundante para optimizar la respuesta a la consulta. En este caso, es responsabilidad del software del SGBD proveer soporte para gestionar la confidencialidad de los datos en todos los niveles de la arquitectura. Este método es implementado de diversa manera según el desarrollador: en Microsoft SQL Server se usan vistas indexadas, en lote productos de Oracle vistas materializadas, etc. Una vista representa la información en un formato conveniente para consultar, y el índice asegura que las consultas contra la vista estén optimizadas.

Sin embargo, la aproximación más usual es desnormalizar el diseño de datos lógico. Realizado con cuidado, esto puede alcanzar una mejora similar en respuesta de consulta, pero complica la tarea de los usuarios que modifiquen contenido del sistema: ahora es la responsabilidad del diseñador de la base de datos el asegurarse de que la base de datos desnormalizada no llegue a ser inconsistente. Esto se realiza mediante reglas en la base de datos llamadas restricciones, que especifican cómo las copias redundantes de información se deben mantener sincronizadas. Es el aumento en la complejidad lógica del diseño de la base de datos y la complejidad añadida de las restricciones adicionales lo que hacen de esta una alternativa delicada. Por otra parte, debido a los gastos indirectos de evaluación de restricciones al insertar, actualizar, o eliminar datos, una base de datos desnormalizada puede llegar a ofrecer en la práctica un rendimiento inferior al que proporcionaba su versión equivalente normalizada. En cambio, cuando se está seleccionado o leyendo datos a menudo el funcionamiento es más probable que sí se mejore.

Ejemplos de técnicas de desnormalización incluyen:

* Vistas materializadas, que pueden implementar lo siguiente:
* Almacenando la cuenta de "muchos" objetos en una relación uno-a-muchos como un atributo de la relación "uno"
* Agregando atributos a una relación de otra relación con la cual será unida (JOIN)
* Esquemas en estrella que también son conocidos como modelos hecho-dimensión y se han extendido a los esquemas de copo de nieve
* Información de resumen preconstruida (útil para informes, data warehouse o data mining) o cubos OLAP

Sin embargo y aunque este proceso intersecciona ampliamente con el proceso objetivo de investigación de este trabajo hoy una gran diferencia.  
En nuestro proyecto de investigación buscamos la forma de automatizar este proceso y basar su ejecución en el análisis de uso de un sistema y no en la experiencia del desarrollador o el experto en SGBD.

Por lo tanto aunque es interesante de cara a captar información inicial y poder manejar el sistema desde el comienzo necesitamos una forma de manjar el proceso de forma automática y basada en el análisis de uso de una aplicación, es decir, en que consultas se ejecutan en una aplicación y a que partes de la base de datos afectan.

###### PSI 3.1.2.2.1 Denormalizacion vs No normalización

Un modelo de datos desnormalizado no es lo mismo que un modelo de datos que no ha sido normalizado, y la desnormalización debe tomar lugar solamente después de que haya ocurrido un nivel satisfactorio de normalización y de que hayan sido creadas las restricciones y/o reglas requeridas para ocuparse de las anomalías inherentes en el diseño. Por ejemplo, que todas las relaciones estén en la tercera forma normal cualquier relación con dependencias de union (JOIN) y multi-valor sean manejadas apropiadamente.

## Planificación temporal

En este apartado se analizará la planificación temporal de la fase de investigación del proyecto, así como de la implementación de la POC.

### Resumen General

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea | Duración | Inicio | Fin |
| Fase de análisis del Estado del Arte | 30d | - | - |
| Fase de análisis del Sistema | 60d | - | - |
| Fase de implementación de la POC | 50d | - | - |

### Fase de análisis del Estado del Arte

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea | Duración | Inicio | Fin |
| - | - | - | - |

### Fase de análisis del Sistema

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea | Duración | Inicio | Fin |
| - | - | - | - |

### Fase de implementación de la POC

|  |  |  |  |
| --- | --- | --- | --- |
| Tarea | Duración | Inicio | Fin |
| - | - | - | - |

# Capitulo 3 PSI 7 Definición de la Arquitectura tecnológica para la POC

## PSI 7.1 IDENTIFICACIÓN DE LAS NECESIDADES DE INFRAESTRUCTURA TECNOLÓGICA

En este estudio se realizará una prueba de concepto para probar la viabilidad de implementación del sistema propuesto.

No es necesario especificar requisitos concretos debido a que el análisis de resultados se basa en una comparativa y sirve con que todos los cálculos se realicen en el mismo sistema.

Para realizar esta POC solo son necesarios los siguientes requisitos

* Sistema Linux para poder asignar todos los recursos a la ejecución para poder obtener datos sin interferencia de otros procesos
* Base de datos mysql para realizar las mediciones
* Repositorio en GitHub para control de versiones

# Capitulo 4 Estudio de viabilidad del sistema.

N/A

## EVS 4, 5 y 6: Estudio y Valoración de Alternativas de Solución y Selección de Alternativa Final

N/A

# Capítulo 5 Toda la movida de DPPI

Completar con los Anexos, Analisis de riesgos, presupuestos …

# Capítulo 6 Análisis del sistema

## 6.1 Elección del Sistema para la POC

### 6.1.1 Selección del sistema de estructuración de datos

Existen numerosas opciones de estructuración de datos tabulares.

A día de hoy los datos son uno de los centros neurálgicos de las empresas, es ahí donde centran sus esfuerzos por obtener, estructurar y mantener sistemas de modelización de datos.

Algunos de estos sistemas son los siguientes:

* Bases de datos Relacionales
* Bases de datos no Relacionales

El sistema mas utilizado hoy en día para la estructuración de datos de forma tabular son las bases de datos relacionales por lo que partiremos de este sistema para construir nuestro banco de datos de estudio.

A parte de esto, nos sirve para manipular de forma sencilla los datos de cara a realizar pruebas mas adelante.

#### 6.1.1.1 Selección de los datos para la POC

Completar con informacion de la base de datos y las consultas

### 6.1.2 Selección del sistema de análisis de consultas

Dado que vamos a utilizar un sistema de base de datos relacional, utilizaremos consultas SQL para el análisis.

#### 6.1.2.1 Mapeo Consultas/Objeto

Para poder realizar un análisis de las consultas necesitamos transformar estas consultas SQL a un tipo de datos que sea capaz de ser almacenado en un programa de forma recurrible y editable.

Para ello necesitamos organizar las consultas internamente en diferentes secciones.

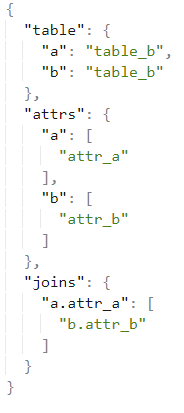
Esto resulta fácil e intuitivo dado que las consultas SQL ya se organizan de esta forma:

* SELECT -> Que estoy buscando?
* FROM -> Donde lo busco?
* WHERE -> Bajo qué condiciones lo busco?

Esto nos permite transformar la siguiente consulta:

SELECT attr\_a, attr\_b FROM table\_a a, table\_b b, WHERE a.attr\_a=b.attr\_b

En la siguiente estructura de objeto:



En este momento estamos listos para construir un mapa de todas las consultas que tengamos en el historico

## 6.2 Estudio de la estructuración de datos

### 6.2.1 Selección del mapeo SGDB/consultas

Para poder realizar un análisis de como las consultas afectan a la base de datos necesitamos poder mapearlas contra la misma.

En este punto hemos estudiado diferentes opciones:

* **Análisis de uso marcando los campos accedidos mediante la ejecución de consultas**: En este caso la manera de proceder seria tener un trigger en cada celda de la base de datos en una base de datos temporal duplicada, donde, cada celda tendría un contador del número de veces que se accede ejecutando todas las consultas.  
    
  En el siguiente ejemplo veremos de forma sencilla como quedaría la siguiente base de datos de un tabla (Persona) tras la ejecución de las siguiente consultas:  
    
  Select nombre from persona where edad > 5  
  Select apellido from persona where edad < 5

|  |  |  |
| --- | --- | --- |
| Nombre | Apellido | Edad |
| Eduardo | garcia | 10 |
| Fernando | Cueva | 2 |

|  |  |  |
| --- | --- | --- |
| Nombre | Apellido | Edad |
| 1 | 0 | 2 |
| 0 | 1 | 2 |

Como podemos ver los campos de la columna edad son accedidos en las dos consultas pues aparecen en el “WHERE” mientras que los campos que aparecen en el “SELECT” solo son accedido tras la comprobación del where.  
Esto seria reproducible para cualquier base de datos y cualquier consulta.

* **Análisis de mapeo por RDF**: RDF es una tecnología de la Web Semántica que se utiliza para relacionar conceptos en tripletas donde el primer elemento es una URI, el segundo elemento es el campo de relación, normalmente identificado por una ontología, y el tercer elemento es otra URI.  
    
  Se indica que el primer y tercer elemento se relacionan entre si a través del segundo, por ejemplo:  
    
  (Joe Biden, presidente, EEUU)  
    
  En nuestro ámbito de investigación nos podría servir para relacionar que campos suelen ser accedidos conjuntamente a través de las columnas a las que pertenecen.  
    
  Por ejemplo en la base de datos de la opción anterior obtendríamos las siguiente tripletas.  
  (10, Edad, Eduardo)  
  (2, Edad, Fernando)  
    
  Aunque era una alternativa prometedora llegamos a un punto de no retorno al comprobar que se podían solapar concetos muy rápidamente ya que, por ejemplo, en el caso anterior la edad refería a un nombre o a un apellido hacían referencia al mismo concepto y no hay ninguna opción viable para añadir información de forma simple para detallar mas la relación ya que la tecnología RDF no lo permite.
* **Mapeo de consultas/SGDB basadas en grafos**: Los grafos una forma muy útil para representar caminos entre entidades, que es justo lo que pretendemos en este caso, expresar las consultas como caminos entre tablas de una base de datos.  
    
  En este análisis partiremos de los siguientes supuestos:
  + Los nodos o vértices del grafo son tablas representadas por sus columnas y campos.
  + Las líneas o aristas del grafos son relaciones entre tablas generadas por los joins de las consultas obtenidos en la fase previa.
  + Los nodos o vértices del grafo tendrán aristas recursivas determinada por aquellas consultas que accedan a un campo de la tabla por valor y no por relación, por ejemplo: … WHERE campo = “valor”

INTRODUCIR GRAFICO DE EJEMPLO  
  
Para esta investigación se ha escogido esta forma de mapear las consultas con la base de datos por los siguientes motivos

* No es necesaria la ejecución de las consultas
* Es una forma visual de representación, un grafo es “visible sobre el papel”
* No es necesaria ninguna tecnología semántica que limite el estudio (caso de RDF)
* Existen tecnologías que nos permitirán obtener información del grafo de forma automática
* Existen algoritmos ampliamente estudiados y documentados de búsqueda en grafos y obtención de caminos.

## 6.3 análisis intermedio de la estructura semántica

Tras haber elegido el análisis de grafos debemos responder a las siguientes cuestiones

* **¿Como damos peso a las aristas del grafo?**  
    
  Como habíamos dicho previamente, las aristas del grafo son las relaciones entre dos campos de dos nodos/tablas, es decir:

Nodo A, campo A 🡪 Nodo B, Campo B

Viendo este hecho y teniendo en cuenta que lo que buscamos es analizar de qué forma se utiliza una aplicación los pesos de las aristas serán las repeticiones de aparición de las relaciones de los nodos, es decir, cuantas veces aparece una relación de dos campos en las consultas. Para elegir un sistema de puntuación se ha evaluado las siguientes opciones:

* **Lineal**: x apariciones -> x peso
* **Fibonacci**: x apariciones -> Fibonacci(x) peso
* **Exponencial**: x apariciones -> exp(x) peso

Tras estudiar la aparición de dichas relaciones entre campos obtenemos el siguiente grafico:

INSERTAR GRAFICO

Por lo tanto, seleccionamos la opción **Lineal** debido a que es suficientemente precisa.

* **¿Como definimos que caminos han de ser analizados y cuáles no?**En este punto hemos de definir una forma de cribar el sistema, es decir, una forma de eliminar relaciones del grafo previamente a la ejecución del algoritmo  
    
  Para ello debemos proveer al sistema de un parámetro que represente el mínimo a por debajo del cual se den por marginales las relaciones.  
    
  En nuestro caso hemos escogido la siguiente formula: **Mayor peso \* 0.3**Esto es debido a que tras realizar esta parte del estudio hemos visto que pesos menores a este son contraproducente en iteraciones altas del sistema
* **¿En qué orden priorizamos dichos caminos?**Dado que cuanto mayor sea el peso de una real acción más afecta en el uso de la base de datos la priorización será de mayor peso a menor

### 6.4 Resumen de elecciones tomadas

Para llevar a cabo la ejecución hemos seleccionado lo siguiente:

* Mapeo de consultas/objeto a través de objetos JSON
* Mapeo de consultas/SGDB en grafo
  + Evaluación de pesos a través de numero de apariciones y de forma lineal
  + Eliminación de relaciones cuyo peso sea menor que -> **Mayor peso \* 0.3**
  + Priorización de mayor a menor peso

## 6.5 Obtención de salidas

Tras Ejecutar la parte de generación del grafo y de evaluación de pesos se obtendrá un grafo con aristas evaluadas que representara que relaciones tienen mas importancia en la base de datos.

El siguiente paso será definir como transformar estas relaciones (joins) en universos semánticos donde la información de una relación sea un concepto semántico.

Aquí tenemos dos métodos que evaluar:

* Generación de un universo semántico basado en la unión de los campos que aparecen en la relación
  + Este método consiste en generar una estructura tabular que contenga la información de la relación, así como los campos extra mas buscados junto con dicha relación.  
    Por ejemplo:   
    Para la arista/relación 🡪   
    Texto, Carta

    Descripción generada automáticamente  
    Obtendríamos el siguiente universo semántico:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo\_A | Campo\_B | Campo\_1 | Campo\_2 | Campo\_3 | Campo\_4 |
| - | - | - | - | - | - |

* Generación del universo semántico basado en la unión de las tablas que son afectadas por la relación:
  + Este método consiste en generar una estructura tabular que contenga la información de la relación, así como los campos extra perteneciente a ambas partes de la relación.  
    Es un caso parecido al anterior salvo por que añadimos todos los campos de las tablas, no solo los afectados por la relación.

Posteriormente y tras obtener el resultado semántico de esta primera iteración se volverá a iterar eliminando la relación y explorada y seleccionando la siguiente en orden de prioridad, guardando los resultados de cada iteración realizada y teniendo en cuenta que a mayor iteración menor impacto.

### 6.5.1 Elección de formato de salida

### 6.5.2 Regeneración del sistema de estructuración de datos

Completar con información en el otro pc

## 6.6 aAálisis de resultados

Completar con información en el otro pc

### 6.6.1 Mediciones

Completar con información en el otro pc

### 6.7.2 Comparativas

Completar con información en el otro pc

# Capitulo 7 Conclusiones

Completar con información en el otro pc

# Capítulo 8 Integración con Sistemas Distribuidos

Completar con información en el otro pc

# Capítulo 9 Anexos

## Anexo. Definiciones y abreviaturas

En este apartado repasaremos las definiciones de abreviaturas y siglas utilizadas.

### Definiciones

* Universo semántico
* Capa semántica

### Abreviaturas

* JSON

Javascript object notation

Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

* RDF

Es un modelo conceptual que sirve para proporcionar información descriptiva sobre los recursos que se encuentran en la Web, permitiendo el intercambio de información a través de diferentes aplicaciones sin que los datos pierdan su significado

* SGDB

Sistema de gestión de bases de datos.

Un SGBD es un conjunto de programas que nos permiten gestionar bases de datos. Es decir, realiza las funciones de modificar, extraer y almacenar información de una base de datos, además de poseer herramientas con funciones de eliminar, modificar, analizar, etc… datos de estas.

* JOIN

Join es el proceso de tomar datos de varias tablas y colocarlos en una vista generada. Por tanto, una instrucción de “SQL JOIN” en un comando Select combina las columnas entre una o más tablas en una base de datos relacional y retorna a un conjunto de datos.

* Mapeo

Es el proceso de transformar un formato de entrada en un formato de salida.

Otra forma de utilizar este acrónimo es para referirnos a como se entrelazan 2 conceptos

* SQL

SQL es un acrónimo en inglés para Structured Query Language. Un Lenguaje de Consulta Estructurado. Un tipo de lenguaje de programación que te permite manipular y descargar datos de una base de datos. Tiene capacidad de hacer cálculos avanzados y álgebra.

* SAP

SAP ayuda a empresas y organizaciones de todos los tamaños e industrias a gestionar sus negocios de manera rentable, a adaptarse continuamente y a crecer de manera sostenible. La empresa desarrolla soluciones de software usadas por pequeñas empresas, compañías medianas y grandes corporaciones.

## Anexo. Análisis de riesgos

Como resultado de la fase de análisis de riesgos hemos obtenido una lista priorizada de riesgos, cada uno de ellos con: sus valores de impacto correspondientes, la respuesta al riesgo y la estrategia tomada frente al riesgo.

|  |  |
| --- | --- |
| ID | 1 |
| Titulo | FALTA DE ENTENDIMIENTO CON EL CLIENTE |
| Descripción | Pérdida de calidad de la investigación por falta de entendimiento en las especificaciones del cliente. |
| Categoría | Externo |
| Probabilidad | Muy Alta |
| Impacto | |  |  | | --- | --- | | Presupuesto | Critico | | Planificación | Critico | | Alcance | Critico | | Calidad | Critico | |
| Impacto total | 0.81 |
| Respuesta | Se realizarán reuniones periódicas para asegurar que todo marcha según la línea base especificada por el cliente |
| Estrategia | Mitigar el riesgo |

|  |  |
| --- | --- |
| ID | 2 |
| Titulo | CAMBIOS DRASTICOS EN EL ESTADO DEL ARTE |
| Descripción | Cambios en la tecnología que provoque un cambio importante en la dirección en la que la misma evoluciona, por ejemplo amplias mejoras en la performance de acceso a datos distribuidos |
| Categoría | Externo- tecnológico |
| Probabilidad | Muy Baja |
| Impacto | |  |  | | --- | --- | | Presupuesto | Critico | | Planificación | Critico | | Alcance | Critico | | Calidad | Inapreciable | |
| Impacto total | 0.09 |
| Respuesta | Se asume el riesgo ya que este hecho a pesar de ser muy poco probable sería imposible de mitigar ya que la tecnología cambia constantemente |
| Estrategia | Asumir el riesgo |

|  |  |
| --- | --- |
| ID | 3 |
| Titulo | FALTA DE CONOCIMIENTOS POR PARTE DEL CLIENTE |
| Descripción | El análisis del estudio posterior a su realización será realizado por personal que no ha realizado el estudio por lo que puede no entender el proceso de análisis |
| Categoría | Externo |
| Probabilidad | Muy Alta |
| Impacto | |  |  | | --- | --- | | Presupuesto | Bajo | | Planificación | Critico | | Alcance | Critico | | Calidad | Critico | |
| Impacto total | 0.81 |
| Respuesta | Se realizarán reuniones periódicas para asegurar que todo marcha según la línea base especificada por el cliente y que todo está correcto en el entendimiento del cliente |
| Estrategia | Mitigar el riesgo |

## Anexo. Presupuestos

Completar con información en el otro pc