

2014



CsP Private API

PROYECTO - EXAMEN GENERAL PRIVADO

VICTOR HARED CIFUENTES MÉNDEZ

0903-08-16287

Contenido

INICIACIÓN	- 1 -
Introducción de propuesta del proyecto.....	- 1 -
Interesados	- 1 -
Descripción de la propuesta inicial	- 1 -
Objetivos del proyecto (estimados)	- 1 -
Listado de interesados (Stakeholders)	- 1 -
Creación de ficha informativa.....	- 1 -
GENERALIDADES	- 2 -
Título del proyecto	- 2 -
Periodo de Ejecución	- 2 -
Localización.....	- 2 -
Autores y Unidad Ejecutora	- 2 -
Autor intelectual.....	- 2 -
Firma desarrolladora	- 2 -
Productor Ejecutivo.....	- 2 -
Beneficiarios.....	- 3 -
Internos.....	- 3 -
Externos.....	- 3 -
Costos y Financiamiento.....	- 3 -
Costos	- 3 -
Financiamiento.....	- 3 -
PLANTEAMIENTO	- 4 -
Planteamiento del proyecto.....	- 4 -
Definición del Proyecto	- 5 -
Antecedentes	- 5 -
Justificación del Proyecto	- 5 -
Recopilación de los Requisitos	- 6 -
Especificación de Requerimientos.....	- 6 -
Objetivos del proyecto	- 7 -
Propuesta final de proyecto.....	- 7 -
Descripción técnica	- 8 -
PLANIFICACIÓN DEL PROYECTO.....	- 8 -

Definición de los Alcances.....	- 10 -
Cronograma de Actividades.....	- 11 -
ESTUDIOS Y FACTIBILIDADES DEL PROYECTO	- 12 -
Estudio del proyecto	- 12 -
Factibilidad Técnica	- 12 -
Factibilidad Económica	- 13 -
Factibilidad Operativa.....	- 13 -
Conclusiones del análisis de factibilidades	- 13 -
Análisis de Costos	- 14 -
Presupuesto económico del proyecto.....	- 16 -
ANÁLISIS GENERAL DEL PROYECTO	- 17 -
Revisión de los Requerimientos.....	- 17 -
FODA del Proyecto.....	- 17 -
Diagrama General de Casos de Uso	- 18 -
Diagrama de Secuencias	- 19 -
Diagramas de Comunicaciones	- 20 -
DISEÑO GENERAL DEL PROYECTO	- 21 -
Diagrama Entidad Relación	- 21 -
Diccionario de Datos	- 22 -
Diagrama de Clases	- 22 -
Diagrama de Componentes.....	- 23 -
Diagrama de Despliegue.....	- 24 -
Descripción de la Arquitectura a utilizar.....	- 25 -
GESTIONES.....	- 27 -
Plan de Gestión de riesgos	- 27 -
Objetivos del plan de gestión de riesgos	- 27 -
Metodología de Gestión de Riesgos.....	- 27 -
Identificación de riesgos potenciales	- 29 -
Clasificación de los riesgos por impacto.....	- 29 -
Plan de Reducción, Supervisión y Gestión de Riesgo (RSGR)	- 30 -
Plan de Recuperación de Desastres.....	- 31 -
Objetivos del plan de recuperación de desastres	- 31 -
Definición del plan.....	- 31 -
Alcance	- 32 -

Protección de Datos y Software	- 32 -
Protección de Hardware	- 32 -
Criticidad por servicios	- 33 -
Informe de riesgos	- 34 -
Estrategias de recuperación ante desastres	- 34 -
Plan de Gestión de Comunicaciones	- 35 -
Matriz de comunicaciones	- 37 -
Factores considerados para el análisis y diseño	- 38 -
PARTE I: SERVIDOR REST	- 38 -
Análisis	- 38 -
Diagrama de Casos de Uso	- 38 -
Diseño	- 39 -
Diagrama de Componentes	- 39 -
Diagrama de Despliegue	- 39 -
Adaptación a Base de Datos	- 40 -
Codificación	- 40 -
Script de base de datos	- 40 -
Verificación y pruebas	- 40 -
Informe de pruebas	- 40 -
Producción	- 40 -
PARTE II: API RESTful	- 41 -
Análisis	- 41 -
Diagrama de Casos de Uso	- 41 -
Descripción de Casos de Uso	- 41 -
Diagrama de Secuencias	- 44 -
Diagrama de Comunicaciones	- 45 -
Diseño	- 45 -
Diagrama de Componentes	- 45 -
Diagrama de Despliegue	- 46 -
Codificación	- 46 -
Pruebas de control	- 46 -
Consultas de prueba a base de datos	- 46 -
Informe de pruebas	- 47 -
Documentación	- 47 -

Producción	- 48 -
PARTE III: CLIENTE API.....	- 48 -
Análisis.....	- 48 -
Diagrama de Casos de Uso	- 48 -
Diagrama de Secuencias	- 49 -
Diagrama de Comunicaciones	- 49 -
Diseño	- 50 -
Diagrama de Componentes	- 50 -
Diagrama de Despliegue	- 50 -
Codificación.....	- 51 -
Vistas de la Interfaz de Usuario	- 51 -
Pruebas de control	- 55 -
Informe de pruebas.....	- 55 -
Documentación	- 56 -
Producción	- 57 -
CIERRE	- 57 -
Cierre del proyecto.....	- 57 -
ANEXOS.....	- 58 -
Diccionario de datos.....	- 58 -
Anexo 2: Script de base de datos.....	- 62 -

INICIACIÓN

Introducción de propuesta del proyecto

Interesados

Ing. Iván Antonio de León
Consultor y Asesor Técnico de Creyent's Productions
Especialista en Servicios Web

Ing. Victor Cifuentes Méndez
CEO Creyent's Productions

Descripción de la propuesta inicial

Desarrollar un servicio web dentro de la organización para permitir a consumidores el acceso a datos de forma controlada mediante una aplicación cliente y a desarrolladores potenciales brindarles lo necesario para que puedan utilizar nuestro servicio de la forma más conveniente según sus necesidades.

Objetivos del proyecto (estimados)

- Crear el servicio web.
- Desarrollar una aplicación cliente.

Listado de interesados (Stakeholders)

<i>ROL GENERAL</i>	<i>STAKEHOLDER</i>
<i>SPONSOR</i>	Ing. Iván Antonio de León
<i>EQUIPO DE PROYECTO</i>	PROJECT MANAGER:
<i>GERENTE DE OPERACIONES</i>	Ing. Victor Hared Cifuentes Méndez Ing. Victor Hared Cifuentes Méndez
<i>USUARIOS</i>	Clientes del proyecto base Consumidores de servicios web (Desarrolladores externos)
<i>PROVEEDORES / SOCIOS DE NEGOCIOS</i>	Comunicaciones: Claro Guatemala

Creación de ficha informativa

Nombre del proyecto:	CsP Private API
-----------------------------	-----------------

Localización geográfica:	Guatemala
Tipo de proyecto:	Servicio Web
Descripción general:	Permitir el acceso a los datos de la empresa mediante servicios controlados sin poner en riesgo los activos desarrollando una aplicación cliente.
Condicionantes y limitantes:	Documentación completa de los avances y aplicativos serán alojados en línea.
Monto de inversión estimado:	\$700.00

GENERALIDADES

Título del proyecto

CsP Private API

Periodo de Ejecución

Del 21 de julio de 2014 al 20 de Agosto de 2013.

Localización

Edificio Central Corporación Creyent's Productions

Autores y Unidad Ejecutora

Autor intelectual

Ing. Iván Antonio de León

Firma desarrolladora

Creyent's Productions

Productor Ejecutivo

Ing. Victor Cifuentes Méndez
CEO Creyent's Productions

Beneficiarios

Internos

Personal administrativo de Creyent's Productions

Externos

Desarrolladores y clientes del Servicio Web

Costos y Financiamiento

Costos

La institución ya cuenta con un servidor donde se ubica el proyecto base, el cual puede servir como servidor web, sin embargo por razones de seguridad se recomienda la adquisición de un servidor exclusivo para el servicio web.

Es necesario realizar el análisis completo del proyecto para establecer con claridad toda la infraestructura a implementar y especificar el costo de la misma.

Financiamiento

Todos los costos serán cubiertos por la empresa misma.

PLANTEAMIENTO

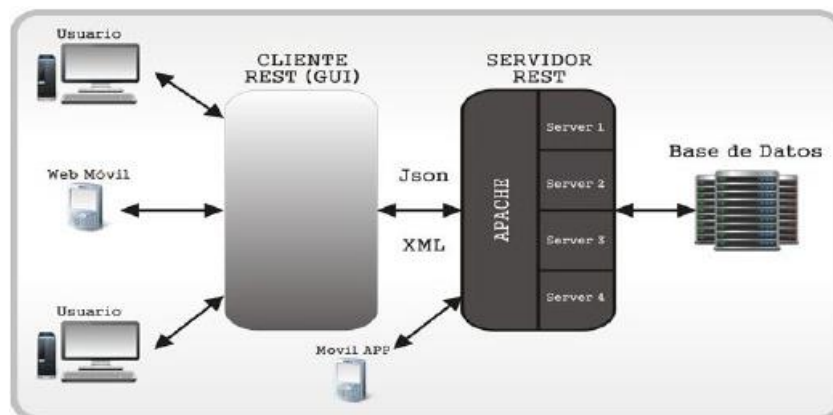
Planteamiento del proyecto

En la actualidad el mundo del desarrollo web cuenta con diferentes matices al momento de realizar transacciones entre proyectos que necesitan acceso a datos, las conexiones se realizan de forma directa entre el servidor web y el servidor de base de datos.

Es por esto que actualmente se recomienda crear una capa intermedia entre dichos servidores, lo que hace que la conexión y transferencia de datos sea transparente y a la vez brinda mayor seguridad y comodidad para los clientes de proyectos de gran alcance o envergadura.



Sin embargo, en proyectos grandes, nuestros clientes o consumidores no son simplemente usuarios comunes, en casos puntuales los consumidores de nuestros productos o servicios son otros desarrolladores, para los cuales debemos ser capaces de brindarles la información que nos soliciten, sin poner en riesgo nuestra propiedad intelectual (llámese código fuente) dándoles acceso a nuestros datos en forma controlada.



Definición del Proyecto

Se cuenta con un proyecto base, el cual sirve para diferentes segmentos de la industria y maneja lo siguiente:

- Clientes y Proveedores
- Productos Genéricos
- Empleados
- Usuarios

Este proyecto es funcional dentro de la empresa, sin embargo se necesita expandir para ir conjuntamente al crecimiento de la organización, razón por la cual se tiene planificado proveer un tipo de servicio mediante el cual más usuarios puedan acceder a los datos del mismo y, tras el desarrollo de nuestro propio cliente externo, la contratación de Programadores FreeLancer para el desarrollo de aplicaciones cliente en diferentes plataformas.

Antecedentes

Se ha visto la necesidad de compartir el acceso a los datos propios de la empresa vía web, pero nunca se ha optado por conceder permiso de tocar el código fuente del proyecto base y menos aún el acceso directo al servidor de bases de datos por políticas de la empresa y por seguridad de los datos.

Justificación del Proyecto

Como empresa se deben contar con activos propios, los cuales incluyen desde planillas de personal hasta el código fuente de las aplicaciones que manejan dichas planillas, en base a esto se opta por emplear un modelo de servicio web conocido como REST (REpresentational State Transfer), mediante el cual se brinda acceso a los datos alojados por la empresa sin que se den a conocer las porciones de código que lo manejan y aunado a esto un control de usuarios en la misma plataforma para limitar los accesos a los datos y generar transacciones seguras con el fin de minimizar los riesgos.

Siendo este el modelo que se requiere para un proyecto como el solicitado, se pretende la creación de una API (*Application Programming Interface* o Interfaz de Programación de Aplicaciones), mediante la cual se hará uso de los servicios web que se estén prestando a través de REST, sin olvidar toda la documentación necesaria para que los desarrolladores que consuman la API puedan hacer uso de la misma.

Concluyendo con el desarrollo del nuevo servicio que prestará la empresa, se llevará a cabo el análisis y desarrollo de una aplicación cliente propia, la cual se pondrá como punta de lanza para la muestra y el uso de nuestros servicios web.

Recopilación de los Requisitos

Por medio de una entrevista con el Sponsor del proyecto, el Ing. Iván de León, se ha generado un documento donde se indican algunos requerimientos para implementar en el proyecto, los cuales se listan a continuación en la especificación de requerimientos.

Especificación de Requerimientos

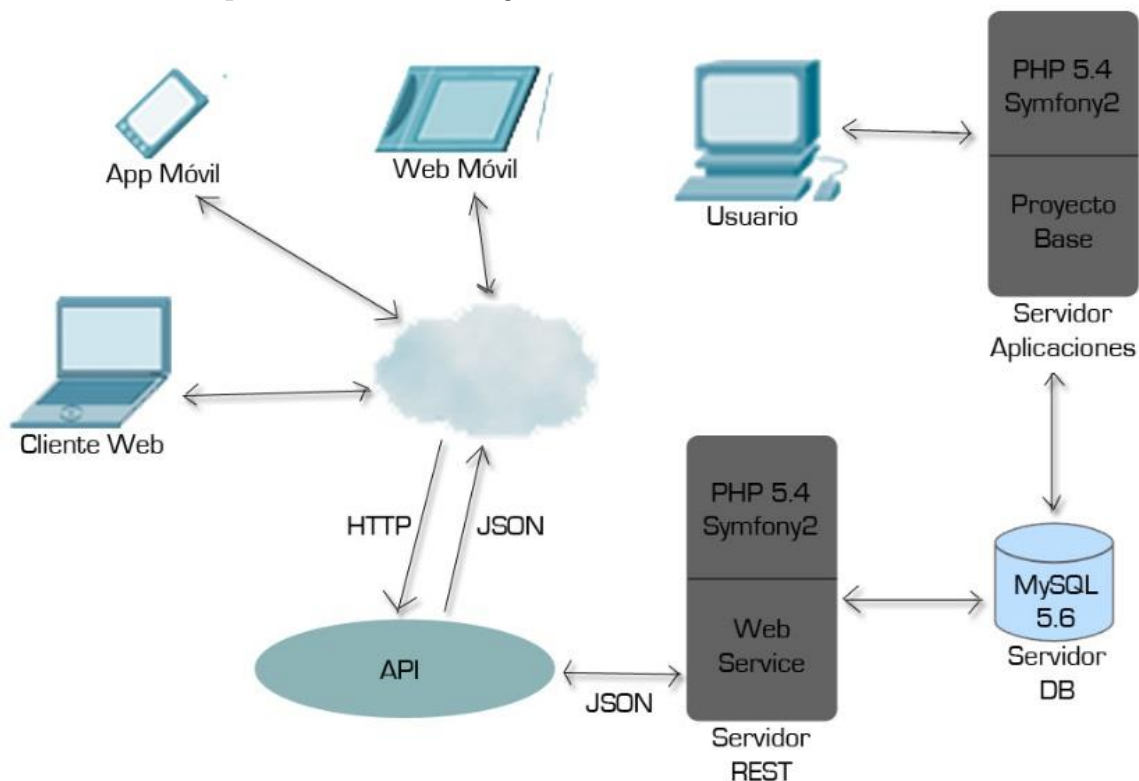
No.	Stakeholder	Requerimientos		Responsable
		Código	Descripción	
1	Sponsor	CSP-R01	LA API se va a desarrollar en el lenguaje de su preferencia (PHP, PYTHON, RUBY).	Equipo de proyecto
2	Sponsor	CSP-R02	LA API debe manejar lo siguiente en el proyecto base: <ul style="list-style-type: none"> - Clientes y Proveedores (Se manejará la información básica de los clientes y proveedores). - Productos (Se manejará la información de los productos genéricos). - Empleados (Se manejará la información de los empleados que laboran dentro de la empresa). - Usuarios (Se manejará la información de los usuarios a los que va a tener acceso la base de datos). 	Equipo de proyecto
3	Sponsor	CSP-R03	La API debe de tener los métodos <ul style="list-style-type: none"> - Insertar (POST) - Actualizar (PUT) - Obtener (GET) 	Equipo de proyecto
4	Sponsor	CSP-R04	Se debe desarrollar una aplicación cliente donde solamente se tendrán funciones básicas para demostrar la funcionalidad de la API.	Equipo de proyecto
5	Sponsor	CSP-R05	Debe presentar la documentación de la API, para que cuando alguien desee usarla tenga toda la documentación lista.	Equipo de proyecto
6	Sponsor	CSP-R06	Debe de manejar todo el código por medio de un gestor GIT o SVN.	Equipo de proyecto

Objetivos del proyecto

- Implementar un servidor REST.
- Diseñar y desarrollar una API RESTful.
- Desarrollar una aplicación cliente para consumir la API.
- Mantener una documentación limpia para los desarrolladores que vayan a utilizar nuestra API.
- Mantenimiento recurrente de un repositorio con la información necesaria.

Propuesta final de proyecto

Teniendo en consideración los puntos anteriores, se plantea implementar una infraestructura de plataforma como la siguiente:



Se pretende implementar un Servidor REST el cual obtendrá acceso directo al Servidor de Bases de Datos, el cual seguirá funcionando sin necesidad de realizar ningún cambio, tanto para él como para el Servidor de Aplicaciones en donde se encuentra instalado el Proyecto Base.

Al Servidor REST se tendrá conexión mediante una API RESTful, se contará con acceso restringido a los usuarios actuales del Proyecto Base y a usuarios nuevos que se creen en cualquiera de las dos plataformas ya que se maneja la misma base de datos.

La API contará con los métodos necesarios para interactuar con los recursos que provea el Servidor REST, la documentación requerida para su correcto funcionamiento será alojada en un repositorio online, GitHub en este caso.

Por último se prevé que las conexiones de los diversos clientes sean por medio de Internet, esto es a causa de que el cliente puede estar en diversas plataformas, ya sea una App Móvil (Smartphone), una Web Móvil (Tablet) o un Cliente Web (Laptop). En cada una de ellas se espera una petición HTTP y el trabajo de la API será devolver una respuesta en JSON como formato predefinido.

Descripción técnica

Componente	Descripción
<i>Servidor REST</i>	Servidor con instalación de Apache y PHP. Servicio Web REST.
<i>API</i>	Symfony2. ORM Entidades de base de datos del proyecto base.
<i>Clientes</i>	GUI Métodos de acceso a la API RESTful
<i>Dirección del repositorio</i>	https://github.com/Creyent/CSP_Private_API

PLANIFICACIÓN DEL PROYECTO

Nombre del proyecto: CsP Private API

Plataforma: A definir.

¿Por qué?: Los clientes de la API pueden ser en diferentes plataformas.

Descripción breve del proyecto: Creación de servicios web para brindar acceso controlado a los datos de la organización.

Recursos	
Humano: <ul style="list-style-type: none"> Ing. Victor Cifuentes Méndez, CEO Creyent's Productions 	Externos: <ul style="list-style-type: none"> Repositorios online Servicio de Internet
De desarrollo: <ul style="list-style-type: none"> IDE's de trabajo Software de diagramación Software de diseño Frameworks Plugins 	De oficina: <ul style="list-style-type: none"> Procesador de texto Servicios de impresión
De evaluación: <ul style="list-style-type: none"> Software de simulación Virtualización de equipos 	Monetarios: <ul style="list-style-type: none"> Según presupuesto a definir
	Temporales: <ul style="list-style-type: none"> 33 días calendario

Actividades a realizar	Tiempo estimado	Recursos necesarios
➤ Planificación	1 día	De oficina
➤ Análisis del proyecto	2 días	De oficina
➤ Desarrollo de propuesta	1 día	De desarrollo y oficina
➤ Servidor REST:		
✓ Análisis	1 día	De desarrollo y oficina
✓ Diagramación	2 días	De desarrollo
✓ Adaptación a BD	1 día	De desarrollo
✓ Codificación	2 días	De desarrollo
✓ Pruebas	1 día	De evaluación
✓ Producción	1 día	De desarrollo y externos
➤ Actualización general del repositorio	1 día	Externos
➤ Revisión de avance con altos ejecutivos	1 día	De oficina
➤ API RESTful		
✓ Análisis	1 día	De desarrollo y oficina
✓ Diseño y desarrollo	1 día	De desarrollo
✓ Codificación	2 días	De desarrollo
✓ Pruebas	1 día	De evaluación
✓ Documentación	2 días	De desarrollo y oficina
✓ Producción	1 día	De desarrollo y externos
➤ Actualización general del repositorio	1 día	Externos
➤ Revisión de avance	1 día	De oficina
➤ Cliente API		
✓ Análisis	1 día	De desarrollo y oficina
✓ Diagramación	2 días	De desarrollo
✓ Codificación	2 días	De desarrollo
✓ Pruebas	2 días	De evaluación
✓ Producción	1 día	De desarrollo y externos
➤ Actualización general del repositorio	1 día	Externos
➤ Revisión general con altos ejecutivos	1 día	De oficina
➤ Actualización general del repositorio antes de producción general y liberación	1 día	Externos
➤ Puesta en producción del proyecto y liberación del Servicio Web RESTful	1 día	Externos

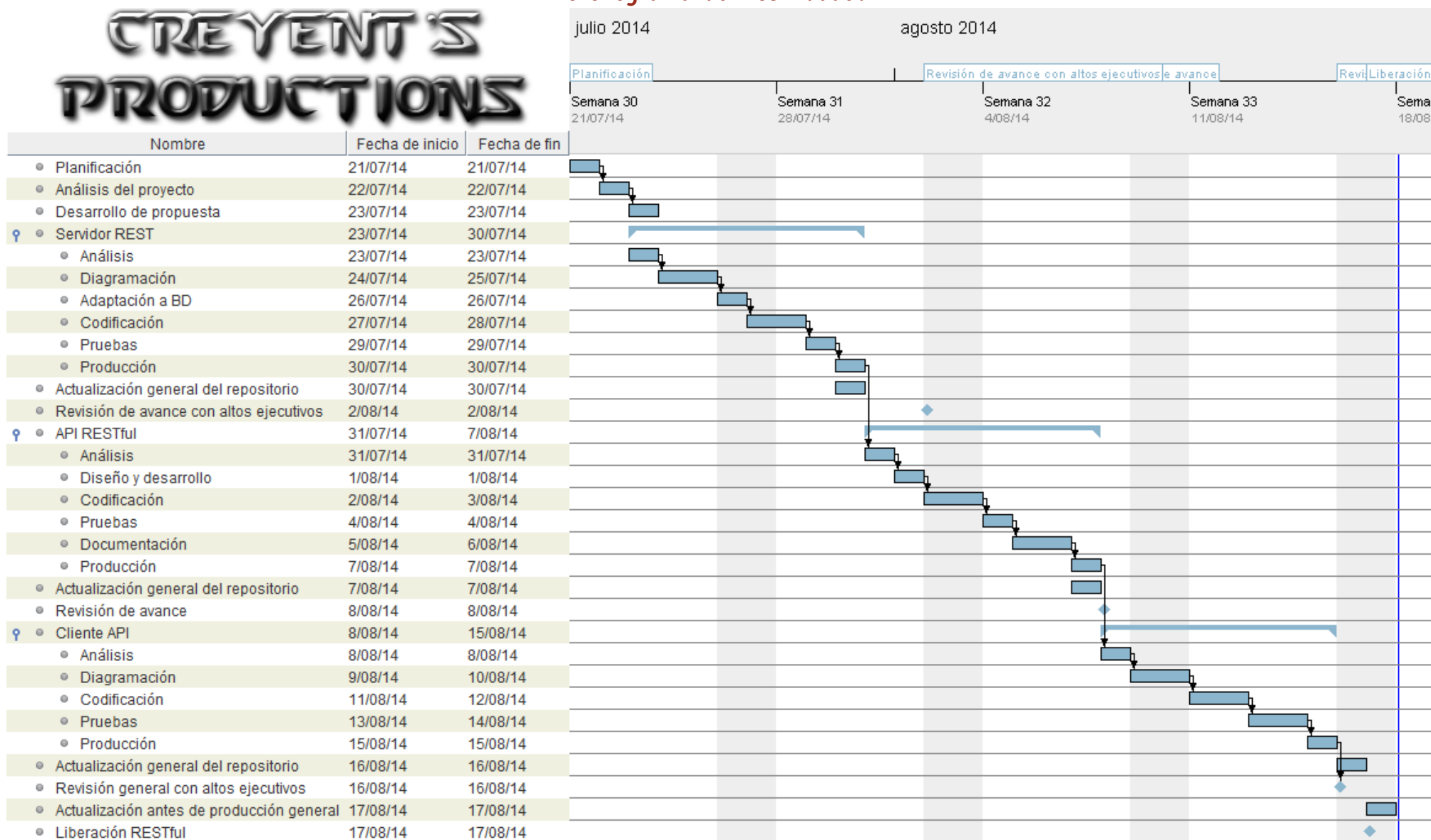
Definición de los Alcances

El proyecto de desarrollo del Servicio Web REST, será implementado bajo las siguientes indicaciones, bajo las cuales se verá delimitado en qué y cómo se realizará.

- a. Script de estructura de la Base de Datos a la que tendrá acceso el Servidor REST.
- b. Recopilación de la información necesaria de la Base de Datos mediante el Diccionario correspondiente.
- c. En el área de diseño se desarrollaran los diagramas básicos necesarios que demuestran la interacción del sistema, consistente en los Diagramas UML:
 1. Diagrama de casos de uso.
 2. Diagrama de secuencia.
 3. Diagrama de comunicaciones.
 4. Diagrama de clases.
 5. Diagrama de componentes.
 6. Diagrama de despliegue.
- d. Implementación completa del servidor REST.
- e. Desarrollo de la API con los verbos básicos de HTTP.
- f. Creación de un Cliente REST que hará uso de la API.
- g. Documentación de la API online.
- h. Código completo en repositorio online (GitHub).

CREYENT'S PRODUCTIONS

Cronograma de Actividades



ESTUDIOS Y FACTIBILIDADES DEL PROYECTO

Estudio del proyecto

En la actualidad las empresas se basan en vender servicios no productos como antaño, es en base a esto que la organización plantea expandir el negocio al crear un Servicio Web mediante el cual los clientes o consumidores puedan acceder a los activos informáticos.

Con este proyecto se realiza una serie de estudios destinados a conocer la viabilidad y factibilidad de su análisis, desarrollo, implementación y mantenimiento en un futuro próximo, tanto para el personal administrativo de la organización como para cualquier empresa de desarrollo o soporte que tome el proyecto una vez se haya realizado.

Factibilidad Técnica

En relación al estudio técnico, se determina que para el correcto funcionamiento del servicio que se pretende implementar, se requiere un equipo de cómputo con perfil de servidor y equipos o dispositivos para ser utilizados como clientes.

Especificaciones técnicas de hardware para el Servidor REST:

- Procesador mínimo de 2 núcleos a 3 GHz.
- Memoria RAM mínima de 4GB.
- Conexión a Internet (Ethernet).
- Sistemas de respaldo de discos (RAD o Espejado).
- Sistemas de respaldo de energía.

Especificaciones técnicas de hardware para clientes:

- Conexión a la Internet (Ethernet, WIFI o Módem).

Software requerido para el Servidor:

- Apache 2.4.7 o posterior.
- MySQL v5.6.14 o posterior.
- PHP v5.5.6 o posterior.
- OpenSSL v1.0.1e o posterior.

Software necesario en las estaciones de trabajo:

- Navegador (recomendado Firefox o Chrome).
- APIClient (aplicación a desarrollar).

Al momento de la realización de este documento es claramente identificable que la mayor parte de servidores cuentan con los requisitos mencionados y, dado que el proyecto

base se encuentra ya en producción, se determina que el equipo requerido ya está disponible dentro de las instalaciones de la organización o es fácilmente adquirible para su implementación.

En términos de infraestructura, la empresa ya cuenta con su red propia y acceso a Internet, por lo que no es necesario realizar ningún cambio concerniente a este aspecto.

Factibilidad Económica

En este punto se observa cuál es el requerimiento de inversión para la realización del proyecto, el cual al ser analizado pretende generar gastos a la organización para la obtención de equipo faltante u obsoleto. Al inicio de este documento se mencionó un costo estimado de \$700, el cual es valuado únicamente para el servidor que se debe adquirir. Sin embargo cabe recalcar el hecho de que sin necesidad de invertir más, se puede llevar a cabo e implementar el servicio con una funcionalidad del 100% utilizando el mismo equipo físico donde se ubica actualmente el proyecto base.

Esto implica que queda a disposición administrativa el optar o no por dicha adquisición, aunque indiferentemente, para el proyecto se presupuestará la compra del servidor a un costo debidamente indicado, aunado a los costos de análisis, diseño y servicios profesionales.

Factibilidad Operativa

En este apartado se estudia las implicaciones que tiene el desarrollo del presente proyecto dentro del personal de la organización, el impacto en términos de aceptación, uso e innovación. Del mismo modo se evalúan las opciones que se han de brindar para los desarrolladores que consuman los servicios que se están por implementar en cuanto a complejidad o documentación para el correcto uso y funcionamiento.

Tras una breve revisión y en introspectiva, es claro que se cumplen operativamente las capacidades o requisitos necesarios para este sistema, dado que los consumidores del servicio serán desarrolladores internos o externos y, las aplicaciones cliente son de uso común, por lo que no se aprecia ningún inconveniente.

Conclusiones del análisis de factibilidades

Al concluir el análisis de factibilidades, se logra establecer que el proyecto que se analiza, diagrama y diseña en el presente documento, es totalmente factible, dado que no se han encontrado obstáculos ni límites relevantes para su desarrollo. Esto implica gastos para la organización, sin embargo es una inversión recuperable mediante la expansión de los procesos y servicios que se prestan dentro de la empresa, aunque no es un gasto obligatorio y puede ser omitido parcialmente.

Análisis de Costos

Es de aclarar que este proyecto cuenta con un sistema base ya en producción, por lo que el análisis de costos se enfoca únicamente en el análisis y desarrollo del servicio web y el cliente, a lo que se le suma la implementación directa en el servidor.

Determinación de Entradas Externas (EE), Salidas Externas (SE) y Consultas Externas (CE):

Entradas Externas

1. Ingreso de categorías
2. Modificación de categorías
3. Borrado de categorías
4. Ingreso de empleados
5. Modificación de empleados
6. Eliminación de empleados
7. Ingreso de proveedores
8. Modificación de proveedores
9. Borrado de proveedores
10. Ingreso de clientes
11. Modificación de clientes
12. Borrado de clientes
13. Ingreso de productos
14. Modificación de productos
15. Borrado de productos

Salidas Externas

1. Formulario de categorías
2. Listado de categorías
3. Formulario de empleados
4. Listado de empleados
5. Formulario de proveedores
6. Listado de proveedores
7. Formulario de clientes
8. Listado de clientes
9. Formulario de productos
10. Listado de productos

Consultas Externas

1. Entidad categorías en proyecto base
2. Entidad empleados en proyecto base
3. Entidad proveedores en proyecto base
4. Entidad clientes en proyecto base
5. Entidad productos en proyecto base

Archivos Lógicos Internos

1. Base de datos temporal (Android)

Archivos de Interfaz Externos

1. Login Oauth

Es de aclarar que en esta estimación se toma en cuenta la aplicación cliente únicamente, dado que el servicio REST reutiliza las entidades ya implementadas en el proyecto base, por lo que se utiliza un método de estimación para proyectos de ingeniería web, el cual toma base de Puntos de Función modificados en conjunto con los pasos de la estimación en proyectos ágiles.

Tabla de Puntos de Función

Valor de dominio de información	Conteo		Factor ponderado							
			Simple		Promedio		Complejo			
Entradas Externas (EE)	15	x	5	3	10	4		6	=	825
Salidas Externas (SE)	10	x	5	4	5	5		7	=	450
Consultas Externas (CE)	5	x	5	3		4		6	=	75
Archivos Lógicos Internos (ALI)	1	x		7	1	10		15	=	10
Archivos de Interfaz Externos (AIE)	1	x		5	1	7		10	=	7
Conteo Total										1367

Respuestas:	Valor
1: Respaldo y recuperación	5
2: Comunicaciones de datos	5
3: Procesamiento distribuido	2
4: Rendimiento crítico	4
5: Existencia de entorno operativo	3
6: Entrada de datos en línea	5
7: Transacción de entrada sobre múltiples pantallas	3
8: Archivos maestros actualizados en línea	5
9: Complejo de valores de dominio de información	2
10: Complejo de procesamiento interno	2
11: Código diseñado para reuso	4
12: Conversión/instalación en diseño	1
13: Instalaciones múltiples	0
14: Aplicación diseñada para cambio	3

PF= 1490

Autor:
Ing. Victor Cifuentes Méndez

Utilizando la técnica para la estimación basada en puntos de función y teniendo en cuenta el tiempo de desarrollo de proyecto y el esfuerzo utilizado (personal), la fórmula para estimar el costo del proyecto está dada por la siguiente formula:

Costo del Software

Costo total = PF * Costo unitario por PF

Costo = Numero de PF * Q.125.00

Costo = según los cálculos y los datos que alimentaran la formula

Presupuesto económico del proyecto

Para la ejecución del proyecto dentro y fuera de la organización, se ha establecido el siguiente presupuesto.

Ítem	Valor	
Administrativo		
Útiles y Papelería	Q50.00	\$6.39
Comunicación (Internet, saldo de celular)	Q200.00	\$25.58
Energía eléctrica	Q200.00	\$25.58
Gastos generales (transporte, imprevistos, etc.)	Q200.00	\$25.58
Análisis		
Factibilidad técnica	Q1,000.00	\$127.88
Factibilidad económica	Q700.00	\$89.51
Factibilidad operativa	Q1,000.00	\$127.88
Hardware (opcional)		
Servidor PowerEdge R220 Rack: Intel® Celeron® G1820 2.7GHz, 2M Cache, 4GB UDIMM, 500GB 7.2k RPM SATA	Q4,915.00	\$628.52
Seguridad (opcional)		
Sistema de respaldo de energía	Q1,500.00	\$191.82
Servicios Profesionales		
Diseño de sistema	Q1,000.00	\$127.88
Adaptación (al proyecto base)	Q500.00	\$63.94
Implementación del sistema	Q1,500.00	\$191.82
Total	Q12,765.00	\$1,632.35

Los gastos que se especifican en este presupuesto serán cubiertos en su totalidad por la organización Creyent's Productions, esto es debido a que el proyecto se realiza con fines de expansión empresarial.

ANÁLISIS GENERAL DEL PROYECTO

Revisión de los Requerimientos

Durante las respectivas depuraciones de requerimientos, se ha determinado que todos los requerimientos son totalmente factibles y su implementación será completa, siempre con la finalidad de mejorar el proyecto y sus respectivos entregables.

FODA del Proyecto

Interiores	Fortalezas <ul style="list-style-type: none"> • Beneficios intangibles • Optimización de tiempos • Expansión empresarial 	Oportunidades <ul style="list-style-type: none"> • Mejoramiento de procesos institucionales • Reducción de problemáticas de ubicación 	Exteriores
	Debilidades <ul style="list-style-type: none"> • Tiempo de desarrollo • Costo de implementación • Tiempo de adaptación 	Amenazas <ul style="list-style-type: none"> • Resistencia al cambio • Falta de aptitudes 	

Diagrama General de Casos de Uso

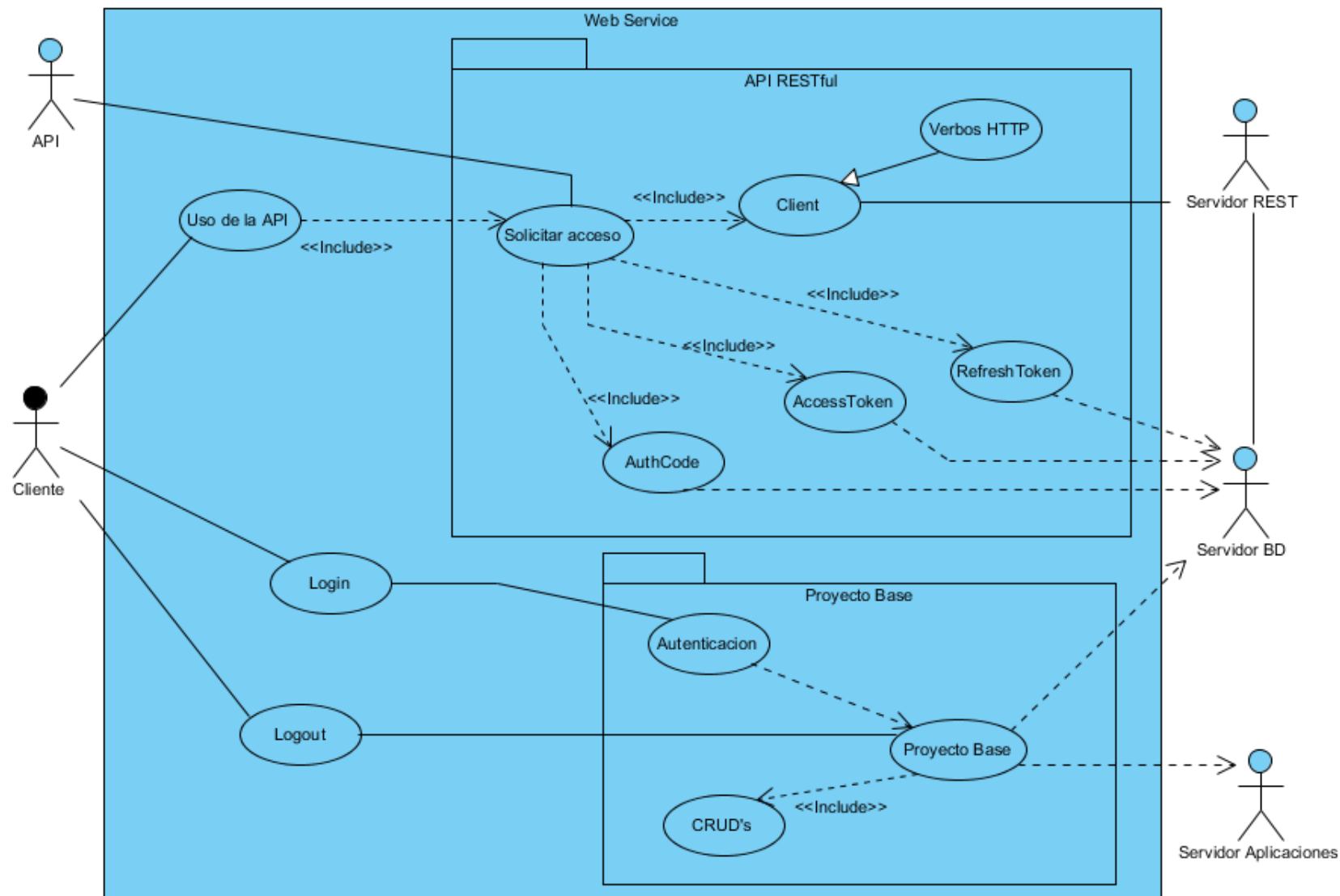
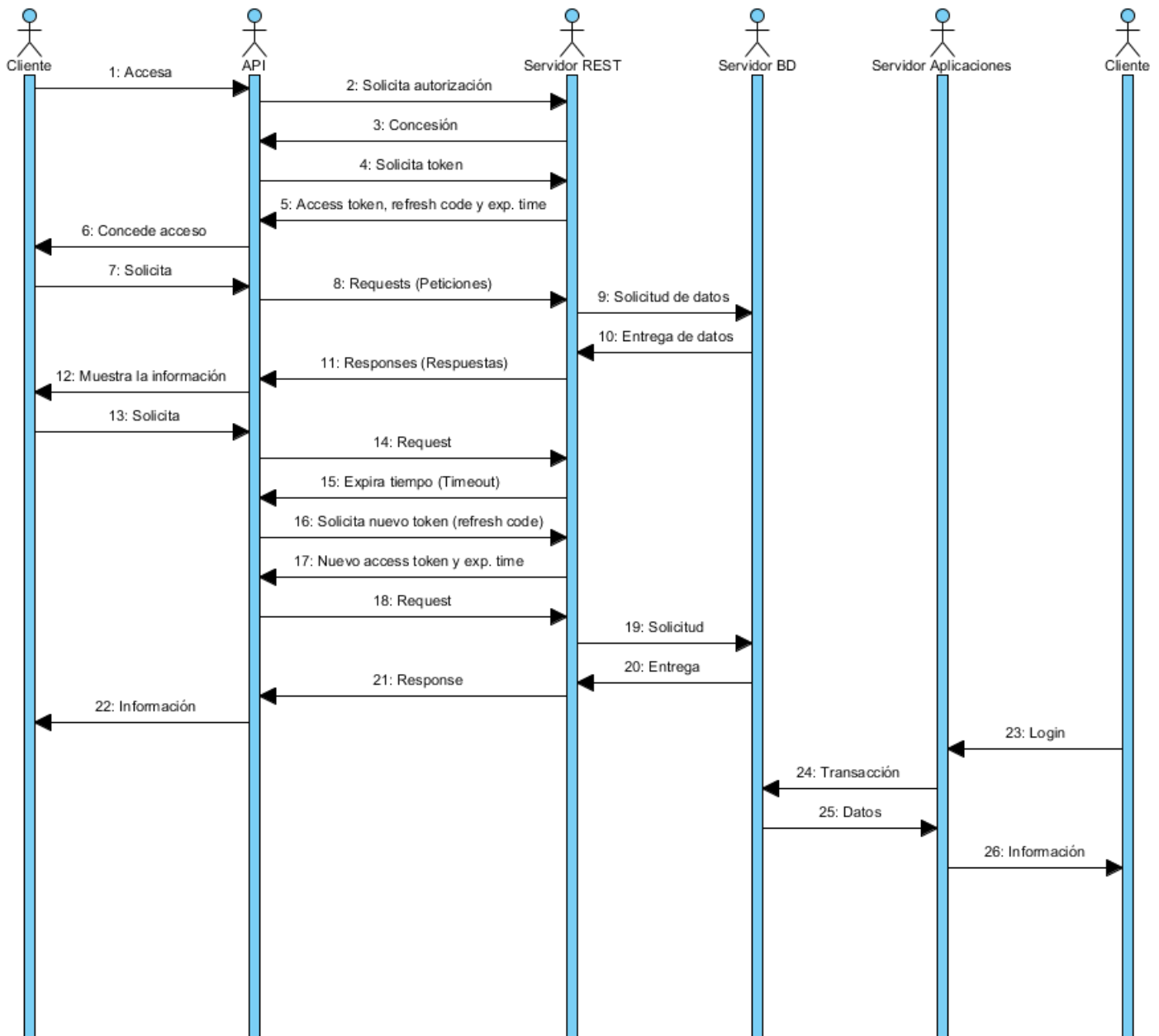
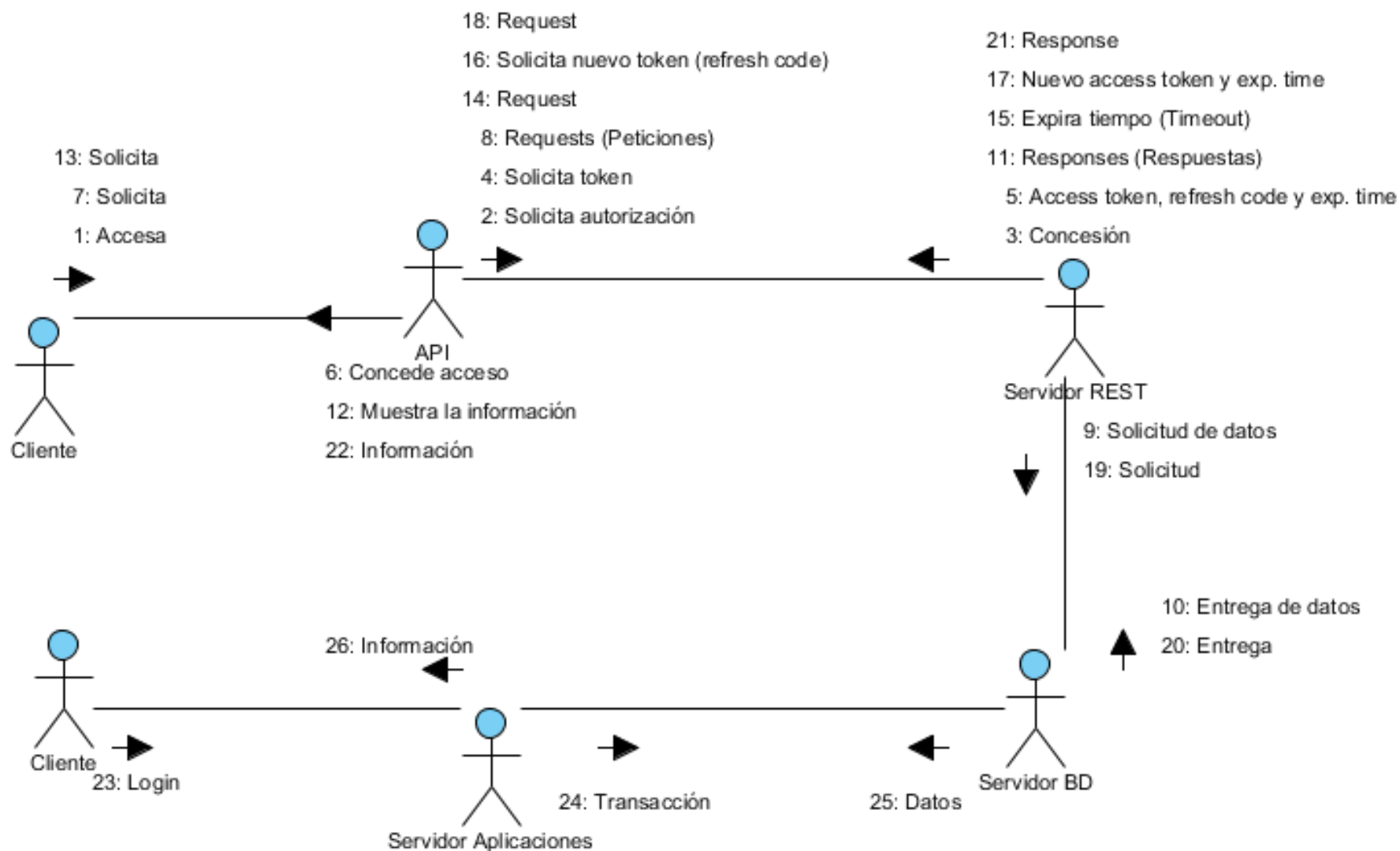


Diagrama de Secuencias

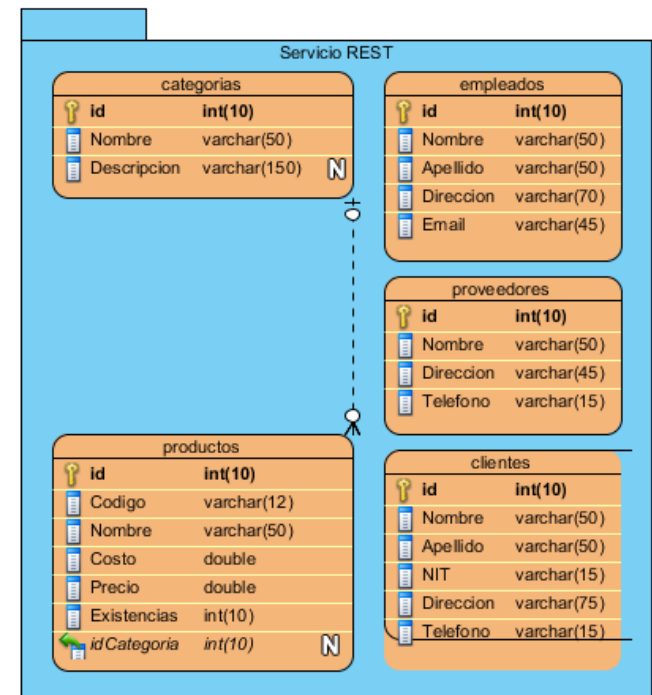
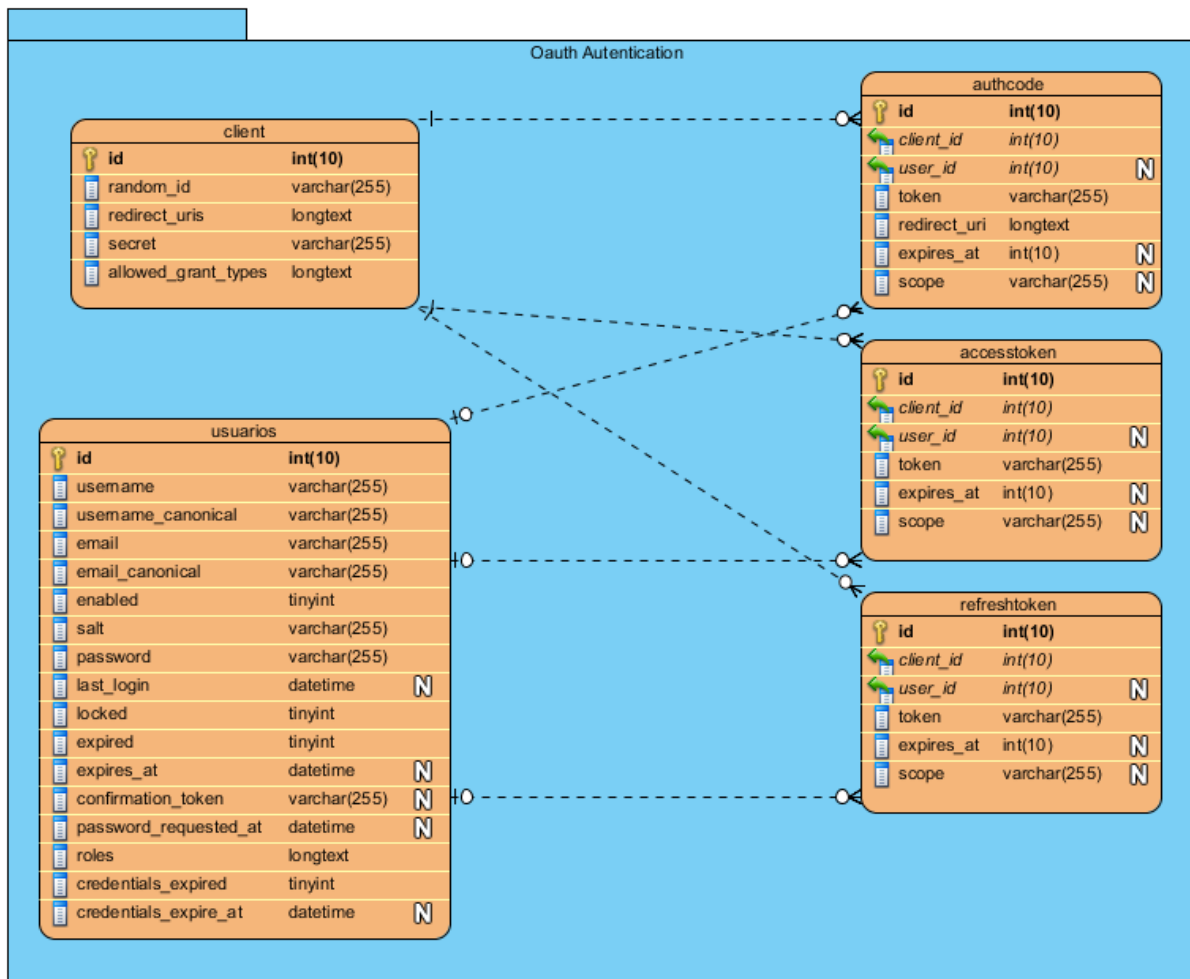


Diagramas de Comunicaciones



DISEÑO GENERAL DEL PROYECTO

Diagrama Entidad Relación



Diccionario de Datos

Diagrama de Clases

Ver Anexo 1.

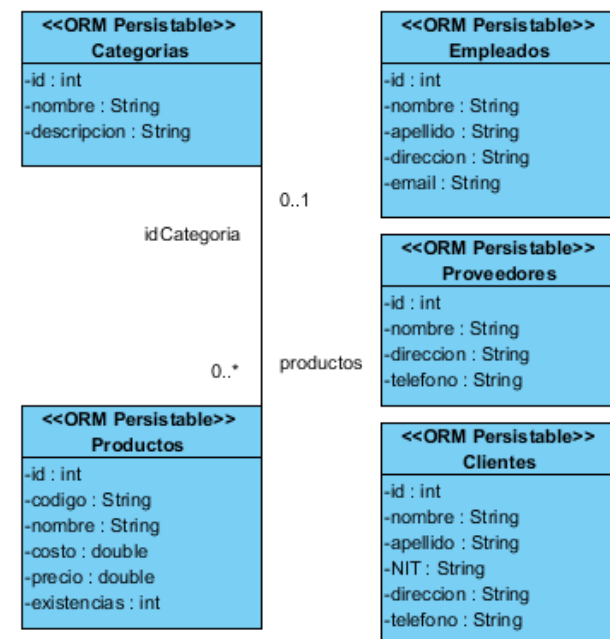
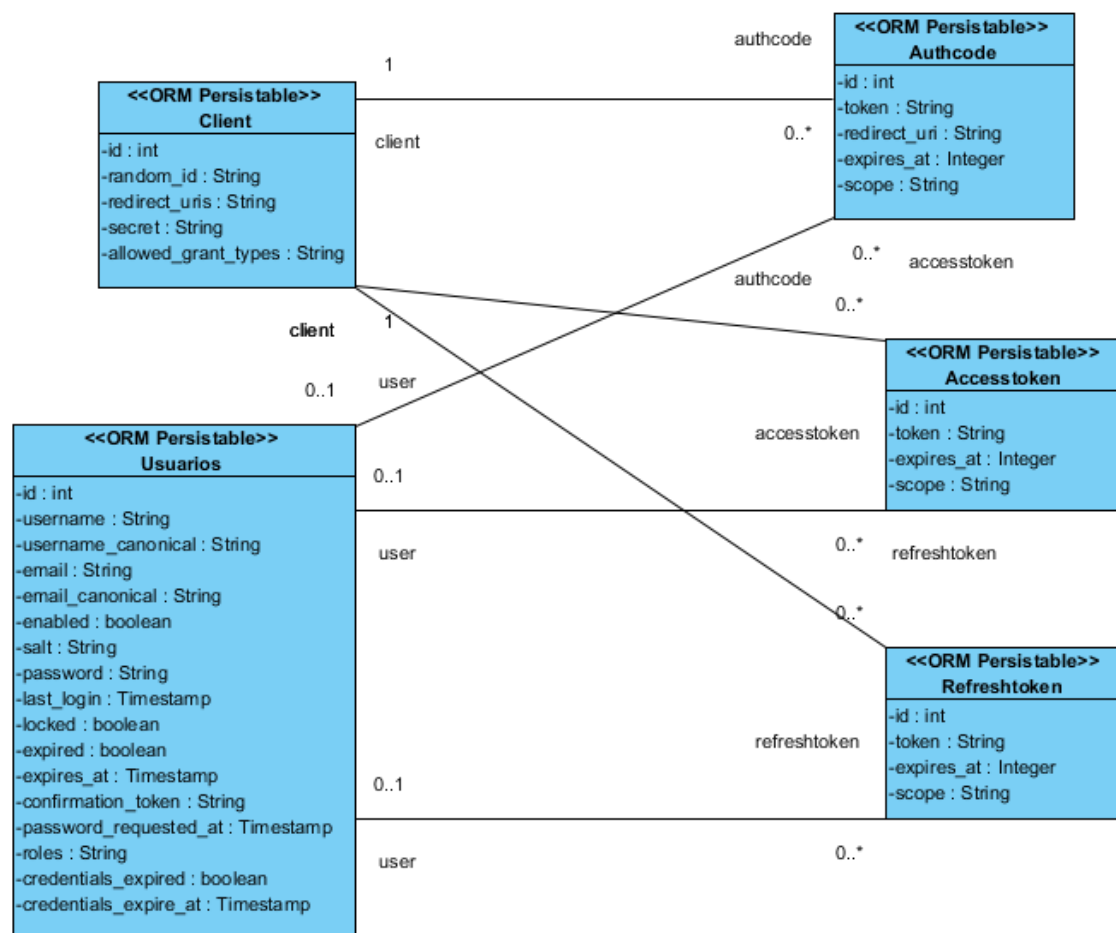


Diagrama de Componentes

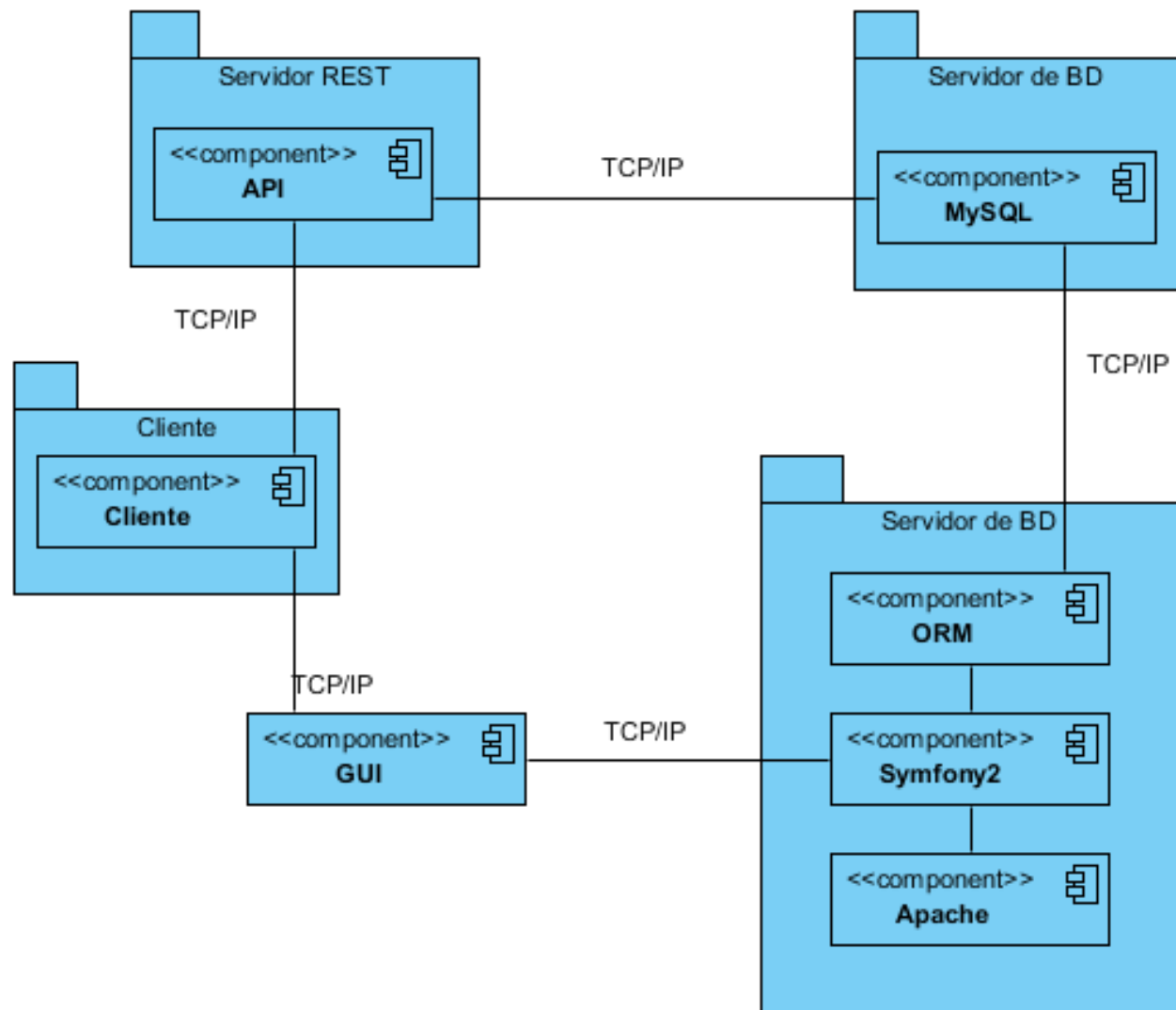
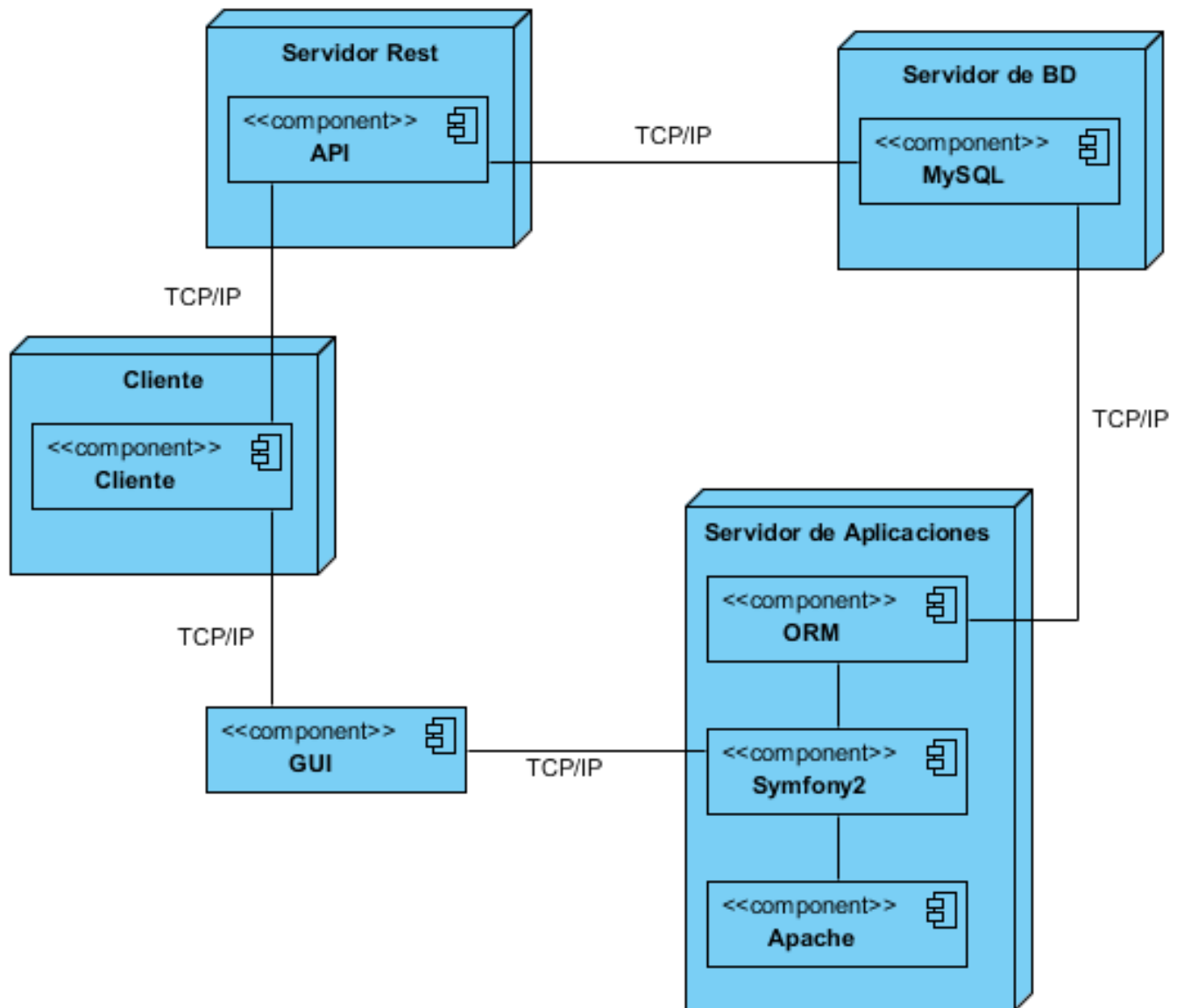
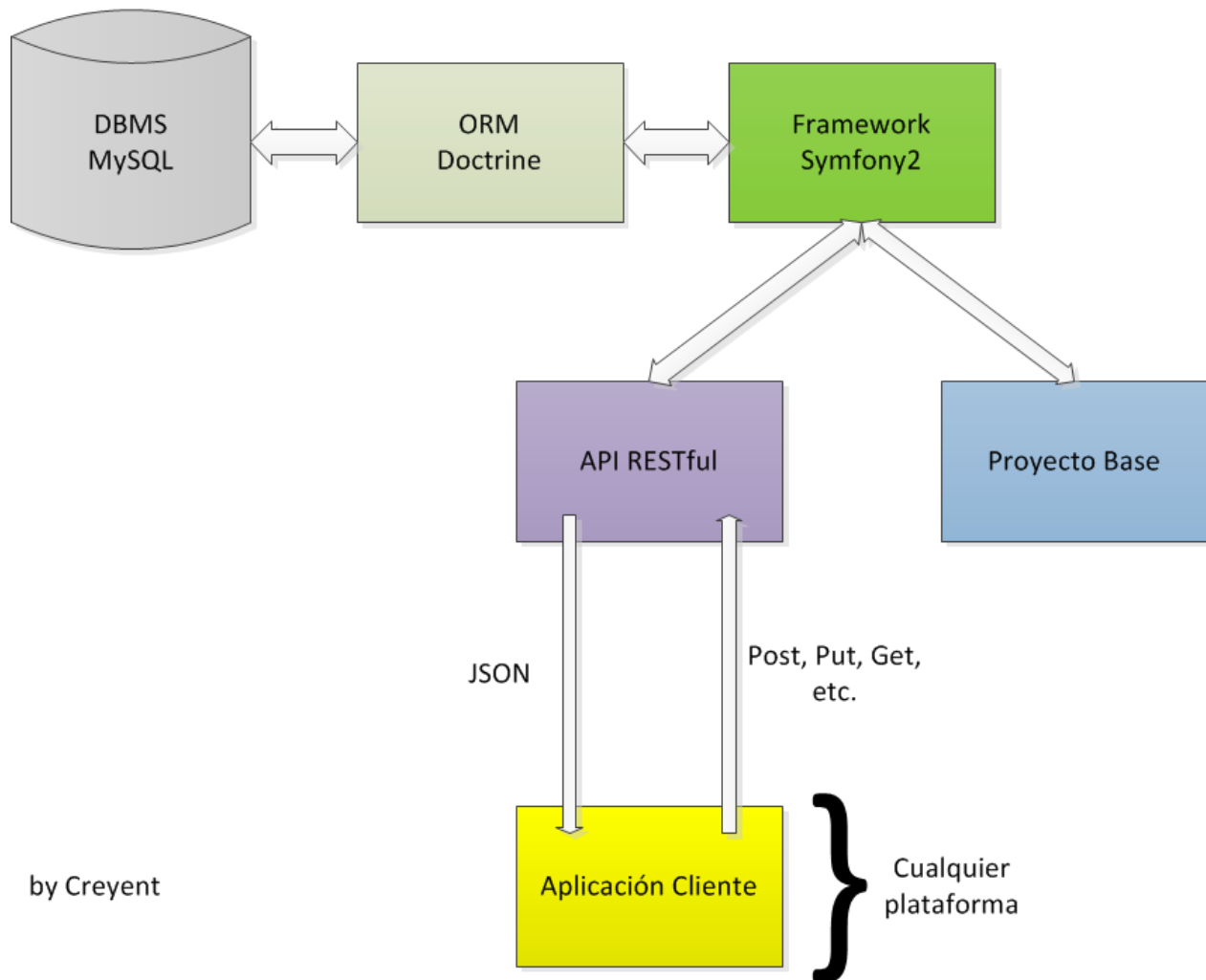


Diagrama de Despliegue



Descripción de la Arquitectura a utilizar

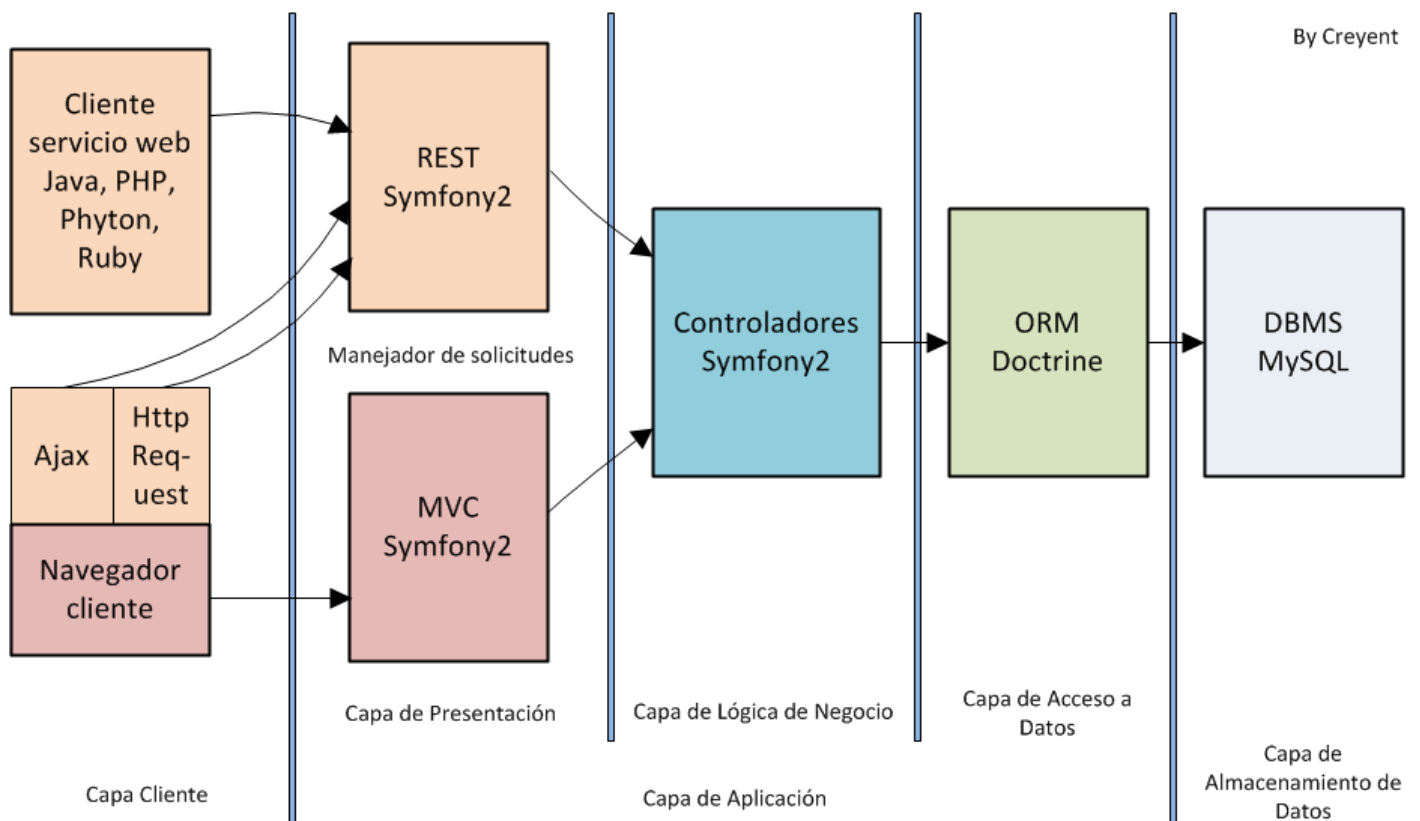
A grandes rasgos y como se había comentado anteriormente, el proyecto en general cuenta con tres componentes principales, los cuales son el proyecto base, la API RESTful y el Cliente API. A continuación se muestra un diagrama de su composición base:



El sistema cuenta con un Proyecto Base Genérico y un servicio REST implementados en PHP con el Framework de trabajo Symfony2, conjuntamente se encuentra el Cliente que hará uso de la API para el servicio REST, el cual puede estar implementado en diversas plataformas.

Para este cliente se han escogido dos plataformas de implementación, las cuales serán Mobile y Web, desarrolladas en Android y el mismo Symfony.

En general el proyecto llevará a cabo su construcción en diversas capas, como por ejemplo las tres capas del modelo MVC, siendo estas las entidades (tablas en bases de datos) representadas en el Modelo (Capa de Acceso a Datos), las Vistas o pantallas de ingreso de datos o informativas generadas (Capa de Presentación) y los Controladores que se encargan de toda la Lógica del Negocio. Aunado a éstas se cuentan la Capa Cliente y la Capa de Almacenamiento de Datos. A continuación se muestra un diagrama el cual servirá para darse una idea más clara de lo mencionado:



GESTIONES

Plan de Gestión de riesgos

Objetivos del plan de gestión de riesgos

- Minimizar el impacto de los eventos adversos que arriesguen el desarrollo del proyecto.
- Maximizar la probabilidad e impacto de eventos positivos para lograr éxito y calidad en el desarrollo de software.
- Identificar eventos de riesgo para darle tratamiento adecuado de forma anticipada con el fin de evitar que repercutan grandemente en el desarrollo del proyecto.
- Clasificar los eventos de riesgo para proveer un tratamiento y observación de dichos ítems.
- Estar preparados ante cualquier eventualidad adversa que ponga en riesgo el desarrollo del proyecto.

Metodología de Gestión de Riesgos

PROCESO	DESCRIPCIÓN	HERRAMIENTAS	FUENTES DE INFORMACIÓN
Planificación de Gestión de los Riesgos	Elaborar Plan de Gestión de los Riesgos.	PMBOK PMI	Sponsor
Identificación de Riesgos	Identificar que riesgos pueden afectar el proyecto y documentar sus características.	Checklist de riesgos	Sponsor Archivos históricos de proyectos
Análisis Cualitativo de Riesgos	Evaluar probabilidad e impacto. Establecer ranking de importancia.		Sponsor
Planificación de Respuesta a los Riesgos	Definir respuesta a riesgos. Planificar ejecución de respuestas.		Sponsor Archivos históricos de proyectos.
Seguimiento y Control del Riesgos	Verificar la ocurrencia de riesgos. Supervisar y verificar la ejecución de respuestas. Verificar aparición de nuevos riesgos.		Sponsor

PROCESO	ROLES	RESPONSABILIDADES
Planificación de Gestión de los Riesgos	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones
Identificación de Riesgos	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones
Análisis Cualitativo de Riesgos	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones
Análisis Cuantitativo de Riesgos	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones
Planificación de Respuesta a los Riesgos	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones
Seguimiento y Control del Riesgos.	Equipo de G. Riesgos Líder Apoyo Miembros	Dirigir actividad, responsable directo Proveer definiciones

PERIODICIDAD DE LA GESTIÓN DE RIESGOS			
Proceso	Momento De Ejecución	Entregable EDT	Periodicidad De Ejecución
Planificación de Gestión de los Riesgos	Al inicio del proyecto.	Plan del Proyecto	Una vez
Identificación de Riesgos	Al inicio del proyecto. En cada reunión del equipo del proyecto.	Plan del Proyecto Reunión de Coordinación Quincenal	Dos veces Quincenal
Análisis Cualitativo de Riesgos	Al inicio del proyecto. En cada reunión del equipo del proyecto	Plan del Proyecto Reunión de Coordinación Quincenal	Dos veces Quincenal
Análisis Cuantitativo de Riesgos	Al inicio del proyecto. En cada reunión del equipo del proyecto	Plan del Proyecto Reunión de Coordinación Quincenal	Dos veces Quincenal
Planificación de Respuesta a los Riesgos	Al inicio del proyecto. En cada reunión del equipo del proyecto	Plan del Proyecto Reunión de Coordinación Quincenal	Dos veces Quincenal
Seguimiento y Control del Riesgos	En cada fase del proyecto.	Reunión de Coordinación Quincenal	Dos veces Quincenal

FORMATOS DE LA GESTIÓN DE RIESGOS	
Planificación de Gestión de los Riesgos	Plan de Gestión de Riesgos
Identificación de Riesgos	Identificación y Evaluación Cualitativa de Riesgos
Análisis Cualitativo de Riesgos	Identificación y Evaluación Cualitativa de Riesgos
Análisis Cuantitativo de Riesgos	Identificación y Evaluación Cualitativa de Riesgos
Planificación de Respuesta a los Riesgos	Plan de Respuesta a Riesgos
Seguimiento y Control del Riesgos	Informe de Monitoreo de Riesgos Solicitud de Cambio Acción Correctiva

Identificación de riesgos potenciales

Para poder identificar y mejorar la gestión de riesgos los vamos a clasificar de acuerdo a la siguiente división y el ámbito donde podría surgir.

- **Riesgo por tamaño del producto (PS)**
 - Mala estimación del tamaño (tiempo mal estimada)
- **Impacto en el negocio (BI)**
 - Precios imperantes en el mercado del software.
 - Aceptación del costo del proyecto.
- **Riesgo por características del cliente: (CC)**
 - Comunicación oportuna con el cliente.
- **Definición del proceso (PD)**
 - ~
- **Entorno de desarrollo (DE)**
 - Poco tiempo para aprender el uso de las herramientas.
- **Cuestiones Técnicas (TI)**
 - ~
- **Tecnología a construir (TB)**
 - Mala adaptación de las bases de datos.
- **Tamaño en la experiencia del equipo de trabajo (SS)**
 - Poca experiencia en uso de Frameworks.

Clasificación de los riesgos por impacto

1. Catastrófico
2. Critico
3. Marginal
4. Despreciable

De acuerdo a la clasificación anterior veremos en la siguiente tabla, ya con una clasificación formal los riesgos identificados, para esto se han utilizado los formatos ampliamente recomendados por los ingenieros en sistemas.

Riesgos	Categoría	Probabilidad	Impacto	RSGR*
Fecha de entrega muy ajustada.	PS	90%	2	1
Precios imperantes en el mercado del software.	BI	30%	2	~
Rechazo del costo del proyecto.	BI	30%	2	~
Comunicación oportuna con el cliente.	CC	30%	2	~
Poco tiempo para aprender uso de las herramientas.	DE	40%	1	2
Mala adaptación de las bases de datos.	TB	40%	2	3
Poca experiencia en uso de Frameworks.	SS	50%	1	4

*** RSGR: Reducción, Supervisión y Gestión de Riesgo**

A los riesgos que cuenten con una probabilidad mayor a 30% se les dará seguimiento a través del plan de Reducción, Supervisión y Gestión de Riesgo (RSGR), sin implicaciones de falta de significancia con los riesgos con porcentaje inferior a este umbral.

Plan de Reducción, Supervisión y Gestión de Riesgo (RSGR)

En este plan, como fue mencionado en el párrafo anterior, se observará los riesgos identificados para crear planes de contingencia. Para lo cual se necesita evaluar cada ítem listado y posteriormente se detallaran los aspectos necesarios para minimizar el impacto del riesgo. Dicha evaluación y especificación se puede apreciar en la siguiente tabla.

No.	Riesgos	Probabilidad	Contingencia a seguir	Supervisión	
				Inicio	Fin
1	Fecha de entrega muy ajustada.	90%	Dentro de la planificación se tienen contempladas fechas límites y de holgura para la entrega final.	Desde inicio	
2	Poco tiempo para aprender uso de las herramientas.	40%	Se cuenta con documentación suficiente en línea para poder aplicar durante el desarrollo.		

3	Mala adaptación de las bases de datos.	40%	Teniendo en cuenta que se trabaja con un proyecto base ya en producción, la adaptación debe realizarse con extremo detalle, preferiblemente utilizando ORM's con acceso directo al DBMS.	Desde inicio	
4	Poca experiencia en uso de Frameworks.	50%	Diversidad de documentación y experiencia personal se debe proveer a los nuevos miembros en caso de expansión.	Desde inicio	

Conclusión

La gestión de riesgos, es una herramienta muy útil para el éxito del proceso de software, con esto se garantiza tener el producto con calidad, en tiempo y de acuerdo a los requerimientos del cliente, identificando los riesgos potenciales que puedan afectar los aspectos mencionados y poderlos controlar de forma proactiva, es decir anticipándose a.

Si se crean planes de contingencia y se supervisa el riesgo en todo el proceso del software se garantiza calidad y cumplimiento del tiempo establecido de entrega.

Plan de Recuperación de Desastres

Objetivos del plan de recuperación de desastres

- Estar preparados ante cualquier desastre, creando planes para tomar acción cuando el evento se presente, protegiendo el hardware y software.
- Garantizar la puesta en producción del sistema en el menor tiempo posible.
- Asegurar y no poner en riesgo el negocio, el cual soporta el software en desarrollo.

Definición del plan

Basándonos en los objetivos definidos del presente plan, se enfocará básicamente en dos ejes, el primero se concentrará en asegurar el software y los datos, mientras que el segundo lo hará por el equipo de hardware. Se establece en ese orden estratégicamente dado que actualmente es más valioso el patrimonio intangible de la organización que el equipo físico, es por esto que se debe tomar como principal objetivo la información.

Alcance

El plan tendrá un alcance para la protección de los datos, hardware y software, a través de procedimientos y planes de contingencia para la pronta recuperación del sistema para que las operaciones que se realicen en el mismo no se vean afectadas y por ende afecte los objetivos empresariales.

Protección de Datos y Software

El software y los datos deben de protegerse prioritariamente debido a que son los pilares más importantes actualmente en los sistemas, así como el hardware, también debe de tenerse el cuidado de proteger el software y los datos realizando backups, con periodicidades establecidas de acuerdo al número de transacciones diarias, esto con el fin de no tener inconsistencia de datos al retornar el sistema a su normalidad ante un desastre.

Protección de Hardware

El hardware debe protegerse tanto física como digitalmente, identificando los riesgos potenciales a los cuales se están acostumbrados, principalmente equipos que funcionen como servidor así como sus conexiones de red y equipos de red relacionados. Todo se debe estimar para poder tener respaldos de energía o de dispositivos (Discos Espejados), sin olvidar los equipos terminales, todo esto con la finalidad de reestablecer los servicios empresariales en caso de sucesos imprevistos.

De igual forma se ha identificado que pueden existir temporadas críticas para el negocio, lo que puede implicar repercusiones potenciales dentro de las actividades normales de la organización (tiempos de alto movimiento), por lo que deben considerarse dentro del análisis de criticidad del sistema. En las siguientes tablas se puede observar lo antes mencionado.

Criticidad por servicios

No.	Servicio	Criticidad	Período Crítico	Procedimientos Alternativos	Tiempo de fuera de servicio	Plataforma	Usuarios afectados
1	Servidor de BD	Alta		Almacenar los pedidos en un archivo local y cargarlos cuando el sistema se ha restablecido (catch-up)	1 día	MySQL	Todos
2	Servidor HTTP	Alta		Almacenar los pedidos en un archivo local y cargarlos cuando el sistema se ha restablecido (catch-up)	1 día	Apache	Todos
3	Servidor PHP	Alta		Almacenar los pedidos en un archivo local y cargarlos cuando el sistema se ha restablecido (catch-up)	1 día	PHP	Todos
4	Servidor REST	Alta		Almacenar los pedidos en un archivo local y cargarlos cuando el sistema se ha restablecido (catch-up)	1 día	Apache	API

Informe de riesgos

Luego del análisis del cuadro anterior presentamos el informe de riesgos identificados de acuerdo a la tabla que continúa, en este caso se pondera la probabilidad de ocurrencia del riesgo y su impacto para el sistema y los objetivos de la empresa.

No.	Amenaza	Riesgo	Criticidad	Probabilidad de ocurrencia	Impacto
1	Falla de sistemas de energía.	Apagones y fallos en equipos de cómputo y comunicaciones.	Media	0.4	Equipos fuera de servicio, daño físico en equipos provocados por la variabilidad de servicio eléctrico.
2	Descargas electro atmosféricas.	Daño total en equipos.	Alta	0.1	Daños irreparables en equipos de cómputo y de comunicaciones.
3	Cables de red expuestos.	Desconexión del sistema	Media	0.2	Indisponibilidad del sistema en uno o todos sus componentes
4	Equipos expuestos.	Apagado, desconfiguración o daño.	Alta	0.2	Indisponibilidad del sistema, daño de equipos
5	Falla de Disco Duro.	Perdida de datos, software.	Alta	0.3	Indisponibilidad del sistema debido a fallas catastrófica

Estrategias de recuperación ante desastres

Si el plan de prevención fallas y se está ya en un desastre se debe actuar, en el plan de recuperación se adoptan guías y estrategias para recuperar el sistema de acuerdo a como se presenten los desastres ya sean estos de tipo natural, provocados o debido a elementos externos no naturales.

Las estrategias a seguir en el caso de la recuperación de equipos de cómputo es comprar de forma inmediata las partes dañadas y/o la sustitución completa de equipo, una alternativa es que debido al bajo costo de los equipos utilizados como servidor es la adición y la configuración de un servidor B, el cual estará disponible para activación y luego de su activación se debe de restaurar las BD, con el ultimo backup hecho, mientras se recupera el sistema se debe de almacenar en cada terminal las transacciones mediante un cache de datos, los cuales deben de descargarse al momento de que el sistema se estabilice.

Dentro del plan se deben de prever y programar simulacros de desastres esto con el fin que el personal que debe estar involucrado ante un desastre tenga la experiencia y el conocimiento necesario, ya que dé, idealmente, no darse una emergencia en un tiempo prolongado, se corre el riesgo que cuando sea necesario el personal ya ha perdido la práctica y el tiempo de recuperación puede prolongarse más del necesario.

Plan de Gestión de Comunicaciones

Nombre del Proyecto	Siglas del Proyecto
CsP Private API	CSPAPI

Comunicaciones del Proyecto: Especificar la Matriz de las Comunicaciones del Proyecto

Ver Matriz de Comunicaciones del Proyecto

Procedimiento para Tratar Polémicas: Defina el procedimiento para procesar y resolver las polémicas, especificando la forma de capturarlas y registrarlas, el modo en que se abordará su tratamiento y resolución, la forma de controlarlas y hacerles seguimiento, y el método de escalamiento en caso de no poder resolverlas.

1. Se captan las polémicas que puedan surgir en el sistema a través de la observación y conversación, o de alguna persona o grupo que los exprese formalmente.
2. Se codifican y registran las polémicas en el Log de Control de Polémicas que a continuación se da a conocer.

LOG DE CONTROL DE POLÉMICAS

Código de Polémicas	Descripción	Involucrados	Enfoque Solución	Acciones de Solución	Responsable	Fecha	Resultado Obtenido

3. Se revisa el Log de Control de Polémicas en una reunión programada que tenga como fin lo siguiente:
 - a. Determinar las soluciones a aplicar a las polémicas pendientes por analizar, designando un responsable por su solución, un plazo de solución, y registrar la programación de estas soluciones en el Log de Control.
 - b. Revisar si las soluciones programadas se están aplicando, de no ser así se tomarán acciones correctivas al respecto.
 - c. Revisar si las soluciones aplicadas han sido efectivas y si la polémica ha sido resuelta, de no ser así se diseñarán nuevas soluciones.
4. En caso que una polémica no pueda ser resuelta o en caso que haya evolucionado hasta convertirse en un problema, deberá ser abordada con el siguiente método de escalamiento:
 - a. En primera instancia será tratada de resolver por el Director del proyecto y el Equipo de Gestión de Proyecto, utilizando el método estándar de resolución de problemas.
 - b. En segunda instancia será tratada de resolver por el Director del proyecto, el Equipo de Gestión de Proyecto, y los miembros pertinentes del Equipo de Proyecto, utilizando el método estándar de resolución de problemas.
 - c. En tercera instancia será tratada de resolver por el Sponsor, el Director del proyecto, y los miembros pertinentes del proyecto, utilizando la negociación y/o la solución de conflictos.
 - d. En última instancia será resuelta por el Sponsor o por el Sponsor y el Comité de Control de Cambios si el primero lo cree conveniente y necesario.

Procedimiento para Actualizar el Plan de Gestión de Comunicaciones: Defina el procedimiento para revisar y actualizar el plan de gestión de comunicaciones.

El Plan de Gestión de las Comunicaciones deberá ser revisado y/o actualizado cada vez que:

1. Hay una solicitud de cambio aprobada que impacte el Plan de Proyecto.
2. Hay una acción correctiva que impacte los requerimientos o necesidades de información de los Stakeholders.
3. Hay personas que ingresan o salen del proyecto.
4. Hay cambios en las asignaciones de personas a roles del proyecto.
5. Hay quejas, sugerencias, comentarios o evidencias de requerimientos de información no satisfechos.
6. Hay evidencias de resistencia al cambio.

La actualización del Plan de Gestión de las Comunicaciones deberá seguir los siguientes pasos:

1. Identificación y clasificación de Stakeholders.
2. Determinación de requerimientos de información.
3. Actualización del Plan de Gestión de las Comunicaciones.
4. Aprobación del Plan de Gestión de las Comunicaciones.
5. Difusión del nuevo Plan de Gestión de las Comunicaciones.

Guía para Eventos de Comunicación: Defina guía para reuniones y conferencias.

Guías para reuniones. Todas las reuniones deberán seguir las siguientes pautas:

1. Debe fijarse la agenda con anterioridad.
2. Debe coordinarse e informarse fecha, hora, y lugar con los interesados.
3. Se debe empezar puntual.
4. Se deben fijar los objetivos de la reunión, los roles, los procesos grupales de trabajo, y los métodos de solución de controversias.
5. Se debe cumplir a cabalidad los roles de facilitador (dirige el proceso grupal de trabajo) y de anotador (toma nota de los resultados formales de la reunión).
6. Se debe terminar puntual.

Matriz de comunicaciones

Nombre del Proyecto	Siglas del Proyecto
CsP Private API	CSPAPI

Información	Contenido	Nivel de Detalle	Responsable	Grupo Receptor	Metodología	Frecuencia
Iniciación del Proyecto	Datos y comunicación sobre la iniciación del proyecto.	Medio	PROJECT MANAGER	Sponsor	Personal	Una sola vez
Iniciación del Proyecto	Datos Preliminares sobre el alcance del proyecto.	Alto	PROJECT MANAGER	Sponsor	Personal	Una sola vez
Planificación del Proyecto	Planificación detallada del Proyecto: <ul style="list-style-type: none"> • Alcance • Tiempo • Costo • Calidad • Riesgo 	Muy Alto	PROJECT MANAGER	Sponsor	Personal	Una sola vez
Estado del Proyecto	Estado Actual, progreso, pronóstico de tiempo y costo, problemas.	Alto	PROJECT MANAGER	Sponsor	Personal	Quincenal
Coordinación del Proyecto	Información detallada de las reuniones de coordinación.	Alto	PROJECT MANAGER	Sponsor	Personal	Quincenal
Cierre del Proyecto	Datos y Comunicación sobre el cierre del proyecto.	Medio	PROJECT MANAGER	Sponsor	Personal	Una sola vez
Informe Final del Proyecto	Objetivos, análisis y resultados, conclusiones, sugerencias, y/o recomendaciones.	Alto	PROJECT MANAGER	Sponsor	Documento Impreso	Una sola vez

Factores considerados para el análisis y diseño

Como se ha mencionado anteriormente, el servidor REST se desarrolla utilizando el Framework Symfony en su versión 2.5, esto quiere decir que físicamente el servidor REST será instalado en el mismo servidor que el de aplicaciones, dado que compartirán los componentes del Framework durante su implementación y utilización.

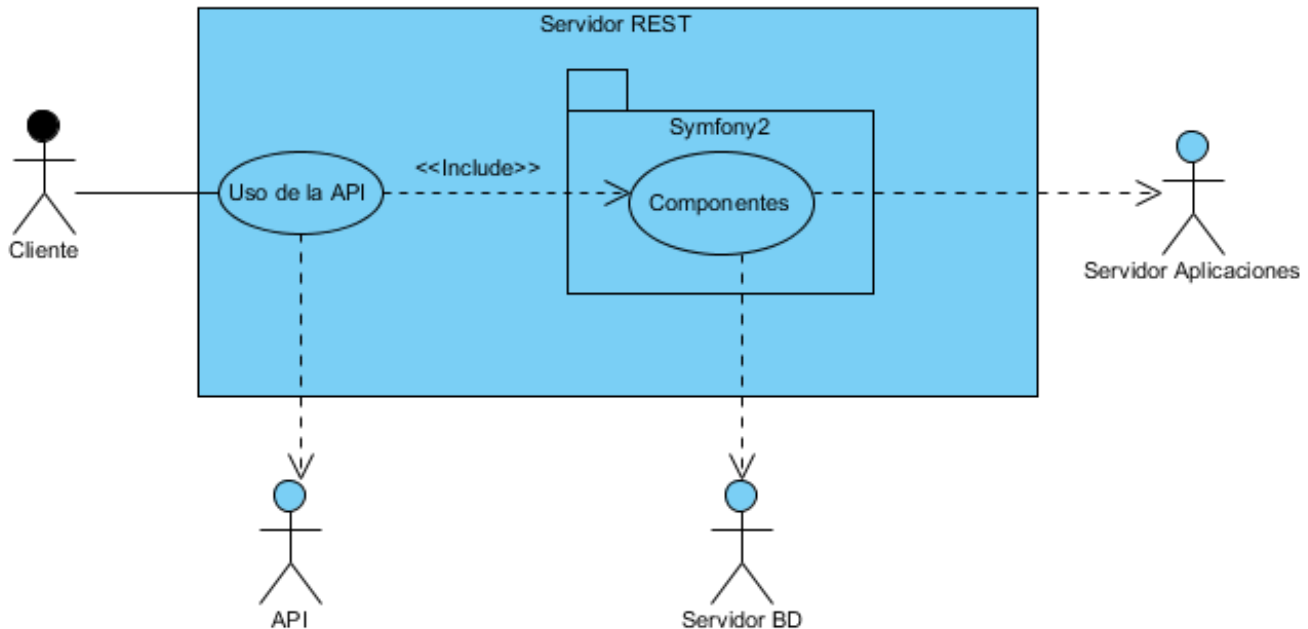
En base a estos detalles se decidió realizar análisis y diseño de cada una de las partes involucradas en el desarrollo total del sistema. Con esto se resuelven dudas en cuanto a porqué se realizó un análisis general y ahora un análisis específico. Cabe aclarar también que ésta es la principal razón por la cual cada análisis y cada diseño son diferentes, ya sea en su tamaño, idea o elementos UML que se presentan.

En específico se analizarán los componentes esenciales del sistema, los cuales se dividirán en tres partes: Servidor REST, API RESTful y Cliente API.

PARTE I: SERVIDOR REST

Análisis

Diagrama de Casos de Uso



Diseño

Diagrama de Componentes

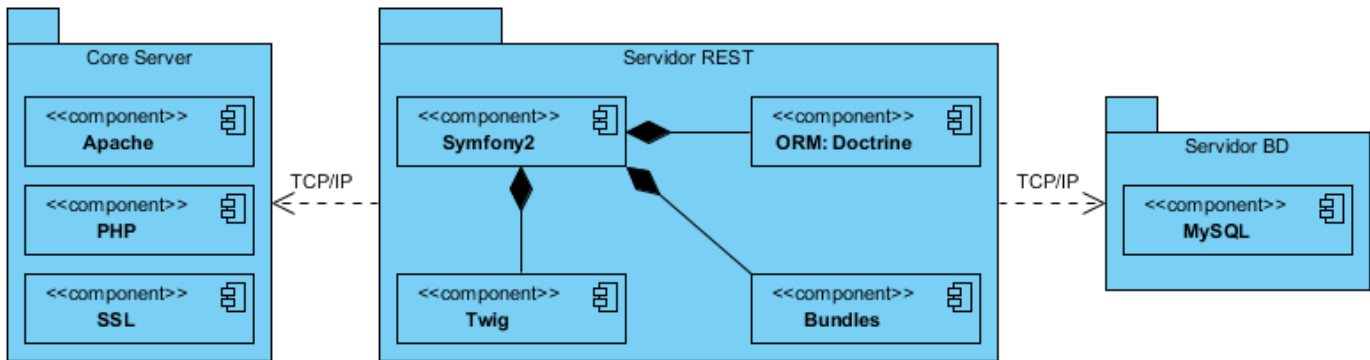
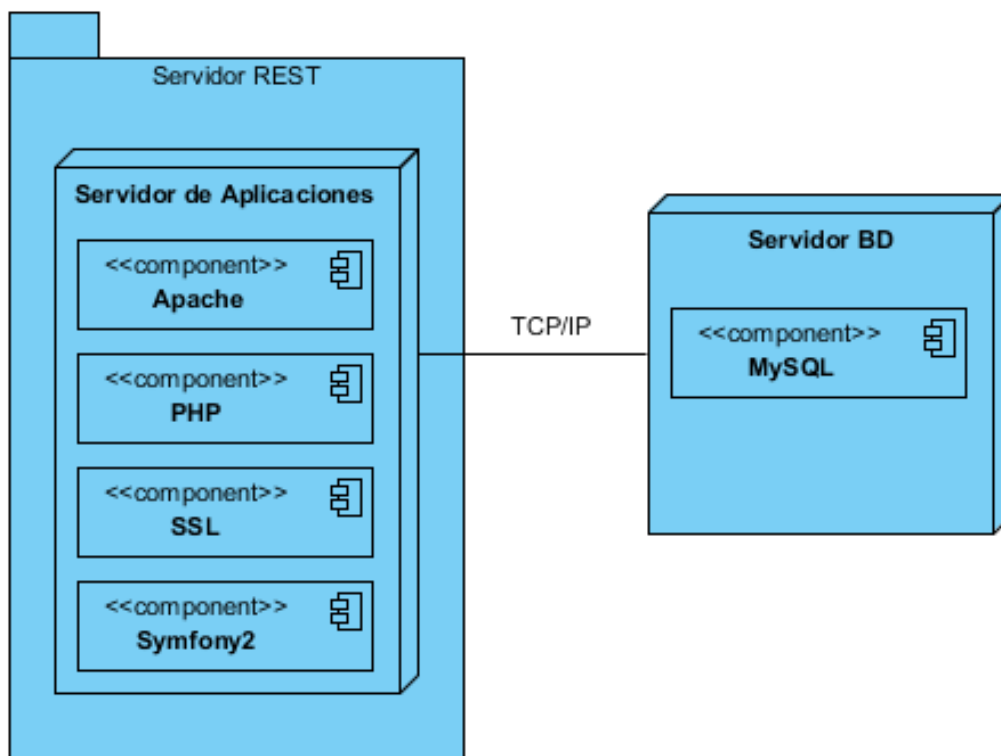


Diagrama de Despliegue



Adaptación a Base de Datos

En este punto se recalca el hecho de que el sistema a implementar para proveer los servicios, se basa en un proyecto base ya en producción, por lo que la base de datos fue diseñada con anterioridad, razón por la cual únicamente se importaron las tablas existentes y se crearon las entidades respectivas en la capa de modelo de negocios dentro del framework.

Estas entidades se pueden observar como clases o tablas en los respectivos diagramas de Clases o Entidad-Relación que se ubican, ya sea en el repositorio o en la sección Diseño General del Proyecto (pág. 21 y 26).

Codificación

Script de base de datos

Ver Anexo 2.

Verificación y pruebas

Informe de pruebas

Al momento de realizar las pruebas de control, se logró identificar que el servidor REST no presenta ningún inconveniente para coexistir dentro del mismo dispositivo físico que el proyecto base, así mismo se creó una conexión estable entre el Servidor REST y el Servidor de Base de Datos.

Producción

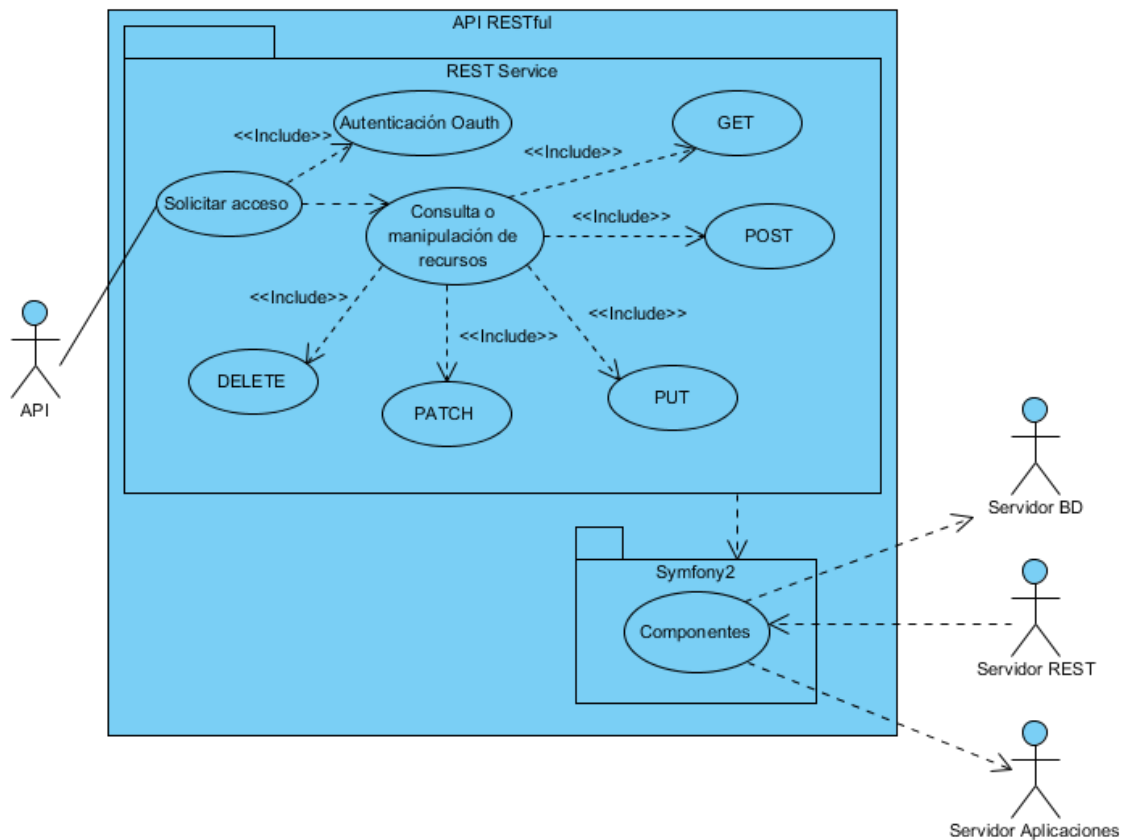
El termino producción o más bien “lanzar a producción” se conoce en el gremio como el hecho de implementar el producto o servicio que se desarrolla para uso de los, valga la redundancia, usuarios del sistema. Esto puede ser con diferentes fines, pero siempre es pasado el punto de corrección de errores, ya que puede ser catastrófico llevar un servicio a producción sin estar un 95% libre de probables filtraciones por agentes de dudosa procedencia.

Sabiendo que el 95% recomendado, es a causa de que nunca se puede llegar al 100% de seguridad en un sistema (sobre todo online), se toman las precauciones necesarias para el presente proyecto y se decide esperar a llevar a producción el servidor hasta tener implementado correctamente el servicio web. De este modo se realizaran más pruebas ya con el servicio en funcionamiento localmente para corregir leaks o exploits que puedan encontrarse y así minimizar los riesgos en un futuro.

PARTE II: API RESTful

Análisis

Diagrama de Casos de Uso



Descripción de Casos de Uso

Con la implementación del presente proyecto se han debido realizar diversos tipos de diagramas, y varios de ellos han sido de casos de uso, pero ninguno de los anteriores o posteriores ha recibido o tendrá una descripción, no porque no se puedan o deban hacer, sino porque se consideran auto-explicativos, sin embargo, el actual diagrama para lo que es la API RESTful, se considera de carácter crítico puesto que en él se dan a conocer los flujos a seguir para la correcta implementación del proyecto en general.

CU-01: Solicitar acceso

Nombre:	CU-01: Solicitar acceso
Autor:	Victor Cifuentes Méndez
Fecha:	Julio de 2,014
Descripción:	Proceso mediante el cual la API se conecta al Servicio REST.
Actores:	API, Servidor REST.
Pre condiciones:	El cliente ha accedido correctamente a la API.
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor ingresa a la fachada de la API. 2. El servicio despliega la interfaz de autenticación. 3. El servicio reconoce al cliente que se ha autenticado en la API. 4. El sistema despliega la información solicitada.
Flujo Alternativo:	<ol style="list-style-type: none"> 5. El servicio no reconoce al cliente que se ha autenticado en la API. 6. El servicio despliega la interfaz de autenticación con los errores encontrados.
Post condiciones:	Se cuenta con los diferentes Tokens según el cliente de la API.

CU-02: Autenticación Oauth

Nombre:	CU-02: Autenticación Oauth
Autor:	Victor Cifuentes Méndez
Fecha:	Julio de 2,014
Descripción:	Proceso mediante el cual la API autentica al usuario del cliente contra el Servicio REST.
Actores:	API, Servidor REST.
Pre condiciones:	La API se ha conectado correctamente con el servicio REST.
Flujo Normal:	<ol style="list-style-type: none"> 1. La API envía los datos del usuario cliente a autenticar en el servidor REST. 2. El servicio concede acceso para autenticar. 3. La API solicita Token enviando datos del Cliente. 4. El servicio devuelve Access Token, Refresh Token y Auth Code.
Flujo Alternativo:	<ol style="list-style-type: none"> 5. El servicio no reconoce los datos del Cliente. 6. El servicio despliega la interfaz de autenticación con los errores encontrados.
Post condiciones:	Se cuenta con los diferentes Tokens según el cliente de la API.

CU-03: Consulta o manipulación de recursos

Nombre:	CU-03: Consulta o manipulación de recursos
Autor:	Victor Cifuentes Méndez
Fecha:	Julio de 2,014
Descripción:	Proceso mediante el cual la API debidamente autenticada en el Servidor REST realiza diversas operaciones consumiendo los servicios RESTful.
Actores:	API, Servidor REST, Servidor de Aplicaciones, Servidor de BD.
Pre condiciones:	La API se ha autenticado correctamente con el servicio REST. Se tiene acceso a los diferentes Tokens del Cliente.
Flujo Normal:	<ol style="list-style-type: none"> 1. Con el Access Token la API realiza peticiones o request al servicio. 2. El servicio transforma los requests para solicitar los datos al Servidor de BD. 3. El Servidor de BD realiza las transacciones necesarias y devuelve los datos requeridos al servicio. 4. El servicio entrega los responses a la API.
Flujo Alternativo:	<ol style="list-style-type: none"> 5. El Access Token expira. 6. La API solicita nuevos Tokens. 7. El servicio autentica nuevamente o realiza un refresh. 8. El servicio devuelve nuevos tokens.
Post condiciones:	Los datos se han almacenado en BD.

CU-04/05/06/07/08: Get, Post, Put, Patch, Delete

Nombre:	CU-04/05/06/07/08: Get, Post, Put, Patch, Delete
Autor:	Victor Cifuentes Méndez
Fecha:	Julio de 2,014
Descripción:	Métodos a través de los cuales la API debidamente autenticada en el Servidor REST ejecuta las diversas operaciones para consumir los servicios REST y así crear el servicio RESTful.
Actores:	API, Servidor REST, Servidor de Aplicaciones, Servidor de BD.
Pre condiciones:	La API se ha autenticado correctamente con el servicio REST. Se tiene acceso a los diferentes Tokens del Cliente. Las URI's de la API deben apuntar correctamente al recurso deseado. El Request enviado por la API debe poseer las cabeceras correctas (GET, POST, PUT, PATCH, DELETE, etc.).
Flujo Normal:	<ol style="list-style-type: none"> 1. La API utiliza algún método HTTP para manipular los recursos del servicio. 2. El Servidor REST interpreta los verbos y los ejecuta. 3. El Servidor REST devuelve el código de respuesta del método HTTP. 4. La API procesa los datos recibidos.

Flujo Alternativo:**Post condiciones:**

Los datos se han almacenado en BD.

Diagrama de Secuencias

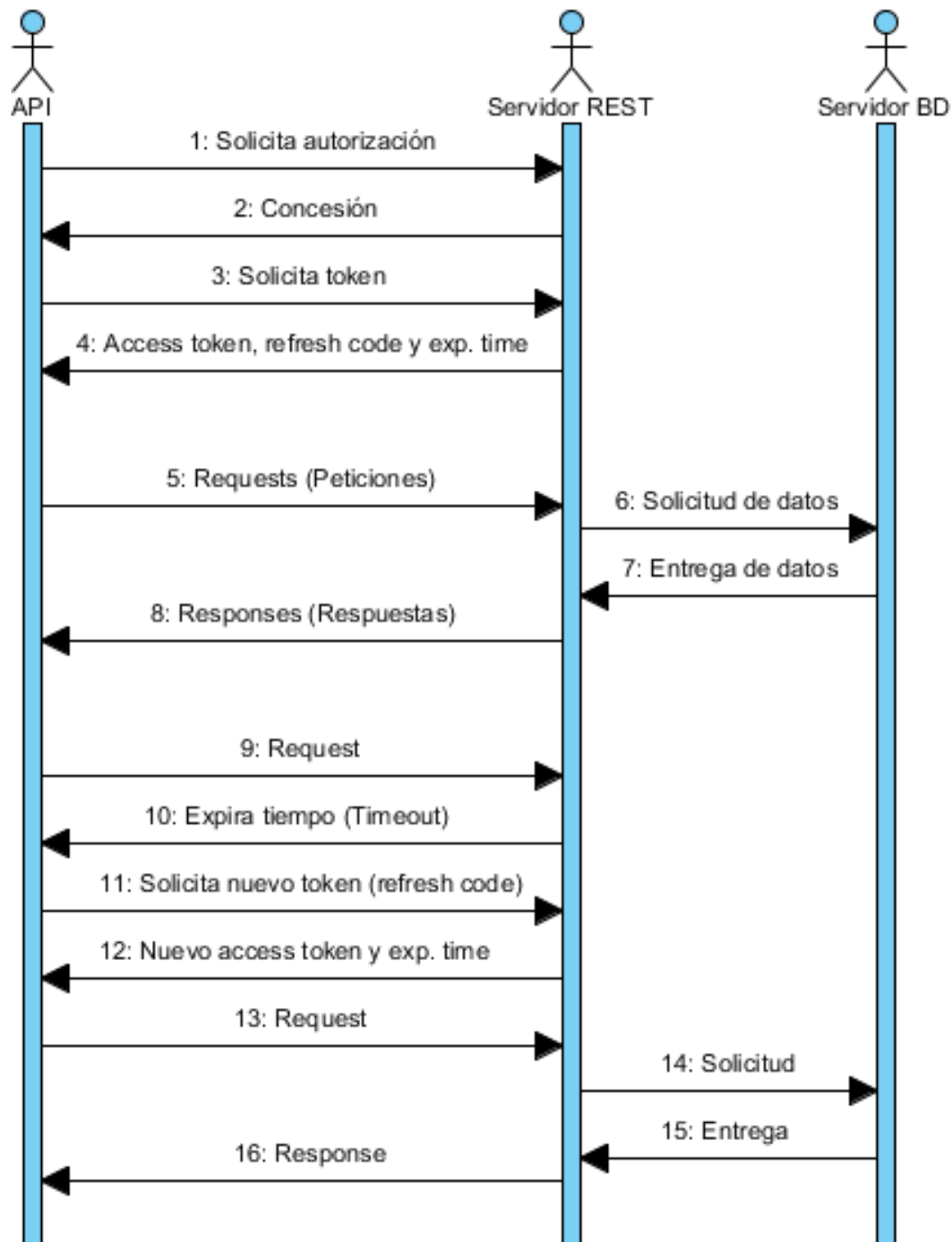
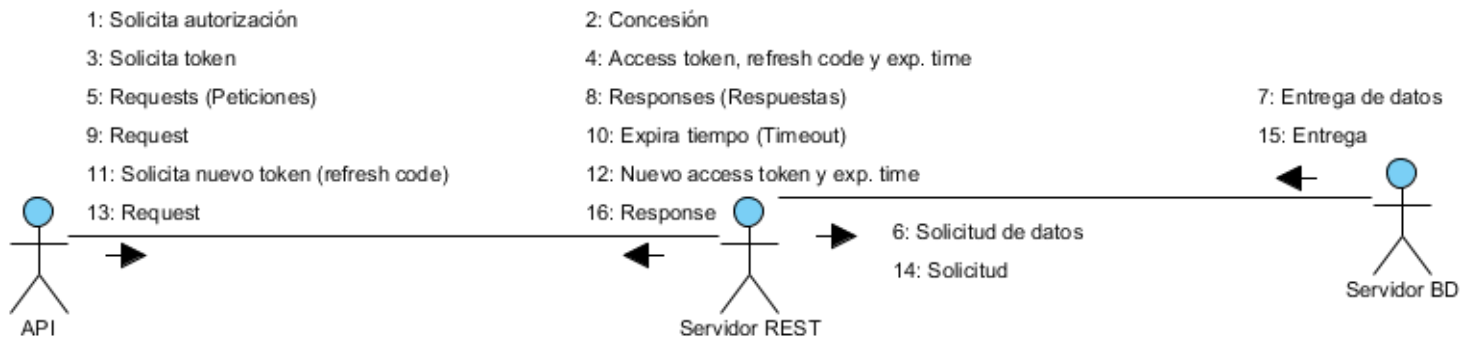


Diagrama de Comunicaciones



Diseño

Al igual que el servidor REST, este apartado cuenta con acceso al Servidor de Bases de Datos, por lo que se vio completamente involucrado en su desarrollo ya que algunos de sus componentes son los que se encargan directamente del tratamiento de los datos desde la API, pasando por las entidades en el modelo de negocios hasta el Gestor de Bases de Datos por medio de ORM. El diagrama Entidad-Relación y el diagrama de clases, así como el respectivo Diccionario de Datos pueden ser vistos para referencia en el apartado Diseño General del Proyecto (a partir de la pág. 21).

Diagrama de Componentes

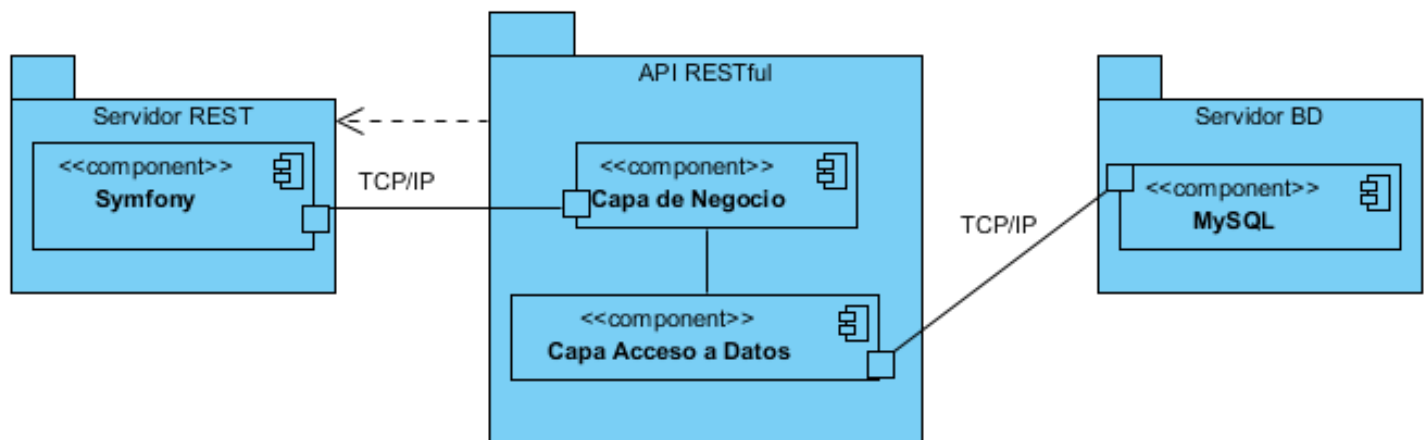
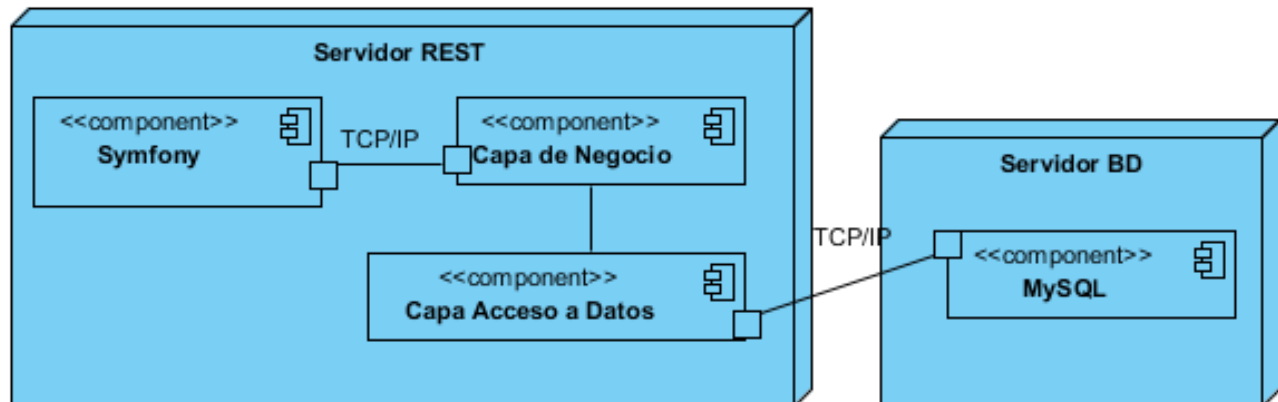


Diagrama de Despliegue



Codificación

Todo el código necesario para la implementación de la API RESTful se encuentra alojado en el repositorio de GitHub.

Pruebas de control

Consultas de prueba a base de datos

Para corroborar el funcionamiento de la API RESTful, es decir, del acceso a las entidades que componen la base de datos del Proyecto Base, ahora que se han creado los controladores que se encargan de redirigir todas las peticiones y de devolver los respectivos response, se realizaron diversos requests o solicitudes HTTP a las direcciones base de la API, siendo ésta:

`http://192.168.1.69/CSP_Private_API/web/api/{recurso}`

La URL base se encuentra en una red local con direcciones IP estáticas, siendo ésta la dirección del puerto Ethernet donde se localiza el servidor; {recurso} es cualquiera de las entidades que se tiene en el proyecto, a éstas peticiones se les agregaron encabezados GET, POST, PUT o DELETE para corroborar la correcta comunicación y transmisión, obteniendo respuestas como:

```
Status
200 OK Loading time: 910
Request headers
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/36.0.1985.143 Safari/537.36
Content-Type: text/plain; charset=utf-8
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-ES, es; q=0.8
Cookie: PHPSESSID=uiavk111f6b16sa3hocpahjn13
```

Response headers

Date: Sun, 17 Aug 2014 02:04:57 GMT

Server: Apache/2.4.7 (Win32) OpenSSL/1.0.1e PHP/5.5.6

X-Powered-By: PHP/5.5.6

Cache-Control: no-store, no-cache, must-revalidate, max-age=0

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: application/json

Content-Length: 106

Pragma: no-cache

Expires: Tue, 01 Jan 1971 02:00:00 GMT

En la respuesta se puede observar el código 200 OK que indica que la petición fue satisfactoria y entrega la siguiente respuesta en formato JSON:

```
[1]
-0: {
  id: 1
  nombre: "Victor"
  apellido: "Cifuentes"
  direccion: "San Marcos"
  email: "ozviccm@hotmail.com"
}
```

Informe de pruebas

Tras la revisión y prueba de cada recurso almacenado en el proyecto base y en cada uno de sus métodos HTTP necesarios para obtener un CRUD (Create, Retrieve, Update, Delete) funcional, se constató que los resultados son los deseados para este tipo de servicios, obteniendo respuestas rápidas y constantes del servidor.

Todo esto puede ser comprobado nuevamente en el Cliente API, el cual depende explícitamente del correcto funcionamiento de este servicio.

Documentación

La documentación para la presente API se encuentra alojada en el repositorio en línea de GitHub, esto con la finalidad de mantenerla lo más actualizada posible, razón por la cual no es incluida en el presente documento para no presentar información desactualizada y/o errónea.

La documentación puede ser encontrada en su versión más actual en la siguiente dirección:

https://github.com/Creyent/CSP_Private_API/blob/master/docs/api/api_doc.html

Producción

Al llegar a este punto se han implementado dos de las tres partes del sistema que se está desarrollando, lo que nos permite dar paso el desarrollo del Cliente API, el cual consumirá los servicios que ya se crearon hasta este punto.

PARTE III: CLIENTE API

Análisis

Diagrama de Casos de Uso

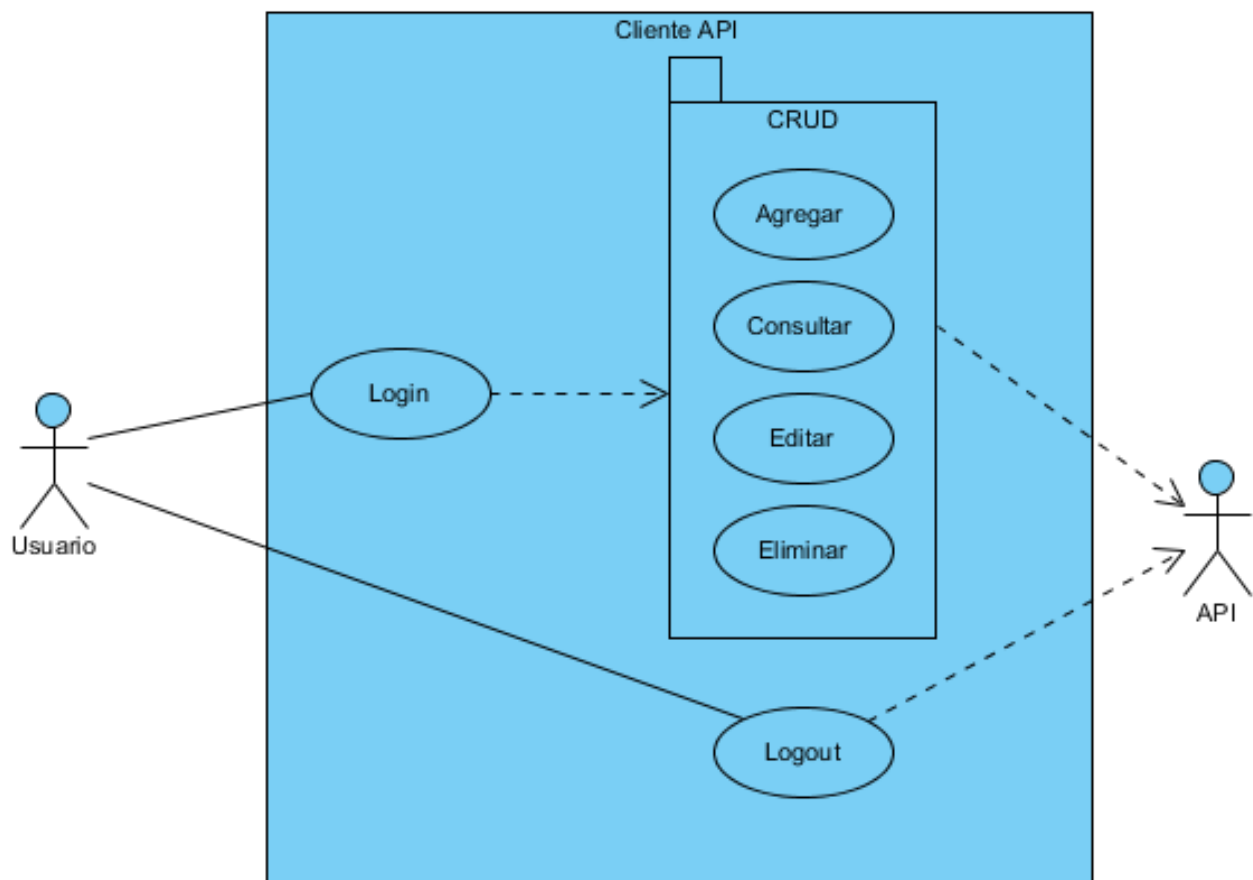


Diagrama de Secuencias

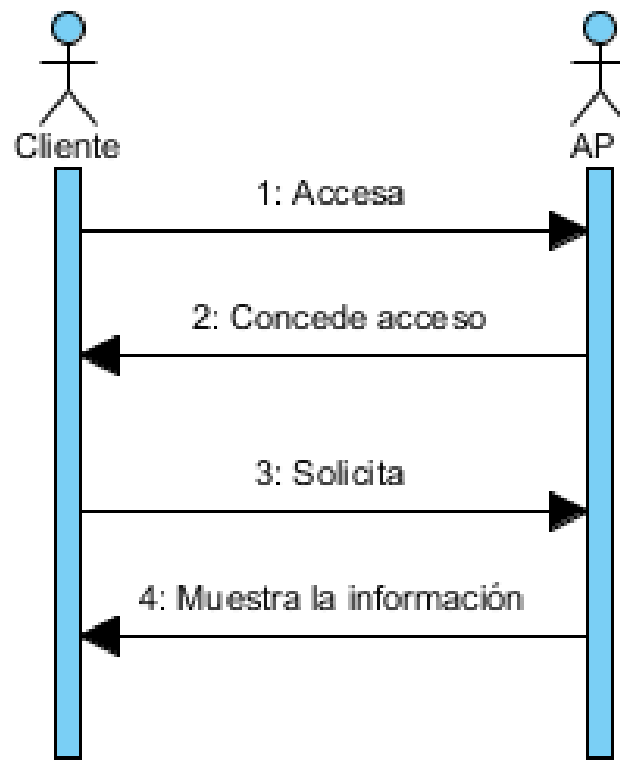
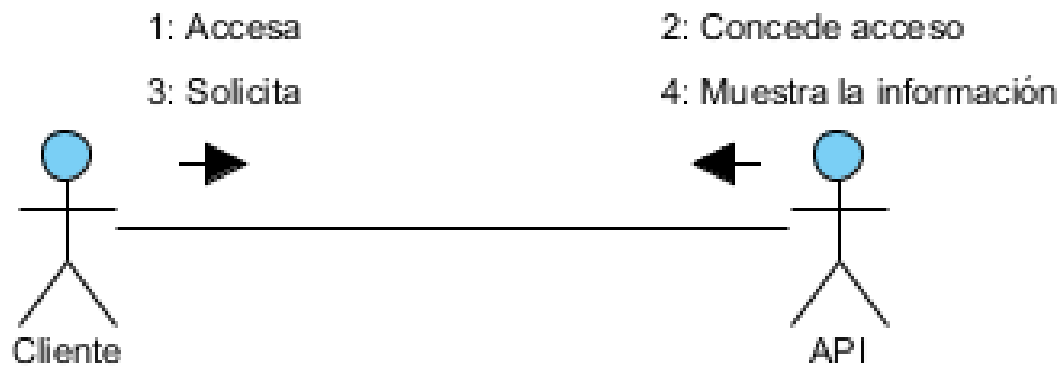


Diagrama de Comunicaciones



Diseño

Diagrama de Componentes

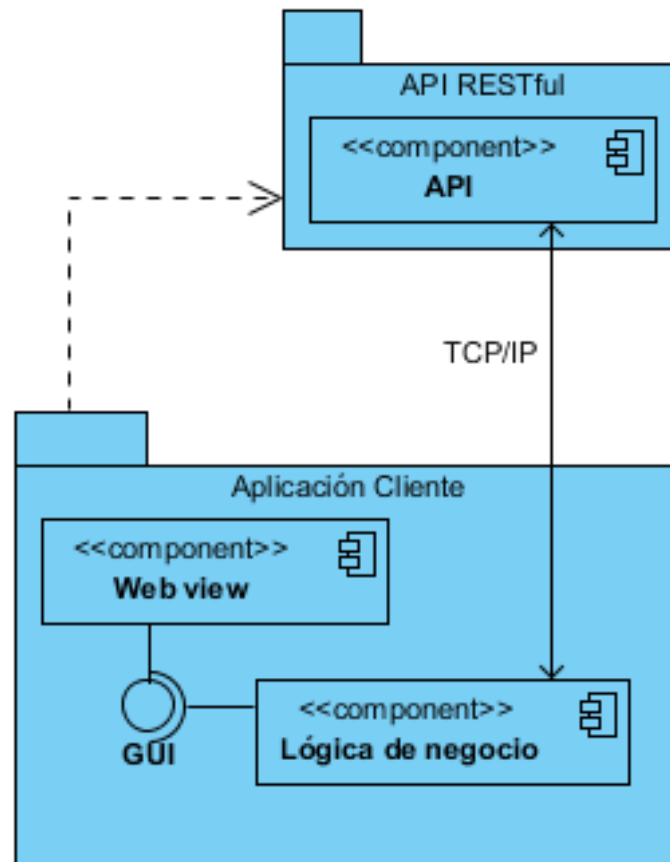
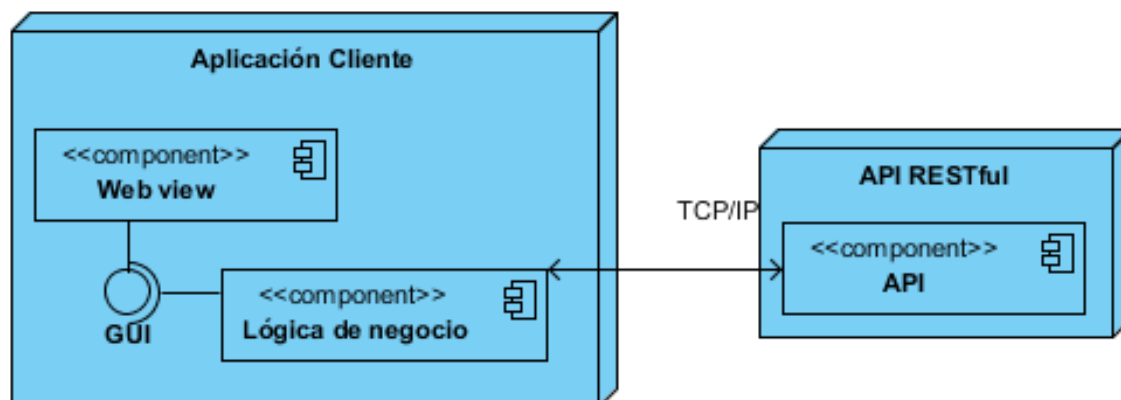


Diagrama de Despliegue

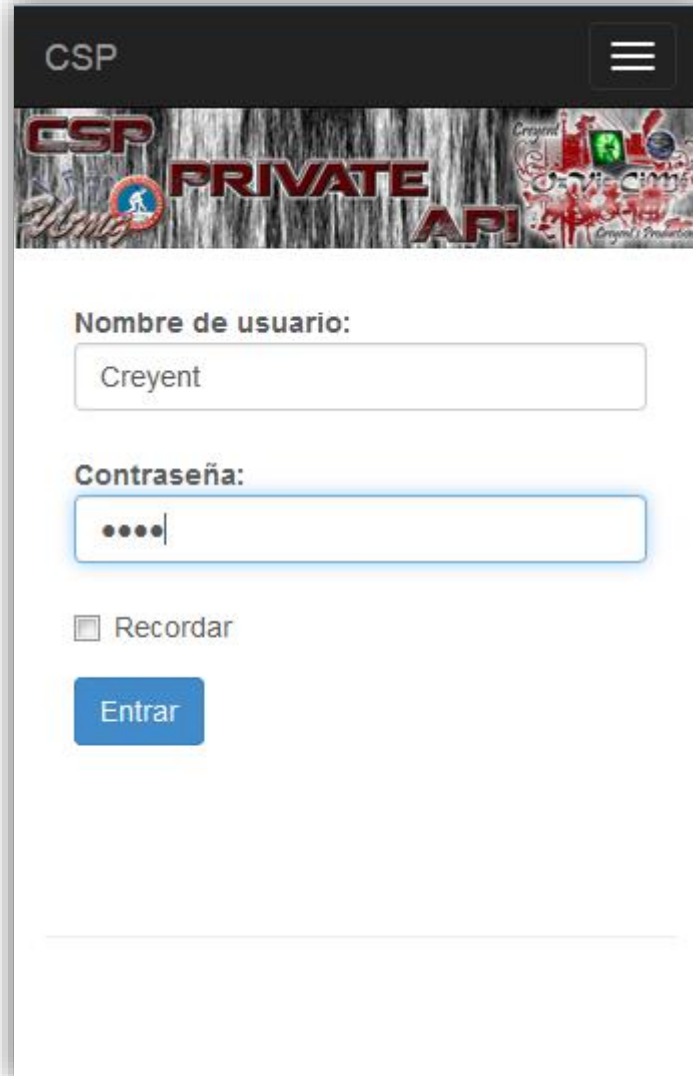


Codificación

Todo el código necesario para la implementación del Cliente API se encuentra alojado en el repositorio de GitHub.

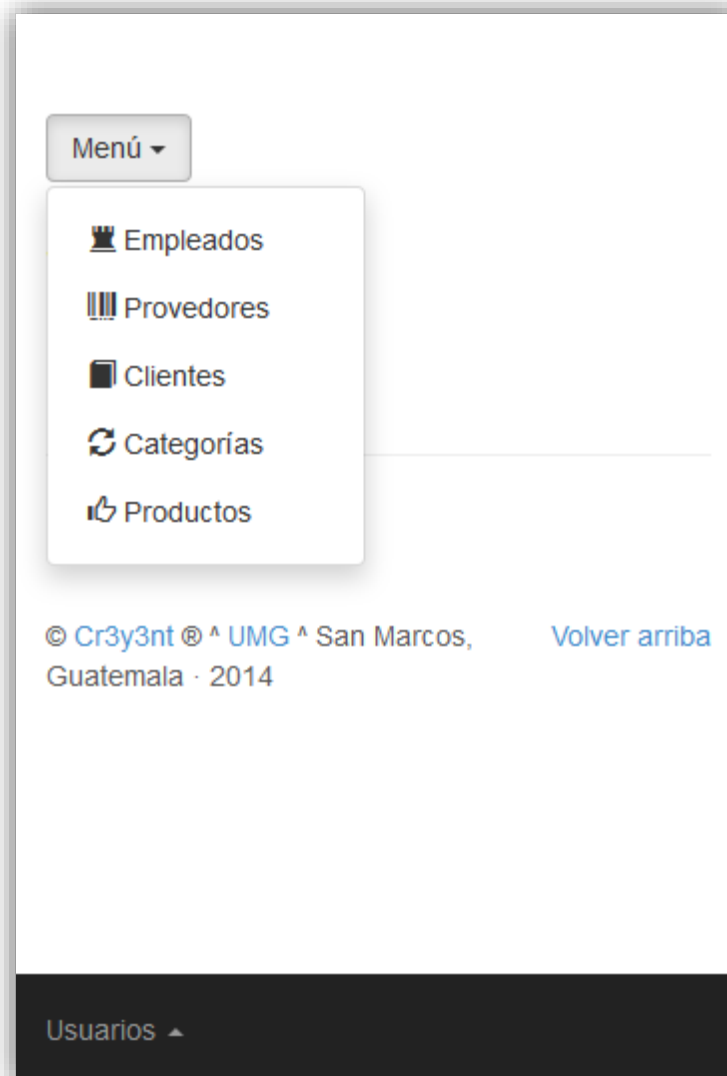
Vistas de la Interfaz de Usuario

Login



The screenshot shows a mobile application interface for 'CSP PRIVATE API'. At the top, there is a dark header with the text 'CSP' on the left and a hamburger menu icon on the right. Below the header is a banner image featuring the 'CSP PRIVATE API' logo and a 'Creyent's Productions' watermark. The main content area is white and contains the following elements: a label 'Nombre de usuario:' followed by a text input field containing the text 'Creyent'; a label 'Contraseña:' followed by a password input field with four dots and a cursor; a checkbox labeled 'Recordar'; and a blue button labeled 'Entrar'.

Vista Base

Menú de recursos

Acceso a Servicio RESTful

Menú ▾

Servicio REST

Buscar

Nombre	Apellido	Dirección	Email	
Victor	Cifuentes	San Marcos	ozviccm@hotmail.com	

© Cr3y3nt ® ^ UMG ^ San Marcos, Guatemala · 2014

[Volver arriba](#)

Usuarios ▲

Pruebas de control

Informe de pruebas

Recurriendo a los descrito anteriormente, tras la revisión y prueba de cada recurso almacenado en el proyecto base y en cada uno de sus métodos HTTP necesarios para obtener un CRUD (Create, Retrieve, Update, Delete) funcional, se constató que los resultados son los deseados para este tipo de servicios, obteniendo respuestas rápidas y constantes del servidor.

Se realizaron diversos requests o solicitudes HTTP a las direcciones base de la API, la cual fue detallada la pagina 46 de este documento, a éstas peticiones se les agregaron de la misma forma encabezados GET, POST, PUT o DELETE para corroborar la correcta comunicación y transmisión, obteniendo respuestas como las mencionadas en el mismo apartado.

Tras las diversas pruebas ya en el Cliente API, se presentan los siguientes resultados, obtenidos mediante diversas herramientas de desarrollo puesto que son desde el Cliente API y no desde mediante acceso directo al Servidor REST.

Consultando datos:

URL	Estado	Dominio	Tamaño	IP Remota
Panel de red activado. No se mostrarán solicitudes recibidas mientras el panel esté desactivado.				
+ GET proveedores	200 OK	192.168.1.69	90 B	192.168.1.69:80
+ GET productos	200 OK	192.168.1.69	297 B	192.168.1.69:80
+ GET categorias	200 OK	192.168.1.69	81 B	192.168.1.69:80
+ GET clientes	200 OK	192.168.1.69	117 B	192.168.1.69:80
+ GET empleados	200 OK	192.168.1.69	106 B	192.168.1.69:80
5 peticiones			691 B	

Agregando datos:

URL	Estado	Dominio	Tamaño	IP Remota	Línea de tiempo
+ GET productos	200 OK	192.168.1.69	297 B	192.168.1.69:80	
+ OPTIONS productos	200 OK	192.168.1.69	0 B	192.168.1.69:80	490ms
+ POST productos	201 Created	192.168.1.69	156 B	192.168.1.69:80	
Encabezados Post Respuesta JSON Caché					
<pre>{ "id":3, "codigo":"CsP-Prod-003", "nombre":"Tenis", "costo":150, "precio":175, "existencias":2, "idcategoria": :{"id":1, "nombre":"Ropa", "descripcion":"Vestimenta"}}</pre>					
3 peticiones			453 B		

Editando datos:

URL	Estado	Dominio	Tamaño	IP Remota	Línea de tiempo
+ GET categorias	200 OK	192.168.1.69	81 B	192.168.1.69:80	
+ GET 2	200 OK	192.168.1.69	27 B	192.168.1.69:80	
+ OPTIONS 2	200 OK	192.168.1.69	0 B	192.168.1.69:80	
- PUT 2	200 OK	192.168.1.69	74 B	192.168.1.69:80	
Encabezados Put Respuesta JSON Caché					
Ordenar alfabéticamente					
id 2					
nombre "Higiene"					
descripcion "Artículos de higiene personal."					
+ GET categorias	200 OK	192.168.1.69	128 B	192.168.1.69:80	
5 peticiones			310 B		

Borrando datos:

URL	Estado	Dominio	Tamaño	IP Remota
+ GET 3	200 OK	192.168.1.69	156 B	192.168.1.69:80
+ OPTIONS 3	200 OK	192.168.1.69	0 B	192.168.1.69:80
+ DELETE 3	204 No Content	192.168.1.69	0 B	192.168.1.69:80
3 peticiones			156 B	

En cada captura se puede observar que cada request genera un código específico que indica el estado de la misma, también se pueden verificar la forma en cómo se envían y se reciben los datos en la respuesta y el método utilizado.

Documentación

La documentación para la implementación completa del proyecto, es la que actualmente lee, sin embargo, se ha incluido también en el repositorio en línea de GitHub, del mismo modo se han incluido de forma ordenada todas y cada una de las imágenes, diagramas y capturas que se hayan mostrado en este documento.

La documentación puede ser encontrada en su versión más actual en la siguiente dirección:

https://github.com/Creyent/CSP_Private_API/blob/master/docs/

Producción

Junto al cierre del proyecto, se llegan a implementar los tres elementos que corresponden al sistema en cuestión, siendo éstos:

- ✓ Servidor REST: Servidor Apache y Framework Symfony2.
- ✓ API RESTful: Desarrollado en Symfony2.
- ✓ Cliente API: Frontend desarrollado en AngularJS y CodenameOne para plataforma Android.

CIERRE

Cierre del proyecto

Para el cierre del proyecto se procede a hacer entrega de este documento al Ingeniero Iván de León, Sponsor del proyecto, el cual procederá a hacer una revisión del mismo y en aceptación del informe, extenderá una nota de respuesta.

Esta actividad será durante una reunión programada, en la que al finalizar se espera cumplir con las expectativas de los interesados y se agradecerá por la oportunidad presentada en esta ocasión para realizar el presente proyecto.

ANEXOS

Diccionario de datos

<i>accesstoken</i>										
<i>Column name</i>	<i>DataType</i>	<i>PK</i>	<i>NN</i>	<i>UQ</i>	<i>BIN</i>	<i>UN</i>	<i>ZF</i>	<i>AI</i>	<i>Default</i>	<i>Comment</i>
<i>id</i>	INT(11)	✓	✓					✓		
<i>client_id</i>	INT(11)		✓							
<i>user_id</i>	INT(11)								NULL	
<i>token</i>	VARCHAR(255)		✓							
<i>expires_at</i>	INT(11)								NULL	
<i>scope</i>	VARCHAR(255)								NULL	
<i>authcode</i>										
<i>Column name</i>	<i>DataType</i>	<i>PK</i>	<i>NN</i>	<i>UQ</i>	<i>BIN</i>	<i>UN</i>	<i>ZF</i>	<i>AI</i>	<i>Default</i>	<i>Comment</i>
<i>id</i>	INT(11)	✓	✓					✓		
<i>client_id</i>	INT(11)		✓							
<i>user_id</i>	INT(11)								NULL	
<i>token</i>	VARCHAR(255)		✓							
<i>redirect_uri</i>	LONGTEXT		✓							
<i>expires_at</i>	INT(11)								NULL	
<i>scope</i>	VARCHAR(255)								NULL	
<i>categorias</i>										
<i>Column name</i>	<i>DataType</i>	<i>PK</i>	<i>NN</i>	<i>UQ</i>	<i>BIN</i>	<i>UN</i>	<i>ZF</i>	<i>AI</i>	<i>Default</i>	<i>Comment</i>
<i>id</i>	INT(11)	✓	✓					✓		
<i>Nombre</i>	VARCHAR(50)		✓							
<i>Descripcion</i>	VARCHAR(150)								NULL	

<i>client</i>										
<i>Column name</i>	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>random_id</i>	VARCHAR(255)		✓							
<i>redirect_uris</i>	LONGTEXT		✓							(DC2Type:array)
<i>secret</i>	VARCHAR(255)		✓							
<i>allowed_grant_types</i>	LONGTEXT		✓							(DC2Type:array)

<i>clientes</i>										
<i>Column name</i>	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>Nombre</i>	VARCHAR(50)		✓							
<i>Apellido</i>	VARCHAR(50)		✓							
<i>NIT</i>	VARCHAR(15)		✓							
<i>Direccion</i>	VARCHAR(75)		✓							
<i>Telefono</i>	VARCHAR(15)		✓							

<i>empleados</i>										
<i>Column name</i>	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>Nombre</i>	VARCHAR(50)		✓							
<i>Apellido</i>	VARCHAR(50)		✓							
<i>Direccion</i>	VARCHAR(70)		✓							
<i>Email</i>	VARCHAR(45)		✓							

<i>productos</i>										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>Codigo</i>	VARCHAR(12)		✓							
<i>Nombre</i>	VARCHAR(50)		✓							
<i>Costo</i>	DOUBLE		✓							
<i>Precio</i>	DOUBLE		✓							
<i>Existencias</i>	INT(11)		✓							
<i>idCategoria</i>	INT(11)								NULL	
<i>proveedores</i>										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>Nombre</i>	VARCHAR(50)		✓							
<i>Direccion</i>	VARCHAR(45)		✓							
<i>Telefono</i>	VARCHAR(15)		✓							
<i>refreshtoken</i>										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>client_id</i>	INT(11)		✓							
<i>user_id</i>	INT(11)								NULL	
<i>token</i>	VARCHAR(255)		✓							
<i>expires_at</i>	INT(11)								NULL	
<i>scope</i>	VARCHAR(255)								NULL	

usuarios										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
<i>id</i>	INT(11)	✓	✓					✓		
<i>username</i>	VARCHAR(255)		✓							
<i>username_canonical</i>	VARCHAR(255)		✓							
<i>email</i>	VARCHAR(255)		✓							
<i>email_canonical</i>	VARCHAR(255)		✓							
<i>enabled</i>	TINYINT(1)		✓							
<i>salt</i>	VARCHAR(255)		✓							
<i>password</i>	VARCHAR(255)		✓							
<i>last_login</i>	DATETIME								NULL	
<i>locked</i>	TINYINT(1)		✓							
<i>expired</i>	TINYINT(1)		✓							
<i>expires_at</i>	DATETIME								NULL	
<i>confirmation_token</i>	VARCHAR(255)								NULL	
<i>password_requested_at</i>	DATETIME								NULL	
<i>roles</i>	LONGTEXT		✓							(DC2Type:array)
<i>credentials_expired</i>	TINYINT(1)		✓							
<i>credentials_expire_at</i>	DATETIME								NULL	
<i>credentials_expired</i>	TINYINT(1)		✓							
<i>credentials_expire_at</i>	DATETIME								NULL	

Anexo 2: Script de base de datos

```
--
-- Estructura de tabla para la tabla 'accesstoken'
--

CREATE TABLE IF NOT EXISTS 'accesstoken' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'client_id' int(11) NOT NULL,
  'user_id' int(11) DEFAULT NULL,
  'token' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'expires_at' int(11) DEFAULT NULL,
  'scope' varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY ('id'),
  UNIQUE KEY 'UNIQ_B39617F55F37A13B' ('token'),
  KEY 'IDX_B39617F519EB6921' ('client_id'),
  KEY 'IDX_B39617F5A76ED395' ('user_id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=3 ;
```

```
-- -----
```

```
--
-- Estructura de tabla para la tabla 'authcode'
--
```

```
CREATE TABLE IF NOT EXISTS 'authcode' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'client_id' int(11) NOT NULL,
  'user_id' int(11) DEFAULT NULL,
  'token' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'redirect_uri' longtext COLLATE utf8_unicode_ci NOT NULL,
  'expires_at' int(11) DEFAULT NULL,
  'scope' varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY ('id'),
  UNIQUE KEY 'UNIQ_F1D7D1775F37A13B' ('token'),
  KEY 'IDX_F1D7D17719EB6921' ('client_id'),
  KEY 'IDX_F1D7D177A76ED395' ('user_id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=1 ;
```

```
-- -----
```

```
--
-- Estructura de tabla para la tabla 'categorias'
--
```

```

CREATE TABLE IF NOT EXISTS 'categorias' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'Nombre' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Descripcion' varchar(150) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=4 ;

-- -----

--
-- Estructura de tabla para la tabla 'client'
--

CREATE TABLE IF NOT EXISTS 'client' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'random_id' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'redirect_uris' longtext COLLATE utf8_unicode_ci NOT NULL COMMENT
  '(DC2Type:array)',
  'secret' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'allowed_grant_types' longtext COLLATE utf8_unicode_ci NOT NULL
  COMMENT '(DC2Type:array)',
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=6 ;

-- -----

--
-- Estructura de tabla para la tabla 'clientes'
--

CREATE TABLE IF NOT EXISTS 'clientes' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'Nombre' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Apellido' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'NIT' varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  'Direccion' varchar(75) COLLATE utf8_unicode_ci NOT NULL,
  'Telefono' varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;

-- -----

```

```
--
-- Estructura de tabla para la tabla 'empleados'
--

CREATE TABLE IF NOT EXISTS 'empleados' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'Nombre' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Apellido' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Direccion' varchar(70) COLLATE utf8_unicode_ci NOT NULL,
  'Email' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;

-----

--
-- Estructura de tabla para la tabla 'productos'
--

CREATE TABLE IF NOT EXISTS 'productos' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'Codigo' varchar(12) COLLATE utf8_unicode_ci NOT NULL,
  'Nombre' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Costo' double NOT NULL,
  'Precio' double NOT NULL,
  'Existencias' int(11) NOT NULL,
  'idCategoria' int(11) DEFAULT NULL,
  PRIMARY KEY ('id'),
  KEY 'IDX_767490E6B2FA397B' ('idCategoria')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=3 ;

-----

--
-- Estructura de tabla para la tabla 'proveedores'
--

CREATE TABLE IF NOT EXISTS 'proveedores' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'Nombre' varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  'Direccion' varchar(45) COLLATE utf8_unicode_ci NOT NULL,
  'Telefono' varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY ('id')
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
-- -----
```

```
--
-- Estructura de tabla para la tabla 'refreshtoken'
--
```

```
CREATE TABLE IF NOT EXISTS 'refreshtoken' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'client_id' int(11) NOT NULL,
  'user_id' int(11) DEFAULT NULL,
  'token' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'expires_at' int(11) DEFAULT NULL,
  'scope' varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY ('id'),
  UNIQUE KEY 'UNIQ_7142379E5F37A13B' ('token'),
  KEY 'IDX_7142379E19EB6921' ('client_id'),
  KEY 'IDX_7142379EA76ED395' ('user_id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=3 ;
```

```
-- -----
```

```
--
-- Estructura de tabla para la tabla 'usuarios'
--
```

```
CREATE TABLE IF NOT EXISTS 'usuarios' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'username' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'username_canonical' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'email' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'email_canonical' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'enabled' tinyint(1) NOT NULL,
  'salt' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'password' varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  'last_login' datetime DEFAULT NULL,
  'locked' tinyint(1) NOT NULL,
  'expired' tinyint(1) NOT NULL,
  'expires_at' datetime DEFAULT NULL,
  'confirmation_token' varchar(255) COLLATE utf8_unicode_ci DEFAULT
NULL,
  'password_requested_at' datetime DEFAULT NULL,
```

```

    'roles' longtext COLLATE utf8_unicode_ci NOT NULL COMMENT
    '(DC2Type:array)',
    'credentials_expired' tinyint(1) NOT NULL,
    'credentials_expire_at' datetime DEFAULT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'UNIQ_EF687F292FC23A8' ('username_canonical'),
    UNIQUE KEY 'UNIQ_EF687F2A0D96FBF' ('email_canonical')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=3 ;

--
-- Restricciones para tablas volcadas
--
-- Filtros para la tabla 'accesstoken'
--
ALTER TABLE 'accesstoken'
  ADD CONSTRAINT 'FK_B39617F519EB6921' FOREIGN KEY ('client_id')
REFERENCES 'client' ('id'),
  ADD CONSTRAINT 'FK_B39617F5A76ED395' FOREIGN KEY ('user_id')
REFERENCES 'usuarios' ('id');

--
-- Filtros para la tabla 'authcode'
--
ALTER TABLE 'authcode'
  ADD CONSTRAINT 'FK_F1D7D17719EB6921' FOREIGN KEY ('client_id')
REFERENCES 'client' ('id'),
  ADD CONSTRAINT 'FK_F1D7D177A76ED395' FOREIGN KEY ('user_id')
REFERENCES 'usuarios' ('id');

--
-- Filtros para la tabla 'productos'
--
ALTER TABLE 'productos'
  ADD CONSTRAINT 'FK_767490E6B2FA397B' FOREIGN KEY ('idCategoria')
REFERENCES 'categorias' ('id');

--
-- Filtros para la tabla 'refreshtoken'
--
ALTER TABLE 'refreshtoken'
  ADD CONSTRAINT 'FK_7142379E19EB6921' FOREIGN KEY ('client_id')
REFERENCES 'client' ('id'),
  ADD CONSTRAINT 'FK_7142379EA76ED395' FOREIGN KEY ('user_id')
REFERENCES 'usuarios' ('id');
```