INSTITUCION UNIVERSITARIA POLITECNICO GRANCOLOMBIANO

MIGRACION DE SERVICIOS DE NEGOCIOS

Presentado por

CAMILO ANDRES PIZA HERNANDEZ

Asignatura

OPCION DE GRADO

AÑO

2020

`

TABLA DE CONTENIDO

DATOS GENERALES - PRÁCTICA EMPRESARIAL

Facultad:	Ingeniería y Ciencias Básicas
Departamento Académico:	INGENIERIA DE SISTEMAS
Nombre completo practicante:	CAMILO ANDRES PIZA HERNANDEZ
Código estudiante:	1610011255
Cédula No.:	1032501864
Correo electrónico:	CAPIZAHE@POLIGRAN.EDU.CO
Cargo asignado:	PRACTICANTE UNIVERSITARIO
Empresa:	BBVA COLOMBIA
Fecha inicio:	02 DE DICIEMBRE DE 2019
Fecha finalización:	02 DE JUNIO DE 2020
Jefe inmediato:	CAMILO AGUIRRE
Cargo Jefe inmediato:	TECH MANAGER I
Asesor práctica empresarial:	
Título del anteproyecto:	RESTRUCTURACION E IMPLEMENTACION DE SERVICIOS DE NEGOCIO
Área de conocimiento:	INGENIERIA DE SISTEMAS
Palabras clave:	INGENIERIA, ARQUITECTURA, DESARROLLO, ANALISIS

INTRODUCCIÓN

Los servicios de negocio cumplen una funcionalidad esencial en banco BBVA ya que se encargan de ser el puente entre el **BACKEND** y el **FRONTEND** como los son los canales digitales (BANCA MOVIL, BBVA NET, BBVA WALLET entre otras aplicaciones) Los en la actualidad estos servicios de negocio se encuentran construidos y en ejecución localmente. Las geografías en las que **BBVA** se encuentra actualmente, de igual manera los servicios de negocio cumplen y tienen la misma importancia como en Colombia, por esto nace la idea de reutilizar de esa manera poder evitar redundancias y agilizar la ejecución de futuros proyectos.

PALABRAS CLAVE:

Backend, BBVA, servicios, reutilizar

MARCO TEORICO

Los servicios son aplicaciones encargadas de la comunicación entre maquinas a través de la red, se encuentran diseñados mediante protocolos de comunicación. En la actualidad contamos con varios protocolos entre ellos los es **SOAP (**Simple Object Access Protocol**)** el cual está basado en especificación de XML del cual nace el principio **REST** (Representational State Transfer) donde se busca mejorar la trasmisión de datos**.**

SERVICIO WEB SOAP

Sus siglas significan Protocolo Simple de Acceso a Objetos en un protocolo de comunicación cuyo lenguaje está basado en **XML** entre sus características permite la extensibilidad ya que puede ser usado en cualquier proceso de negocio, a su vez es también es neutro ya que, aunque el

protocolo de comunicación por lo general es **HTTP** puede ser usado con otros protocolos de comunicación como lo es el **FTP**.

SERVICIO WEB REST

Sus siglas significan Transferencia de Estado Representacional es una arquitectura cuyo aliado es el protocolo **HTTP.** Es una interfaz para la comunicación entre maquinas haciendo uso del protocolo **HTTP** y sus verbos como lo son GET, POST, PUT, DELETE, PATCH que son los mas comunes y mas usados bajo esta arquitectura y cuyo formato más común de respuesta es **JSON (**JavaScript Object Notation**)** no obstante está abierto a otros formatos como lo es **XML**.

API

Sus siglas significan Interfaz de programación de aplicaciones, API es un conjunto de protocolos y definiciones que se usan para integrar el software del lado del Backend, su principal objetivo es comunicar maquinas sin la necesidad de saber cómo se implementa mediante endpoints.

SPRING FRAMEWORK

Framework de código abierto para la creación de aplicaciones empresariales Java, cuenta con una estructura modular para implementar diferentes tipos de arquitecturas según las necesidades de la aplicación. Entre sus características se encuentran:

- Inyección de dependencias
- Desarrollo con POJOS (Plain Object Data Objects)
- Simplifica el acceso a datos.

RAML (RESTful API Modeling Lenguaje)

Es una manera simple y estructurada de describir APIs RESTful, basado en HTTP ya que involucra los principios REST. Genera automáticamente la documentación mediante documentos interactivos.

DESCRIPCIÓN DE LA EMPRESA

El banco BBVA COLOMBIA S.A cuyas siglas hacen referencia a Banco Bilbao Vizcaya Argentaria es una entidad bancaria de primer nivel; Hace parte de un grupo financiero global, con un negocio que ofrece servicios financieros en más de 30 países a 53 millones de clientes. BBVA con una trayectoria de más de 50 años, en Colombia es destacada por ser una entidad comprometida con el desarrollo social y económico del país. Con presencia en más de 122 municipios a lo largo del territorio nacional. Además de ser conocida como una entidad ágil y moderna, caracterizada por su constante vocación de liderazgo e innovación. La cual le ha permitido ocupar siempre los primeros lugares en el sector financiero. [1]

FORMULACIÓN DEL PROBLEMA

PROBLEMÁTICA GENERAL

En la actualidad la Banca Móvil está construida con un sin número de piezas locales lo cual desvía el camino de la reutilización. En todas las geografías en las cuales BBVA se encuentra la Banca Móvil se encuentra construida de manera similar a como esta en Colombia. Por lo tanto, estos componentes permiten abrir campo a la reutilización de estas tecnologías y servicios que esta misma ofrece.

AREA DE CONOCIMIENTO

Entre las áreas de conocimiento podemos identificar las áreas de ingeniería tales como el desarrollo, análisis y optimización de esa manera poder obtener los mejores resultados puesto que varios servicios cuentan con implementaciones las cuales se pueden optimizar y reorganizar de esa manera mejorar su mantenibilidad y reutilización.

PROPUESTA DE PRÁCTICA

JUSTIFICACIÓN

Se busca realizar la migración de la Banca móvil a tecnologías globales BBVA para poder redundar el beneficio de la reutilización de componentes implementados por los diferentes países en los cuales BBVA hace presencia. Este proyecto se está trabajando por bloques y en la actualidad estamos sobre el bloque 0 el cual cuenta con 25 servicios

Las áreas de conocimiento a usar son aquellas que se encuentran de manera explícita como implícita dentro de su implementación, ya que varias de ellas y sus mejores prácticas se forman por la academia. Se busca que cada uno de los desarrollos sean implementados de manera limpia y ordenada de esa manera evitar errores que puedan conllevar a un fallo interno del servicio o pueda causar alguna vulnerabilidad en el sistema. El principal objetivo es realizar cada uno de ellos con basados en las buenas prácticas de POO y patrones de diseño

ALCANCE

Cada uno de los servicios de negocio cumplen un objetivo importante dentro de la organización tanto para empleados como clientes por lo tanto es importarte que cada uno de ellos se actualicen en un tiempo prudente con sus desarrollos y pruebas de esa manera poder realizar su paso a producción.

OBJETIVO PRINCIPAL

Desarrollar y actualizar los servicios y microservicios de negocio del banco los cuales se encuentran actualmente implementados localmente y se busca volverlos servicios globales para poderlos implementar en las diferentes geografías.

OBJETIVOS SECUNDARIOS

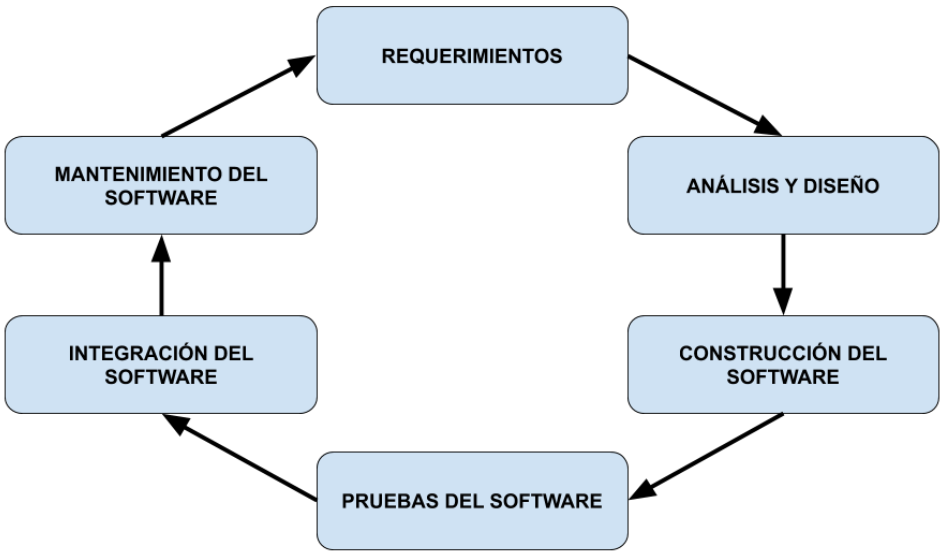
- Diseñar los nuevos servicios de negocio mediante el lenguaje de modelo de Apis **RAML** el cual nos da una visión más clara y concisa de las propiedades que tendrá el servicio y su objetivo como también permitir la reutilización de este.
- Desarrollar los servicios de negocio basados en los criterios diseñados en el **RAML** haciendo uso del framework **Springboot** que es la herramienta que en la cual están desarrollados los servicios actualmente.
- Realizar las pruebas y ajustes necesarios a cada uno de los servicios verificar que cumplan con el objetivo para el cual se encuentran implementados de esa manera evitar posibles errores durante el desarrollo haciendo uso de **Junit** y **Mockito**.
- Dar soporte al área de calidad y producción a los servicios que presentan inconsistencias.

METODOLOGÍA

DESCRIPCIÓN DE ACTIVIDADES

Se define ciclo de vida de software como la herramienta que describe el desarrollo de software, desde su fase inicial hasta su fase final.

Figura 1: Ciclo de vida de software



Fuente: Elaboración propia

El proyecto cuenta con diferentes etapas para su desarrollo; La primera etapa cuenta con una fase de entendimiento que consiste en comprender los requerimientos y basados en ellos establecer una posible fecha de entrega estimada basada en la experiencia de los proyectos anteriores. La siguiente fase consiste en el análisis la cual nos permite identificar la funcionalidad principal del servicio a desarrollar. Posteriormente la etapa de desarrollo en la cual es necesario realizar la codificación correspondiente para el funcionamiento técnico del servicio teniendo en cuenta las diferentes recomendaciones y siguiendo los patrones de diseño ya implementados. Después de la "finalización" de estas etapas sigue es necesario realizar las pruebas unitarias de cada uno de los

servicios anteriormente desarrollados de esa manera descartar posibles fallos en el desarrollo y las pruebas locales las cuales me permiten hacer pruebas en un ambiente de desarrollo. Finalmente, las fases de implementación y mantenimiento las cuales consisten en su paso a calidad y estar al tanto de posibles fallos durante su despliegue para dar soporte al servicio implementado.

CRONOGRAMA DE ACTIVIDADES

El cronograma funciona de manera iterativa puesto que para cada desarrollo

ACTIVIDADES	FASES	TAREAS	MESES(SEMANAS)															
			FEBRERO				MARZO				ABRIL				MAYO			
CAPACITACION	**Capacitación**	Entendimiento herramientas de desarrollo																
PRIMER SERVICIO (Desde Cero)	**Entendimiento**	Definición del problema																
		Identificar requerimientos																
		Establecer fecha entrega																
		Identificar documentación relacionada																
	Análisis	Identificar funcionalidad del servicio.																
	Desarrollo	Implementación JSON descriptor																
		Implementación servicios correspondientes.																
	Pruebas	Test unitarios																
		Pruebas locales																
SEGUNDO SERVICIO (Actualizar)	**Entendimiento**	Definición problema																
	Análisis	Identificar funcionalidad																
	Desarrollo	Implementación servicios correspondientes																
	Pruebas	Pruebas unitarias																
		Pruebas locales																
TERCER SERVICIO (Actualizar)	**Entendimiento**	Definición problema																
	Análisis	Identificar funcionalidad																
	Desarrollo	Implementación servicios correspondientes																

	Pruebas	Pruebas unitarias																	

ENTREGABLES

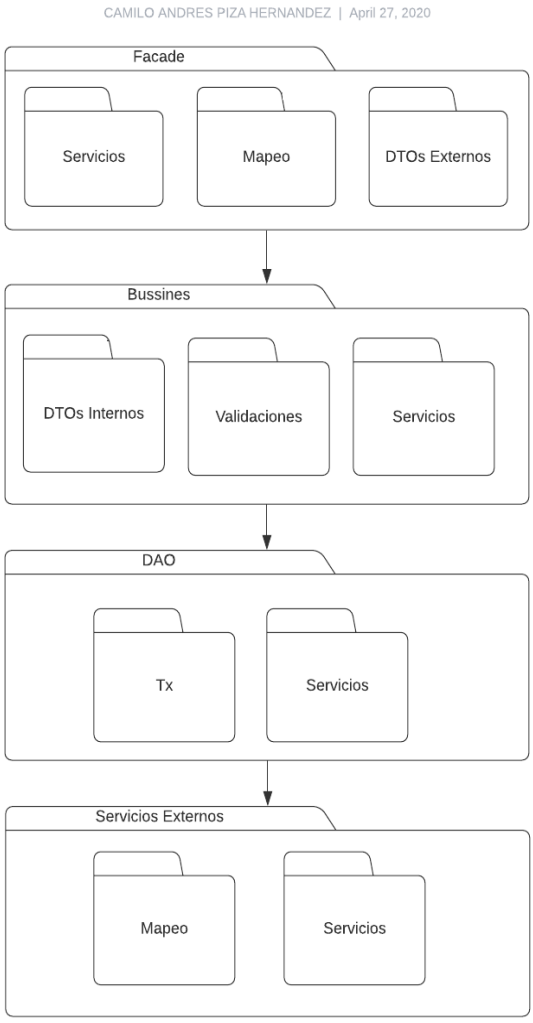
Basados en el ciclo de vida de software y cada una de sus fases, los entregables se realizan mediante el uso de la metodología **SCRUM** puesto que es necesario mantener una comunicación constante con el supervisor de esa manera asegurar que se está desarrollando de manera asertiva por lo tanto día tras día se hace un Daily Sprint para aclarar posibles dudas, al finalizar cada semana mediante un Sprint Review se hace entrega de avance del proyecto se realizan los cambios de ser necesario y finalmente para entrega se realiza una reunión final con el supervisor para verificar el correcto funcionamiento del servicio de esa manera cerrar la tarea correspondiente.

FASE DE IMPLEMENTACION

DESARROLLO

Durante el proceso de desarrollo se presentaron diferentes áreas de conocimiento en el desarrollo de software como lo es la programación orientada a objetos y el uso de patrones de diseño, ya que es necesario que el código cuente con la calidad necesaria para su paso a producción, consiste en 4 capas las cuales cumplen un objetivo específico (Véase **figura 2**)

Figura 2. Diagrama de paquetes.



Fuente: Elaboracion propia

FACADE SERVICIOS	Es la capa la cual se encarga de comunicar los servicios con el consumidor FRONTEND (Banca móvil)
MAPEO	Consiste en recibir el input y mapearlo a datos basado en el modelo de negocio de DTO Externo a DTO interno
DTO'S EXTERNOS	Son los objectos que llegan desde el consumidor mediante los métodos **HTTP** los cuales contienen información para realizar la solicitud.
DTO'S INTERNOS	Son objetos internos del servicio lo cual permite establecer la correcta comunicación los objetos dentro del servicio.
VALIDACIONES	Valida si los datos ingresados y los datos de respuesta cuentan con todos sus campos requerido
TX	Transacción directa con el **BACKEND** principal del banco
SERVICIOS EXTERNOS	Se encarga de realizar la petición a servidores externos al banco. Ejemplo: Baloto

Cada una de estas capas corresponde a una funcionalidad en el sistema para poder lograr el principal objetivo que es completar satisfactoriamente la solicitud evitando errores en las capas externas como las son "DAO" y "Servicios Externos". Estas capas son:

FACADE (FACHADA)

En la primera capa se encuentra el paquete **Facade** el cual hace referencia al patrón de diseño ya que es un patrón estructural cuyo objetivo es simplificar la complejidad para el uso del servicio de esa manera el cliente solo tenga un punto de acceso.

En el paquete "Mapper" dentro de la capa Facade consiste en recibir los **DTOs (**Data Transfer Object**)** externos como input y como output se retornan **DTOs** internos para poder manejar la misma estructura con los servicios. Se validan que los datos no sean nulos para evitar errores en la ejecución del servicio puesto que algunos de los parámetros de entrada no son requeridos.

BUSINESS (NEGOCIO)

En esta capa se realizan las validaciones correspondientes en los inputs y outputs. Se verifica que los campos requeridos no sean nulos o que no manejen una estructura adecuada. Dado el caso que el input sea incorrecto ser retornará un **BUSINESS EXCEPTION** informando al cliente que los datos ingresados son erróneos o no cumplen con el formato necesario para continuar con el proceso.

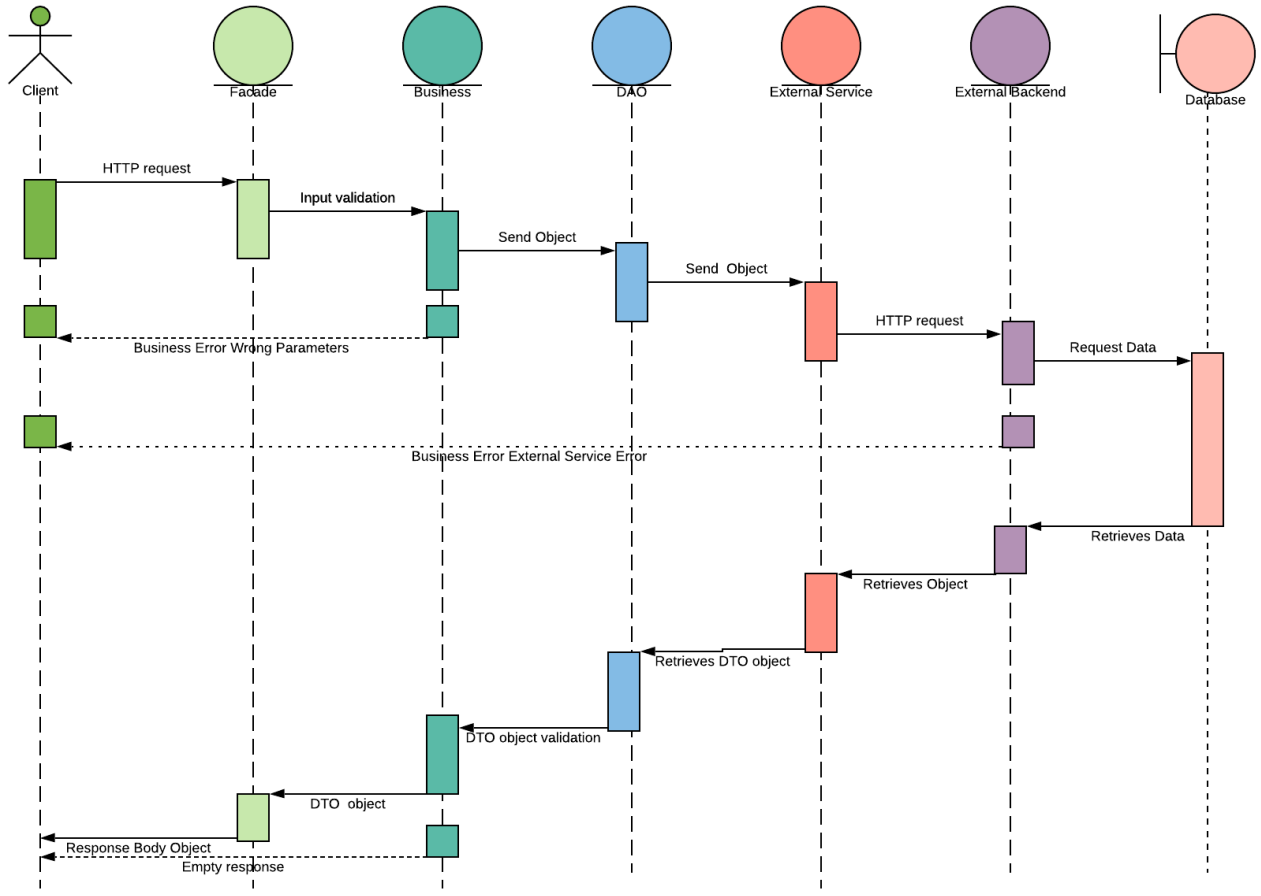
DAO (OBJETO DE ACCESO A DATOS)

Esta capa hace referencia al patrón de diseño **DAO** cuyo objetivo es proporcionar un objeto de acceso a datos. Esta capa para transacciones directas con el Backend (Host) del banco la cual se encuentra definida mediante métodos con las operaciones CRUD **(**Create Read Update Delete**)** u otras consultas que sean necesarias para el funcionamiento de dicho servicio.

SERVICIOS EXTERNOS

La función principal de esta capa es realizar peticiones a servidores externos al banco. Consiste en realizar la petición http o https al servidor y mapear su respuesta que viene en formato JSON a los **DTOs** internos ya implementados mediante librerías que me permite des serializar el objeto JSON a un objeto definido en el modelo.

Figura 3. Diagrama de secuencia



Fuente: Elaboración propia

PRUEBAS

En esta etapa próximo a finalizar el desarrollo se empiezan a hacer pruebas unitarias para verificar el correcto funcionamiento del servicio haciendo uso de librerías tales como Junit y el framework Mockito el cual nos permite aislar las cada uno de los componentes así poder evaluarlo de manera independiente.

Figura 3. Ejemplo prueba mockito

```
import static org.fest.assertions.Assertions.assertThat;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;

import org.junit.Test;

public class MyTest {

    @Test
    public void testMocking() {
        final MyServiceClientImpl client = mock(MyServiceClientImpl.class);
        final String expected = "Success!";
        when(client.callService()).thenReturn(expected);
        assertThat(client.callService()).isEqualTo(expected);
    }
}
```

Fuente: http://www.zapp.city/kotlin/testing/java/2016/02/16/kotlin-mockito.html

Cuando la fase de pruebas unitarias funciona correctamente y cada una de las pruebas se compilan satisfactoriamente se procede a compilar el proyecto mediante la herramienta **Maven** el cual es una herramienta de gestión y comprensión de proyectos de software de esa manera validad la compilación exitosa entre cada uno de los componentes del servicio, si el proceso culmina satisfactoriamente generará un Jar con en el proyecto. Siguiente a esto se hace uso de servidor de pruebas en ambiente de desarrollo el cual se encuentra en un contenedor **Docker** el cual cuenta con las funcionalidades del servidor en producción. Se realizan las configuraciones

necesarias y se procede a desplegar el servidor en local. Finalmente, mediante un formato de pruebas unitarias evidenciar que el servicio funciona correctamente (Véase **figura 4**).

Figura 4. Formato pruebas unitarias

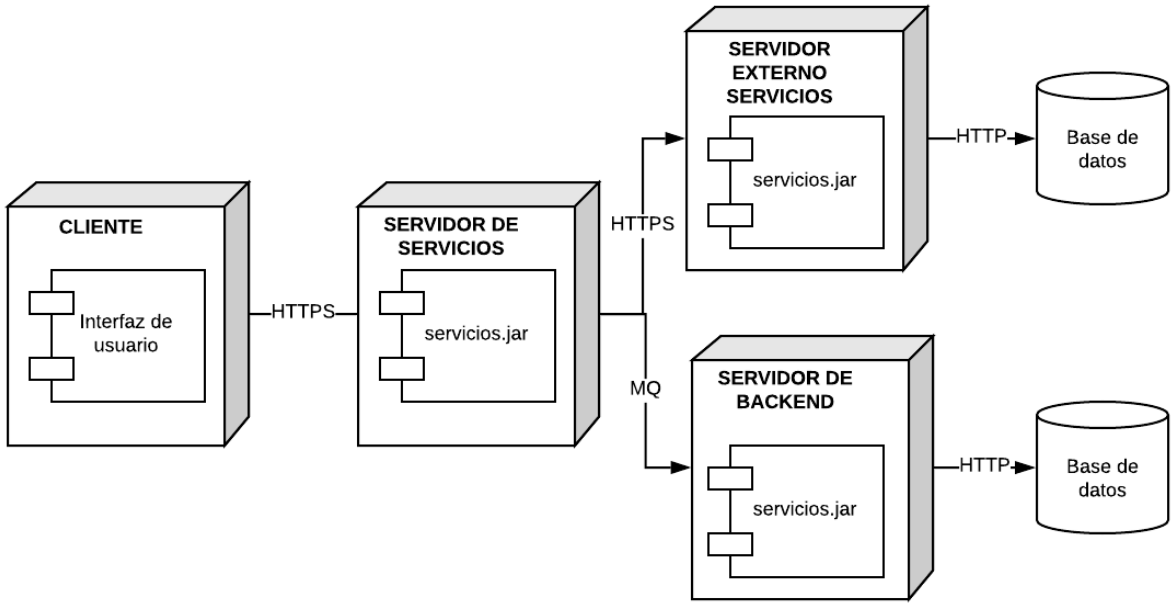
Fuente 4. Documento banco BBVA

DESPLIEGUE

El despliegue consiste en su paso a producción, después de haber culminado con el desarrollo, las pruebas unitarias y en ambiente de desarrollo el servicio pasa al área de calidad para revisión

y finalmente si el servicio cumple con los objetivos requeridos se procederá a su paso a producción (**Véase** figura 5).

Figura 4. Diagrama de componentes



Fuente. Elaboración propia

RETROSPECTIVA

En la finalización de cada servicio se comenta la experiencia desarrollando cada uno de ellos en términos de dificultad y de posibles opciones de mejora para optimizar el tiempo de entrega de cada uno de desarrollos manteniendo la balanza entre tiempo y calidad. También se discuten las diferentes implementaciones en diferentes puntos específicos del servicio puesto que pueden ser mejoras para próximos proyectos.

CONCLUSIONES

Durante el desarrollo de varios de los servicios fue necesario revisar la documentación de algunos de las librerías puesto que no se contaba con el conocimiento suficiente para su uso y eran herramientas indispensables ya que su funcionalidad reduce el tiempo y la complejidad durante el desarrollo. Entre estas librerías tales como Mockito el cual es un framework que aísla el servicio para hacer las pruebas unitarias con mayor precisión y exactitud para prevenir fallos en el sistema. Son conocimientos adquiridos de gran valor ya que son herramientas que se usan en el mundo laboral. Otra de ellas es la librería “Jackson” que es una librería que se encarga de des serializar los elementos en formatos JSON, varios de los objetos entrantes eran complejos y necesitaban de un modelo abstracto para poder mapear los objetos correctamente por lo cual era necesario recibir información detallada de la funcionalidad de los métodos para lograr el objetivo.

Dada las condiciones del país mayor parte del trabajo fue remoto, lo cual ayudo en mi formación profesional, en el manejo de los tiempos, responsabilidad y compromiso.

El tiempo en la oficina me permitió estar rodeado de gente muy profesional y hábil por eso entendí que la universidad es un pequeño paso para lograr tus sueños y que la experiencia es un factor fundamental.

BIBLIOGRAFÍA

Apache Maven Project. (s.f.). *What is maven?* Obtenido de Maven: https://maven.apache.org/what-is-maven.html

BBVA. (s.f.). *BBVA Móvil*. Obtenido de BBVA: https://www.bbva.com.co/personas/servicios-digitales/movil.html

BBVA COLOMBIA. (s.f.). *bbva.com.co*. Obtenido de https://www.bbva.com.co/personas/historia.html

Dept. Ciencia de la computacion. (2012). *Casos de prueba: JUnit.* Obtenido de http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.pdf

FasterXML. (s.f.). *Jackson*. Obtenido de github: https://github.com/FasterXML/jackson

Json. (s.f.). *Introducing JSON*. Obtenido de JSON: https://www.json.org/json-en.html

Mockito. (s.f.). *Why drink it? Mockito* . Obtenido de Mockito: https://site.mockito.org/

Mozilla. (s.f.). *HTTP*. Obtenido de https://developer.mozilla.org/: https://developer.mozilla.org/es/docs/Web/HTTP

Platzi. (s.f.). *Qué es Frontend y Backend*. Obtenido de Platzi: https://platzi.com/blog/que-es-frontend-y-backend/

RAML. (s.f.). *The Simple Way To Design APIs*. Obtenido de RAML: raml.org

SCRUM. (s.f.). *¿Qué es SCRUM?* Obtenido de scrum.org: www.scrum.org/resources/blog/que-es-scrum

W3C. (11 de February de 2004). *Web Services Glossary*. Obtenido de W3C: https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice

GLOSARIO

Api

Conjunto de definiciones y protocolos para integrar un software de aplicaciones. Permiten la comunicacion entre productos., 8

Banca Móvil

Aplicación que permite reaizar operaciones bancarias sin necesidad de ir a un banco, 7

CRUD

Termino que se usa para referirse a operaciones basicas de base de datos, 13

framework

Conjunto estandarizado de conceptos en un entorno de trabajo para facilitar el desarrollo de un producto o servicio, 8

Jar

Tipo de archivo que permite ejecutar aplicaciones y herramientas escritas en lenguaje java., 15

Petición

Indica la accion que se desea realizar para un recurso determinado. Tambien llamados "HTTP verbs", 14