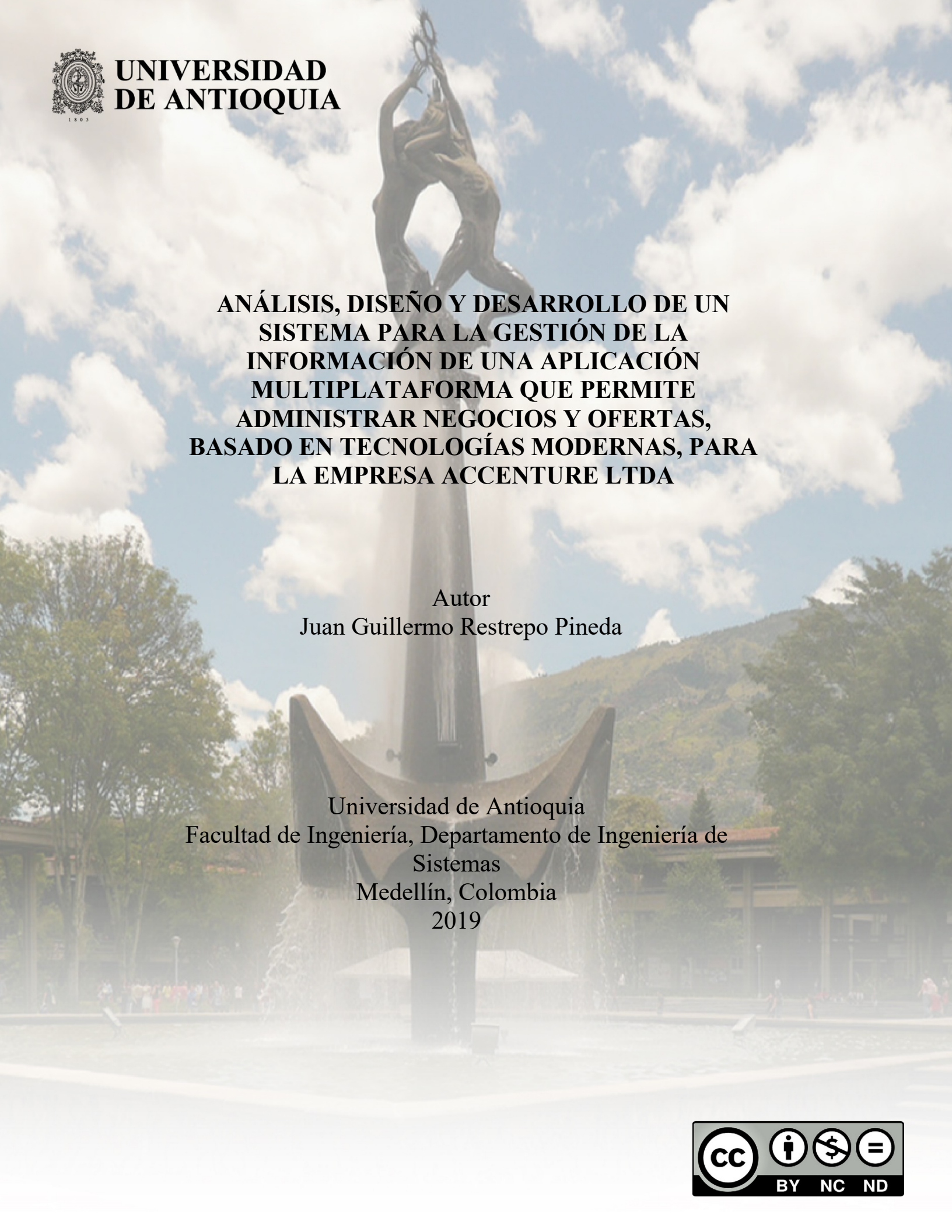


UNIVERSIDAD DE ANTIOQUIA

ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE UNA APLICACIÓN MULTIPLATAFORMA QUE PERMITE ADMINISTRAR NEGOCIOS Y OFERTAS, BASADO EN TECNOLOGÍAS MODERNAS, PARA LA EMPRESA ACCENTURE LTDA

Autor
Juan Guillermo Restrepo Pineda

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas
Medellín, Colombia

2019

ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE UNA APLICACIÓN MULTIPLATAFORMA QUE PERMITE ADMINISTRAR NEGOCIOS Y OFERTAS, BASADO EN TECNOLOGÍAS MODERNAS, PARA LA EMPRESA ACCENTURE LTDA

Juan Guillermo Restrepo Pineda

Informe de práctica
como requisito para optar al título de:
Ingeniero de Sistemas

Carlos Mario Sierra Duque

Juan Guillermo Restrepo Pineda, Ingeniero de Sistemas

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas
Medellín, Colombia
2019

ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE UNA APLICACIÓN MULTIPLATAFORMA QUE PERMITE ADMINISTRAR NEGOCIOS Y OFERTAS, BASADO EN TECNOLOGÍAS MODERNAS, PARA LA EMPRESA ACCENTURE LTDA

Resumen

Accenture LTDA requiere construir un aplicativo multiplataforma para un cliente interno que le permita administrar usuarios, negocios y ofertas de productos de forma efectiva, utilizando sistemas de filtrado y geolocalización fáciles de manejar. La aplicación debe utilizar herramientas y tecnologías solicitadas por el cliente y, a su vez, debe implementar nuevas tecnologías que le proporcionen ventajas competitivas con respecto a productos similares de sus competidores. Como resultado se obtiene una plataforma web y móvil que cumple con las expectativas del cliente, satisfaciendo criterios de calidad y usabilidad que les facilitan a sus usuarios su utilización.

Introducción

La empresa Accenture LTDA es una compañía multinacional dedicada a la prestación de servicios profesionales, con un enfoque en soluciones para las áreas de estrategia, consultoría, digital, tecnología y operaciones [1]. En el mercado tecnológico, Accenture aprovecha el poder de las tecnologías emergentes con el fin de obtener niveles de rendimiento óptimos en los sistemas construidos para sus clientes [2]. Analizando su alcance a nivel de negocios, el cual contempla un amplio abanico de contextos para la prestación de sus servicios, se observa de manera particular las capacidades globales con las que cuenta la empresa para apoyar productos a nivel financiero, en donde destaca en la realización del diseño e implementación de pruebas automatizadas de aplicaciones de todo tipo (móvil, web, escritorio y aplicaciones por demanda en general). Mediante el análisis, diseño y desarrollo de herramientas y aplicaciones que hagan uso de tecnologías modernas demandas por sus clientes, Accenture busca potenciar su mercado en el área de desarrollo y acceder a nuevos y potenciales clientes. Por tal motivo, la empresa inicia en Medellín un Centro de Tecnología Avanzada (Advanced Technology Center o ATC por sus siglas en ingles), para proveer servicios de desarrollo e innovación utilizando herramientas, procesos, modelos de automatización y de delivery más avanzados a nivel mundial [3]. Como parte integral de esta estrategia, el ATC inicia la construcción de diferentes sistemas tecnológicos para llenar las necesidades de sus clientes, entre los que se encuentra el diseño de un sistema multiplataforma que permita, a grandes rasgos, administrar las ofertas de múltiples negocios, junto con sus clientes y administradores de negocios.

En este proyecto se busca desarrollar una aplicación multiplataforma, que permita registrar a usuarios, clientes, o a administradores. Los administradores tendrán la capacidad de registrar, gestionar y publicar sus negocios y ofertas. Por otra parte, los clientes podrán buscar los negocios y ofertas registrados en el sistema, geolocalizarlos en un mapa y acceder a sus detalles.

Esta nueva aplicación hace uso de tecnologías vigentes como Spring, Spring Boot, Xamarin Forms, Google Maps, Angular y servicios de Amazon AWS Cloud. El enfoque de este desarrollo será la realización del Backend o la capa de datos del sistema, junto con el desarrollo móvil con Xamarin Forms.

Objetivo General

Analizar, diseñar e implementar el Backend o capa de acceso de datos para dos aplicaciones homólogas (en versiones web y móvil), el cual les permita a los clientes buscar ofertas comerciales de todo tipo, y a administradores de negocios registrar y gestionar sus establecimientos junto con sus ofertas.

Objetivos Específicos

- Realizar el análisis y diseño del Backend, haciendo uso de una arquitectura basada en microservicios.
- Concretar las unidades básicas de información o entidades de datos que integran el sistema de información.
- Hacer uso de la base de datos no relacional DynamoDB (por petición del cliente), para estructurar y almacenar la información de la aplicación.
- Analizar, diseñar y documentar una API Restful que permita la comunicación de los datos con la interfaz de usuario.
- Realizar una especificación inicial de los recursos, URLs o endpoints que se expondrán al Frontend, los cuales representarán las posibles operaciones que puede realizar la aplicación sobre la información, como puede ser la creación, lectura, actualización y/o eliminación de datos, con el fin de ser utilizados por las aplicaciones (web y móvil).
- Implementar una API Restful que haga uso de la arquitectura basada en microservicios web, la cual permita administrar operaciones de registro, consulta, actualización y desactivación de clientes.
- Implementar una API Restful que haga uso de la arquitectura basada en microservicios web, la cual permita administrar operaciones de creación, lectura, actualización y eliminación de negocios.
- Implementar una API Restful que haga uso de la arquitectura basada en microservicios web, la cual permita administrar operaciones de registro, lectura, actualización y eliminación de ofertas.
- Validar las operaciones de todas las APIs, haciendo pruebas sobre el código escrito previamente.
- Interconectar las APIs desarrolladas.
- Construir una aplicación móvil que permita el consumo de los microservicios anteriormente implementados. La aplicación debe ser multiplataforma.

Marco Teórico

En el desarrollo de aplicaciones multiplataforma basados en tecnologías emergentes, es indispensable la especificación y construcción de un sistema de información que se encargue

de las tareas relativas a la configuración y gestión de los datos necesarios para el correcto funcionamiento de dichas aplicaciones. Son múltiples las aproximaciones que permiten abordar el proceso de elaboración de este sistema, entre los que se encuentra el diseño basado en microservicios.

Dicha aproximación será utilizada en este proyecto haciendo uso de nuevas tecnologías, con el fin de desarrollar un sistema que facilite administrar la información para aplicaciones multiplataforma que permite registrar, administrar y buscar tanto negocios como ofertas. En la Tabla 1 se da una breve explicación de estas tecnologías y otros conceptos esenciales.

Tabla 1. Arquitectura y tecnologías necesarias para el desarrollo del producto

Tecnologías	**Descripción general**
Diseño basado en microservicios	El diseño o arquitectura basado en microservicios es un enfoque de desarrollo que surge como respuesta al diseño monolítico de aplicaciones y que busca la segmentación de sistemas de información y sus funcionalidades en un conjunto de pequeños servicios independientes e intercomunicados y con responsabilidades altamente diferenciadas. Este planteamiento reduce los tiempos de desarrollo y la corrección de errores debido al bajo acoplamiento de las funciones [4].
Spring y Spring Boot	Spring es un framework de desarrollo de aplicaciones en Java, fundamentado en el principio de inversión de dependencias y que facilita la construcción de servicios web. Por otra parte, Spring Boot es una extensión de Spring que facilita la inicialización de proyectos Spring, proveyendo las dependencias y la estructura básica. [5].
Apache Camel	Apache Camel es un marco de integración versátil de código abierto basado en conocidos patrones de integración empresarial, provee una biblioteca con dependencias mínimas para una fácil integración en cualquier aplicación Java [6].
AWS DynamoDB	DynamoDB es un servicio de bases de datos no relacional proporcionado por Amazon Web Services para aplicaciones que necesitan un alto rendimiento a cualquier escala, administrada totalmente en la nube, rápida y que soporta estructuras de datos basadas en documentos y pares de valor y clave. Una de las características por las que destaca este servicio, es la posibilidad de estructurar datos en una base de datos no relacional haciendo uso de tablas [7].
Swagger y Swagger Editor	Swagger es un marco de software de código abierto respaldado por un gran ecosistema de herramientas que ayuda a los desarrolladores a diseñar, crear, documentar y consumir servicios

	web RESTful. Entre estas herramientas se encuentra Swagger Editor, el cual utiliza el estándar OpenAPI y el lenguaje YAML para la descripción de APIs. Una API es un conjunto de especificaciones y reglas en código que las aplicaciones utilizan para comunicarse entre ellas; en síntesis, consiste en una interfaz que facilita la comunicación entre los programas. Por otra parte, YAML es un lenguaje descriptivo que utiliza un formato que permite serializar datos legibles para humanos. [8].
Xamarin	Xamarin es un conjunto de herramientas o toolkit multiplataforma para construir aplicaciones escritas en .NET. Su mayor ventaja es la posibilidad de desarrollar sistemas totalmente nativos en Android, iOS y Universal Windows Platform, utilizando el lenguaje de programación C# [9].
Azure DevOps	Azure DevOps es una herramienta de Microsoft que permite abarcar equipos de desarrollo con roles especiales y herramientas para arquitectos de software, especialistas en desarrollo y testers [10].

Metodología de trabajo

Se hizo uso de la metodología de trabajo ágil SCRUM [11], utilizando las técnicas y herramientas que proporciona para facilitar la realización de todas las actividades del proyecto. En concreto se definieron y refinaron un conjunto moderado de tareas o actividades que se ejecutaron durante un periodo de dos semanas. Luego de esto, se realizó una revisión de los resultados obtenidos en todo el grupo de desarrollo y se procedió a hacer una retrospectiva de todo el proceso, con el fin de mejorar la metodología de trabajo de ser necesario. Al final, se le entregan al dueño del producto los resultados obtenidos y se repitió todo este proceso seleccionando un nuevo conjunto de actividades.

Metodología de desarrollo

Debido a la naturaleza del proyecto, y apoyándose en la metodología de trabajo, se identifican las etapas de desarrollo descritas en la Tabla 2, las cuales facilitaron el logro eficaz de los objetivos.

Tabla 2. Etapas de desarrollo

Etapa	**Descripción general**
Análisis del problema	Se realiza una revisión y descomposición de los requerimientos provistos por la empresa, con el fin de obtener y aclarar las necesidades que se deben superar. Estas necesidades se plasman en tareas y actividades.
Diseño de la	Se evalúan las tecnologías que deben ser utilizadas para la

solución	implementación de la solución. Se define y construye la arquitectura de la solución. Se realiza el diseño arquitectónico de la solución.
Capacitaciones	Se capacitan a los integrantes del grupo en las tecnologías necesarias.
Implementación o codificación	Se hace uso de las tecnologías seleccionadas en la etapa anterior y se codifica la solución para cada necesidad y problema.
Pruebas de la implementación	Se valida el desarrollo realizado hasta el momento.
Entrega del artefacto de valor	Se hace entrega del producto al cliente y se recibe su retroalimentación.

Análisis del problema

En esta etapa del proyecto se especificaron las funcionalidades requeridas por el cliente, las cuales fueron plasmadas en historias de usuario junto con sus criterios de aceptación. Se definió el alcance del proyecto y se estimaron los tiempos y necesidades de este.

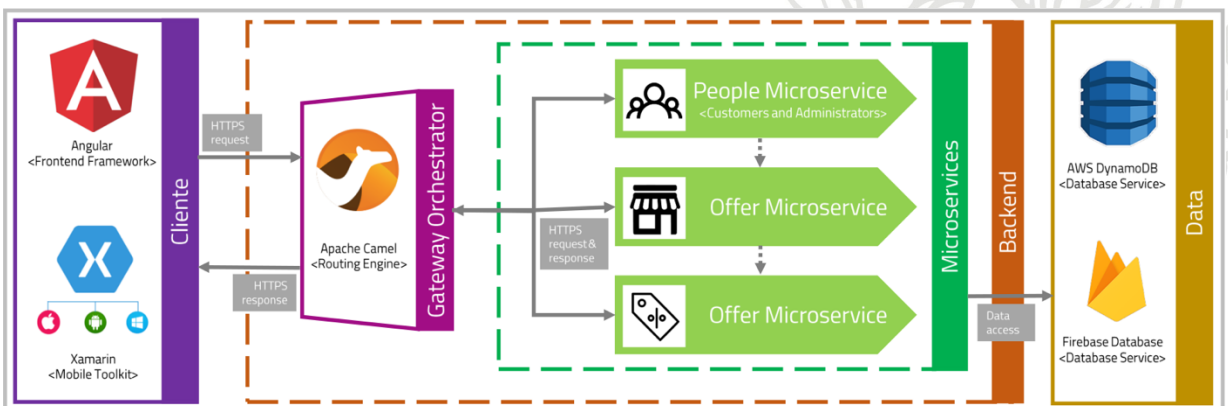
Para coordinar al equipo durante la realización del proyecto, se utilizaron los servicios de Azure DevOps para organizar las historias de usuario en un tablero Kanban, almacenar el código en un repositorio remoto y automatizar el despliegue de la aplicación de forma local.

Diseño de la solución

En esta etapa se evaluaron las tecnologías que debían utilizarse para la construcción del producto final, ya que estas fueron impuestas por el cliente, en lo que se refiere a nivel de datos. Entre las tecnologías a utilizar estaban Spring, AWS DynamoDB y Swagger. Al realizar este ejercicio, se determinó la necesidad de capacitar a los miembros del equipo en estas tecnologías.

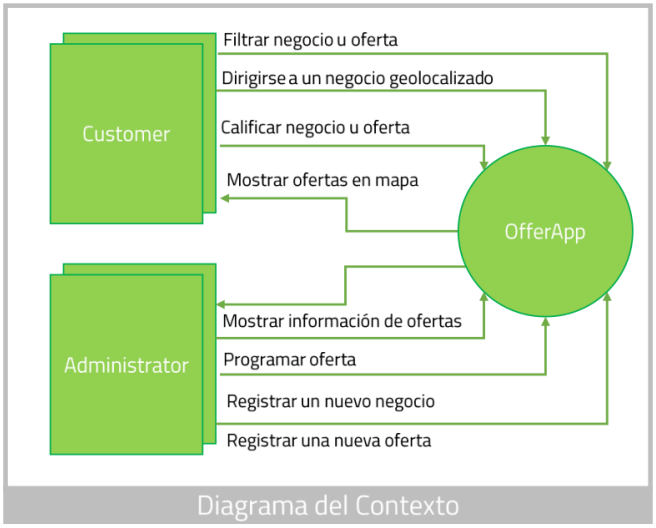
Teniendo en cuenta las precondiciones del proyecto, el equipo procedió a delinear la arquitectura del producto, la cual está basada en microservicios. En la Figura 1 se contempla de forma general la arquitectura construida, profundizando en la capa de acceso a datos.

Figura 1. Diseño arquitectónico general de la aplicación



El usuario ingresa a la aplicación a través de su navegador web o su dispositivo móvil. Entre las operaciones que puede solicitar el cliente se encuentra el registro de usuario como cliente o administrador, inicio de sesión en la aplicación, registro, búsqueda y administración de negocios y ofertas, programación de inicio y fin de ofertas, geolocalización, desplazamiento y vista panorámica hacia los negocios de interés haciendo uso de la API de Google Maps, búsqueda de negocios y ofertas por filtrado, calificación de ofertas, entre otras opciones, tal y como se observa en la Figura 2.

Figura 2. Diagrama del contexto de la aplicación



Capacitaciones

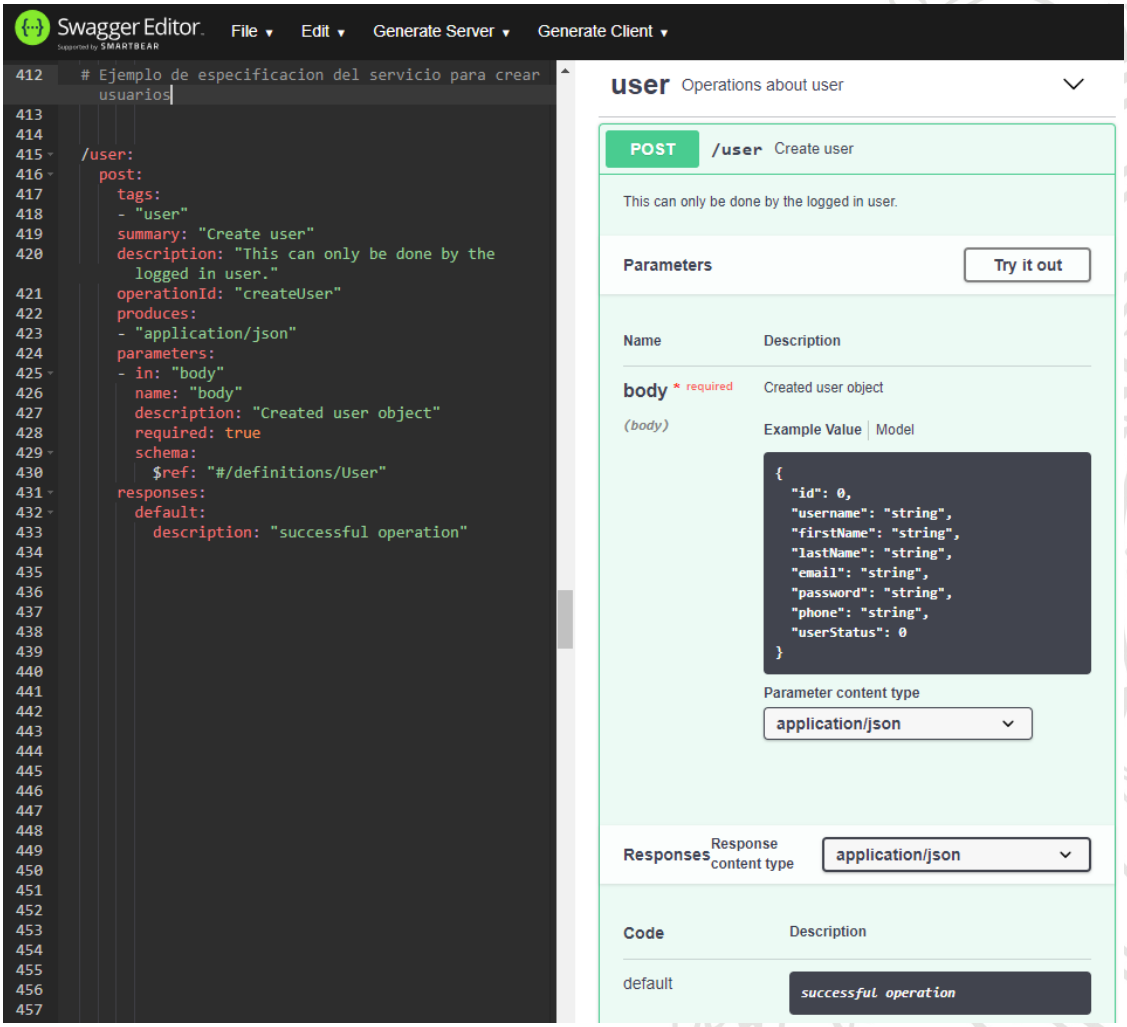
Teniendo clara la arquitectura de la aplicación, en esta etapa se capacitó a los integrantes del equipo de desarrollo en las tecnologías necesarias, dependiendo de sus responsabilidades en el proyecto. Para los integrantes encargados del Backend se hizo énfasis en Spring con Spring Boot para la construcción de los microservicios, Swagger para la documentación del producto, Apache Camel para la orquestación de microservicios y Xamarin Forms para la versión móvil.

Implementación y pruebas

En esta etapa se ejecutó la construcción de la aplicación multiplataforma, comenzando con la especificación de los microservicios haciendo uso de Swagger Editor, en el cual se describen las solicitudes que se pueden realizar al sistema, los formatos y modelos de datos soportados por la aplicación, que para este son datos serializados en formato JSON, facilitando su manipulación y almacenamiento en bases de datos no relacionales y estableciendo los recursos y tipos de respuestas que entrega el aplicativo a la capa de vista o Frontend. Teniendo esta descripción, se crean cada uno de los microservicios como proyectos independientes, haciendo uso de Spring Boot. Cada microservicio es construido con normas de calidad y métodos de validación que son propios de la empresa, además de contar con las reglas delimitadas en la documentación de Swagger Editor. Consecutivamente, se utilizan los modelos de datos previamente definidos para construir estructuras de datos en AWS

DynamoDB, que posteriormente son vinculados y configurados para ser utilizadas en el aplicativo. También se configura una base de datos en Firebase Realtime Database para almacenar las imágenes y fotografías de los negocios y las ofertas. Los datos utilizados son simulados, sin embargo, los nombres de las tablas junto con sus campos son definidos haciendo uso de estándares propios del cliente. En la Figura 3 se observa un ejemplo de la especificación de un servicio que permite crear usuarios.

Figura 3. Especificación de un servicio con Swagger Editor

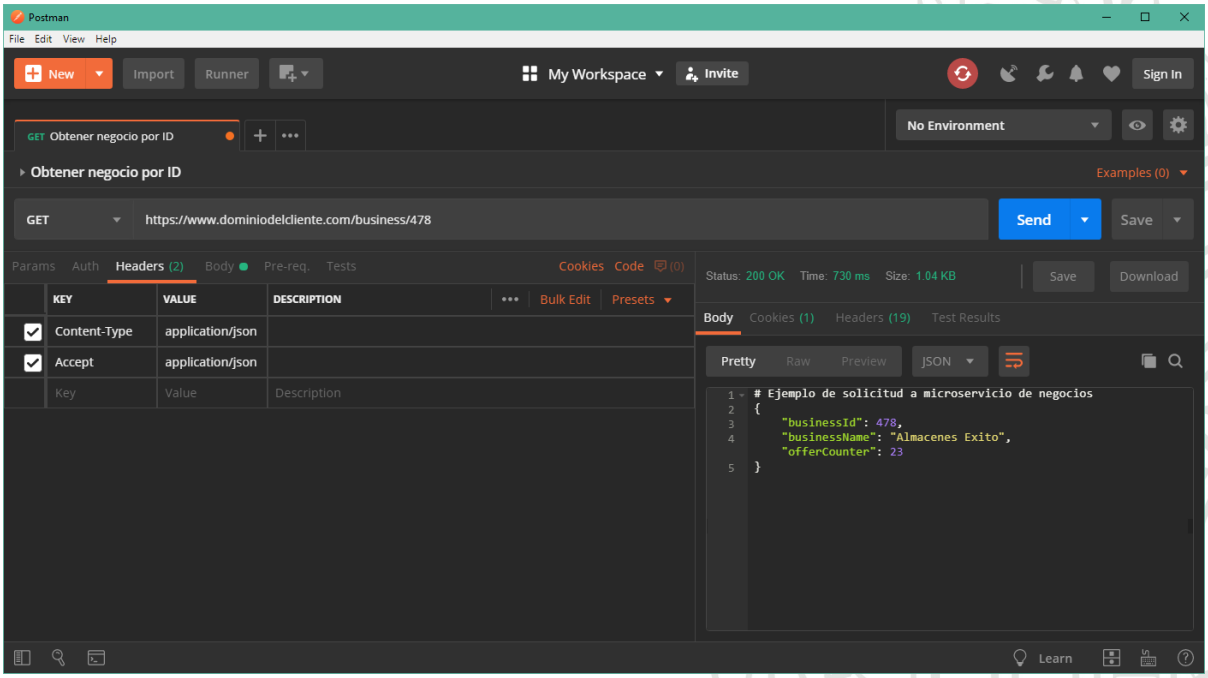


Durante la realización de cada microservicio se observó la necesidad de probar su funcionamiento, pero debido a que la capa de vista de la aplicación se estaba desarrollando a la par de la capa de acceso a datos, se hizo uso del ambiente de desarrollo de pruebas para APIs llamada Postman, el cual permite simular ser el Frontend y permite realizar solicitudes al Backend. En la Figura 4 se observa un ejemplo de solicitud a una versión inicial del microservicio encargado de la administración de la información de los negocios.

A continuación, se orquestó la comunicación del Frontend con el Backend y de los microservicios entre ellos mismos, haciendo uso de la biblioteca de patrones de integración

Apache Camel para construir un puente entre los componentes que requieren comunicarse. Este puente intercepta las solicitudes y respuestas, las reinterpreta en nuevas entidades conocidas como mensajes y las redirige al componente adecuado según su estructura e información.

Figura 4. Solicitud con Postman al microservicio de negocios



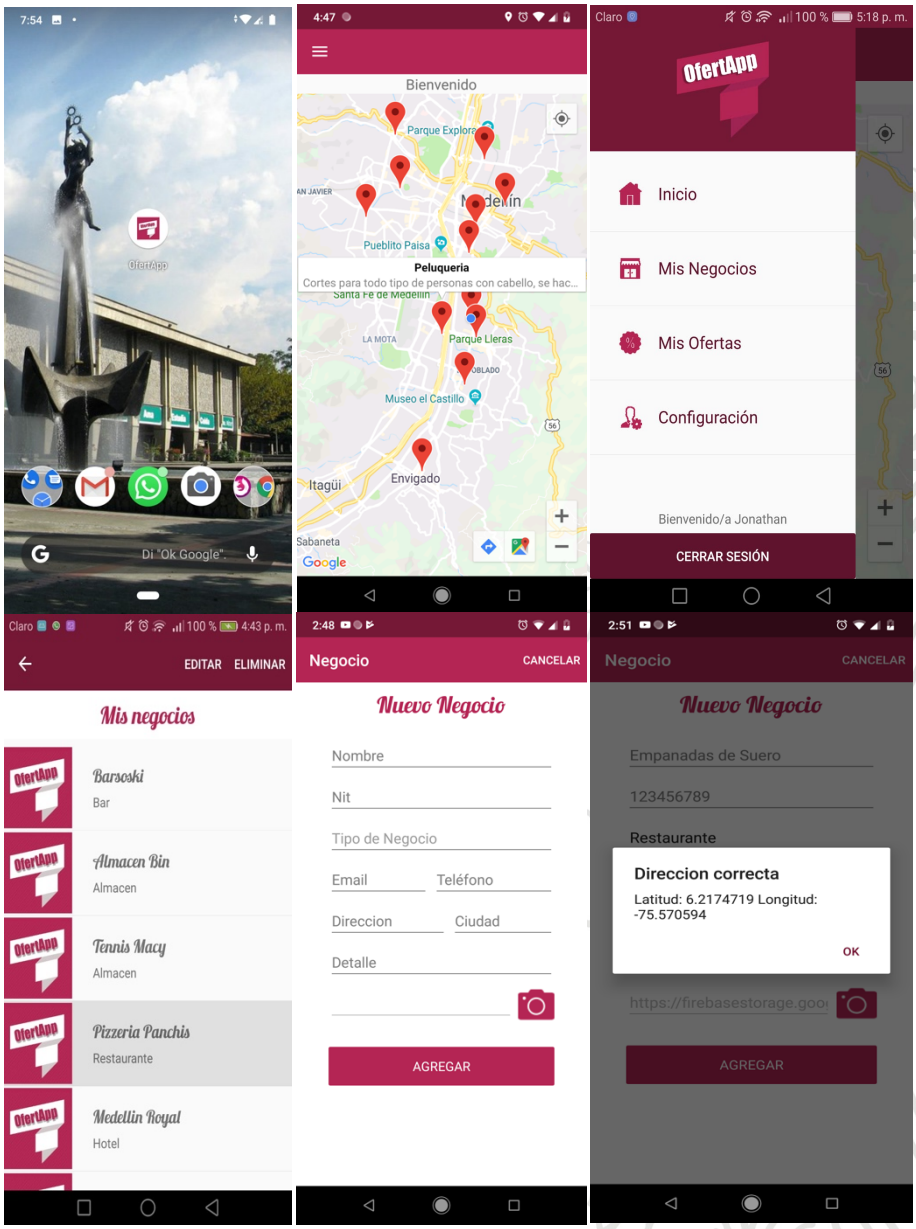
Posteriormente, se procede a construir la versión móvil de la aplicación haciendo uso de Xamarin Forms. Se crean los modelos, vistas y servicios necesarios para consumir los microservicios construidos previamente y se realiza una adaptación adicional en el proyecto para poder utilizar la API de Google Maps en el sistema de geolocalización.

Resultados y análisis

Entre los resultados del proyecto se tiene la aplicación que permite la gestión integral de negocios y ofertas de todo tipo, junto con sus clientes y administradores. En las Figuras 5 y 6 se observa el resultado final en las versiones web y móvil.

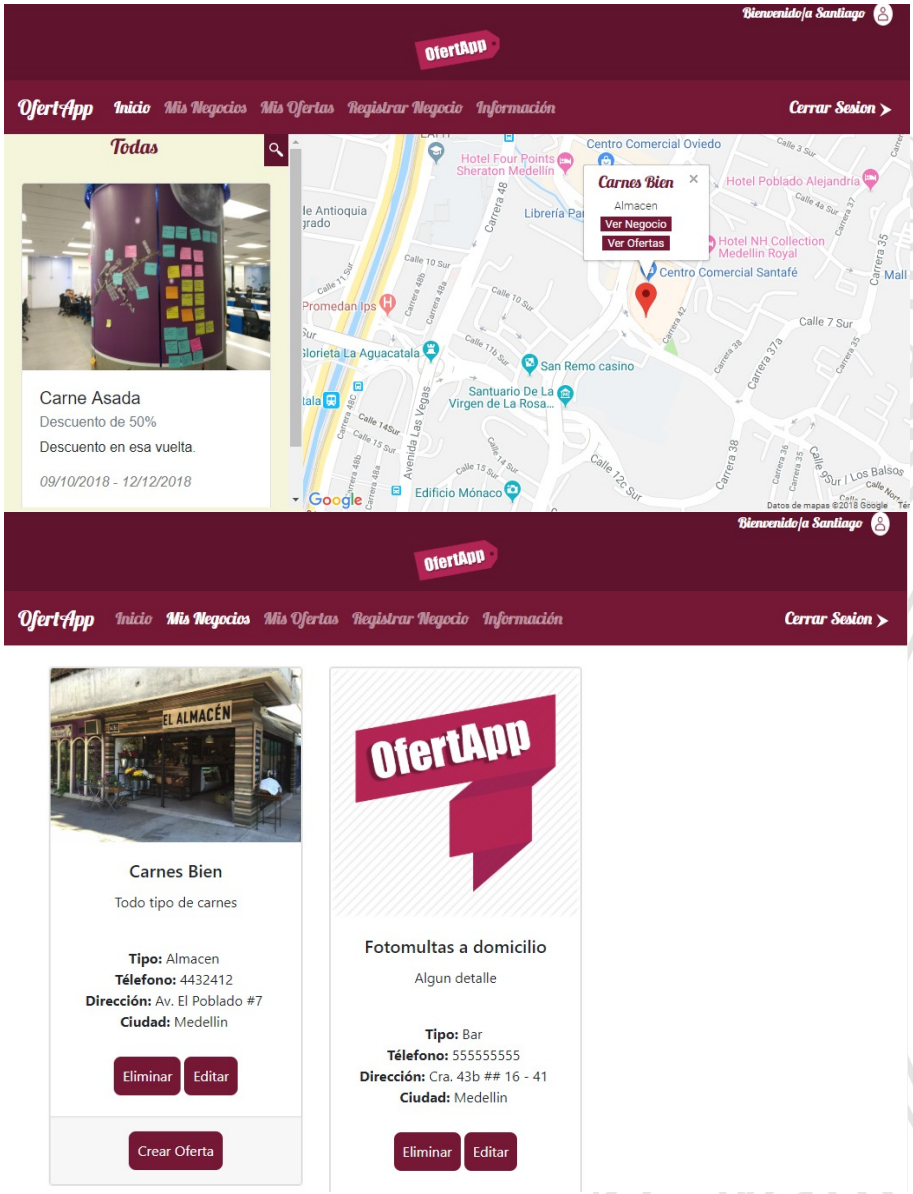
En la Figura 5 se observa respectivamente el lanzador de la aplicación, la geolocalización de los negocios, el menú de opciones para un administrador, el registro y listado de negocios y una prueba de conversión de direcciones físicas a coordenadas de latitud y longitud.

Figura 5. Aplicación en su versión móvil con Xamarin Forms



En la Figura 6 se observa el resultado obtenido en el Frontend, haciendo uso de los microservicios construidos con Spring.

Figura 6. Aplicación en su versión web



La aplicación utiliza una base de datos no relacional ensamblada en AWS DynamoDB, además de estar completamente documentada con Swagger, tal y como fue solicitado por el cliente.

Estos resultados fueron aprobados mediante los resultados obtenidos en las pruebas realizadas sobre cada uno de los microservicios desarrollados, haciendo uso del kit de pruebas unitarias provisto por Spring. Estas pruebas validaron los tipos de respuestas obtenidos por los servicios bajo diferentes solicitudes.

Conclusiones

Finalizando el desarrollo del proyecto, se analizó, diseñó e implementó la capa de acceso de datos para una aplicación multiplataforma con versiones web y móvil, la cual les permita a los clientes buscar ofertas comerciales y negocios de todo tipo, geolocalizarlos y puntuarlos; y a administradores de negocios registrar y gestionar sus establecimientos junto con sus ofertas. La base de datos utilizada fue DynamoDB, tal como fue solicitado por el cliente interno. Se diseñaron, documentaron, implementaron y entregaron APIs Restful interconectadas, basadas en microservicios web, y con sus respectivas especificaciones de recursos, las cuales permiten la administración de la información de usuarios, clientes, negocios y ofertas, validando todas sus operaciones mediante el uso de pruebas unitarias. Además, se construyó una versión móvil utilizando Xamarin para consumir los microservicios expuestos por las diferentes APIs.

El producto final es completamente funcional, esta totalmente documentada con el lenguaje y formato especificado por el cliente y cumple con las normas internas de la empresa en cuanto a criterios de calidad y usabilidad.

Referencias Bibliográficas

[1] Accenture, "Acerca de Accenture", 2019. [En línea]. Disponible en: https://www.accenture.com/co-es/company.

[2] Accenture, "Accenture Technology", 2019. [En línea]. Disponible en: https://www.accenture.com/co-es/technology-index.

[3] Accenture, "Advance Technology Center", 2019. [En línea]. Disponible en: https://www.accenture.com/co-es/service-advanced-technology-center-medellin.

[4] López D. y Maya E. "Arquitectura de software basada en microservicios para desarrollo de aplicaciones web", 2017. [En línea]. Disponible en: http://dspace.redclara.net:8080/bitstream/10786/1277/1/93%20Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf.

[5] Pivotal Software. "Spring Boot - Overview", 2018. Disponible en: http://spring.io/projects/spring-boot.

[6] The Apache Software Foundation, "Apache Camel", 2019. Disponible en: http://camel.apache.org/.

[7] Amazon, "DynamoDB - Servicio de base de datos gestionada NoSQL", 2019. Disponible en: https://aws.amazon.com/es/dynamodb/.

[8] Smartbear, “Swagger editor”, 2018. Disponible en: https://swagger.io/tools/swagger-editor/.

[9] Microsoft, “Entregue aplicaciones nativas para Android, iOS y Windows, con un único código base compartido .NET”, 2018. Disponible en: https://visualstudio.microsoft.com/es/xamarin/.

[10] Microsoft, “Azure DevOps - Plan smarter, collaborate better, and ship faster with a set of modern dev services”, 2019. Disponible en: https://azure.microsoft.com/en-us/services/devops/.

[11] Francia, J. “Scrum Una mejor manera de construir productos”, 2016. Disponible en: https://www.scrum.org/resources/blog/que-es-scrum.