



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE ORIZABA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

REPORTE FINAL DE RESIDENCIA PROFESIONAL

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

NOMBRE DE PROYECTO:

DESARROLLO DE UNA APP MÓVIL HÍBRIDA CON IONIC PARA LA GESTIÓN DE CASAS A TRAVÉS DEL CONSUMO DE SERVICIOS WEB REST Y LA PLATAFORMA GOOGLE FIREBASE

LUGAR DE REALIZACIÓN:

GRUPO MARSSOFT

ASESOR EXTERNO:

M.S.C. JOSÉ FERNANDO CARREÓN DÍAZ DE LEÓN

ASESOR INTERNO:

DR. ULISES JUÁREZ MARTÍNEZ

RESIDENTE:

C. FÉLIX ANTONIO SERNA OLGUÍN

NÚMERO DE CONTROL: 14011127

AGRADECIMIENTOS.

El amor recibido, la dedicación y la paciencia con la que cada día se preocupaban por mi avance y desarrollo a lo largo de mis años como universitario, es simplemente único y se refleja en la vida de un hijo.

Gracias a mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas, gracias a mi madre por estar dispuesta a acompañarme cada larga y agotadora noche de estudio, agotadoras noches en las que su compañía era de mucho aprecio para mí; gracias a mi padre por siempre desear y anhelar siempre lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante mi vida.

Gracias a Dios por la vida de mis padres, también porque cada día bendice mi vida con la hermosa oportunidad de estar y disfrutar al lado de las personas que sé que más me aman, y a las que yo sé que más amo en mi vida, gracias a Dios por permitirme amar a mis padres, gracias a mis padres por permitirme conocer de Dios y de su infinito amor.

Gracias a la vida por este nuevo triunfo, gracias a todas las personas que me apoyaron y creyeron en la realización de este proyecto de residencias profesionales.

Gracias a mi universidad, gracias por haberme permitido formarme y en ella, gracias a todas las personas que fueron partícipes de este proceso, ya sea de manera directa o indirecta, gracias a todos ustedes, profesores y amigos, fueron ustedes los responsables de realizar su pequeño aporte, que el día de hoy se vería reflejado en la culminación de mi paso por la universidad.

Este es un momento muy especial que espero, perdure en el tiempo, no solo en la mente de las personas a quienes agradecí, sino también a quienes invirtieron su tiempo para echarle una mirada a mi proyecto de residencias profesionales; a ellos asimismo les agradezco con todo mi ser.

RESUMEN.

El presente proyecto tiene como objetivo implementar una aplicación móvil para la renta de casas y de administración para las inmobiliarias.

En la cual, dentro de esta aplicación el arrendador podrá publicar casas agregando las especificaciones exactas, además de incluir fotos y videos en las publicaciones correspondientes a dicha propiedad.

El presente proyecto de Ingeniería en Sistemas Computacionales implica varias ramas. La primera de estas ramas es la Programación de Aplicaciones Móviles siendo esta la base principal del proyecto se apoyará en la rama de Programación Web acompañada del Diseño Web y sustentará toda su información en la rama de Base de Datos.

Para la realización de este proyecto es necesario apoyarse en una metodología Scrum...

Las herramientas utilizadas en el desarrollo son de Microsoft y Google, para la programación se utiliza Ionic 4 como Framework para desarrollar aplicaciones híbridas, para la integración con la base de datos se maneja consultas directas con Angular TypeScript el cual permite un uso eficiente de la memoria interna para facilitar el acceso a datos, y se implementó en el motor de base de datos Firebase que es una plataforma muy extensa donde ofrecen servicios como **Google Firebase Realtime Database**, para almacenar datos en la nube, **Google Firebase Cloud Storage**, para guardar imágenes y videos, además de **Google Firebase Cloud Messaging** para la gestión de notificaciones PUSH.

ÍNDICE.

Contenido indizado del reporte.

Toda imagen en el reporte llevará un pie de imagen indizado e incluido en un índice de figuras.

Toda tabla en el reporte llevará un pie de imagen indizado e incluido en un índice de tablas.

AGRADECIMIENTOS.	2
RESUMEN.	3
CAPÍTULO 1. GENERALIDADES DEL PROYECTO	6
1.1 Introducción.	6
1.2 Descripción de la empresa u organización y del puesto o área del trabajo el estudiante.	6
1.3 Problemas a resolver, priorizándolos.	6
1.4 Objetivos.	6
1.4.1 Objetivo General.	6
1.4.2 Objetivos Específicos.	7
1.5 Justificación.	7
CAPÍTULO 2. MARCO TEÓRICO	8
2.1 Metodología de desarrollo	8
2.1.1 Metodología SCRUM.	8
2.2 Herramientas de desarrollo	11
2.2.1 Ionic Framework	11
2.2.1.1 Angular	12
2.2.2 Node.js	13
2.2.2.1 NPM (Node Package Manager)	15
2.2.3 Visual Studio Code	16
2.2.4 Google Firebase	17
2.2.4.1 Google Firebase Realtime Database	18
2.2.4.2 Google Firebase Cloud Storage	20
2.2.4.3 Google Firebase Cloud Messaging	22
CAPÍTULO 3. DESARROLLO	25
3.1 Procedimiento y descripción de las actividades realizadas.	25
CAPÍTULO 4. RESULTADOS	25

4.1 Resultados.....	25
4.2 Actividades Sociales realizadas en la empresa u organización (si es el caso).	25
CAPÍTULO 5. CONCLUSIONES	25
5.1 Conclusiones de Proyecto, recomendaciones y experiencia personal profesional adquirida.....	25
5.2 Competencias desarrolladas y/o aplicadas.....	25
FUENTES DE INFORMACIÓN	26

CAPÍTULO 1. GENERALIDADES DEL PROYECTO

1.1 Introducción.

El proyecto **DESARROLLO DE UNA APP MÓVIL HÍBRIDA CON IONIC PARA LA GESTIÓN DE CASAS A TRAVÉS DEL CONSUMO DE SERVICIOS WEB REST Y LA PLATAFORMA GOOGLE FIREBASE**, intenta cumplir con las expectativas propuestas al momento de planear este proyecto.

Esta implementación que se tiene pensada realizar hará más eficiente el servicio que la empresa tiene planeada ofrecer a los diversos clientes que tiene contemplados.

1.2 Descripción de la empresa u organización y del puesto o área del trabajo el estudiante.

Grupo Marssoft es una consultoría de software conformada por GSI Soluciones y Armesis Software SA especializada en la generación de soluciones creativas para tu negocio o empresa. Con más de 8 años de experiencia y cerca de 15 proyectos finalizados, Marssoft es tu aliada para comenzar ese proyecto que tanto necesitas.

1.3 Problemas a resolver, priorizándolos.

Grupo Marssoft con domicilio en Norte 14 No. 138, Col. Centro en la ciudad Orizaba, Veracruz. Quiere implementar una plataforma donde arrendatarios y arrendadores puedan interactuar en ella como visualizar publicaciones de casas en renta con sus diversas características para que sean descritas en dichas publicaciones

1.4 Objetivos.

1.4.1 Objetivo General.

Desarrollar una aplicación móvil híbrida basada en Ionic que permita la gestión de inmuebles a través de la conexión con servicios web basados en REST de la versión web de la misma plataforma de inmuebles utilizando las funcionalidades adicionales como el envío de notificaciones PUSH que faciliten a los arrendadores y arrendados la interacción a través de una casa en particular, utilizando las plataformas de Google Firebase Realtime Database (GFRD) para la conexión a la base de datos, Google Firebase Cloud Storage (GFCS) para almacenar los recursos de los arrendadores como fotografías y

videos, así como Google Firebase Cloud Messaging (GFCM) para la gestión de notificaciones PUSH.

1.4.2 Objetivos Específicos.

- Analizar y comprender el comportamiento de las herramientas Ionic 4 para el desarrollo de aplicaciones nativas y Google Firebase, como lo son Google Firebase Realtime Database (GFRD) y Google Firebase Cloud Messaging (GFCM).
- Diseñar una interfaz gráfica a través de los componentes nativos que brinda Ionic 4 que cumpla con la navegación intuitiva y fácil de usar.
- Desarrollar la aplicación móvil a través de la implementación del patrón MVVM (Modelo-Vista, Vista-Modelo) en Angular con TypeScript, así como la estructura de la base de datos con Google Firebase Realtime Database (GFRD) y los servicios web basados en REST con PHP.
- Generar un caso de estudio en una inmobiliaria de la ciudad para validar la funcionalidad del producto obtenido.

1.5 Justificación.

Al auge de los teléfonos inteligentes han provocado que muchas herramientas estén disponibles para plataformas como iOS y Android como una forma de no perder un mercado potencial de clientes y garantizar el consumo de sus publicaciones para no quedarse atrás.

Con el desarrollo de una aplicación web enriquecida para la gestión de casas en la modalidad de renta supone un reto dado que los arrendadores y arrendados necesitan estar en un equipo de escritorio para darle seguimiento al índice de búsqueda en las que aparece la casa que está rentando y de los arrendados por buscarlo en motores de búsqueda desde una computadora.

Por lo tanto, se propone el desarrollo de una aplicación híbrida que permita a los arrendadores y arrendados la interacción entre ellos a través de una casa en particular, ofreciendo herramientas de búsqueda basadas en su ubicación actual y aplicando filtros personalizados que garanticen una mejor búsqueda.

Aunado a ellos, y aprovechando las ventajas de una app móvil instalada en los teléfonos inteligentes, se propone una interacción entre la app y el dispositivo a través de notificaciones push vivas que permitan una acción rápida sin abrir la aplicación móvil.

CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se aborda el fundamento teórico que soporta a este proyecto, tales como herramientas y metodología con la cual se llevó a cabo el desarrollo.

2.1 Metodología de desarrollo

2.1.1 Metodología SCRUM.

La metodología Scrum es tendencia en la gestión de proyectos. Si trabajas en un sector en el que el nivel de incertidumbre es alto y tu trabajo ágil, quizás tengas que aplicar Scrum para gestionar tus proyectos.

El sector del desarrollo de software es el principal representante de este tipo de metodología ágil. Se trata de planificar tus proyectos en pequeños bloques o Sprints, e ir revisando y mejorando el anterior. Y es el propio término Scrum proviene del mundo del rugby.

Scrum es un método para trabajar en equipo a partir de iteraciones o Sprints. Así pues, Scrum es una metodología ágil, por lo que su objetivo será controlar y planificar proyectos con un gran volumen de cambios de última hora, en donde la incertidumbre sea elevada.

Se suele planificar por semanas. Al final de cada Sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de esto, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente Sprint.

La metodología Scrum se centra en ajustar sus resultados y responder a las exigencias reales y exactas del cliente. De ahí, que se vaya revisando cada entregable, ya que los requerimientos van variando a corto plazo. El tiempo mínimo para un Sprint es de una semana y el máximo es de cuatro semanas.

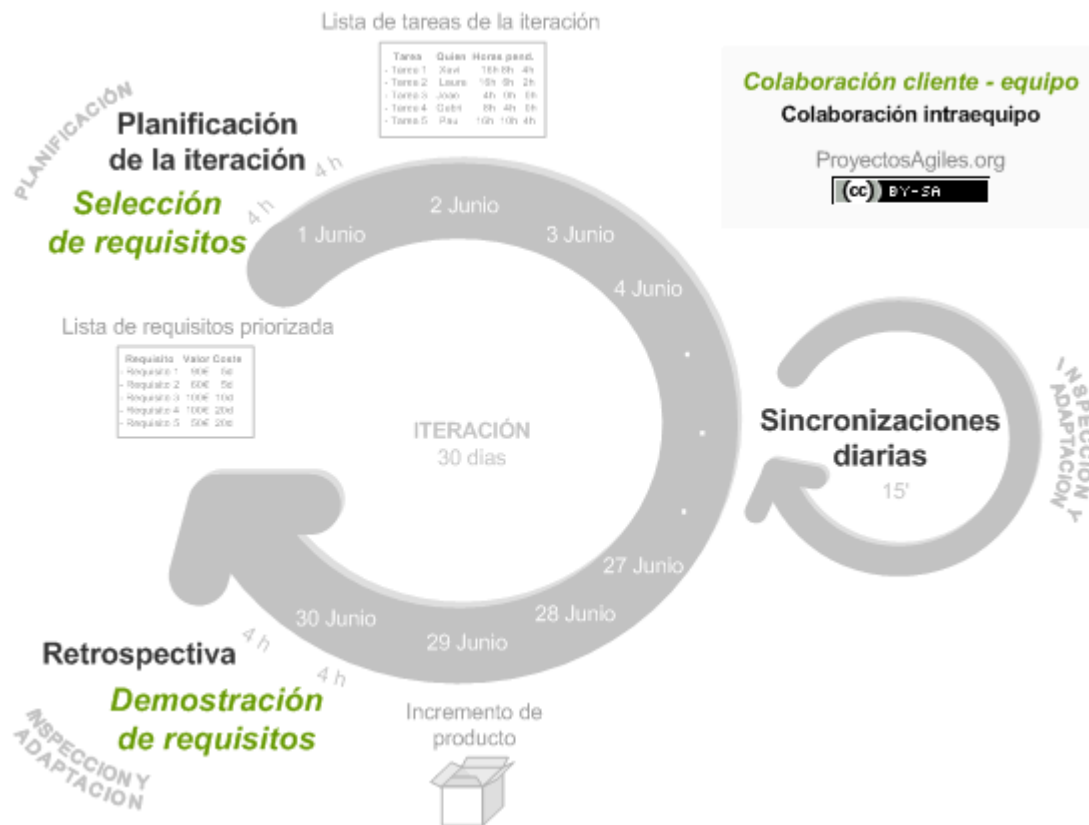
Entre las principales características de la metodología Scrum, destaca que es un desarrollo incremental en lugar de la clásica planificación del desarrollo completo de un producto o servicio. Los equipos Scrum se caracterizan por ser autoorganizados. Y se centra en el producto final, en la calidad del mismo.

Además, en la metodología Scrum se solapan diferentes fases de desarrollo, en lugar de llevar a cabo una planificación secuencial o de cascada.

El proceso

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión).

Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.



El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes (*los tiempos indicados son para iteraciones de 2 semanas*):

Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. **Selección de requisitos** (2 horas). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
2. **Planificación de la iteración** (2 horas). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las

tareas, se autoorganizan para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

Ejecución de la iteración

Cada día el equipo realiza una **reunión de sincronización** (15 minutos), normalmente delante de un tablero físico o pizarra (Scrum Taskboard).

El equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración.

En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo se refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. **Revisión (demostración)** (1,5 horas). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los

cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

2. **Retrospectiva** (1,5 horas). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

2.2 Herramientas de desarrollo

2.2.1 Ionic Framework

Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS.

Historia.

Ionic fue creado por Drifty Co. en 2013. Los productos anteriores de Drifty incluye Codiqa y Jetstrap.

Tomando la retroalimentación de los clientes que intentaron construir aplicaciones móviles, el equipo decidió construir su propio marco que se centre en el rendimiento y en la construcción con los estándares web modernos.

Después de lanzar una versión alfa del marco en noviembre de 2013, una beta 1.0 fue lanzado en marzo de 2014, con una final en 1.0 de mayo de 2015.

Principales características.

1.- Alto rendimiento

La velocidad es importante. Tan importante que sólo se nota cuando no está en tu app. Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

2.- AngularJS & Ionic

Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.

3.- Centro nativo

Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios.

4.- Bonito diseño

Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

5.- Un potente CLI

Con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.

2.2.1.1 Angular

Angular (comúnmente llamado "Angular 2+" o "Angular 2"), es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.

Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

Angular es la evolución de AngularJS aunque incompatible con su predecesor.

Principales características de Angular.

Velocidad y rendimiento

- **Generación de código:** Angular convierte tus plantillas en código altamente optimizado para las máquinas virtuales de JavaScript de hoy en día, ofreciéndote todas las ventajas del código escrito a mano con la productividad de un framework.
- **Universal:** Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS. También abre posibilidades para la optimización del SEO del sitio.

- **División del código:** Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Éste ofrece una división automática de códigos para que los usuarios sólo carguen el código necesario para procesar la vista que solicitan.

Productividad

- **Plantillas:** Permite crear rápidamente vistas de interfaz de usuario con una sintaxis de plantilla simple y potente.
- **Angular CLI:** Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como previsualizar de forma instantánea la aplicación.
- **IDEs:** Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs.

Historia completa del desarrollo

- **Testing:** Utiliza Karma para realizar pruebas unitarias, y Protractor para realizar pruebas end-to-end de forma rápida y estable.
- **Animación:** Permite crear animaciones complejas y de alto rendimiento con muy poco código a través de la intuitiva API de Angular.
- **Accesibilidad:** Posee características para crear aplicaciones accesibles con los componentes disponibles para ARIA.

2.2.2 Node.js

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.³ Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.⁴⁵

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y Java EE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket.

Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS.6 Node.js incluye un entorno REPL para depuración interactiva.

Aspectos técnicos.

Concurrencia

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto.

Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback.

Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como cluster8 para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta.

V8

V8 es el entorno de ejecución para JavaScript creado para Google Chrome. Es software libre desde 2008, está escrito en C++ y compila el código fuente JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

Node.js contiene libuv para manejar eventos asíncronos. Libuv es una capa de abstracción de funcionalidades de redes y sistemas de archivo en sistemas Windows y sistemas basados en POSIX como Linux, Mac OS X y Unix.

El cuerpo de operaciones de base de Node.js está escrito en JavaScript con métodos de soporte escritos en C++.

Módulos

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y otros módulos fundamentales, como por ejemplo Path, FileSystem, Buffer, Timers y el de propósito más general Stream. Es posible utilizar módulos desarrollados por terceros, ya sea como archivos ".node" precompilados, o como archivos en javascript plano.

Los módulos Javascript se implementan siguiendo la especificación CommonJS para módulos, utilizando una variable de exportación para dar a estos scripts acceso a funciones y variables implementadas por los módulos.

Los módulos de terceros pueden extender node.js o añadir un nivel de abstracción, implementando varias utilidades middleware para utilizar en aplicaciones web, como por ejemplo los frameworks connect y express. Pese a que los módulos pueden instalarse como archivos simples, normalmente se instalan utilizando el Node Package Manager (npm) que nos facilitará la compilación, instalación y actualización de módulos, así como la gestión de las dependencias.

Además, los módulos que no se instalen en el directorio por defecto de módulos de Node necesitarán la utilización de una ruta relativa para poder encontrarlos. El wiki Node.js proporciona una lista de varios de los módulos de terceros disponibles.

Desarrollo homogéneo entre cliente y servidor

Node.js puede ser combinado con una base de datos documental (por ejemplo, MongoDB o CouchDB) y JSON lo que permite desarrollar en un entorno de desarrollo JavaScript unificado. Con la adaptación de los patrones para desarrollo del lado del servidor tales como MVC y sus variantes MVP, MVVM, etc.

Node.js facilita la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor.

Bucle de eventos

Node.js se registra con el sistema operativo y cada vez que un cliente establece una conexión se ejecuta un callback. Dentro del entorno de ejecución de Node.js, cada conexión recibe una pequeña asignación de espacio de memoria dinámico, sin tener que crear un hilo de ejecución.

A diferencia de otros servidores dirigidos por eventos, el bucle de gestión de eventos de Node.js no es llamado explícitamente, sino que se activa al final de cada ejecución de una función callback. El bucle de gestión de eventos se termina cuando ya no quedan eventos por atender.

2.2.2.1 NPM (Node Package Manager)

NPM es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

Desde la versión 0.6.3 de Node.js npm es instalado automáticamente con el entorno. npm se ejecuta desde la línea de comandos y maneja las dependencias para una aplicación.

Además, permite a los usuarios instalar aplicaciones Node.js que se encuentran en el repositorio. npm está escrito enteramente en JavaScript y fue desarrollado por Isaac Z. Schlueter a raíz de la frustración que experimentó mientras trabajaba con CommonJS y

considerando las deficiencias de otros proyectos similares como PHP (PEAR) y Perl (CPAN).

En noviembre 2014, Collin Winter escribió en GitHub que "npm" no es una abreviatura para "Node Package Manager", sino un retroacrónimo recursivo para "npm is not an acronym". Aun así, en el archivo README.md encontrado en npm-0.0.1 se puede leer en la primera línea: "The Node Package Manager".

2.2.3 Visual Studio Code

Visual Studio Code. Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código.

También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

Historia.

Visual Studio Code fue anunciado el 29 de abril de 2015 por Microsoft en la conferencia Build de 2015. Una versión preliminar fue lanzada poco después. El 18 de noviembre de 2015, Visual Studio Code fue lanzado bajo la licencia MIT y su código fuente fue publicado en GitHub.

También se anunció el soporte de extensión. El 14 de abril de 2016, Visual Studio Code graduó la etapa de vista previa pública y se lanzó a la web.

Características.

El código combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code, cuenta con herramientas de Debug hasta opciones para actualización en tiempo real de nuestro código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran (por ejemplo, en el caso de SASS a CSS).

Además de las extensiones, tendremos la posibilidad de optar por otros themes o bien configurarlo a nuestro gusto. Para modificar el esquema de colores y los iconos.

Ventajas.

- Se puede utilizar como lenguajes de programación.

- Visual Studio Code es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar los principales del desarrollo Web: HTML, CSS, y JavaScript, entre otros.
- Posibilidad de configurar la vista a nuestro gusto. De esta forma, podremos tener más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal o un detalle de problemas, entre otras posibilidades.

2.2.4 Google Firebase

Firebase se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades.

Firebase se inició cuando Google la compró en 2014, y seguidamente la fue mejorando mediante la compra del equipo de Divshot.

Principales características de Firebase.

- **Desarrollo:** Firebase permite la creación de mejores aplicaciones, minimizando el tiempo de optimización y desarrollo, mediante diferentes funciones, entre las que destacan la detección de errores y de testeo, que supone poder dar un salto de calidad a la app. Poder almacenar todo en la nube, testear la aplicación o poder configurarla de manera remota, son características destacables de la plataforma.
- **Analítica:** Tener un control máximo del rendimiento de la app mediante métricas analíticas, todo desde un único panel y de forma gratuita, es una de las ventajas que ofrece Firebase respecto a la analítica web. Los datos analíticos que facilita Firebase, facilita la toma de decisiones basadas y fundamentadas en datos reales.
- **Poder de crecimiento:** Permite gestionar de manera fácil todos los usuarios de las aplicaciones, con el añadido de que se pueden captar nuevos usuarios, mediante invitaciones o notificaciones.
- **Monetización:** Mediante AdMob, Firebase permite que puedas ganar dinero.
- **Rapidez:** Implementar Firebase puede ser fácil y rápido, gracias a su API que es muy intuitiva, sostenida en un solo SDK. Con Firebase puedes centrar tus esfuerzos en resolver los problemas de tus clientes y así poder evitar la pérdida de tiempo en la creación de una infraestructura compleja.

- **Agilidad:** Firebase ofrece aplicaciones multiplataforma con una APIs integradas a SDK individuales para iOS, Android y Javascript, de tal forma que se puede gestionar diferentes aplicaciones sin necesidad de salir de la propia plataforma.

2.2.4.1 Google Firebase Realtime Database

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando compilas apps multiplataforma con nuestros SDK de iOS, Android y JavaScript, todos tus clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

Funciones clave

Tiempo real	En lugar de solicitudes HTTP típicas, Firebase Realtime Database usa la sincronización de datos (cada vez que cambian los datos, los dispositivos conectados reciben esa actualización en milisegundos). Proporciona experiencias colaborativas y envolventes sin pensar en el código de red.
Sin conexión	Las aplicaciones de Firebase continúan respondiendo, incluso sin conexión, dado que el SDK de Firebase Realtime Database hace que tus datos persistan en el disco. Cuando se restablece la conexión, el dispositivo cliente recibe los cambios que faltaban y los sincroniza con el estado actual del servidor.
Acceso desde dispositivos cliente	Se puede acceder a Firebase Realtime Database directamente desde un dispositivo móvil o un navegador web; no se necesita un servidor de aplicaciones. La seguridad y la validación de datos están disponibles a través de las reglas de seguridad de Firebase Realtime Database: reglas basadas en expresiones que se ejecutan cuando se leen o se escriben datos.
Escalamiento en varias bases de datos	Se puede acceder a Firebase Realtime Database directamente desde un dispositivo móvil o un navegador web; no se necesita un servidor de aplicaciones. La seguridad y la validación de datos están disponibles a través de las reglas de seguridad de Firebase Realtime Database: reglas basadas en expresiones que se ejecutan cuando se leen o se escriben datos.

¿Cómo funciona?

Con Firebase Realtime Database, puedes compilar aplicaciones ricas y colaborativas, ya que permite el acceso seguro a la base de datos directamente desde el código del cliente. Los datos persisten de forma local.

Además, incluso cuando no hay conexión, se siguen activando los eventos en tiempo real, lo que proporciona una experiencia adaptable al usuario final. Cuando el dispositivo vuelve a conectarse, Realtime Database sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, lo que combina los conflictos de forma automática.

Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones, llamado reglas de seguridad de Firebase Realtime Database, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir.

Integrar Firebase Authentication permite que los desarrolladores definan quién tiene acceso a qué datos y cómo acceden a ellos.

Realtime Database es una base de datos NoSQL y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente.

Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta. Es importante pensar cómo deben acceder a los datos los usuarios y estructurarlos según corresponda.



2.2.4.2 Google Firebase Cloud Storage

Cloud Storage se creó para programadores de apps que necesitan almacenar y publicar contenido generado por usuarios, como fotos o videos.

Cloud Storage para Firebase es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google.

Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para tus apps de Firebase, sin importar la calidad de la red.

Puedes usar nuestros SDK para almacenar imágenes, audio, video y otros tipos de contenido generado por el usuario. En el servidor, puedes usar Google Cloud Storage para acceder a los mismos archivos.

Funciones clave

Operaciones robustas

Los SDK de Firebase para Cloud Storage realizan las operaciones de carga y descarga sin importar la calidad de la red. Las cargas y descargas son robustas, lo que significa que se reinician en el punto en el que se interrumpieron para así ahorrar

	tiempo y ancho de banda a los usuarios.
Seguridad sólida	Los SDK de Firebase para Cloud Storage se integran con Firebase Authentication a fin de brindar autenticación intuitiva y sencilla para los programadores. Puedes usar nuestro modelo de seguridad declarativa para permitir el acceso según el nombre de archivo, el tamaño, el tipo de contenido y otros metadatos.
Gran escalabilidad	Cloud Storage para Firebase está diseñado para escalar a exabytes si tu app se vuelve viral. Pasa fácilmente de la fase prototipo a la de producción con la misma infraestructura que respalda a Spotify y Google Fotos.

¿Cómo funciona?

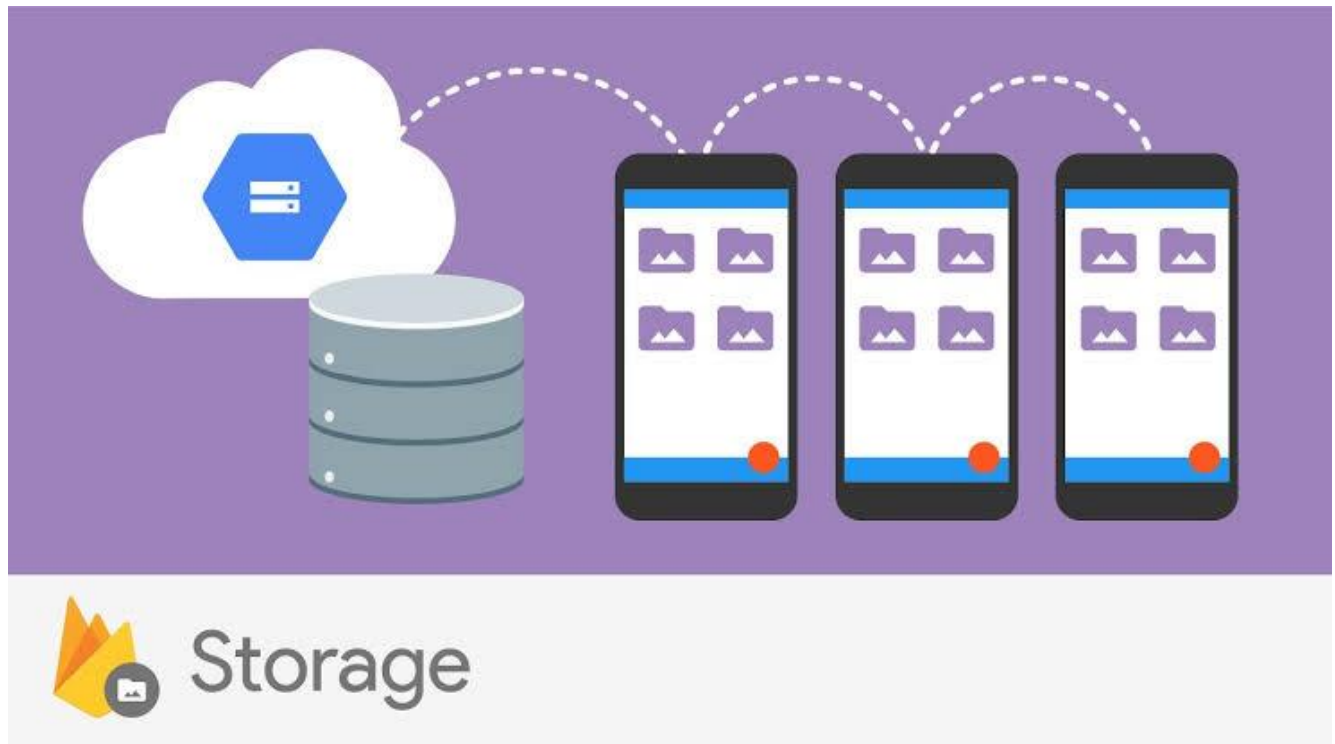
Los programadores usan los SDK de Firebase para Cloud Storage para subir y descargar archivos directamente de los clientes. Si la conexión a la red es deficiente, el cliente puede reintentar la operación donde la dejó de inmediato, lo cual les ahorra tiempo y ancho de banda a los usuarios.

Cloud Storage almacena tus archivos en un depósito de Google Cloud Storage y los hace accesibles a través de Firebase y Google Cloud.

Esto te permite tener la flexibilidad para subir y descargar archivos de clientes móviles a través de los SDK de Firebase y realizar procesamiento en el servidor, como el filtrado de imágenes o la transcodificación de video mediante Google Cloud Platform.

Cloud Storage se escala automáticamente, lo que significa que no es necesario migrar a ningún otro proveedor. Obtén más información acerca de todos los beneficios de nuestra integración con Google Cloud Platform.

Los SDK de Firebase para Cloud Storage se integran perfectamente en Firebase Authentication a fin de identificar a los usuarios. Además, ofrecemos un lenguaje de seguridad declarativo que te permite configurar controles de acceso para archivos individuales o grupos de archivos, de manera que puedas hacer que los archivos sean públicos o privados según lo que desees.



2.2.4.3 Google Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que te permite enviar mensajes de forma segura y gratuita.

Con FCM, puedes notificar a una app cliente que un correo electrónico nuevo o que otros datos están disponibles para la sincronización.

Puedes enviar mensajes de notificación para volver a atraer a más usuarios y aumentar su retención.

Para los casos prácticos de mensajería instantánea, un mensaje puede transferir una carga de hasta 4 KB a una app cliente.

Funciones clave

Envía mensajes de notificación o mensajes de datos

Envía mensajes de notificación que se muestran a tu usuario. También puedes enviar mensajes de datos y determinar completamente lo que ocurre en el código de tu aplicación.

Orientación versátil

Distribuye mensajes a tu app cliente en cualquiera de las siguientes tres formas: a dispositivos individuales, a grupos de dispositivos o a dispositivos suscritos a

Envía mensajes desde apps cliente

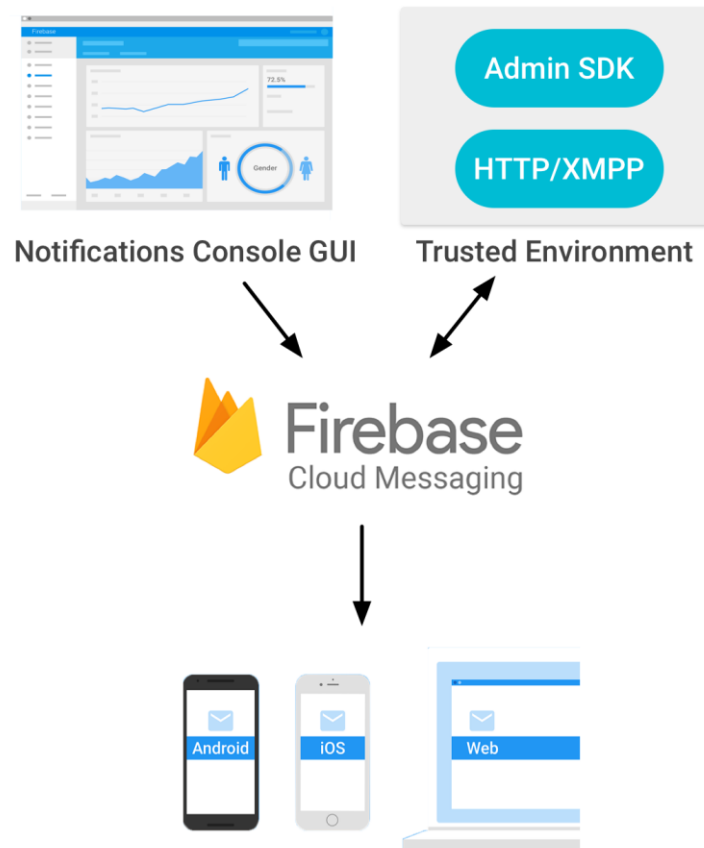
temas.

Envía mensajes de confirmación, de chat y de otros tipos desde los dispositivos a tu servidor a través del canal de conexión confiable de FCM que consume poca batería.

¿Cómo funciona?

Una implementación de FCM incluye dos componentes principales para enviar y recibir datos:

1. Un entorno de confianza como Cloud Functions para Firebase o un servidor de apps para generar, orientar y enviar mensajes
2. Una app cliente de iOS, Android o Web (JavaScript) que reciba mensajes





CAPÍTULO 3. DESARROLLO

3.1 Procedimiento y descripción de las actividades realizadas.

Descripción de las actividades de acuerdo a la metodología utilizada. Diagramas UML, etc.

CAPÍTULO 4. RESULTADOS

4.1 Resultados.

Cuál fue el producto resultante, pantallas con descripción, códigos relevantes, imágenes, evidencias de su producto terminado y cómo cumple con los objetivos planteados y requerimientos.

4.2 Actividades Sociales realizadas en la empresa u organización (si es el caso).

Brigadas, campañas, eventos. Solo en caso de haber participado en ellos.

CAPÍTULO 5. CONCLUSIONES

5.1 Conclusiones de Proyecto, recomendaciones y experiencia personal profesional adquirida.

5.2 Competencias desarrolladas y/o aplicadas.

Seleccionar de la siguiente lista, o pueden incluir alguna otra, especificando porque creen haberla adquirido o aplicado, en qué momento del proceso, etc.

Competencias Instrumentales

- Capacidad de análisis y síntesis.
- Capacidad de organizar y planificar.
- Conocimientos generales básicos.
- Conocimientos básicos de la carrera.
- Comunicación oral y escrita en su propia lengua.
- Conocimiento de una segunda lengua.
- Habilidades básicas de manejo de la computadora.
- Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas.
- Solución de problemas.
- Toma de decisiones.

Competencias Interpersonales

- Capacidad crítica y autocrítica.
- Trabajo en equipo.
- Habilidades interpersonales.
- Capacidad de trabajar en equipo interdisciplinario.
- Capacidad de comunicarse con profesionales de otras áreas.
- Apreciación de la diversidad y multiculturalidad.

- Habilidad para trabajar en un ambiente laboral.
- Compromiso ético.

Competencias Sistémicas

- Capacidad de aplicar los conocimientos en la práctica.
- Habilidades de investigación.
- Capacidad de aprender.
- Capacidad de adaptarse a nuevas situaciones.
- Capacidad de generar nuevas ideas (creatividad).
- Liderazgo.
- Conocimiento de culturas y costumbres de otros países.
- Habilidad para trabajar en forma autónoma.
- Capacidad para diseñar y gestionar proyectos.
- Iniciativa y espíritu emprendedor.
- Preocupación por la calidad.
- Búsqueda del logro.

FUENTES DE INFORMACIÓN

Electrónicas y bibliográficas, en formato APA.

Qué es SCRUM. (2018, 9 octubre). Recuperado 7 noviembre, 2019, de <https://proyectosagiles.org/que-es-scrum/>

EcuRed. (s.f.). Ionic - EcuRed. Recuperado 10 noviembre, 2019, de <https://www.ecured.cu/Ionic>

Colaboradores de Wikipedia. (2019, 5 noviembre). Angular (framework). Recuperado 10 noviembre, 2019, de [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

Colaboradores de Wikipedia. (2019a, 5 noviembre). Angular (framework). Recuperado 10 noviembre, 2019, de [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

Colaboradores de Wikipedia. (2019d, 6 noviembre). Npm. Recuperado 10 noviembre, 2019, de <https://es.wikipedia.org/wiki/Npm>

EcuRed. (s.f.-b). Visual Studio Code - EcuRed. Recuperado 10 noviembre, 2019, de http://www.ecured.cu/Visual_Studio_Code

Manuel Pérez Cardona, S. E. O. (2018, 4 octubre). Firebase, qué es y para qué sirve la plataforma de Google. Recuperado 10 noviembre, 2019, de <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>

Firebase Realtime Database | Firebase Realtime Database. (s.f.). Recuperado 12 noviembre, 2019, de <https://firebase.google.com/docs/database?hl=es>

Cloud Storage | Firebase. (s.f.). Recuperado 12 noviembre, 2019, de <https://firebase.google.com/docs/storage/?hl=es-419>

ANEXOS (Opcional)

Anexos (carta de autorización por parte de la empresa u organización para la titulación y otros si son necesarios).

Registros de Productos (patentes, derechos de autor, compra-venta del proyecto, etc.).