FACULTAD DE INGENIERIA

Asignatura: Desarrollo de Software para Móviles (DSM03L)

CICLO ACADEMICO: 01-2021

Título:

"Primer avance, proyecto de Catedra DSM"

Docente:

Ing. Alexander Alberto Sigüenza Campos

Presentado por:

Apellidos, Nombres	Carné
Valle Serrano, Oswaldo Alberto	vs161940
Pleitez Hércules, Kevin Eliu	Ph161929
Roberto Fernando Funes Morán FM180247	FM180247

Soyapango, 5 de marzo de 2021

Índice

Contenido

Objetivos

Objetivo General:

- Desarrollar una aplicación móvil de citas médicas para pacientes con pocas posibilidades de viajar.

Objetivos Específicos:

- Implementar un chat real con el uso de la API Strapi.
- Aplicar MongoDB para el almacenamiento remoto de la información.
- Realizar diferentes pruebas de funcionamiento para garantizar su optimización y buen uso.

Introducción

El desarrollo de aplicaciones móviles o softwares de computadora posee diferentes ambientaciones, como lo es la financiera, económica, administrativa, social, etc. Para el caso de las aplicaciones de ambientación social, son sistemas que tienen un fin social, que puede ser lucrativo o no, este dependiendo de la decisión de los desarrolladores o de la empresa que los contrato. Un ejemplo puede ser el uso de un sistema para calcular la calidad del aire en cierta zona, un sistema de riego computarizado para una siembra agrícola, etc. Las aplicaciones son muy variadas, para este caso será una aplicación hacia las personas que desean recibir una consulta médica pero que no tienen los medios para desplazarse por motivos de salud a causa de la pandemia mundial por el COVID-19. Ya que dicha enfermedad ha hecho que muchos pacientes no estén en contacto con sus médicos y ni puedan realizar sus citas médicas presencialmente. Por lo que el uso de una aplicación de mensajería privado ayudara a mantener a dichos pacientes comunicados con sus doctores.

La aplicación consta del uso de mensajería y de roles, donde los doctores y los pacientes son los roles principales, los pacientes podrán pedir una cita en una fecha determinada y con un doctor especifico, aparte de que este también disponible; dicho doctor podrá aceptar la cita y empezar a tener contacto con el paciente, organizarse para realizar una video llamada en cualquier plataforma, responder dudas, compartir recetas médicas, etc. Con ello los pacientes no tendrán que correr el riesgo de viajar hacia el hospital y de contagiarse de COVID-19.

Diseños UX/UI

Imagen 1.0: Prototipo de Login

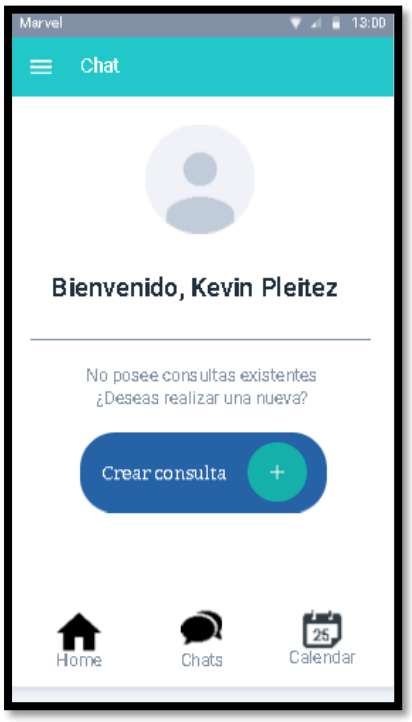


Imagen 1.1: Diseño de interfaz principal

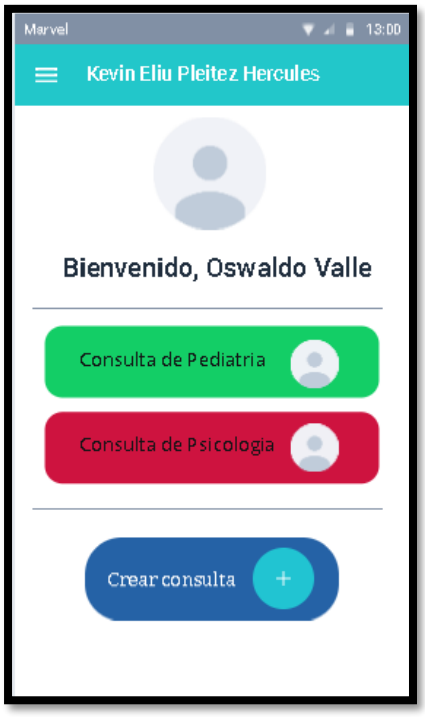


Imagen 1.2: Prototipo de interfaz con consultas creadas

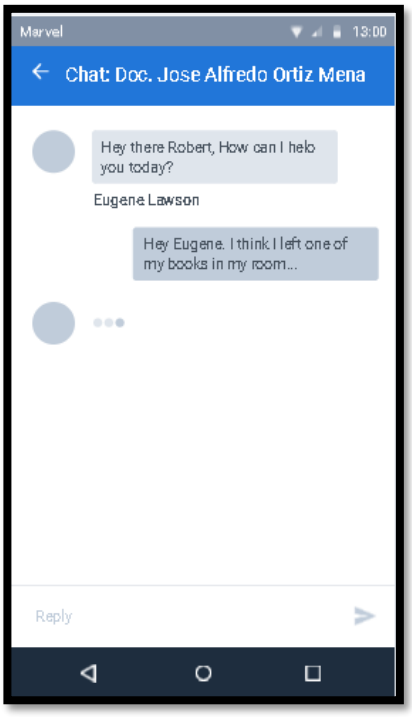


Imagen 1.3: Prototipo de vista del chat

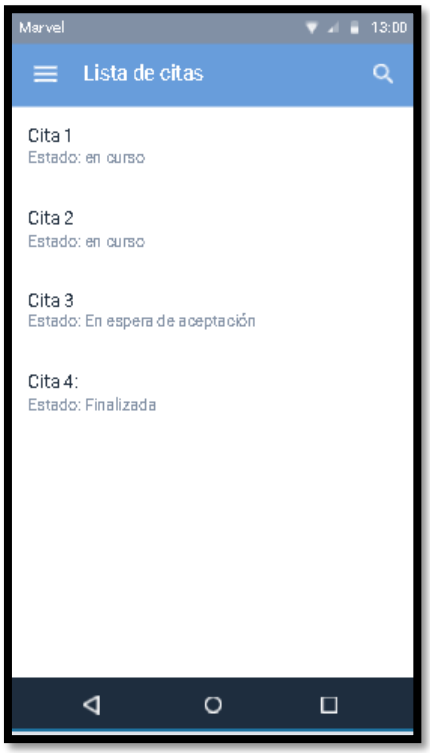


Imagen 1.4: Vista de ordenamiento de citas citas.

Imagen 1.5: Diseño de formulario de creación de

Lógica de solución del problema

Comenzando con el desarrollo de la aplicación, será por medio de React Native, un framework de código abierto ambientado en JavaScript y Java, para desarrollar aplicaciones móviles para Android y iOS, con ello se programarán las distintas funcionalidades de la aplicación.

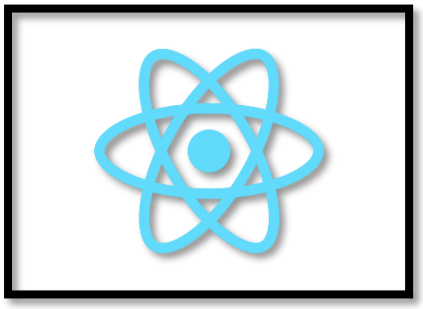
Para el tratamiento y almacenamiento de la información será por medio de la API Back End STRAPI, que guardará la información en formato Json, y así mandarla a una Base de Datos en MongoDB, por medio de un Clúster que brinda la página en conjunto con AWS.

Primero la aplicación mandara los datos obtenidos a Strapi, que el mismo puede crear tablas y relacionarlas, así nos facilitaría el hecho de crear de cero la base de datos y empezar a crear las consultas y hacer código desde cero. Luego manda los datos a MongoDB para ser almacenados en la nube a la hora de que la aplicación esté funcionando de manera global.

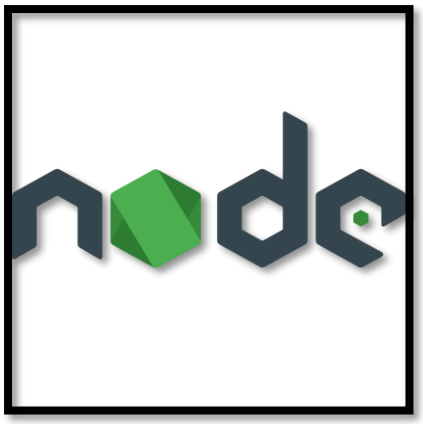
Así la aplicación tendrá acceso a la información dada por Strapi y Strapi a la información que guardo a la base de datos de MongoDB

Android Studio servirá para simular el funcionamiento de la aplicación únicamente, ya que se utilizará React Native como ambiente de programación.

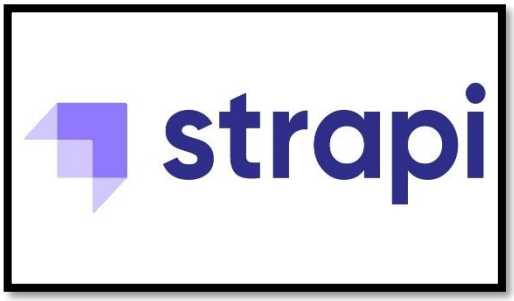
Herramientas a utilizar durante el desarrollo



React Native: React Native es un marco de aplicación móvil de código abierto creado por Facebook, Inc. Se utiliza para desarrollar aplicaciones para Android, Android TV, iOS, macOS, tvOS, Web, Windows y UWP al permitir a los desarrolladores utilizar el marco de React junto con la plataforma nativa.



Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Servirá para la instalación de componentes de React Native, el propio React Native y STRAPI.



CMS STRAPI: Strapi es un CMS sin encabezado de código abierto basado en Node.js para todas las necesidades de API y gestión de contenidos de los desarrolladores. Permite crear API funcionales en cuestión de minutos y desarrollar software sin la molestia de las complejidades de un CMS. Todos los datos están disponibles a través de una API 100 % personalizable para adaptarse a cualquier necesidad. Se encarga del todo el backend.

Presupuesto de costo del proyecto

Para el cálculo de los costos del proyecto se hará uso del modelo basado en puntos de casos de uso.

Primero calcularemos el peso de los actores no ajustado, donde los actores son los que estaran usando el sistema, en este caso son 3:

- Administrador.
- Doctor.
- Paciente.

Ahora aplicaremos la fórmula del peso de los actores no ajustado.

$$UAW = \sum_{i=1}^{n-actores}(cantidadtipoactor_i * peso)$$

El cálculo seria el siguiente:

$$UAW = (3 * 3) = 15$$

Luego necesitamos el peso de los casos de uso que tendrá el sistema, basado en la siguiente tabla:

Tipo de caso de uso	Descripción	Peso
Simple	El caso de uso contiene menos de 3 transacciones	5
Medio	El caso de uso contiene de 4 a 7 transacciones	10
Complejo	El caso de uso tiene más de 7 transacciones	15

En este caso el sistema tendría un tipo de caso de uso “complejo” por tener más de 7 transacciones, por lo que el peso es de 15.

Ahora calcularemos el peso de los casos de estudio con la siguiente formula, tomando en cuenta que la aplicación tiene 6 casos de estudio.

$$UUCW = \sum(tipocasodeuso_i * peso)$$

$$UUCW = (4 * 10) = 40$$

Ahora calcularemos los puntos de caso de estudio:

$$UUCP = AUW + UUCW$$

$$UUCP = 15 + 40 = 55$$

Ahora armaremos la tabla de Factores técnicos, basándonos en esta tabla:

DESCRIPCION	PESO
Sistema Distribuido	2
Objetivos de Rendimiento o Tiempos de respuesta	1
Eficiencia del usuario final	1
Procesamiento interno complejo	1
Código debe ser reutilizable	1
Facilidad de instalación	0.5
Facilidad de uso	0.5
Portabilidad	2
Facilidad de cambio	1
Concurrencia	1
Incluye objetivos especiales de seguridad	1
Provee acceso directo a terceras partes	1
Se requiere facilidades especiales de entrenamiento a usuarios	1

Armamos nuestra tabla para sumar cada valor obtenido en cada facto, el valor de cada factor lo pondremos del 0 al 5:

Descripción	Peso	Valor	Resultado
Sistema distribuido	2	2	4
Tiempo de respuesta	1	5	5
Eficiencia de usuario final	1	1	1
Procesamiento interno complejo	1	3	3
Codigo Reutilizable	1	1	1
Facilidad de instalación	0.5	1	0.5
Facilidad de uso	0.5	2	1
Portabilidad	2	1	2
Facilidad de cambio	1	3	3
Concurrencia	1	5	5
Objetivos de seguridad	1	4	4
Acceso a terceras partes	1	3	3
Facilidad de entrenamiento a usuario	1	0	0
Total:			32.5

Ahora usaremos la fórmula:

$$TCF = 0.6 + (0.01 * TFactor)$$

$$TFactor = 0.6 + (0.01 * 32.5) = 0.925$$

Ahora calcularemos los puntos de Factores de Ambiente, al igual que los puntos de factores, este tendrá la siguiente tabla:

DESCRIPCION	PESO
Familiaridad con el modelo de proyecto utilizado	1.5
Experiencia en la aplicación	0.5
Experiencia en orientación a objetos	1
Capacidad del analista líder	0.5
Motivación	1
Estabilidad de los requerimientos	2
Personal a tiempo parcial	-1
Dificultad del lenguaje de programación	-1

Y sumaremos cada uno de los factores que multipliquemos con su valor de 0 a 5 que asignemos:

Descripción	Peso	Valor	Resultado
Familiaridad con el proyecto	1.5	1	1.5
Experiencia en la aplicación	0.5	0	0
Experiencia en orientación a objetos	1	4	4
Capacidad del analista líder	0.5	4	2
Motivación	1	5	5
Estabilidad de los requerimientos	2	3	6
Personal a tiempo parcial	-1	2	-2
Dificultad del lenguaje de programación	-1	4	-4
Total:			12.5

$$EF = 1.4 + (-0.03 * 12.5) = 1.025$$

Ahora se calcularán los puntos de caso de uso:

$$\boldsymbol{UCP = UUCP \times TCF \times EF = 33 \times 0.905 \times 0.95}$$

$$UCP = 55 * 0.925 * 1.025 = 52.15$$

$$Horas - empleados = UCP * 20 = 99.55 * 20 = 1042.92\ horas$$

Cada empleado en un plazo de 1043 horas se acabará el proyecto, en 8 horas al día, alrededor de 4 meses de trabajo.

Basándonos en el precio de cada lenguaje de programación según esta tabla:

Tabla 12. Salario por lenguaje de programación (Galván, 2014).

Lenguaje	Muestra	Mediana	Media	Desviación estándar
Objetive C	21	$33,000	$40,141	$29,611
C	52	$25,725	$24,989	$13,762
Java	321	$25,000	$26,227	$16,342
VB	103	$25,000	$23,922	$11,809
C#	293	$24,000	$24,637	$13,549
Node JS	30	$23,970	$28,952	$20,037
PL/SQL	226	$23,450	$25,379	$16,436
Ruby	58	$23,250	$26,796	$19,870
Javascript	429	$22,700	$24,463	$14,372
Bash	47	$21,500	$24,959	$11,466
Python	40	$20,000	$23,910	$19,481
PHP	167	$18,000	$20,074	$14,016
Delphi	18	$18,000	$20,028	$9,681

Java y JS tienen un valor de $58 hora, por lo que el costo por empleado es de:

= 1043 * 58 = $60, 494 en todo el proyecto * 2 trabajadores = $ 120,988 en total para 130 días de desarrollo, alrededor de 4 meses y medio.

Fuentes de consulta

- Introduction · React Native. (2021). Retrieved 6 March 2021, from https://reactnative.dev/docs/getting-started
- Hernández, D., & Hernández. (2021). ¿Cuáles son los programadores que mejor cobran?. Retrieved 6 March 2021, from https://computerhoy.com/noticias/tecnologia/cuales-son-programadores-mejor-cobran-709631
- Antúnez Barbosa, T., Valdovinos Rosas, R., Marcial Romero, J., Ramos Corchado, M., & Herrera Arriaga, E. (2021). Estimación de costos de desarrollo, caso de estudio: Sistema de Gestión de Calidad del Reactor TRIGA Mark III. Retrieved 6 March 2021, from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000100018#t10
- Welcome to the Strapi developer documentation! (2021). Retrieved 6 March 2021, from https://strapi.io/documentation/developer-docs/latest/getting-started/introduction.html