



ESCUELA POLITÉCNICA NACIONAL

INTEGRANTES: JEAN CARLOS SUÁREZ

JORGE PÉREZ

ARNOLD IZA

GÉNESIS CUICHÁN

MATERIA: APLICACIONES WEB Y MÓVILES (GR1)

FECHA DE ENTREGA: 13-09-2022

| Reunión 1 | | | |
|--------------------------------|--|------------------|---------|
| Fecha: | Sábado, 03 de septiembre de 2022 | | |
| Horas inicio: | 2:00 pm | Hora fin: | 7:00 pm |
| Actividades realizadas: | <ul style="list-style-type: none">Reestructuración del backend de acuerdo a lo implementado en el proyecto del primer bimestre | | |
| Logros obtenidos: | <ul style="list-style-type: none">Se reestructuro el configSe reestructuro el model | | |
| Resumen: | <p>A través de una reunión de manera virtual se reunieron todos los integrantes para analizar de que manera quedo configurado el proyecto inicial, para estructurar de acuerdo a lo analizado en clases, es decir, tener un backend y frontend.</p> <p>Luego se creó la carpeta config y carpeta model. Dentro de la carpeta config se creó el archivo <i>mongoose.config.js</i>.</p> <p>Por otro lado, en la carpeta <i>model</i> se creó los archivos <i>order.model.js</i>, <i>producto.model.js</i> y <i>user.model.js</i></p> | | |
| Ideas obtenidas: | <p>Entre las ideas fueron las siguientes:</p> <ul style="list-style-type: none">Buscar ejemplos realizados en clases para la estructura del proyecto.Asignar nombre a la base de datos.Establecer parámetros para cada uno de los modelos.Establecer validaciones en cada uno de los modelos. | | |
| Problemas obtenidos: | <p>Los problemas fueron los siguientes:</p> <ul style="list-style-type: none">La mayor dificultad fue migrar del backend inicial a la estructura que se realizó en clases. | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none">Adaptar por parte a la nueva estructura | | |

| Reunión 2 | | | |
|--------------------------------|--|------------------|---------|
| Fecha: | Domingo, 04 de septiembre del 2022 | | |
| Horas inicio: | 5:00 pm | Hora fin: | 7:00 pm |
| Actividades realizadas: | <ul style="list-style-type: none"> Reestructuración del backend de acuerdo a lo implementado en el proyecto del primer bimestre | | |
| Logros obtenidos: | <ul style="list-style-type: none"> Se reestructuró controllers | | |
| Resumen: | <p>En una reunión virtual, se comenzó a configurar la estructura del backend, en la que la anterior reunión faltaba la configuración de los controllers . Para esto se creó la carpeta <i>controller</i> y <i>routes</i> en la carpeta backend. En la carpeta <i>controller</i> se crea los siguientes archivos <i>order.controller.js</i>, <i>producto.controller.js</i>, <i>seed.controller.js</i>, <i>upload.controller.js</i> y <i>user.controller.js</i>.</p> | | |
| Ideas obtenidas. | <p>Entre las ideas fueron las siguientes:</p> <ul style="list-style-type: none"> Establecer el CRUD para cada uno de los controllers. | | |
| Problemas obtenidos: | <p>Los problemas fueron los siguientes:</p> <ul style="list-style-type: none"> La mayor dificultad fue migrar del backend inicial a la estructura que se realizó en clases. | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none"> Adaptar por parte a la nueva estructura | | |

| Reunión 3 | | | |
|--------------------------------|---|------------------|---------|
| Fecha: | Viernes, 09 de septiembre de 2022 | | |
| Horas inicio: | 8:00 pm | Hora fin: | 9:30 pm |
| Actividades realizadas: | <ul style="list-style-type: none"> • Reestructuración del backend de acuerdo a lo implementado en el proyecto del primer bimestre | | |
| Logros obtenidos: | <ul style="list-style-type: none"> • Se reestructuró routes • Creación del archivo server.js • Se comprobó en POSTMAN las rutas | | |
| Resumen: | <p>En una reunión virtual, se comenzó a configurar la estructura del backend, en la que la anterior reunión faltaba la configuración de las routes. Para esto se creó la carpeta <i>routes</i> en la carpeta backend. En la carpeta <i>routes</i> se creó los archivos <i>order.routes.js</i>, <i>producto.routes.js</i>, <i>seed.routes.js</i>, <i>upload.routes.js</i> y <i>user.routes.js</i>.</p> <p>Luego de realizarla configuración y creación de routes, se creó el archivo <i>server.js</i> en el que se llama a la base de datos, se establece el puerto, se define un archivo de configuración y se define las rutas.</p> <p>Finalmente, se pasó a comprobar el ingreso de datos de manera correcta a la base de datos mediante POSTMAN para las peticiones GET, POST, PUT Y DELETE.</p> | | |
| Ideas obtenidas. | <p>Entre las ideas fueron las siguientes:</p> <ul style="list-style-type: none"> • Establecer las rutas para llamar a la API. • Realizar pruebas en postman • Ingreso de datos de manera correcta | | |
| Problemas obtenidos: | <p>Los problemas fueron los siguientes:</p> <ul style="list-style-type: none"> • La mayor dificultad fue migrar del backend inicial a la estructura que se realizó en clases. | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none"> • Adaptar por parte a la nueva estructura | | |

| Reunión 4 | | | |
|--------------------------------|--|------------------|----------|
| Fecha: | Sábado, 10 de septiembre de 2022 | | |
| Horas inicio: | 2:00 pm | Hora fin: | 10:00 pm |
| Actividades realizadas: | <ul style="list-style-type: none"> • Reestructuración del frontend • Configuración del frontend | | |
| Logros obtenidos: | <ul style="list-style-type: none"> • Funcionalidad completa del frontend con la utilización del backend | | |
| Resumen: | <p>En la reunión realizada se estableció estructura del frontend, visto en clases, debido a que el proyecto inicial tenía el frontend con la estructura pedida lo único que se hizo fue unirlo al backend creado.</p> <p>El frontend funcionó debido a que se uso las mismas rutas definidas en el proyecto inicial.</p> | | |
| Ideas obtenidas. | <ul style="list-style-type: none"> • Crear frontend • Copiar las carpetas del proyecto inicial del frontend en el nuevo proyecto • Instalar el frontend • Corregir errores que surgieron, | | |
| Problemas obtenidos: | <p>Los problemas fueron los siguientes:</p> <ul style="list-style-type: none"> • Existieron errores debido a ciertos controllers que se definieron de diferente manera con respecto al backend inicial. | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none"> • Se corrigió variables de manera que estén iguales que en los controllers. | | |

| Reunión 5 | | | |
|--------------------------------|--|------------------|---------|
| Fecha: | Domingo, 11 de septiembre de 2022 | | |
| Horas inicio: | 3:00 pm | Hora fin: | 7:00 pm |
| Actividades realizadas: | <ul style="list-style-type: none"> • Creación del chatbot de Telegram • Configuración de la comunicación de Telegram en el backend | | |
| Logros obtenidos: | <ul style="list-style-type: none"> • Funcionalidad completa del chatbot con la utilización de Telegram a través del backend | | |
| Resumen: | <p>En la reunión realizada se configuró la api de Telegram mediante la configuración de un chatbot en el cual, se requirió crear a través del BotFather obteniendo el token para habilitar la comunicación, luego se estructuró dentro del archivo server.js en donde realiza la comunicación desde un dispositivo móvil hacia nuestro servidor.</p> <p>El chatbot "Farmacias_Vitality" funcionó debido a que muestra la información de los productos de manera detallada.</p> | | |
| Ideas obtenidas. | <ul style="list-style-type: none"> • Utilizar Telegram • Crear un chatbot y obtener la API • Instalar la librería Telegraf para utilizar en el frontend • Configurar los mensajes de ayuda y bienvenida • Configurar los mensajes en caso de que el usuario requiere más información acerca de un producto • Corregir errores que surgieron, | | |
| Problemas obtenidos: | <p>El problema fue el siguiente:</p> <ul style="list-style-type: none"> • Identificar cuál método funciona de acuerdo a la solicitud realice el cliente, debido a que en la guía de Telegraf existen algunos métodos similares, sin embargo, los resultados varían al momento de mostrar en pantalla | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none"> • Utilizar los métodos .start() y .help() para mostrar los mensajes iniciales, además, de utilizar .reply() para mostrar el mensaje en texto y .hears() en caso de que el usuario envíe el mensaje solicitado | | |

| Reunión 6 | | | |
|--------------------------------|---|------------------|----------|
| Fecha: | Lunes, 12 de septiembre de 2022 | | |
| Horas inicio: | 2:00 pm | Hora fin: | 10:00 pm |
| Actividades realizadas: | <ul style="list-style-type: none"> • Creación del frontend para móviles basado en React Native • Implementación de la aplicación móvil | | |
| Logros obtenidos: | <ul style="list-style-type: none"> • Funcionalidad completa de la aplicación móvil en React Native asimilando la función de frontend | | |
| Resumen: | En la reunión realizada se construyó la app móvil basada en React Native asimilando la función de frontend en la cual, su función principal, es mostrar los productos del negocio dentro del dispositivo móvil | | |
| Ideas obtenidas. | <ul style="list-style-type: none"> • Instalar React Native • Configurar la aplicación móvil • Mostrar una interfaz amigable con el usuario dentro de la aplicación móvil • Utilizar Telegram • Corregir errores que surgieron | | |
| Problemas obtenidos: | <p>El problema fue el siguiente:</p> <ul style="list-style-type: none"> • Identificar la manera de adaptar de acuerdo al frontend elaborado a base de React, a la aplicación móvil de React Native obteniendo la API de nuestro servidor para mostrar dentro del dispositivo. | | |
| Solución: | <p>La solución planteada fue la siguiente:</p> <ul style="list-style-type: none"> • Utilizar una plantilla de ejemplo consultada en la página oficial de React Native, además de instalar algunas librerías para obtener las APIs tal como lo es Axios para mostrar en pantalla la lista de los productos disponibles en nuestro negocio | | |