APOYO EN EL DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE SOLUCIONES SOFTWARE MEDIANTE METODOLOGÍAS AGILES EN EL SECTOR DE TECNOLOGÍA DEL GRUPO ALGAR S.A.S

POR:

SANTIAGO CARDONA ARBOLEDA

INFORME FINAL

PRÁCTICAS INTERINSTITUCIONALES

FACULTAD DE INGENIERÍA

TECNOLOGIA EN SISTEMAS

TECNOLÓGICO DE ANTIOQUIA- INSTITUCIÓN UNIVERSITARIA

MEDELLIN

2021

CONTENIDO

1. INTRODUCCIÓN

La realización de las practicas no son solo un proceso requerido para titulación como Tecnólogo en Sistemas, sino también un proceso fundamental en la formación profesional del estudiante.

La organización propone el desarrollo de un aplicativo software como medio por el cual validar la adquisición de conocimientos al final del periodo de práctica, este deberá ser llevado a cabo bajo el marco de metodología ágil SCRUM con tecnologías front end y back end especificadas por la organización, además, este deberá tener funciones aplicables a un ambiente laboral del sector, y será debidamente calificado y aprobado por la misma organización.

En el presente documento se describen los componentes de las experiencias, proyectos y conocimientos adquiridos durante el proceso de prácticas, entre estos se hace necesario mencionar el apoyo brindado en el desarrollo del proyecto Electro Lux, el apoyo al desarrollo de un Bot para atención al cliente, la trayectoria en el aprendizaje del idioma portugués, y demás tareas realizadas que se han de mencionar a lo largo del documento.

2. DESCRIPCIÓN DEL LUGAR DE PRÁCTICA

2.1 Descripción de la empresa

Nombre o razón social:	ALGAR TECNOLOGIA S.A. S
Actividad principal:	Procesos de negocios
Dirección:	CALLE 49 SUR # 45 A 300 EDIF S48 OFICINA 2214
Ciudad:	Medellín
Teléfono:	480 86 66
Página web:	https://algartech.com/es

2.2 Información del cooperador

Nombres y apellidos:	Juan Camilo Flórez
Cargo:	Senior Software Developer – Tutor
Profesión:	Ingeniero de Sistemas y Computación
Teléfono:	322 860 24 33
Correo electrónico:	jcflorezv@unal.edu.co

2.3 Misión

Nuestra misión es conectar gente y organizaciones en una forma única, somos el socio adecuado para quien está buscando resultados, de ahí nuestro compromiso por transformar la relación entre clientes y compañías a través de soluciones inteligentes.

2.4 Visión

Nuestra visión es crecer por encima del promedio en la industria, con transformación digital como pilar clave, inversiones en modelos de negocio innovadores, y ser reconocida como la mejor opción para clientes y accionistas hasta 2022.

2.5 Principios y/o valores corporativos

1. El cliente es la razón de nuestra existencia
2. Integrated
3. Valorar el talento humano
4. Sustentabilidad
5. Emprendimiento

2.6 Reseña histórica de la empresa

Somos parte del Grupo Algar, que tiene 90 años y opera en las Tecnologías de la Información y la Comunicación (TIC), Entretenimiento y Agro.

Unos 19.000 colaboradores sobresalen por ganarse la confianza de sus 2 millones de clientes, por su compromiso de servir con agilidad y desarrollar soluciones innovadoras que contribuyan a lograr mejores resultados.

Durante 22 años hemos estado repensando la experiencia sus clientes y gestionando el entorno tecnológico con un único propósito: conectar personas y organizaciones de una manera única para ser el mejor socio para sus mejores resultados.

Actualmente operamos en toda Latinoamérica y nuestro mayor diferencial es ¡Nuestra gente! Aquí todos se sienten como en casa. Nuestro entorno es cool, nuestra diversidad es nuestro condimento y nuestro conocimiento es nuestra fuerza.

2.7 Descripción del área de la práctica

La práctica tiene lugar dentro del equipo de desarrollo ágil de proyectos del sector de tecnología del grupo Algar, este equipo está compuesto por un líder técnico o Scrum Master, un líder de proyecto o Product owner (este tiene contacto directo con el cliente para saber si lo que se hace, cumple con las expectativas del mismo), un asesor o Senior developer, y desarrolladores coordinados de manera virtual.

Específicamente, dentro del equipo asignado por la empresa como entorno de práctica, este llevó a cabo el desarrollo del proyecto Electrolux durante la mayor parte del proceso, este buscaba la implementación de una plataforma que permitiera registrar atenciones a clientes, registrar y listar productos adquiridos e instalados por los mimos, y proporcionar facilidades que ayudaran a agilizar el proceso de soporte a usuarios finales.

3. DESCRIPCIÓN DE LA PRÁCTICA

La práctica toma lugar en un entorno de producción de software compuesto por un equipo que bajo el marco de trabajo de metodologías agiles, emprenden el desarrollo de proyectos mediante adición de funcionalidades por fases hasta completar el producto al final.

Es de particular importancia mencionar que cada proyecto se ha de fraccionar en pequeñas partes u objetivos a cumplir llamados Sprints, cada uno de estos, ha de tener prioridad, sea alta; media o baja; una lista de subtareas a realizar para la conclusión del mismo, y un de plazo de 1 a 4 semanas dependiendo de su complejidad. Una vez el objetivo del Sprint es definido, el Scrum Master lista, reparte y asigna según roles y capacidades las tareas a desarrollar (Estas han de ser catalogadas dentro del Scrum board dependiendo si están pendientes por ejecución, en proceso, en pruebas o completadas).

Finalmente, cuando las tareas dispuestas para el sprint son completadas e integradas al producto, este es probado como un conjunto de todas las actividades anteriores y recién integradas, si este es aprobado, se da por terminado el sprint y se pasa al siguiente ciclo hasta culminar el proyecto.

Específicamente, dentro del equipo asignado por la empresa como entorno de práctica, este llevó a cabo el desarrollo del proyecto Electrolux de manera virtual durante la mayor parte del proceso, el cual buscaba la implementación de una plataforma que permitiera registrar atenciones a clientes, registrar y listar productos adquiridos e instalados por los mimos, y funciones que ayudaran a agilizar el proceso de soporte brindado por operarios a usuarios finales.

Al final del proceso de práctica se habrá evaluado mediante un rubrica presentada por la organización, la capacidad para adaptarse a los diferentes problemas que se le plantearon como Full Stack Developer Junior.

4. OBJETIVOS DE LA PRÁCTICA

4.1 General

Apoyo en el diseño, desarrollo e implementación de soluciones software bajo el marco de metodologías agiles en el sector de tecnología del grupo Algar S.A.S.

4.2 Específicos

- Desarrollo de arquitecturas para sitios web.
- Desarrollo de aplicativos backs end.
- Desarrollo de UI / front end.

5. FUNCIONES REALIZADAS

5.1. Función 1: Desarrollar arquitecturas para sitios web.

Determinar gráficamente el flujo que la información ha de seguir dentro del sitio web, sin llevar a cabo ningún tipo de implementación.

Esta parte se basa en graficar la estructura que los componentes del sitio web han de seguir, esto se realiza con el fin de visualizar las rutas que se crearan y la forma en la que la información ha de fluir por las mismas.

5.2. Función 2: Desarrollar aplicativos Back End.

Desarrollar de rutas para el acceso a la información, con el propósito de realizar operaciones C.R.U.D sobre esta.

5.3. Función 3: Desarrollar interfaces Front End.

Desarrollar la apariencia del sitio web y establecer las conexiones que permitan al usuario la manipulación de la información.

5.4. Función 4: Desarrollar servicios.

Desarrollar componentes con el fin de independizar y modularizar proyectos para el cumplimiento de funciones específicas, que ayudan a la durabilidad, seguridad y estabilidad del aplicativo.

6. DESARROLLO METODOLÓGICO DE LA PRÁCTICA

Dentro del equipo asignado por la empresa como entorno de práctica, se llevó a cabo el desarrollo del proyecto Electrolux durante la mayor parte del proceso, este buscaba la implementación de una plataforma que permitiera registrar atenciones a clientes, registrar y listar productos adquiridos e instalados por los mimos, y proporcionar facilidades que ayudaran a agilizar el proceso de soporte a usuarios finales.

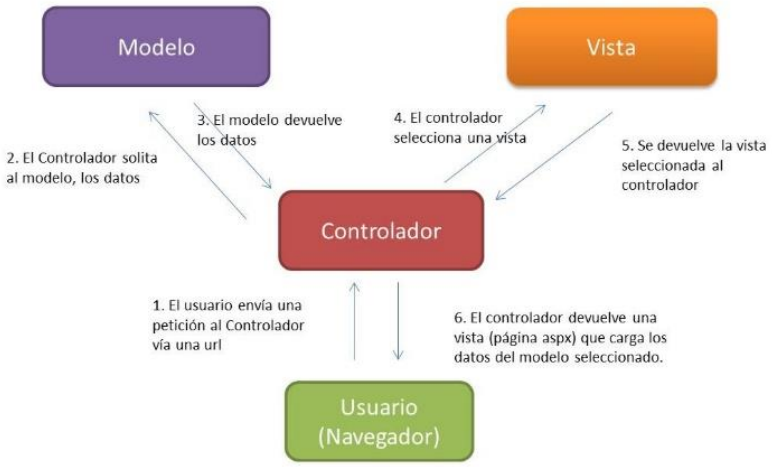
Se debe mencionar que, en la elaboración de este proyecto para el tiempo en el que se llevaron a cabo las practicas, este ya contaba con significativos avances referentes al diseño de su arquitectura, desarrollo del back, front end y otras funcionalidades. sin embargo, dentro del este periodo, se llevó a cabo el desarrollo de múltiples pantallas UI, se implementaron funciones para el BackEnd y mínimos ajustes a funciones del sistema.

Desarrollar arquitecturas para sitios web

Referente al desarrollo de arquitecturas web, se hizo uso del modelo Vista / Controlador en donde, este contiene una contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. Siendo la vista, la interfaz de usuario que compone la información que se envía al cliente y los mecanismos interacción con éste. Y el Controlador, que actúa como intermediario entre el Modelo y la Vista [1], gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Figura 1

Modelo Vista / Controlador



Tomado de: Modelo Vista – Controlador por QualityDev

https://external content.duckduckgo.com/iu/?u=https%3A%2F%2Fsites.google.com%2Fsite%2Faunaris2%2F_%2Frsrc%2F1468878015667%2Fprogramacion%2Fmodelo-vista---controlador%2Fmvc.jpg&f=1&nofb=1

Desarrollar aplicativos Back End.

Frente al desarrollo BackEnd se opta por Node.Js en gran parte de los proyectos debido a su funcionalidad, flexibilidad, escalabilidad y simpleza en temas como el desarrollo de APIs (Interfaz de programación de aplicaciones), Aplicaciones de Streaming, Aplicaciones que utilizan tiempo real y Microservicios.

En este apartado se crearon rutas para cada operación CRUD (Create, Read, Update y Delete respectivamente) realizada sobre los datos en Front End.

Figura 2

Ejemplo basico de especificacion de rutas Back End

```
const express = require('express')
const app = express()

app.get('/',function(req,res){
  res.json({title :'hello world'})
})

app.get('/user',function(req,res){
  res.json({firstName :'John',{lastName
  :'Doe',age :33}})
})

app.listen(3000, function () {
  console.log('Example app listening on port
  3000!')
})
```

Cabe mencionar que cada ruta especificada dentro de una aplicación, explicita el tipo de operación que se ha de realizar con la misma, por lo que a su vez se tuvieron en cuenta los diferentes tipos de métodos de acceso a rutas (GET, POST, PUT, DELETE, PATCH, etc.)

Desarrollar interfaces Front End.

Para el desarrollo de front end se decidió hacer uso de librerías y frameworks que facilitaran la construcción de proyectos a través del agilismo, entre estas se encuentran las librerías de React.js, Angular CLI y frameworks como Material UI, Ant design y Boostrap.

Al momento de escoger la librería a utilizar de entre la gran variedad de estas, se tuvieron en cuenta múltiples opciones, pero en comparación a estas, React aporta una serie de posibilidades

muy importantes. Al tener las vistas asociadas a los datos, no se hace necesario escribir código para manipular la página cuando los datos cambian. Esta parte en librerías sencillas es muy laboriosa de conseguir y es algo que React hace automáticamente. Además, en comparación a otras como jQuery, React permite una arquitectura de desarrollo más avanzada, con diversos beneficios como la encapsulación del código en componentes, que ofrecen una serie de ventajas más importantes que los plugin, como la posibilidad de que esos componentes conversen e interaccionen entre sí.

Respecto al uso de esta herramienta, se implementaros interfaces de usuario basadas en mock-ups y campos definidos dentro de historias de usuario presentadas para cada pantalla por el scrum master. Las historias de usuario usualmente fueron presentadas en un formato parecido al siguiente:

Figura 3

Formato estândar para história de usuário

Historia de Usuario	
Numero: 4	Nombre: Contratos con Empresas
Usuario: Administrador	
Modificación de Historia Numero:	Iteración:
Prioridad en Negocio: Media	Puntos Estimados:
Riesgo en Desarrollo:	Puntos Reales:
Descripción: En la opción de Contratación con empresas, se llenara un formulario de registro de firma de contrato con el cliente que sería la empresa, se tendrá la siguiente información, como Nombre de la empresa del cliente, Números de referencia, Nombre del cliente, Monto por el cual se realiza el contrato, Fecha de inicio de la obra, Fecha de conclusión de obra.	
Observaciones:	

Tomado de: https://sites.google.com/site/fentocelsrl/plantillas-de-historias-de-usuario?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

Es importante mencionar que estas historias generalmente definen tanto los campos que deberán contener las interfaces de usuario, como la forma en la que estos han de interactuar con la lógica de negocio / Back End. Estas historias de usuario son representadas en estructuras de código que más tarde son compiladas para entregar un producto visual.

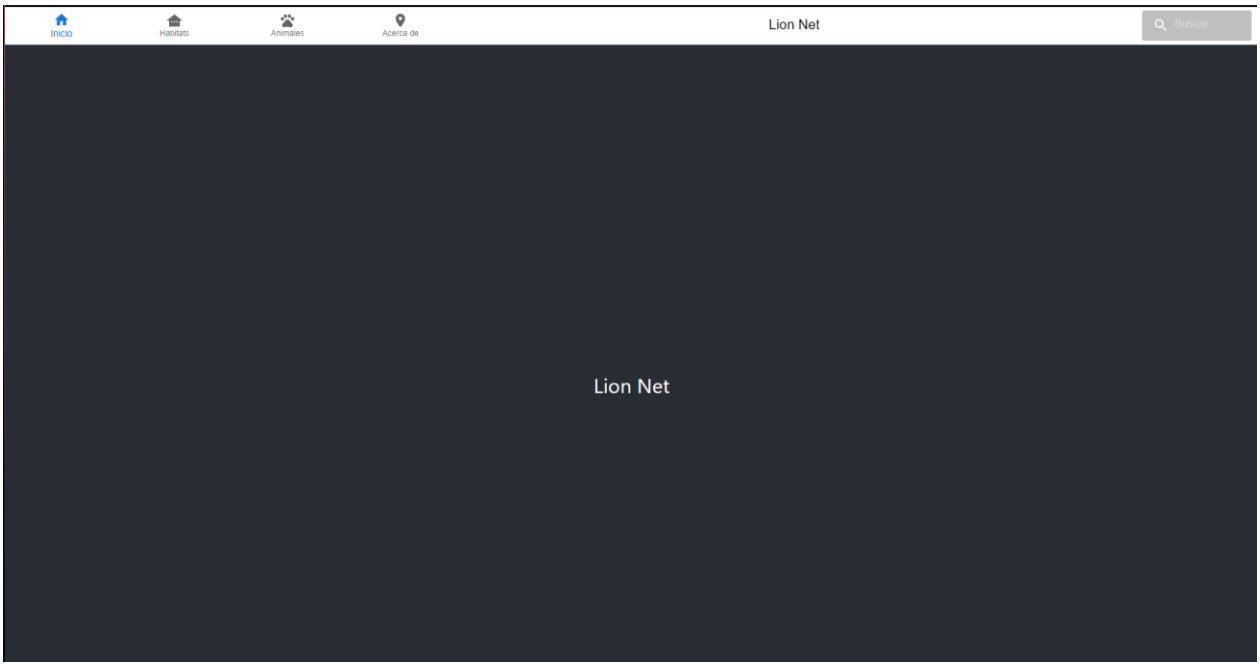
Figura 3

Código fuente de aplicativo Front End

```
export default function NavBar() {
    const [value, setValue] = React.useState(0);
  return (
    <Box sx={{ flexGrow: 1}}>
      <AppBar position="static">
            <BottomNavigation
                showLabels
                value={value}
                onChange={(event, newValue) => {
                setValue(newValue);
                }}
                >
                <BottomNavigationAction label="Inicio" icon={<HomeIcon />} />
                <BottomNavigationAction label="Habitats" icon={<OtherHousesIcon />} />
                <BottomNavigationAction label="Animales" icon={<PetsIcon />} />
                <BottomNavigationAction label="Acerca de" icon={<LocationOnIcon />} />
                <Typography
                    variant="h6"
                    noWrap
                    component="div"
                    sx={{ flexGrow: 2, display: { xs: 'none', sm: 'block' }, color: 'black', marginTop:'10px'}}
                >
                    Lion Net
                </Typography>

                <Search>
                <SearchIconWrapper>
                    <SearchIcon />
                </SearchIconWrapper>
                <StyledInputBase
                    placeholder="Buscar"
                    inputProps={{ 'aria-label': 'Buscar' }}
                />
                </Search>
            </BottomNavigation>
      </AppBar>
    </Box>
  );
}
```

Figura 4

Representación visual del código



Desarrollar servicios.

Por último, se hizo uso de la tecnología de contenedores Docker para el empaquetado de funciones independientes que, mediante rutas y puertos, prestaron servicios específicos a aplicativos de diferentes nichos.

En este proceso, se empaquetaron tanto aplicativos Back como Front end, que mediante contenedores modulares e independientes los unos de los otros, corrieron servicios con fines específicos pero necesarios para otras aplicaciones.

Figura 5

Estructura general utilizada para los archivos docker-compose.yml

```yaml
version: '3.9'

services:
    back:
        build: ./restservices .
        ports:
            - 8080:8080
        networks:
            - some-net

    front:
        build: ./PokeAPI-Front .
        ports:
            - 3000:3000
        networks:
            - some-net
networks:
    some-net:
        driver: bridge
```

7. RESULTADOS OBTENIDOS

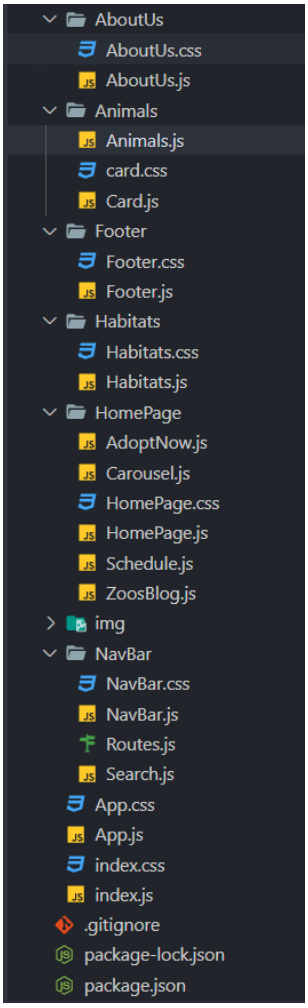
7.1 Función 1 Desarrollar arquitecturas para sitios web.

Se adquirieron las competencias necesarias para la creación de estructuras web basadas en requerimientos propuestos por el cliente.

Para este desarrollo, se implementó una estructura que permitiera al usuario navegar a través de las pantallas de “Home”, “Hábitats”, “Animals” y “About Us”; cada una de ellas con funciones referentes al nombre de las mismas.

Figura 6

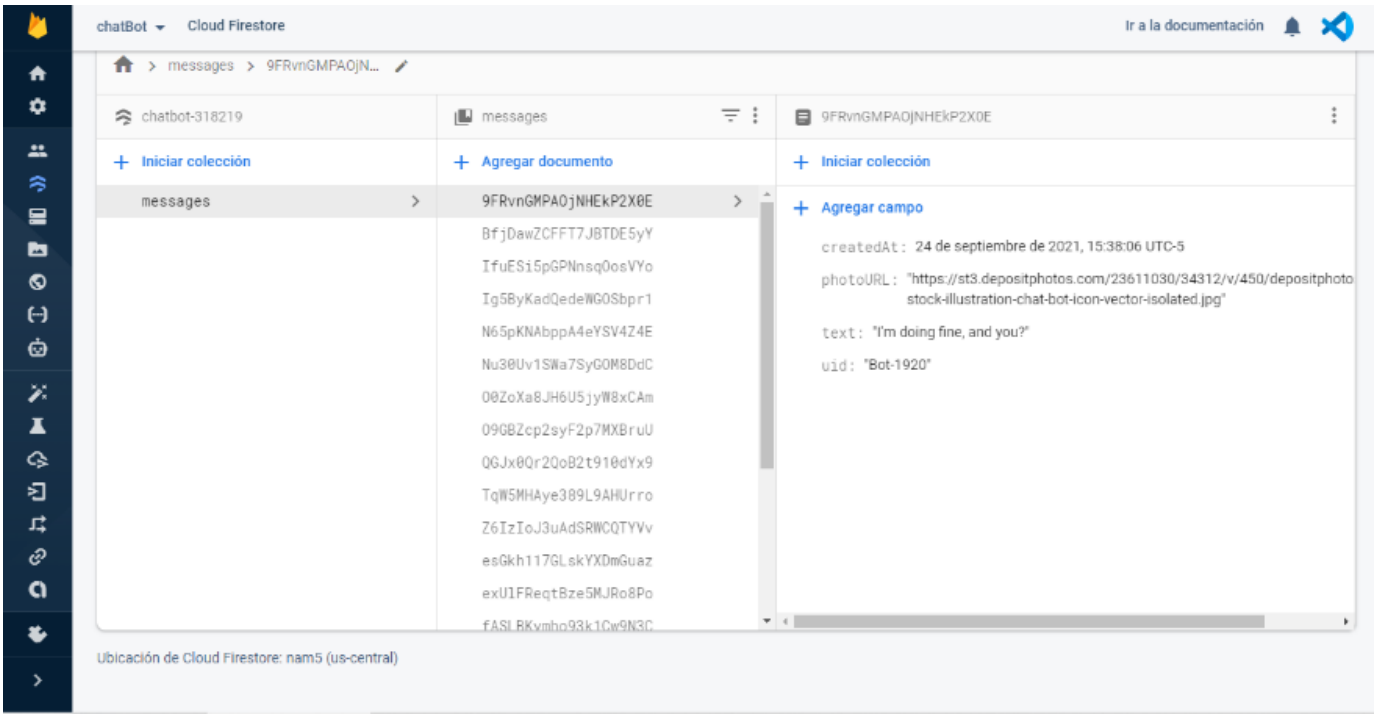
Arquitectura Web del Aplicativo Lion Net



Además de esto, se diseñaron estructuras para colecciones de bases de datos a través de la consola de Firebase, esto luego fue usado para crear repositorios de datos en la nube para mensajes de un ChatBot. Estas colecciones fueron diseñadas en base a los campos que se consideraron requeridos para la creación de mensaje y usuarios registrados.

Figura 7

Colección creada para el proyecto ChatBot



7.2 Función 2: Desarrollar aplicativos Back End.

Se codificaron rutas que permitieran realizar operaciones como:

1. Listar Animales

```
crudRoute.get('/getAnimals', getAnimals)


export const getAnimals = (req, res) => {


   try {


      async function getCollection() {
         const queryAnimal = collection(db, 'animals');
         const animalSnapshot = await getDocs(queryAnimal);
         const animalList = animalSnapshot.docs.map(doc => { const docSnapshot = doc.data(); return {
...docSnapshot, id: doc.id } });
         res.status(200).json({ data: animalList });
      }


      getCollection();


   } catch (error) {


      res.status(404).json({ error });


   }
}
```

2. Actualizar Animales

```
crudRoute.put('/updateAnimal/:id', updateAnimal)


export const updateAnimal = async (req, res) => {


  const schema = Joi.object({
    id_animal: Joi.string().min(1).max(255).required(),
    name: Joi.string().min(4).max(255).required(),
    born: Joi.string().min(10).max(255).required(),
    description: Joi.string().required(),
    origin: Joi.string().min(1).max(30).required(),
    id_habitat: Joi.number().min(1).max(10).required(),
    id_blood: Joi.number().min(1).max(10).required(),
    id_genre: Joi.number().min(1).max(10).required(),
    status: Joi.string().min(1).max(10).required(),
  });


  try {


    const validate = schema.validate(req.body);


    if (!validate.error) {


      const docRef = doc(db, 'animals', `${req.params.id}`);


      await updateDoc(docRef, req.body);
```

```
            res.status(200).json({ message: 'Se ha actualizado correctamente' })
        } else {
            res.status(422).json({ message: 'Te faltan campos por llenar', error: validate.error.details[0].message })
        }

    } catch (error) {
        res.status(404).json({ message: 'Error al intentar validar la informacion', error })
    }
}
```

3. Añadir animal

```
crudRoute.post('/addAnimal', addAnimal)
export const addAnimal = async (req, res) => {
    const schema = Joi.object({
        id_animal: Joi.string().min(1).max(255).required(),
        name: Joi.string().min(4).max(255).required(),
        born: Joi.string().min(10).max(255).required(),
        description: Joi.string().required(),
        origin: Joi.string().min(1).max(30).required(),
        id_habitat: Joi.number().min(1).max(10).required(),
        id_blood: Joi.number().min(1).max(10).required(),
        id_genre: Joi.number().min(1).max(10).required(),
        status: Joi.string().min(1).max(10).required(),
    });
```

```
  try {
    const validate = schema.validate(req.body);


    if (!validate.error) {


      const options = {
        'method': 'GET',
        'url': `${process.env.URL_BASE}${process.env.PORT}/api/getAnimal/${req.body.id_animal}`,
        'headers': {
          'Content-Type': 'application/json'
        }
      };


      request(options, async (error, response) => {
        if (!error && response.statusCode === 404) {
          const docRef = await addDoc(collection(db, 'animals'), req.body);
          res.status(201).json({ id: docRef._key.path.segments[1], data: `Registrado correctamente` })
        } else {
          res.status(200).json({ message: `El animal con ${req.body.id_animal} ya se encuentra registrado`
})
        }
      });


    } else {
      res.status(422).json({ message: 'Te faltan campos por llenar', error: validate.error.details[0].message })
    }
  } catch (error) {
```

```
    res.status(404).json({ message: 'Error al intentar validar la informacion', error })
}
```

Dichas rutas, fueron utilizadas por el Front de la aplicación Lion Net a través de consultas de la siguiente manera:

```
React.useEffect(() => {
    getAnimals();
}, []);


const getAnimals = () => {
    const URL = 'http://localhost:4000/api/getAnimals';
    fetch(URL)
        .then(response => response.json())
        .then(data => setAnimals(data.data));
}
```

7.3 Función 3: Desarrollo de interfaces Front End.

Se codificaron los siguientes componentes para el aplicativo Lion Net, estos fueron desarrollados a través de módulos de la creación de funciones JavaScript que tuvieran como retorno componentes JSX que al ser renderizados dieran como resultado las vistas que se observan a continuación:

Figura 8

Carousel principal, diseñado para la pantalla de Home de Lion Net

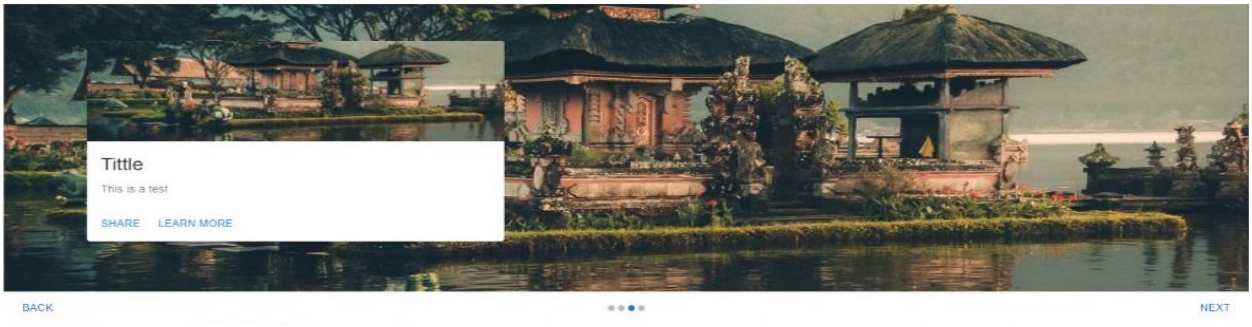
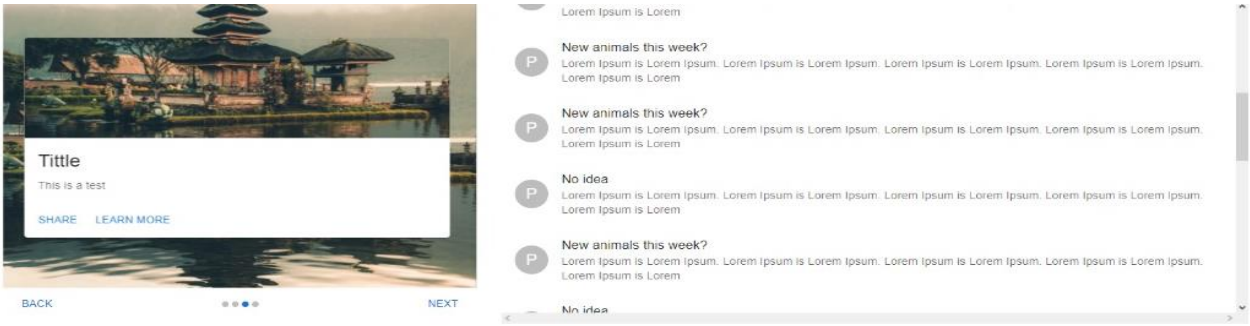


Figura 9

Carousel & Blog List diseñados para secciones Zoos Blog y Adopt Now de Lion Net



Si bien los componentes visuales se codificaron bajo el marco de React.js, la mayoria de las etiques utilizadas fueron importadas de la librería Matrial UI, tal y como se muestra en la siguiente porcion de codigo sacada de uno de los componentes mencionados.

```
import Card from '@mui/material/Card';
import CardActions from '@mui/material/CardActions';
import CardContent from '@mui/material/CardContent';
import CardMedia from '@mui/material/CardMedia';
import Typography from '@mui/material/Typography';
import Button from '@mui/material/Button';


function MediaCard({tittle: Tittle, text: Text, image: Image}) {
```

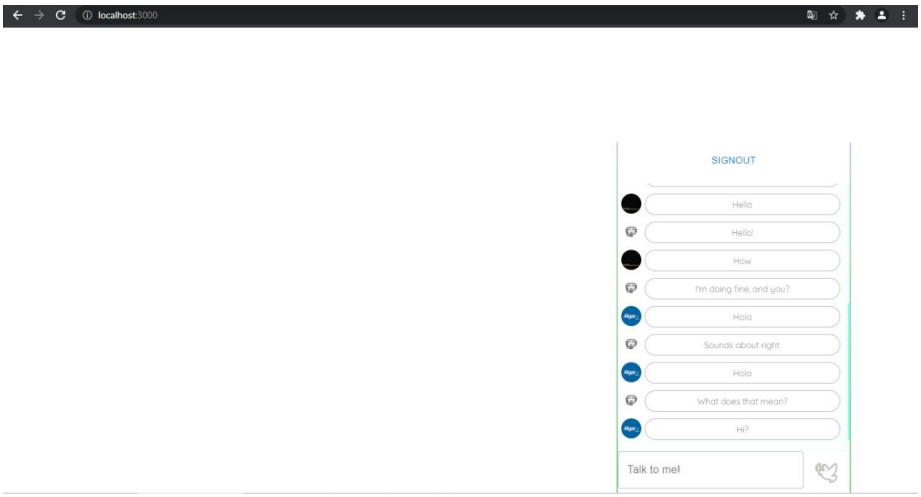
```
return (
  <Card sx={{
     width: '500px',
     position: 'absolute',
     transform: "translate( 20%, -125%)"
     }}>

    <CardMedia
     component="img"
     height="140"
     image={Image}
     alt="green iguana"
    />


    <CardContent>
     <Typography gutterBottom variant="h5" component="div">
        {Tittle}
     </Typography>
     <Typography variant="body2" color="text.secondary">
        {Text}
     </Typography>
    </CardContent>
```

```
    <CardActions>
      <Button size="small">Share</Button>
      <Button size="small">Learn More</Button>
    </CardActions>
  </Card>
 );
}
```

Cabe mencionar que se llevo a cabo la implementacion de un ChatBot enlazado a colecciones de Firebase previamente implementatdas.

Figura 10

Pestaña principal de ChatBot realizado con Firebase



Para la codificación de este ChatBot se hizo uso del Api de Autenticación de Google para el control de los usuarios y estados de autorización para la validación de la sesión como se muestra en la siguiente porción de código:

```
import firebase from 'firebase/app';
import 'firebase/firestore';
import 'firebase/auth';

import { useCollectionData } from 'react-firebase-hooks/firestore';
import { useAuthState } from 'react-firebase-hooks/auth'; Estado de autorizacion

firebase.initializeApp({Llaves del Api});

// Initializing Firebase
const auth = firebase.auth();
const firestore = firebase.firestore();

function App() {
  const [
    user //Isuario actual
    // , setUser
  ] = useAuthState(auth)


  return (
    <div className="App">
        {user ?
```

```
      <ChatRoom />
      : <SignIn />}
    </div>
  );
}
```

7.4 Función 4: Desarrollar servicios.

Se codifico un Midleware en Python con el fin de que actuara como punto intermediario entre el texto enviado por un usuario y la respuesta proporcionada por un Bot después del procesamiento de dicho texto.

Figura 11

BackEnd codificado en Python codificado para conseguir las respuestas

```python
from getResponses import get_response
from flask import Flask, json, jsonify, request
from flask_cors import CORS, cross_origin
# py -3 pip install pip install -U flask-cors

app = Flask(__name__)
cors = CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'

@app.route('/', methods = ['GET'])
@cross_origin()
def home_page():
    #access your DB get your results here
    data = { "data":"This is the home page, wellcome!" }
    return jsonify(data)


@app.route('/message', methods = ['GET'])
@cross_origin()
def getResponses():
    message_query = str(request.args.get('message')) # http://127.0.0.1:3001/message?message=how
    response = {'response': str(get_response(message_query))}
    return jsonify(response)

if __name__ == '__main__':
    app.run(port=3001)
```

Para el procesamiento de dicho texto, se hizo uso de la siguiente función creada con el fin de enviar una de las posibles respuestas en base a probabilidades calculadas por coincidencia de palabras.

Figura 12

Algoritmo codificado en Python para el procesamiento del y retorno de respuestas

```python
import long_responses as long


def message_probability(user_message, recognised_words, single_response=False, required_words=[]):
    message_certainty = 0
    has_required_words = True

    # Counts how many words are present in each predefined message
    for word in user_message:
        if word in recognised_words:
            message_certainty += 1

    # Calculates the percent of recognised words in a user message
    percentage = float(message_certainty) / float(len(recognised_words))

    # Checks that the required words are in the string
    for word in required_words:
        if word not in user_message:
            has_required_words = False
            break

    # Must either have the required words, or be a single response
    if has_required_words or single_response:
        return int(percentage * 100)
    else:
        return 0
```

7.5 Dificultades técnicas en el desarrollo de la práctica

Durante el proceso de práctica, en ocasiones se presentaron momentos de hasta una semana de duración en los que no se pudo llevar a cabo la realización de las actividades con normalidad, esto debido a falta de credenciales necesarias para el acceso a plataformas, inconsistencias en la información presentada para la especificación de tareas, dependencia de sectores externos atrasados en actividades, y falta de comunicación por parte de otros miembros. Además, puesto que la organización escogida como centro de prácticas tiene su sede principal de operaciones ubicada en Brasil, fue necesario el estudio del idioma portugués para facilitar la compresión de documentación en este idioma, así como la comunicación con intermediarios de proyectos y personal de la empresa.

8. CONCLUSIONES

Como conclusión, se obtiene que el proceso de practica fue sustancialmente necesario y constructivo para el desarrollo profesional, así como también permitió adquirir y refinar competencias relevantes en el mercado como fueron el diseño de arquitecturas web, la codificación de aplicativos tanto Back como Front End, y la creación de servicios que cumplieran funciones específicas.

Además, durante este periodo se pudo visualizar de primera mano la forma en la que se llevan a cabo las tareas en el sector tecnológico, por lo que también se dieron nociones de las posibles problemáticas que se podrían presentan en futuros desarrollos.

Sin lugar a duda, es de remarcar que esta fue una experiencia que brindó contexto y conocimientos de gran valor que, por último, se agradece inmensamente el medio que la institución proporcionó para la realización de la misma.

9. REFERENCIAS

Agencia de Marketing Online NeoAttack. (2014, 21 diciembre). *▷ ¿Qué es Front end y para qué sirve? - Neo Wiki*. NeoAttack. https://neoattack.com/neowiki/front-end/

Ghoshal, A. (2021, 26 marzo). *What is Firebase?* EDUCBA. https://www.educba.com/what-is-firebase/

Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework. (2019, 20 junio). Universidad de Alicante. https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

¿Qué son los microservicios? | AWS. (2021, 6 octubre). Amazon Web Services, Inc. https://aws.amazon.com/es/microservices/

Team, C. (2021, 4 septiembre). *What is Back End?* Codecademy News. https://www.codecademy.com/resources/blog/what-is-back-end/

What Is Node.js? (2021, 10 febrero). Oracle Developer. https://developer.oracle.com/nodejs/what-is-node-js/

What is User Interface Design? (2018, 12 octubre). The Interaction Design Foundation. https://www.interaction-design.org/literature/topics/ui-design