



PERÍODO 2020



Proyecto Final

Chatbot para el sector de los restaurantes.

GRUPO #5

Baquerizo Marzo_Barros Mero_Castillo Caicedo_Soto Yance_Troya Palma

Descripción del problema.

Debido a la situación actual alrededor del mundo, se han virtualizado todas nuestras actividades, es decir, la gran mayoría de trabajos ahora se desarrollan como teletrabajo y las tiendas tuvieron que crear su propia página web para poder promocionar sus productos o mediante redes sociales.

En el caso de los restaurantes, se ofrece un menú diferente cada día por lo que esta información se envía por WhatsApp o las personas escriben por interno, lo mismo en redes sociales, pero cuando la demanda es muy alta se complica el responderles a tantos usuarios para que conozcan el menú del día. Entonces se decidió crear un Chat Bot que sea capaz de interactuar con los usuarios y darles a conocer las diferentes opciones que tiene el restaurante, esto sin necesidad que este una persona constantemente respondiendo a cada usuario, muchas veces el mismo mensaje.

Diagramas del diseño del proyecto, modelo entidad-relación.

Para la implementación de nuestro proyecto se decidió usar en nuestro backend Node.js, el cual nos permitirá poder conectarnos con el API de slack mediante un websocket, para esto se procederá a utilizar el paquete que la misma plataforma nos brinda para node.js que es el slack bot. Este api nos permite la utilización de diferentes servicios, pero en nuestro caso utilizaremos el servicio del bot al que nos podremos conectar mediante un token que se lo genera en la plataforma de Slack. Para nuestra base de datos se empleará MongoDB, como estamos conectados por medio de un websocket se procederá a responder las solicitudes del usuario en donde el bot podrá acceder a nuestra base y extraer la información que se necesita, procesarla y así mostrársela al usuario. A continuación, se procederá a mostrar una imagen descriptiva del diseño que se implementó.

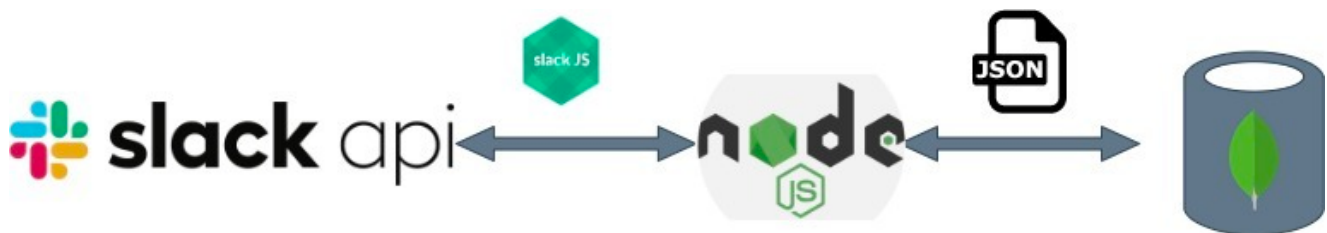


Ilustración 1 Diseño del proyecto

Para que se pueda extraer la información de nuestra base de datos en MongoDB se utilizara mongosse, esta información se lograra extraer en formato JSON. En el código se encontrarán funciones que nos permitirán hacer la conexión con la base de datos. Aquí se mostrará la tabla en la que se guardará toda la información de las diversas opciones que ofrece el restaurante.

Menú
Id
Categoría
Plato
Precio

Recursos que se utilizará.

Para la programación de nuestra propuesta de utilizara Nodejs como se pudo observar en el diseño, implementándose el api de slack, que nos ofrecerá el servicio del boot. Para que los usuarios del restaurante puedan comenzar la interacción con el boot y poder conocer las diversas opciones que ofrece el restaurante se lo podrá hacer de dos maneras.

La primera manera es que el usuario cuente con un dispositivo móvil y pueda obtener la aplicación de slack, en la que simplemente deberá presionar el siguiente enlace, el cual lo agregar al chat del restaurante donde podrá ya interactuar con el bot.

https://join.slack.com/t/chatrestaurant/shared_invite/zt-gge2qquv-kNOviBVaKRdr06cbzsYVMA

La segunda forma es accediendo desde un computador o simplemente desde un navegador, deberá presionar sobre e siguiente enlace <https://app.slack.com/client/T018HEW7C6B/C018HEW7GRM>. En el que ya podrá interactuar con nuestro bot, cabe recalcar que para las dos opciones es necesaria una conexión a internet, ya que. la base de datos esta en la nube y nuestro bot deberá extraer la información de esta para presentar a los usuarios

Como ya se había mencionado se utilizo como base de datos una base de MongoDB, pero se utilizo mongosse para poder acceder a los datos, esta base se encuentra en la nube y es ahí, en donde los administradores de los restaurantes tendrán la posibilidad de acceder y modificar el menú del día, o

también actualizar los platos con su respectivo precio. Para que el bot comience su ejecución se deberá correr nuestro código. Para este proyecto se utilizó el software de visual studio porque nos ofrece una interfaz amigable y podemos hacer el uso de NodeJs. En el archivo async-await.js es donde se podrá modificar los datos en la sección async function main()

```
src > JS async-await.js > ...
1  const db = require('../backend/database')
2  const menu=require('../models/restaurante')
3
4  async function main(){
5      const menus=new menu({
6          categoria: 'Bebidas',
7          nombre: 'Vinos',
8          precio:1.00
9      })
10     const savemenu=await menus.save();
11     console.log(savemenu);
12     return savemenu
13 }
14
15 main()
16 .then(savemenu =>console.log(savemenu))
17 .catch(err =>console.log(err))
```

Esta base de datos estará conectada con slack la cual es un api que nos ofrece el servicio del bot. Una vez ingresado a la aplicación tendremos nuestro bot el cual nos ofrecerá información con respecto al menú del día.

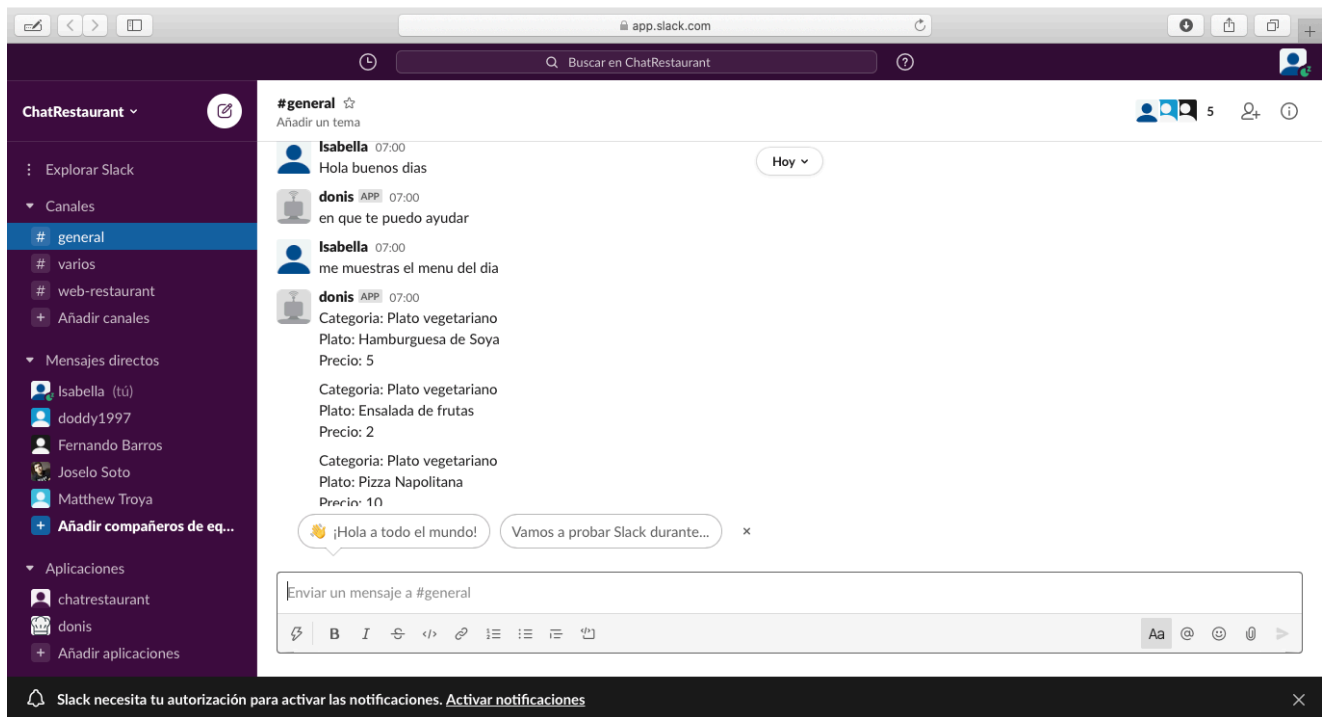
Para que nuestro chatbot se encuentre activo se necesita ejecutar el programa donde en el terminal nos indicará cuando el bot esté activo y si la conexión a la base de datos fue exitosa.

```
macbookpro@MacBook-Pro-de-Macbookpro Chat_Bot % node src/index.js
Bot is ready
Database is connect to mongodb+srv://restaurant:70IuZAIrtY7a6Tya@cluster0.w68qc.mongodb.net/restaurante_db?retryWrites=true&w=majority
```

```

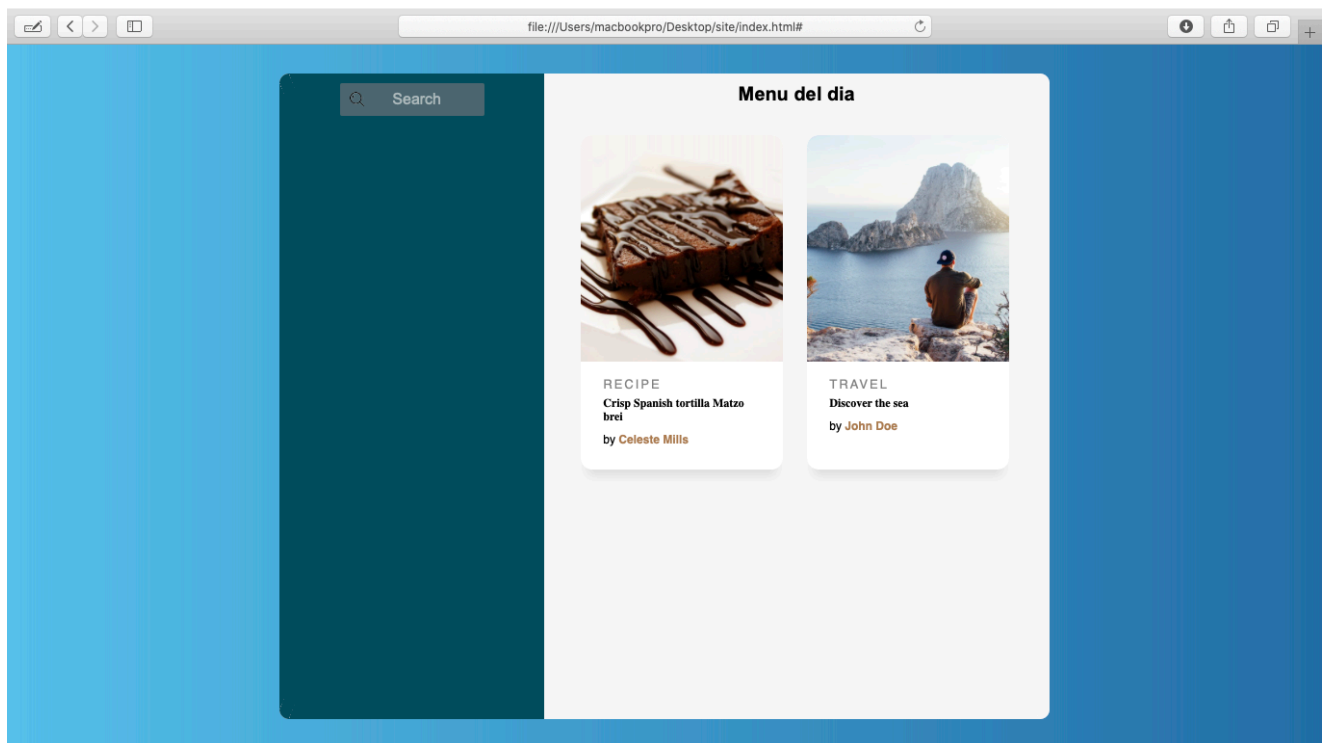
src > JS find.js > ...
1  const db = require('../backend/database')
2  const menu=require('../models/restaurante')
3  const axios = require('axios');
4  async function main(){
5      const menus= await menu.find()
6      for(var i=0;i<menus.length;i++){
7          console.log("Categoria: "+menus[i].categoria+"\n"+
8              "Plato: "+menus[i].nombre+"\n"+
9              "Precio: "+menus[i].precio+"\n")
10     }
11
12 }
13
14 main()
15

```



Recomendaciones

Para mejorar el funcionamiento de nuestro chatbot se desea realizar una página web, es decir un frontend propio del restaurante, donde todos los usuarios podrán acceder sin necesidad de solicitar el permiso del administrador para poder utilizar el chatbot. Esta página web fue creada, pero debido a que slack cuenta con su propio frontend no se pudo realizar la conexión con nuestro frontend.



ANEXOS

Código principal

```
const SlackBots =require('slackbots');
const axios = require('axios');
const db = require('../backend/database')
const menu=require('../models/restaurante')
//Eliminar duplicados
function onlyUnique(value, index, self) {
  return self.indexOf(value) === index;
}
// Funcion menu
async function main(){
  const menus= await menu.find()
  for(var i=0;i<menus.length;i++){
    bot.postMessageToChannel('general',"Categoria: "+menus[i].categoria+"\n"+
    "Plato: "+menus[i].nombre+"\n"+
    "Precio: "+menus[i].precio+"\n")
  }
}

async function categoria(){
  const menus= await menu.find()
  let list_category=[]
  for(var i=0;i<menus.length-1;i++){
    list_category.push(menus[i].categoria) //creando una lista con las categorias
```

```

    }
    const unique = list_category.filter(onlyUnique)

    for(var i=0;i<unique.length;i++){
        bot.postMessageToChannel('general',unique[i]+"\\n")
    }
}

async function vegetariano(){
    const menus=await menu.find()
    for(var i=0;i<menus.length;i++){
        if (menus [i].categoria=="Plato vegetariano"){
            bot.postMessageToChannel('general', "Categoria: "+menus[i].categoria+"\\n"+
            "Plato: "+menus[i].nombre+"\\n"+
            "Precio: "+menus[i].precio+"\\n")
        }
    }
}

}

async function bebidas(){
    const menus=await menu.find()
    for(var i=0;i<menus.length;i++){
        if (menus [i].categoria=="Bebidas"){
            bot.postMessageToChannel('general', "Categoria: "+menus[i].categoria+"\\n"+
            "Plato: "+menus[i].nombre+"\\n"+
            "Precio: "+menus[i].precio+"\\n")
        }
    }
}

}

async function postre(){
    const menus= await menu.find()
    for(var i=0;i<menus.length;i++){
        if(menus[i].categoria=="Postre"){
            bot.postMessageToChannel('general',"Categoria: "+menus[i].categoria+"\\n"+
            "Plato: "+menus[i].nombre+"\\n"+
            "Precio: "+menus[i].precio+"\\n")
        }
    }
}

}

async function carta(){
    const menus= await menu.find()
    for(var i=0;i<menus.length;i++){
        if(menus[i].categoria=="Plato a la carta"){
            bot.postMessageToChannel('general',"Categoria: "+menus[i].categoria+"\\n"+

```

```

        "Plato: "+menus[i].nombre+"\n"+
        "Precio: "+menus[i].precio+"\n")
    }
}

}

const bot=new SlackBots({
  token: 'xoxb-1289506250215-1304282982274-P1aIgsHV37Cn8ezsIWxJJ76N',
  name: 'donis'
});

bot.on('open',()=> console.log('Bot is ready'));

bot.on('start',()=>{
  bot.postMessageToChannel('general','Hola que deseas :smiley:')
});

bot.on('message',data=>{
  if(data.type !== 'message' || data.subtype==='bot_message' || !data.text)return;
  const arg=data.text.split(" ")
  if(arg.includes("menu")){
    main()
  }
  else if(arg.includes("como") && arg.includes("estas")){
    bot.postMessageToChannel('general','Bien y tu')
  }
  else if(arg.includes("categorías") || arg.includes("categorias") ||
arg.includes("opciones")|| arg.includes('opcion') ){
    categoria()
  }
  else if (arg.includes("postres")){
    postre()
  }
  else if ( arg.includes("bebidas") || arg.includes ("tomar") || arg.includes
("refrescos") ){
    bebidas()

  }
  else if (arg.includes("platos") && arg.includes("carta")){
    carta()
  }
  else if ( arg.includes("vegetarianas") || arg.includes('vegetariana') ||
arg.includes('vegetariano') || arg.includes('vegetarianos') ) {
    vegetariano()
  }
  else if ( arg.includes("gracias")) {
    bot.postMessageToChannel('general','Un gusto ayudarte')
  }
}

```



```

    else{
        bot.postMessageToChannel('general','en que te puedo ayudar')
    }
    console.log(arg)
})

```

Conexión a la base de datos

```

const mongoose=require('mongoose')
const
uri='mongodb+srv://restaurant:70IuZAIRtY7a6Tya@cluster0.w68qc.mongodb.net/restaurante_db?r
etryWrites=true&w=majority'
const db=mongoose.connection;
mongoose.connect(uri,{
    useNewUrlParser:true,
    useUnifiedTopology:true
});

db.on('open',__dirname=>{
    console.log('Database is connect to',uri);
})

db.on('error',err=>{
    console.log(err)
})

```

Imporación de la base de datos

```

const {Schema,model}=require('mongoose')

const menu=new Schema({
    categoria: String,
    nombre:String,
    precio:Number
})

module.exports =model('Menu',menu)

```

Agregar información a la base de datos

```

const db = require('.././backend/database')
const menu=require('.././models/restaurante')

async function main(){
    const menus=new menu({
        categoria:'Bebidas',
        nombre:'Vinos',
        precio:1.00
    })
}

```

```

    })
    const savemenu=await menus.save();
    console.log(savemenu);
    return savemenu
  }

  main()
  .then(savemenu =>console.log(savemenu))
  .catch(err =>console.log(err))

```

Extracción de información de la base de datos

```

const db = require('../backend/database')
const menu=require('../models/restaurante')
const axios = require('axios');
async function main(){
  const menus= await menu.find()
  for(var i=0;i<menus.length;i++){
    console.log("Categoria: "+menus[i].categoria+"\n"+
      "Plato: "+menus[i].nombre+"\n"+
      "Precio: "+menus[i].precio+"\n")
  }
}

main()

```

Código página web (Front end alternativo)

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
html,
body {
  height: 100%;
  display: flex; /* atributo para empezar flex container */
  justify-content: center;
  align-items: center;
}
html {
  font-family: Arial, Helvetica, sans-serif;
  background: linear-gradient(to right, #57c1eb 0%,#246fa8 100%);
  font-size: 10px;
}
#chat-container {
  min-width: 800px; /* definir un min width y max width es esencial en un flex container*/

```

```

    max-width: 1000px;
    max-height: 800px;
    height: 95vh;
    background: #FFF;
    border-radius: 10px;
    flex-grow: 1; /* atributo para flex container */
    display: flex;
}

#side-bar {
    background: #004b5efd;
    border-radius: 10px 0 0 10px;
    width: 275px;
    height: 100%;
    flex-basis: 275px;
    display: flex;
    justify-content: center;
}

#chat-window {
    background: rgba(236, 236, 236, 0.397);
    border-radius: 0 10px 10px 0;
    display: flex;
    flex-basis: 525px;
    width: 525px;
    height: 100%;
    flex-direction: column;
}

#search-box {
    background: rgba(128, 128, 128, 0.507);
    border-radius: 2px;
    width: 150px;
    height: 5%;
    margin-top: 10px;
    display: flex;
    align-items: center;
}

#search-icon {
    width: 15px;
    height: 15px;
    margin-left: 10px;
}

#search-text{
    margin-left: 30px;
    font-size: medium;
    color: rgba(255, 255, 255, 0.541);
}

#menu {
    display: flex;

```

```

    justify-content: center;
    width: 525px;
    height: 5%;
    margin-bottom: 20px;
    margin-top: 10px;
}
h1 {
    color: black;
}
#cards{
    /* overflow-y: scroll;
    overflow-x: hidden; */
}

/* ////////////////////////////////////// */

@import url('https://fonts.googleapis.com/css?family=Roboto+Slab:100,300,400,700');
@import
url('https://fonts.googleapis.com/css?family=Raleway:300,300i,400,400i,500,500i,600,600i,7
00,700i,800,800i,900,900i');

*{
    box-sizing: border-box;
}

.cards {
    width: 100%;
    display: flexbox;
    display: -webkit-flex;
    justify-content: center;
    -webkit-justify-content: center;
    max-width: 820px;
    margin-left: 10px;
}

.card--1 .card__img, .card--1 .card__img--hover {
    background-image: url('https://images.pexels.com/photos/45202/brownie-dessert-cake-
sweet-45202.jpeg?auto=compress&cs=tinysrgb&h=750&w=1260');
}

.card--2 .card__img, .card--2 .card__img--hover {
    background-image: url('https://images.pexels.com/photos/307008/pexels-photo-
307008.jpeg?auto=compress&cs=tinysrgb&h=750&w=1260');
}

.card__like {
    width: 18px;
}

```

```
.card__clock {
  width: 15px;
  vertical-align: middle;
  fill: #AD7D52;
}

.card__time {
  font-size: 12px;
  color: #AD7D52;
  vertical-align: middle;
  margin-left: 5px;
}

.card__clock-info {
  float: right;
}

.card__img {
  visibility: hidden;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  width: 100%;
  height: 235px;
  border-top-left-radius: 12px;
border-top-right-radius: 12px;
}

.card__info-hover {
  position: absolute;
  padding: 16px;
  width: 100%;
  opacity: 0;
  top: 0;
}

.card__img--hover {
  transition: 0.2s all ease-out;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  width: 100%;
  position: absolute;
  height: 235px;
  border-top-left-radius: 12px;
border-top-right-radius: 12px;
top: 0;
```

```
}  
.card {  
  margin-right: 25px;  
  transition: all .4s cubic-bezier(0.175, 0.885, 0, 1);  
  background-color: #fff;  
  width: 40%;  
  position: relative;  
  border-radius: 12px;  
  overflow: hidden;  
  box-shadow: 0px 13px 10px -7px rgba(0, 0, 0,0.1);  
}  
.card:hover {  
  box-shadow: 0px 30px 18px -8px rgba(0, 0, 0,0.1);  
  transform: scale(1.10, 1.10);  
}  
.card__info {  
  z-index: 2;  
  background-color: #fff;  
  border-bottom-left-radius: 12px;  
  border-bottom-right-radius: 12px;  
  padding: 16px 24px 24px 24px;  
}  
.card__category {  
  font-family: 'Raleway', sans-serif;  
  text-transform: uppercase;  
  font-size: 13px;  
  letter-spacing: 2px;  
  font-weight: 500;  
  color: #868686;  
}  
.card__title {  
  margin-top: 5px;  
  margin-bottom: 10px;  
  font-family: 'Roboto Slab', serif;  
}  
.card__by {  
  font-size: 12px;  
  font-family: 'Raleway', sans-serif;  
  font-weight: 500;  
}  
.card__author {  
  font-weight: 600;  
  text-decoration: none;  
  color: #AD7D52;
```

```
}

.card:hover .card__img--hover {
  height: 100%;
  opacity: 0.3;
}

.card:hover .card__info {
  background-color: transparent;
  position: relative;
}

.card:hover .card__info-hover {
  opacity: 1;
}
```