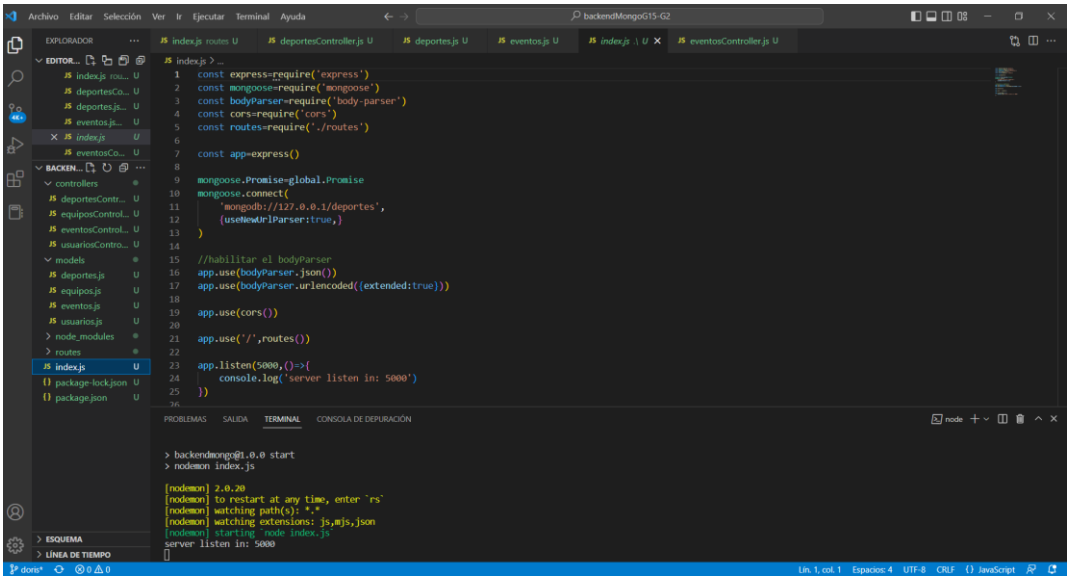
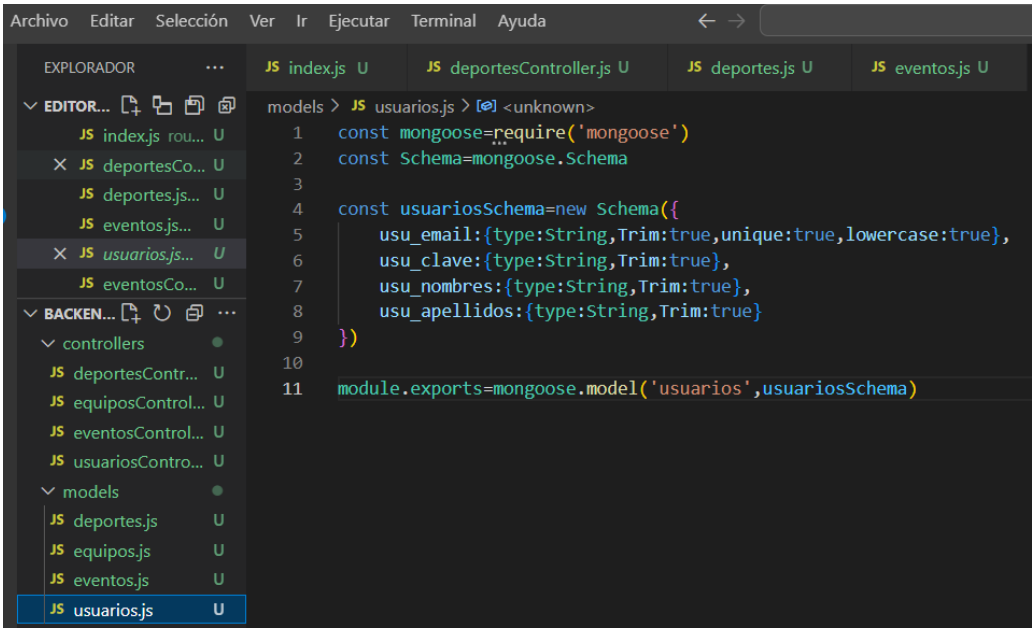
SPRINT 2: Desarrollo del Backend

Identificación Proyecto	
Nombre Proyecto:	Deportes
Número Equipo:	
Integrantes del equipo	
Rol (Líder-Desarrollador – Cliente)	Nombre
Líder	Cristhiam Felipe Mora
Desarrollador	Jorge Moreno
Desarrollador	Juan Álvarez
Desarrollador	Doris Aroca
Desarrollador	Felipe Álvarez
Desarrollador	Brandon Ortiz Herrera
Cliente	Alvaro Stid Bolaños Vásquez

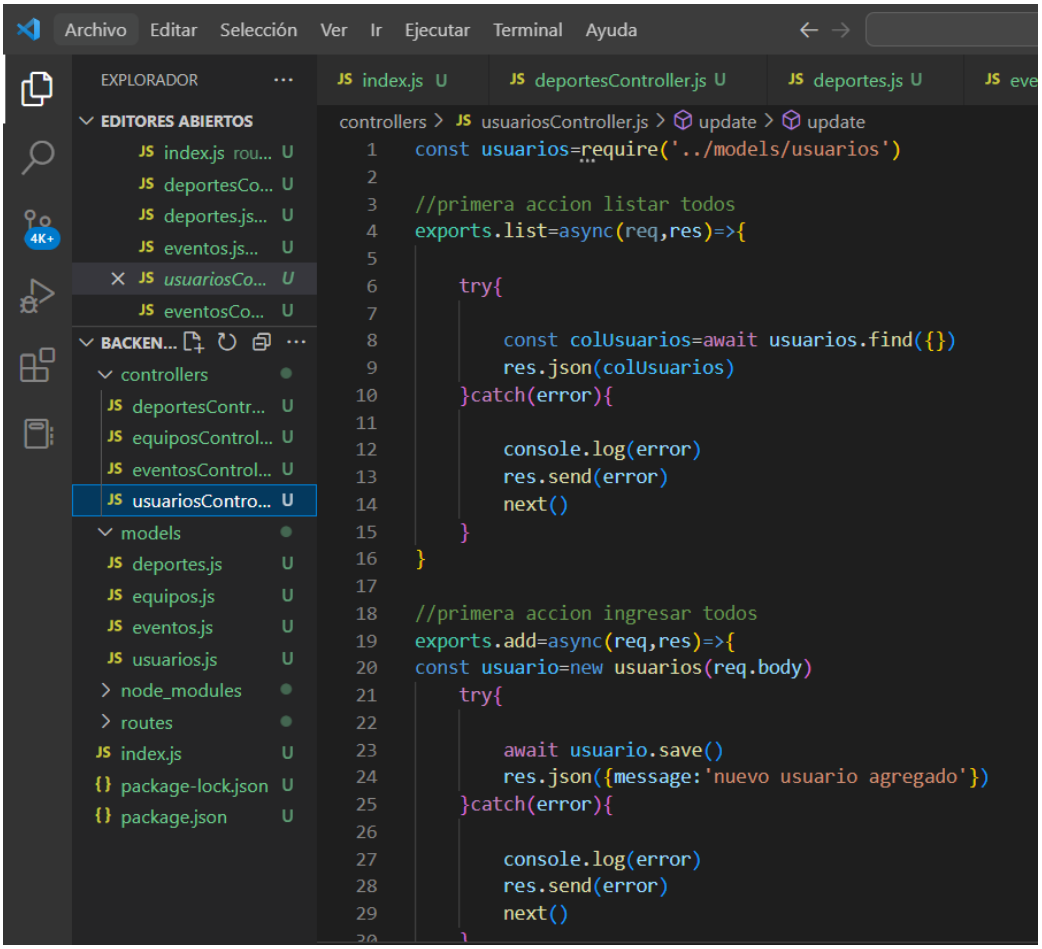
Evidencia construcción del Backend

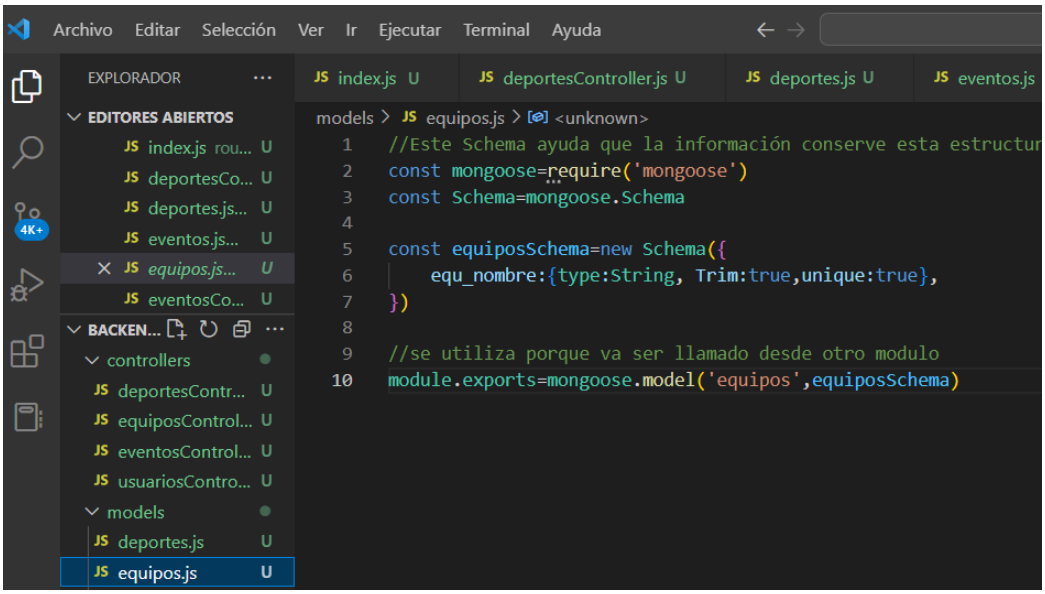
Como evidencia de la construcción del Backend, se debe presentar capturas de pantalla donde se visualice el proceso de construcción del Backend, como la creación en Spring Boot, modelo, controlador, etc.

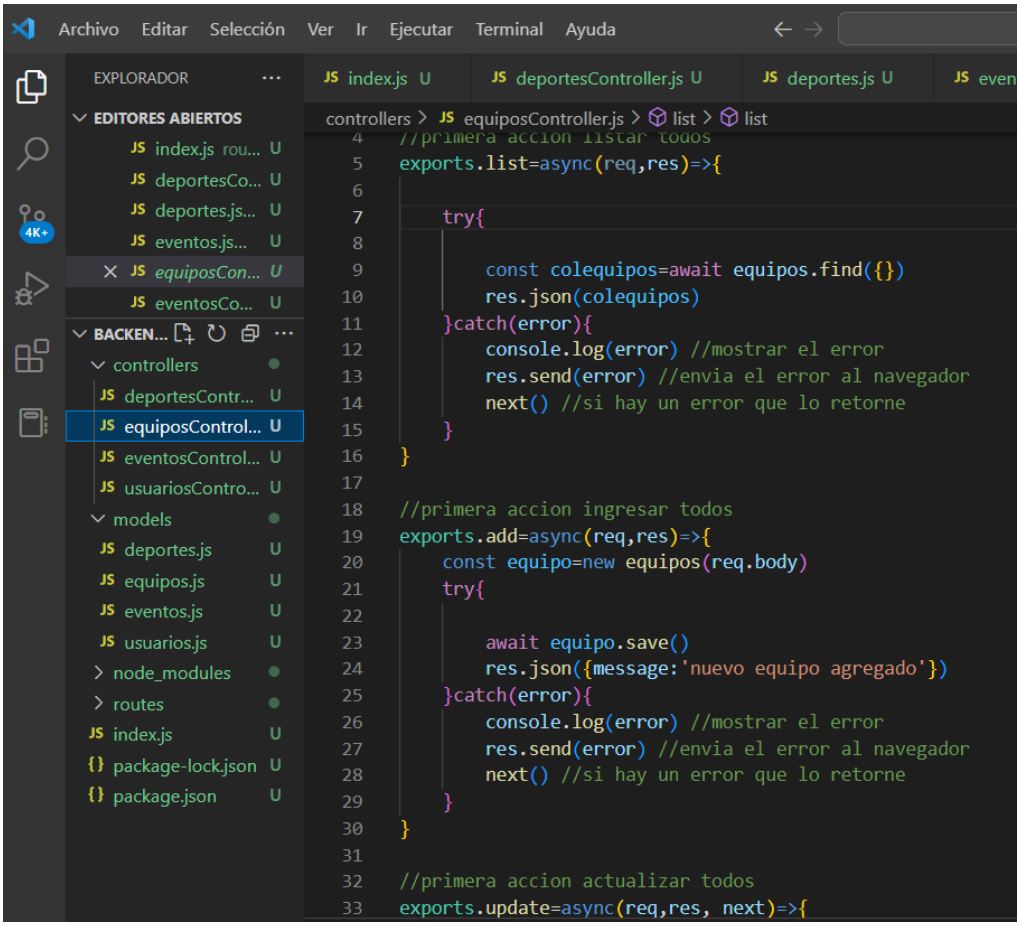



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
index.js U
deportesController.js U
deportes.js U
eventos.js U
models > usuarios.js > <unknown>
1 const mongoose=require('mongoose')
2 const Schema=mongoose.Schema
3
4 const usuariosSchema=new Schema({
5 usu_email:{type:String,Trim:true,unique:true,lowercase:true},
6 usu_clave:{type:String,Trim:true},
7 usu_nombres:{type:String,Trim:true},
8 usu_apellidos:{type:String,Trim:true}
9 })
10
11 module.exports=mongoose.model('usuarios',usuariosSchema)
EDITOR...
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
usuarios.js... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
equipos.js U
eventos.js U
usuarios.js U

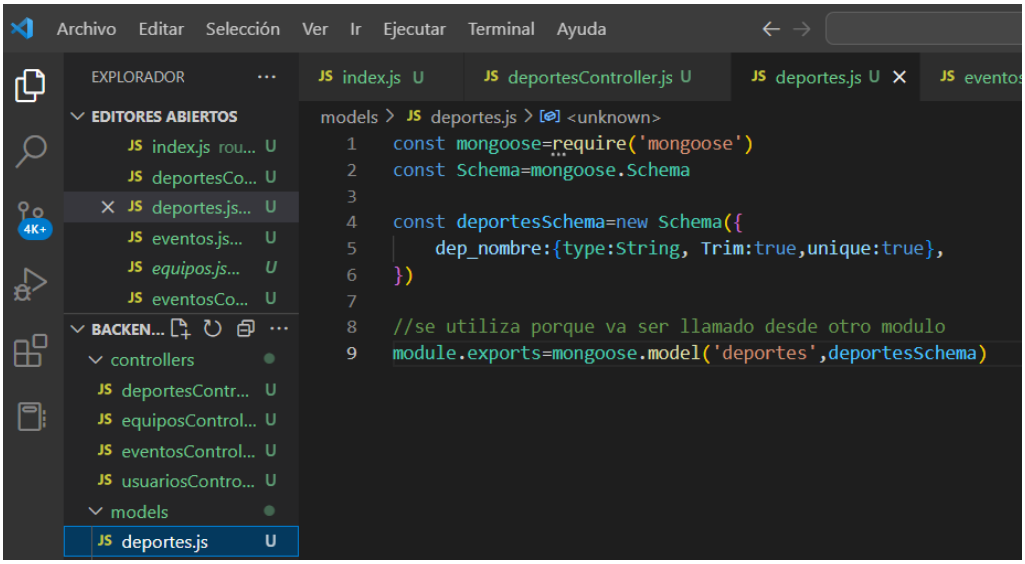

Mision
TIC2022
unab

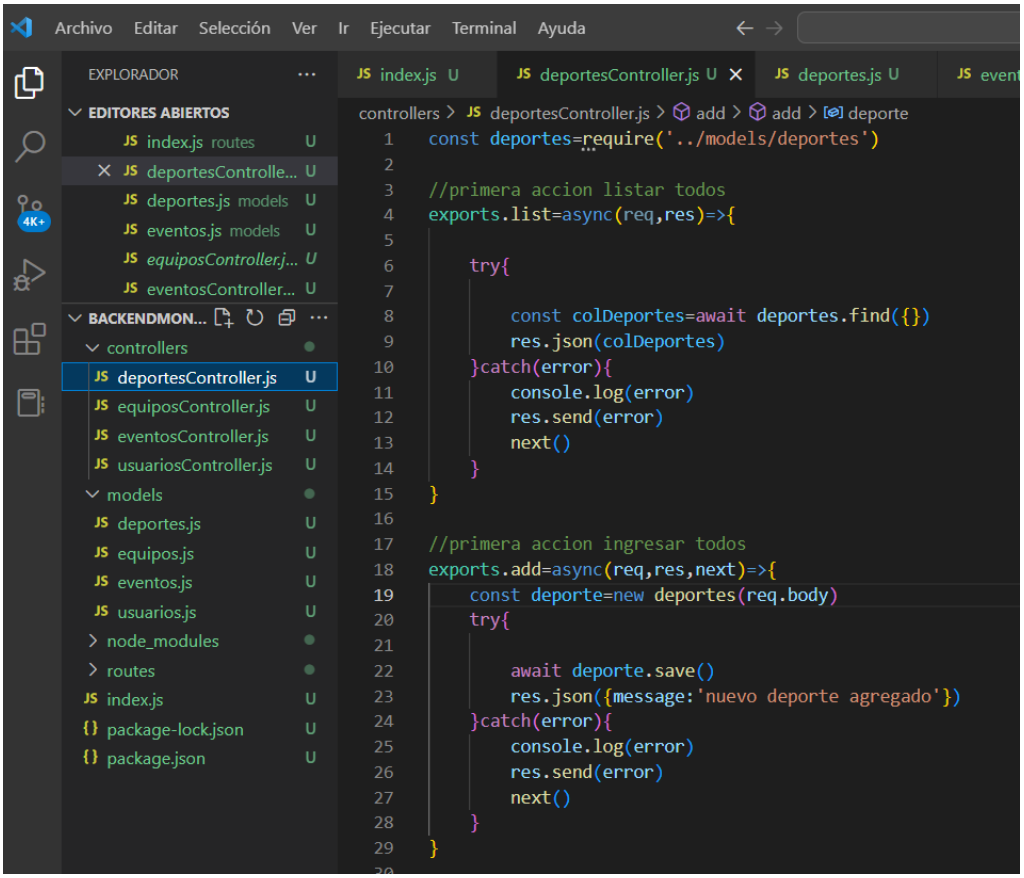

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
usuariosCo... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
equipos.js U
eventos.js U
usuarios.js U
node_modules
routes
index.js U
package-lock.json U
package.json U
index.js U
deportesController.js U
deportes.js U
controllers > usuariosController.js > update > update
const usuarios=require('../models/usuarios')
//primera accion listar todos
exports.list=async(req,res)=>{
try{
const colUsuarios=await usuarios.find({})
res.json(colUsuarios)
}catch(error){
console.log(error)
res.send(error)
next()
}
}
//primera accion ingresar todos
exports.add=async(req,res)=>{
const usuario=new usuarios(req.body)
try{
await usuario.save()
res.json({message:'nuevo usuario agregado'})
}catch(error){
console.log(error)
res.send(error)
next()


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
equipos.js... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
equipos.js U
index.js U
deportesController.js U
deportes.js U
eventos.js
models > equipos.js > <unknown>
//Este Schema ayuda que la información conserve esta estructur
const mongoose=require('mongoose')
const Schema=mongoose.Schema
const equiposSchema=new Schema({
equ_nombre:{type:String, Trim:true,unique:true},
})
//se utiliza porque va ser llamado desde otro modulo
module.exports=mongoose.model('equipos',equiposSchema)

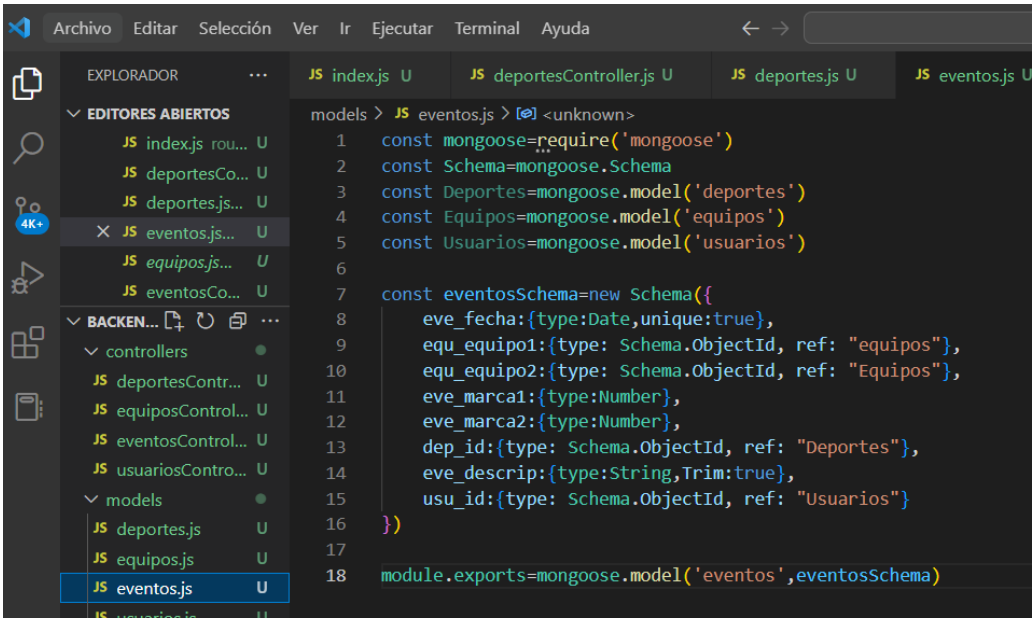

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
equiposCon... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
equipos.js U
eventos.js U
usuarios.js U
node_modules
routes
index.js U
package-lock.json U
package.json U
index.js U
deportesController.js U
deportes.js U
controllers > equiposController.js > list > list
//primera accion listar todos
exports.list=async(req,res)=>{
try{
const colequipos=await equipos.find({})
res.json(colequipos)
}catch(error){
console.log(error) //mostrar el error
res.send(error) //envia el error al navegador
next() //si hay un error que lo retorne
}
}
//primera accion ingresar todos
exports.add=async(req,res)=>{
const equipo=new equipos(req.body)
try{
await equipo.save()
res.json({message:'nuevo equipo agregado'})
}catch(error){
console.log(error) //mostrar el error
res.send(error) //envia el error al navegador
next() //si hay un error que lo retorne
}
}
//primera accion actualizar todos
exports.update=async(req,res, next)=>{

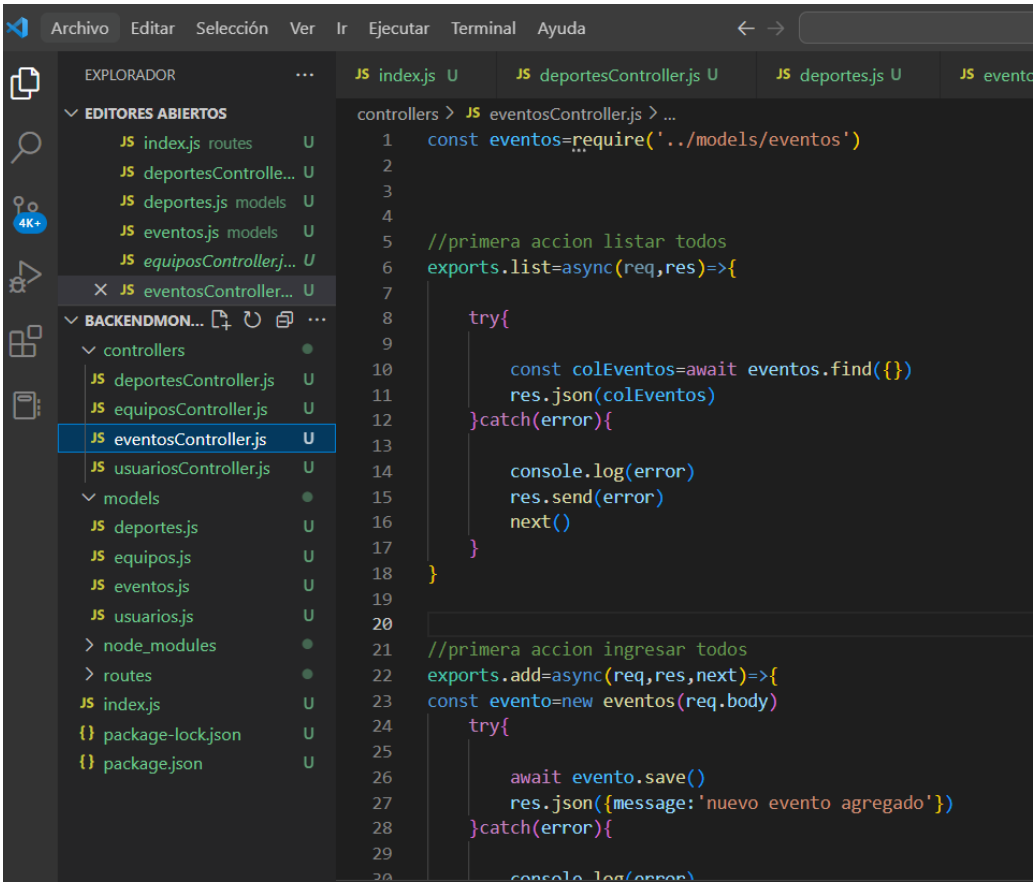

Mision
TIC2022
unab

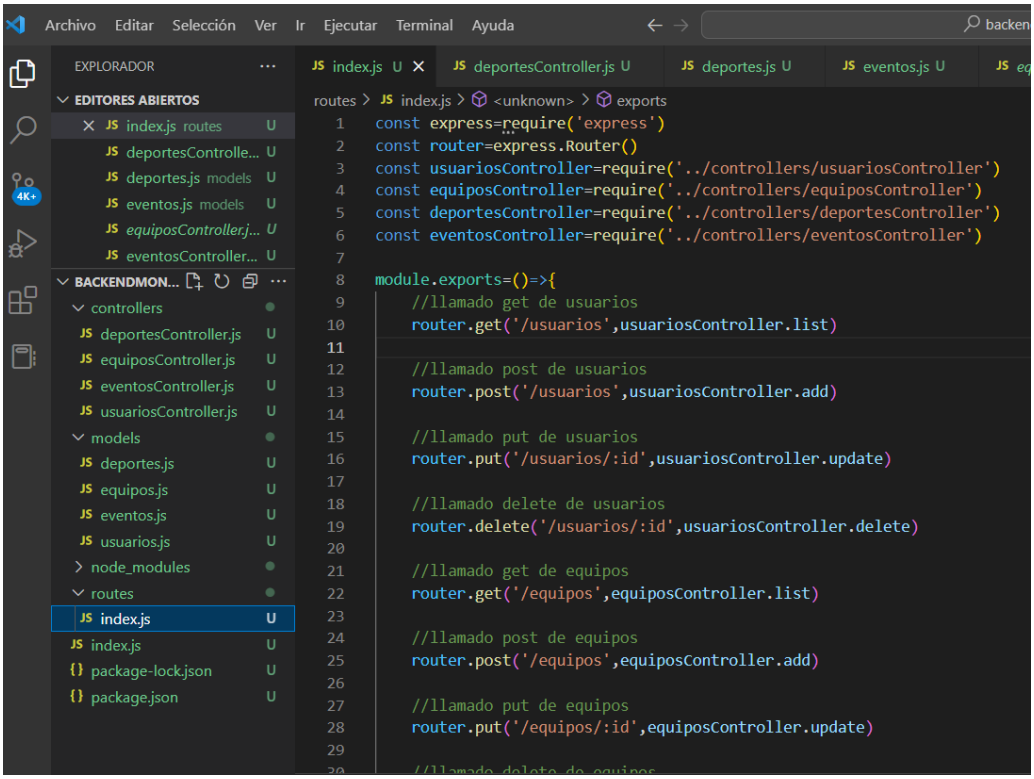

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
equipos.js... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
index.js U
deportesController.js U
deportes.js U
models > deportes.js > <unknown>
1 const mongoose=require('mongoose')
2 const Schema=mongoose.Schema
3
4 const deportesSchema=new Schema({
5 dep_nombre:{type:String, Trim:true,unique:true},
6 })
7
8 //se utiliza porque va ser llamado desde otro modulo
9 module.exports=mongoose.model('deportes',deportesSchema)


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js routes U
deportesControlle... U
deportes.js models U
eventos.js models U
equiposController.j... U
eventosController... U
BACKENDMON...
controllers
deportesController.js U
equiposController.js U
eventosController.js U
usuariosController.js U
models
deportes.js U
equipos.js U
eventos.js U
usuarios.js U
node_modules
routes
index.js U
package-lock.json U
package.json U
index.js U
deportesController.js U
deportes.js U
controllers > deportesController.js > add > add > deporte
1 const deportes=require('../models/deportes')
2
3 //primera accion listar todos
4 exports.list=async(req,res)=>{
5
6 try{
7
8 const colDeportes=await deportes.find({})
9 res.json(colDeportes)
10 }catch(error){
11 console.log(error)
12 res.send(error)
13 next()
14 }
15 }
16
17 //primera accion ingresar todos
18 exports.add=async(req,res,next)=>{
19 const deporte=new deportes(req.body)
20 try{
21
22 await deporte.save()
23 res.json({message:'nuevo deporte agregado'})
24 }catch(error){
25 console.log(error)
26 res.send(error)
27 next()
28 }
29 }


Mision
TIC2022
unab

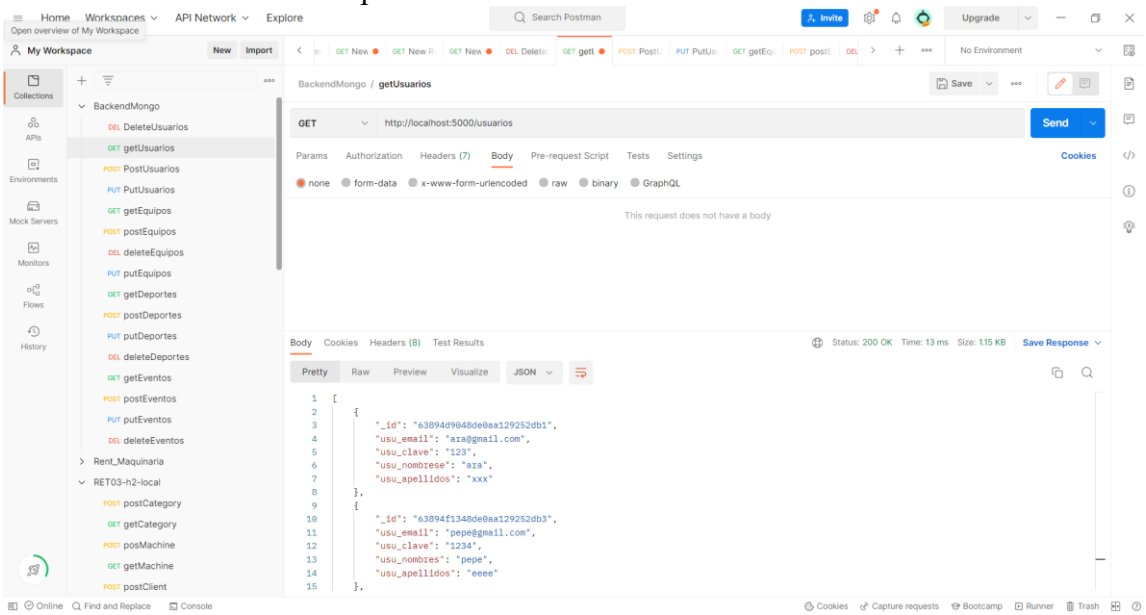

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js rou... U
deportesCo... U
deportes.js... U
eventos.js... U
equipos.js... U
eventosCo... U
BACKEN...
controllers
deportesContr... U
equiposControl... U
eventosControl... U
usuariosContro... U
models
deportes.js U
equipos.js U
eventos.js U
index.js U
deportesController.js U
deportes.js U
eventos.js U
models > eventos.js > <unknown>
1 const mongoose=require('mongoose')
2 const Schema=mongoose.Schema
3 const Deportes=mongoose.model('deportes')
4 const Equipos=mongoose.model('equipos')
5 const Usuarios=mongoose.model('usuarios')
6
7 const eventosSchema=new Schema({
8 eve_fecha:{type:Date,unique:true},
9 equ_equipo1:{type: Schema.ObjectId, ref: "equipos"},
10 equ_equipo2:{type: Schema.ObjectId, ref: "Equipos"},
11 eve_marca1:{type:Number},
12 eve_marca2:{type:Number},
13 dep_id:{type: Schema.ObjectId, ref: "Deportes"},
14 eve_descrip:{type:String,Trim:true},
15 usu_id:{type: Schema.ObjectId, ref: "Usuarios"}
16 })
17
18 module.exports=mongoose.model('eventos',eventosSchema)


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR
EDITORES ABIERTOS
index.js routes U
deportesControlle... U
deportes.js models U
eventos.js models U
equiposController.j... U
eventosController... U
BACKENDMON...
controllers
deportesController.js U
equiposController.js U
eventosController.js U
usuariosController.js U
models
deportes.js U
equipos.js U
eventos.js U
usuarios.js U
node_modules
routes
index.js U
package-lock.json U
package.json U
index.js U
deportesController.js U
deportes.js U
controllers > eventosController.js > ...
1 const eventos=require('../models/eventos')
2
3
4
5 //primera accion listar todos
6 exports.list=async(req,res)=>{
7
8 try{
9
10 const colEventos=await eventos.find({})
11 res.json(colEventos)
12 }catch(error){
13
14 console.log(error)
15 res.send(error)
16 next()
17 }
18 }
19
20
21 //primera accion ingresar todos
22 exports.add=async(req,res,next)=>{
23 const evento=new eventos(req.body)
24 try{
25
26 await evento.save()
27 res.json({message:'nuevo evento agregado'})
28 }catch(error){
29



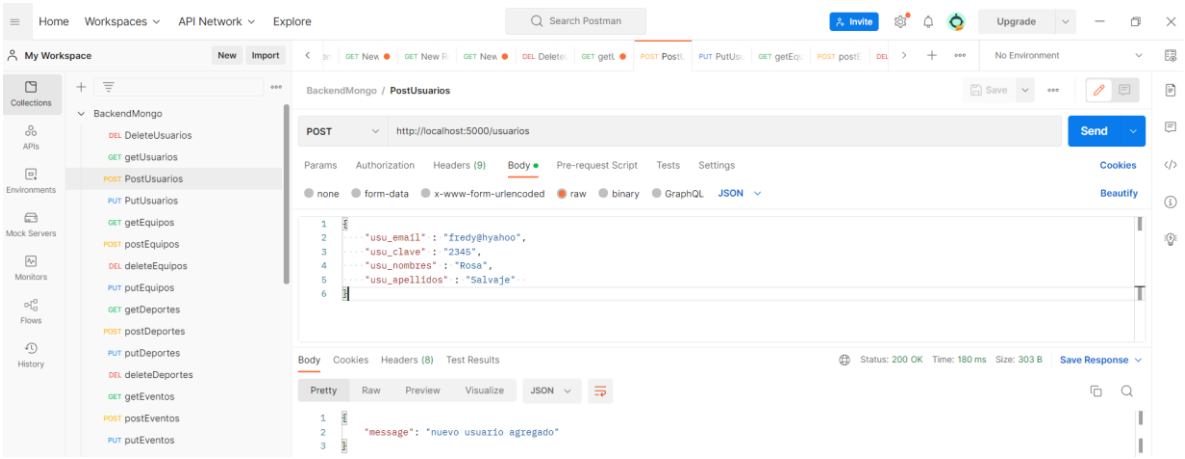
Evidencias de los “endpoint” con el consumo de recursos del API REST

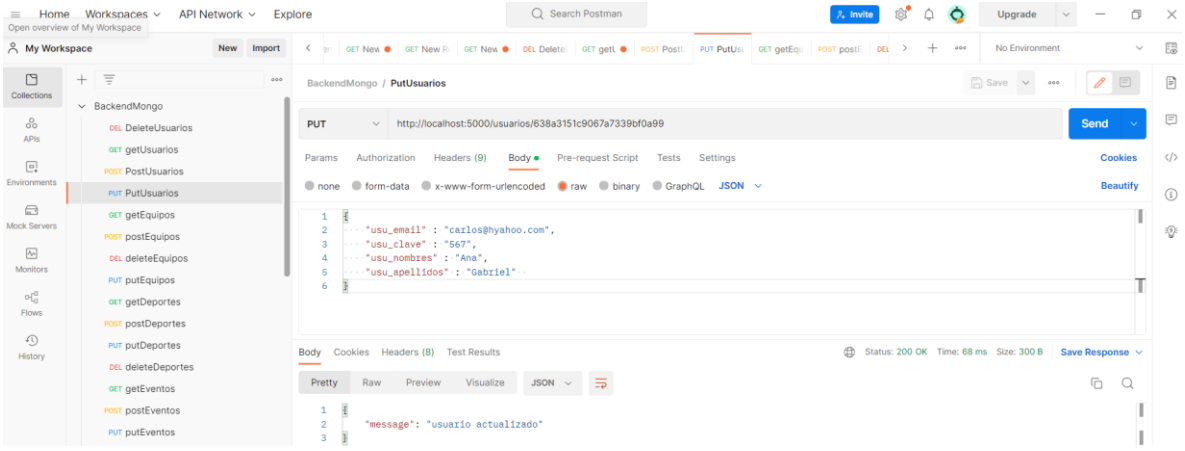
Como evidencia de los “endpoint” donde se visualice el consumo de recursos del API REST.

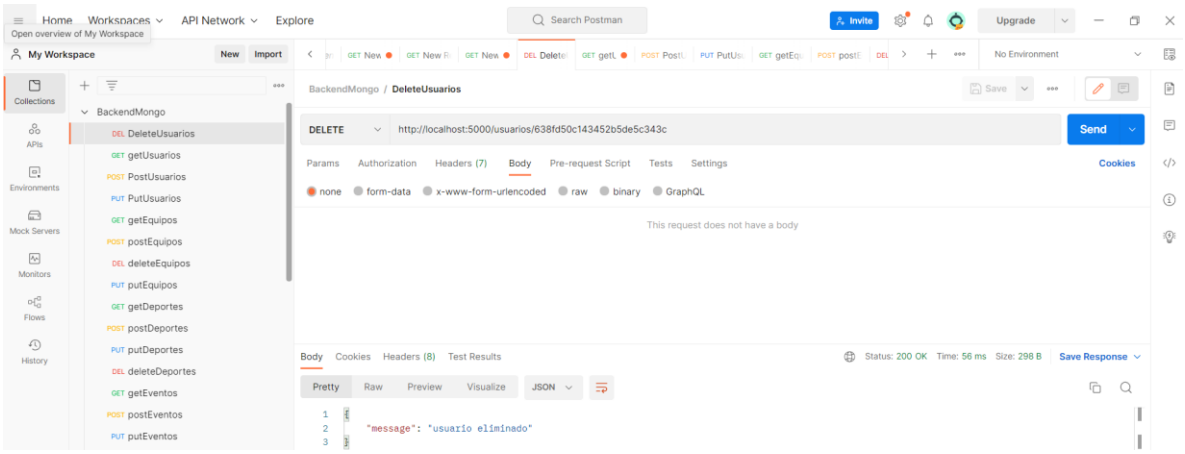


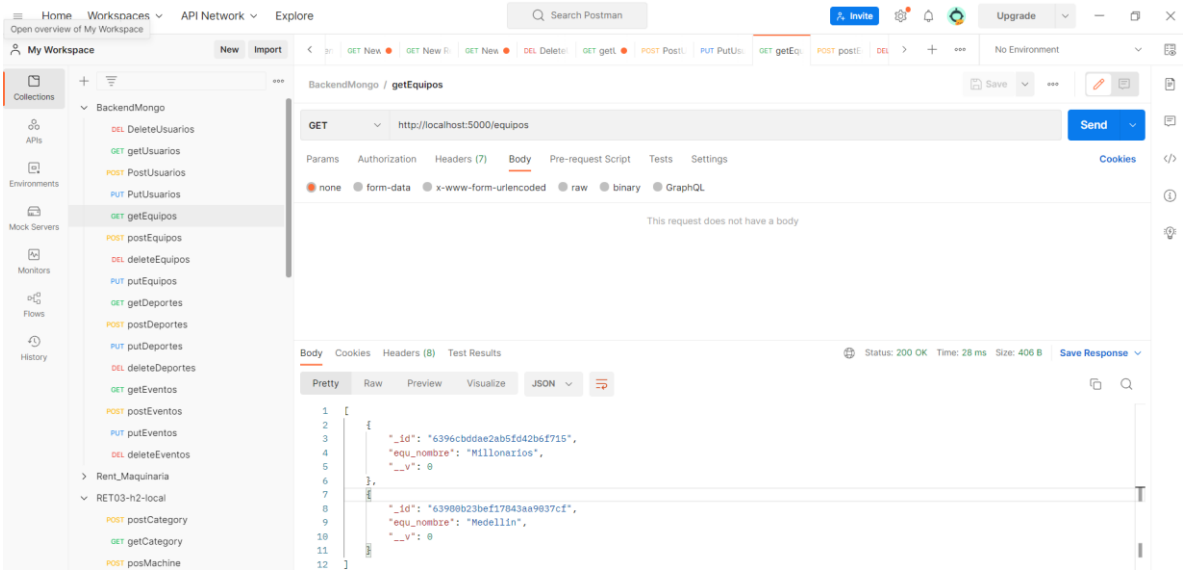

```
[
  {
    "_id": "63894d9048de0aa129252db1",
    "usu_email": "ara@gmail.com",
    "usu_clave": "123",
    "usu_nombrese": "ara",
    "usu_apellidos": "xxx"
  },
  {
    "_id": "63894f1348de0aa129252db3",
    "usu_email": "pepe@gmail.com",
    "usu_clave": "1234",
    "usu_nombres": "pepe",
    "usu_apellidos": "eeee"
  },
  {
    "_id": "638a22fdc9067a7339bf0a96",
    "usu_email": "prueba@com",
    "usu_clave": "123456",
    "usu_nombres": "zeus",
    "usu_apellidos": "aro"
  },
  {
    "_id": "638a2560c9067a7339bf0a97",
    "usu_email": "ejemploOne1@com",
    "usu_clave": "1234567",
    "usu_nombres": "lana",
    "usu_apellidos": "hijo1"
  },
  {
    "_id": "638a3151c9067a7339bf0a98",
    "usu_email": "ejemploMany1@com",
    "usu_clave": "12",
    "usu_nombres": "lanaMany1",
    "usu_apellidos": "hijoMany1"
  },
  {
    "_id": "638a3151c9067a7339bf0a99",
    "usu_email": "carlos@hyahoo.com",
    "usu_clave": "567",
    "usu_nombres": "Ana",
    "usu_apellidos": "Gabriel"
```


```
    },
    {
        "_id": "6396ad1a854b1f55789a83b2",
        "usu_email": "fredy@hyahoo",
        "usu_clave": "2345",
        "usu_nombres": "Rosa",
        "usu_apellidos": "Salvaje",
        "__v": 0
    }
]
```

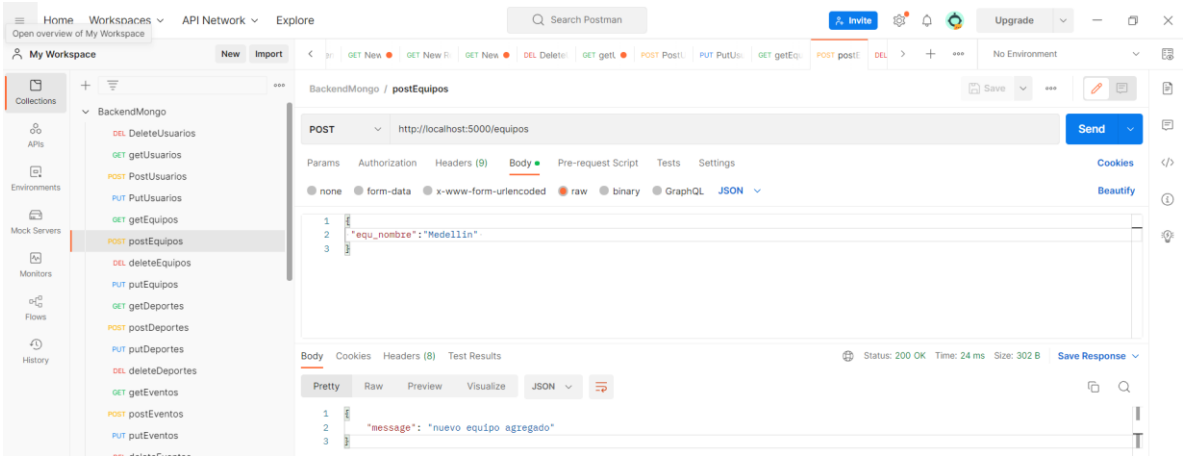


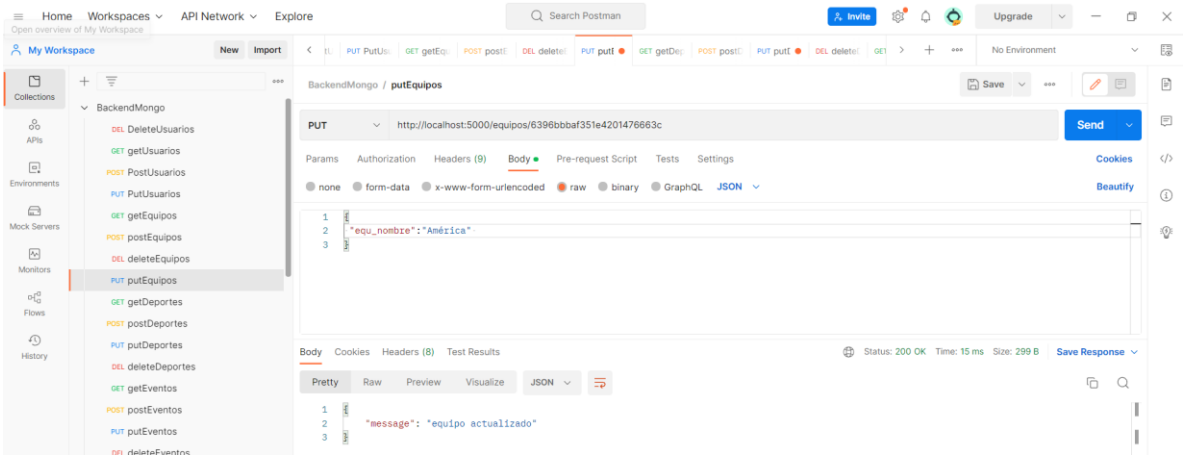


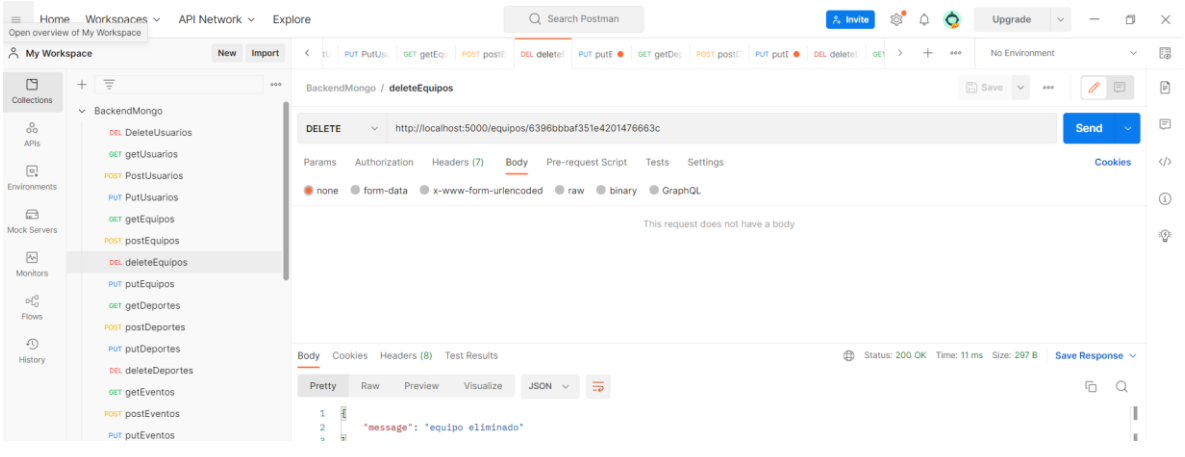


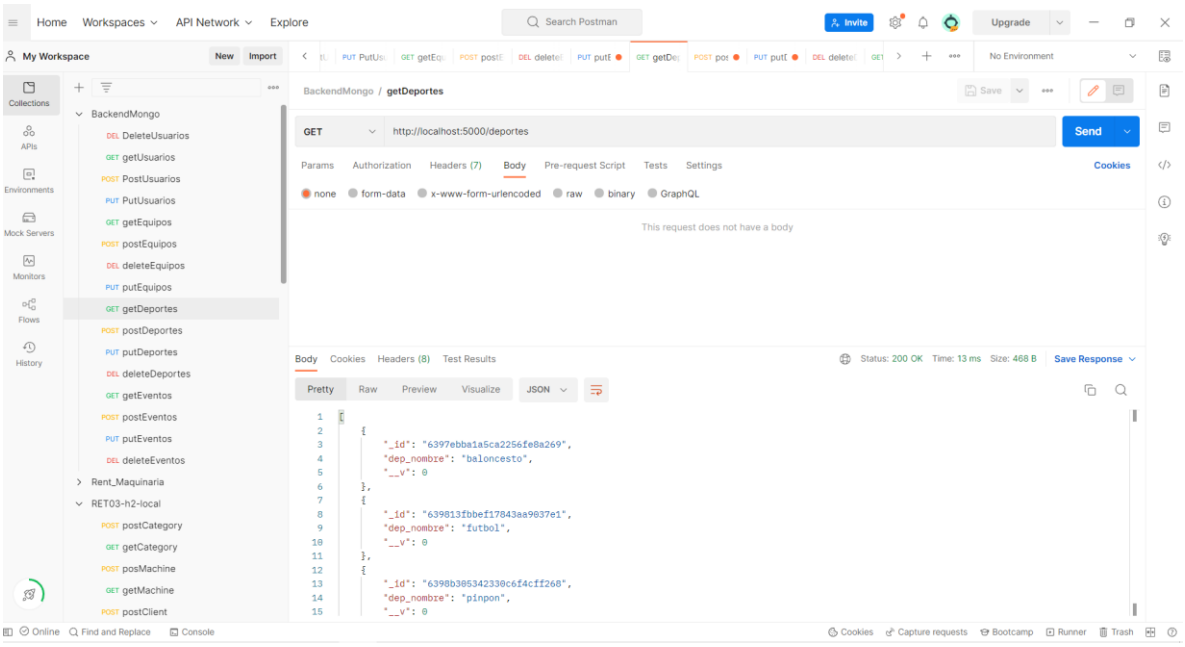

```
[
    {
        "_id": "6396cbddae2ab5fd42b6f715",
        "equ_nombre": "Millonarios",
        "__v": 0
    },
    {
        "_id": "63980b23bef17843aa9037cf",
        "equ_nombre": "Medellin",
        "__v": 0
    }
]
```


Misión
TIC 2022
unab


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DeleteUsuarios
getUsuarios
PostUsuarios
PutUsuarios
getEquipos
postEquipos
deleteEquipos
putEquipos
getDeportes
postDeportes
putDeportes
deleteDeportes
getEventos
postEventos
putEventos
BackendMongo / postEquipos
Save
POST
http://localhost:5000/equipos
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"equ_nombre":"Medellin"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 24 ms
Size: 302 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "nuevo equipo agregado"
}

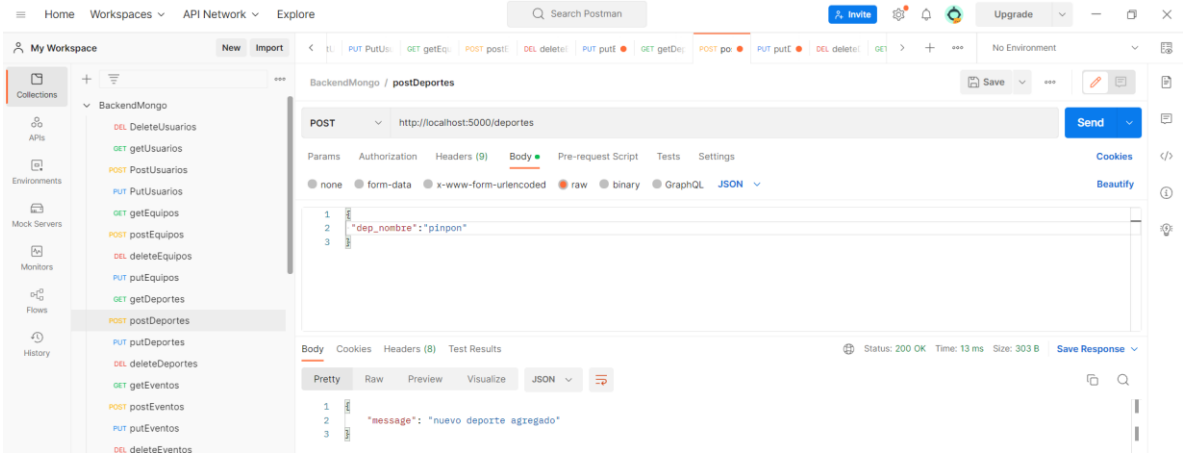

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DeleteUsuarios
getUsuarios
PostUsuarios
PutUsuarios
getEquipos
postEquipos
deleteEquipos
putEquipos
getDeportes
postDeportes
putDeportes
deleteDeportes
getEventos
postEventos
putEventos
BackendMongo / putEquipos
Save
PUT
http://localhost:5000/equipos/6396bbbaf351e4201476663c
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"equ_nombre":"América"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 15 ms
Size: 299 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "equipo actualizado"
}

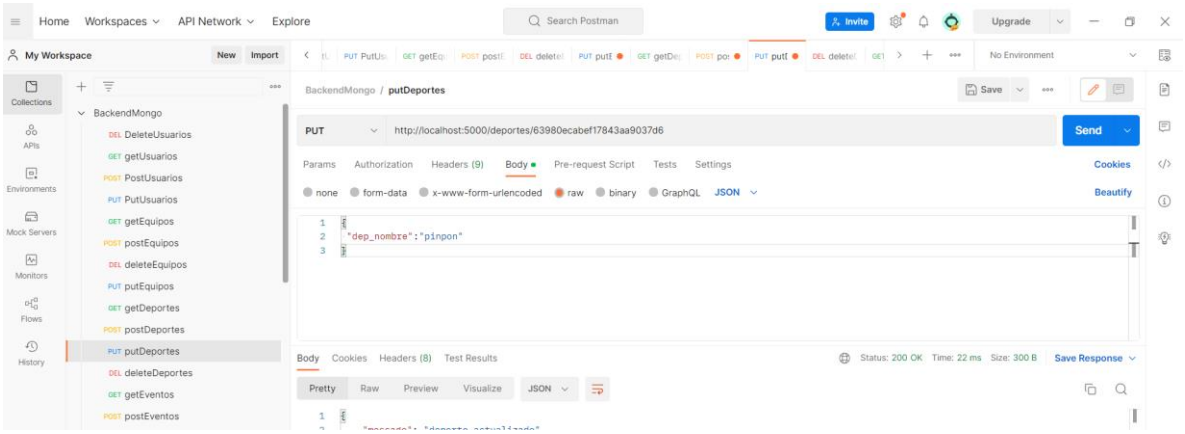

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DeleteUsuarios
getUsuarios
PostUsuarios
PutUsuarios
getEquipos
postEquipos
deleteEquipos
putEquipos
getDeportes
postDeportes
putDeportes
deleteDeportes
getEventos
postEventos
putEventos
BackendMongo / deleteEquipos
Save
DELETE
http://localhost:5000/equipos/6396bbbaf351e4201476663c
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
This request does not have a body
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 11 ms
Size: 297 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "equipo eliminado"
}

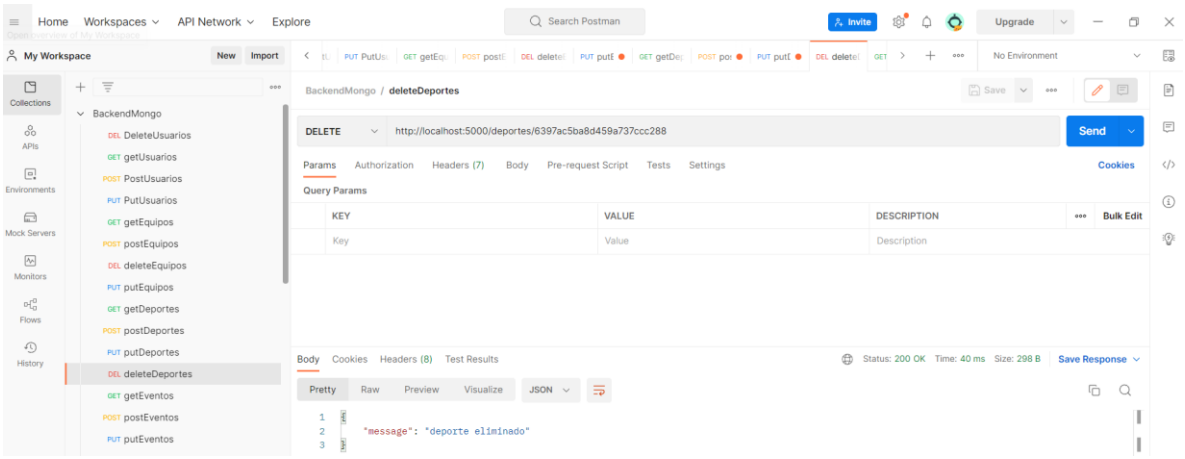



```
[
  {
    "_id": "6397ebba1a5ca2256fe8a269",
    "dep_nombre": "baloncesto",
    "__v": 0
  },
  {
    "_id": "639813fbbef17843aa9037e1",
    "dep_nombre": "futbol",
    "__v": 0
  },
  {
    "_id": "6398b305342330c6f4cff268",
    "dep_nombre": "pinpon",
    "__v": 0
  }
]
```

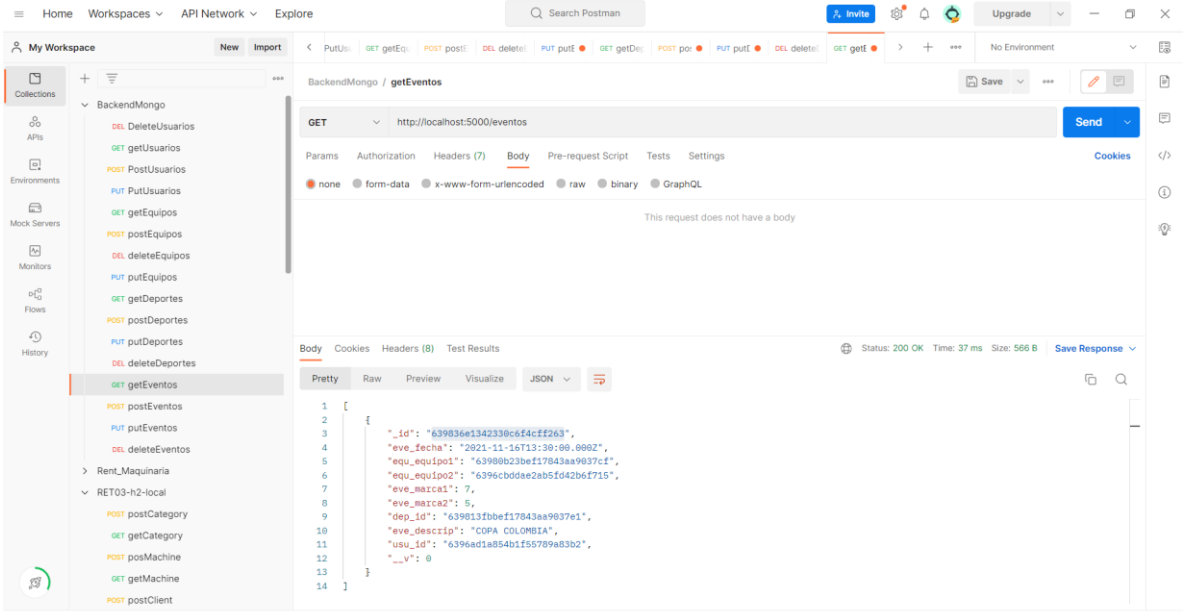

Misión
TIC 2022
unab

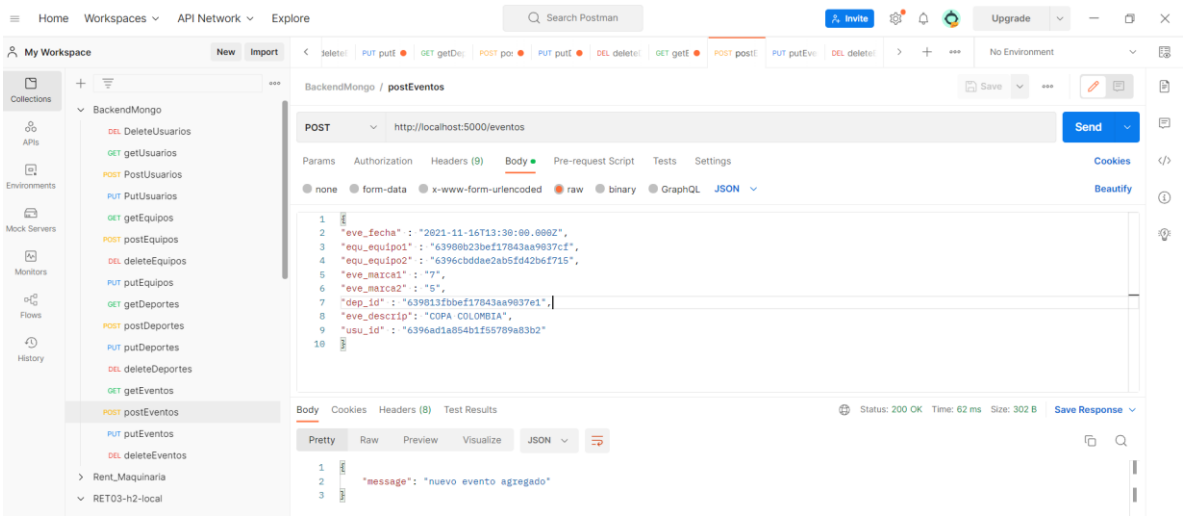

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DEL DeleteUsuarios
GET getUsuarios
POST PostUsuarios
PUT PutUsuarios
GET getEquipos
POST postEquipos
DEL deleteEquipos
PUT putEquipos
GET getDeportes
POST postDeportes
PUT putDeportes
DEL deleteDeportes
GET getEventos
POST postEventos
PUT putEventos
DEL deleteEventos
BackendMongo / postDeportes
Save
POST
http://localhost:5000/deportes
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"dep_nombre":"pinpon"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 13 ms
Size: 303 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "nuevo deporte agregado"
}

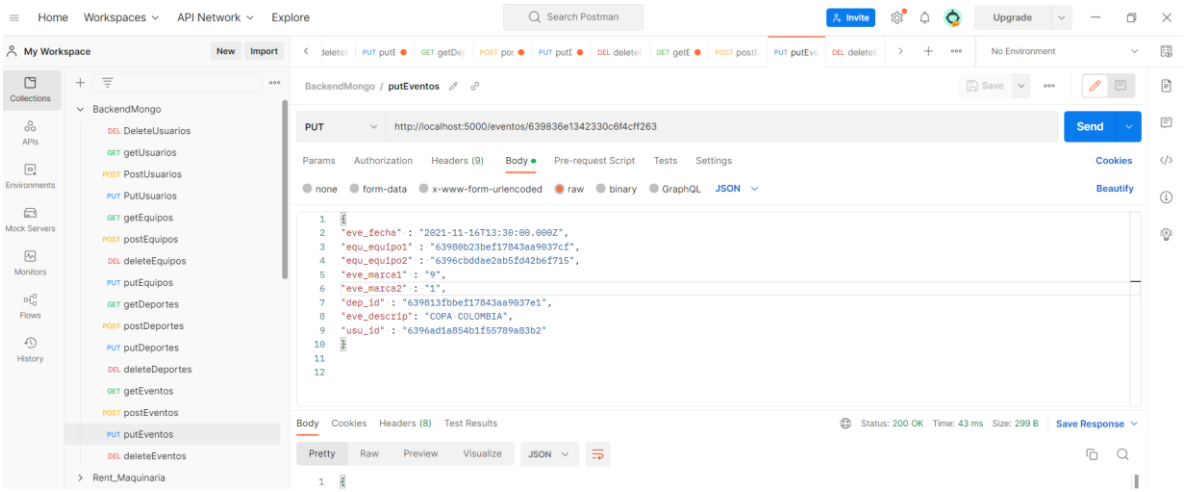

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DEL DeleteUsuarios
GET getUsuarios
POST PostUsuarios
PUT PutUsuarios
GET getEquipos
POST postEquipos
DEL deleteEquipos
PUT putEquipos
GET getDeportes
POST postDeportes
PUT putDeportes
DEL deleteDeportes
GET getEventos
POST postEventos
BackendMongo / putDeportes
Save
PUT
http://localhost:5000/deportes/63980ecabef17843aa9037d6
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"dep_nombre":"pinpon"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 22 ms
Size: 300 B
Save Response
Pretty
Raw
Preview
Visualize
JSON

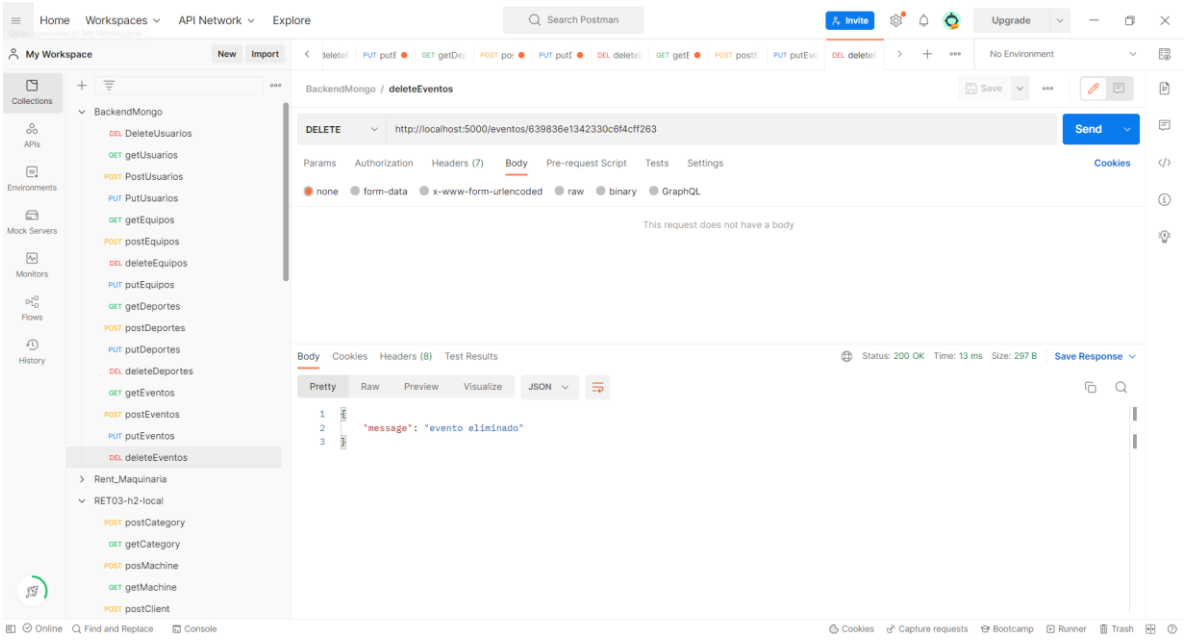

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DEL DeleteUsuarios
GET getUsuarios
POST PostUsuarios
PUT PutUsuarios
GET getEquipos
POST postEquipos
DEL deleteEquipos
PUT putEquipos
GET getDeportes
POST postDeportes
PUT putDeportes
DEL deleteDeportes
GET getEventos
POST postEventos
PUT putEventos
BackendMongo / deleteDeportes
Save
DELETE
http://localhost:5000/deportes/6397ac5ba8d459a737ccc288
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 40 ms
Size: 298 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "deporte eliminado"
}


Mision
TIC2022
unab


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DeleteUsuarios
getUsuarios
PostUsuarios
PutUsuarios
getEquipos
postEquipos
deleteEquipos
putEquipos
getDeportes
postDeportes
putDeportes
deleteDeportes
getEventos
postEventos
putEventos
deleteEventos
Rent_Maquinaria
RET03-h2-local
postCategory
getCategory
posMachine
getMachine
postClient
BackendMongo / getEventos
Save
GET
http://localhost:5000/eventos
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
This request does not have a body
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 37 ms
Size: 566 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"_id": "639836e1342330c6f4cff263",
"eve_fecha": "2021-11-16T13:30:00.000Z",
"equ_equipo1": "63980b23bef17843aa9037cf",
"equ_equipo2": "6396cbddae2ab5fd42b6f715",
"eve_marca1": 7,
"eve_marca2": 5,
"dep_id": "639813fbbef17843aa9037e1",
"eve_descrip": "COPA COLOMBIA",
"usu_id": "6396ad1a854b1f55789a83b2",
"__v": 0
}
]
Online
Find and Replace
Console
Cookies
Capture requests
Bootcamp
Runner
Trash


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
BackendMongo
DeleteUsuarios
getUsuarios
PostUsuarios
PutUsuarios
getEquipos
postEquipos
deleteEquipos
putEquipos
getDeportes
postDeportes
putDeportes
deleteDeportes
getEventos
postEventos
putEventos
deleteEventos
Rent_Maquinaria
RET03-h2-local
BackendMongo / postEventos
Save
POST
http://localhost:5000/eventos
Send
Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"eve_fecha" : "2021-11-16T13:30:00.000Z",
"equ_equipo1" : "63980b23bef17843aa9037cf",
"equ_equipo2" : "6396cbddae2ab5fd42b6f715",
"eve_marca1" : "7",
"eve_marca2" : "5",
"dep_id" : "639813fbbef17843aa9037e1",
"eve_descrip": "COPA COLOMBIA",
"usu_id" : "6396ad1a854b1f55789a83b2"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 62 ms
Size: 302 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"message": "nuevo evento agregado"
}

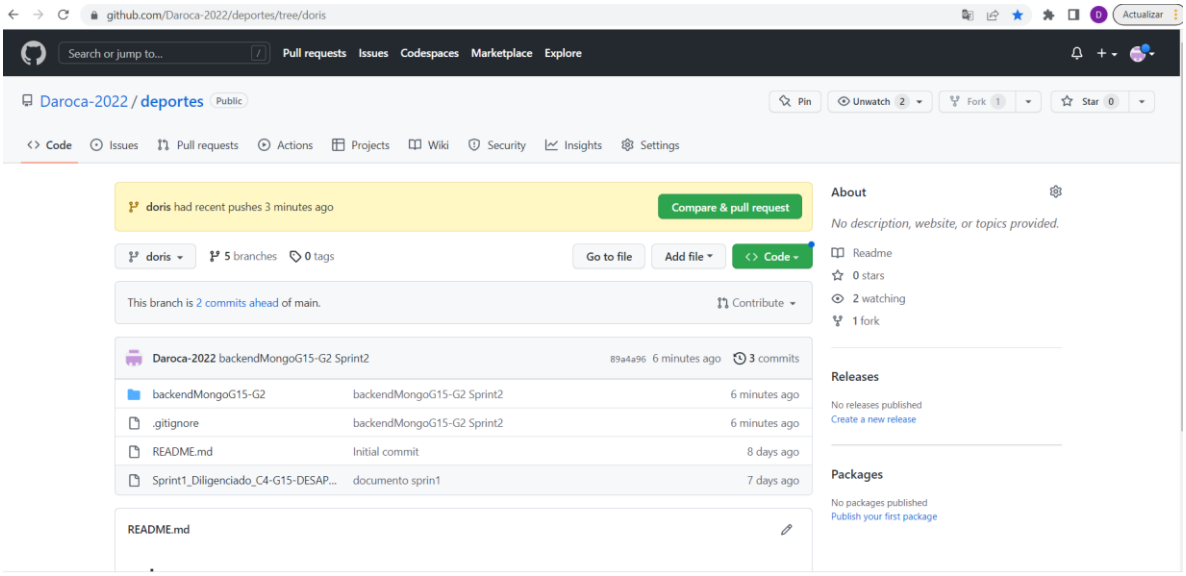




Evidencia GitLab o GitHub

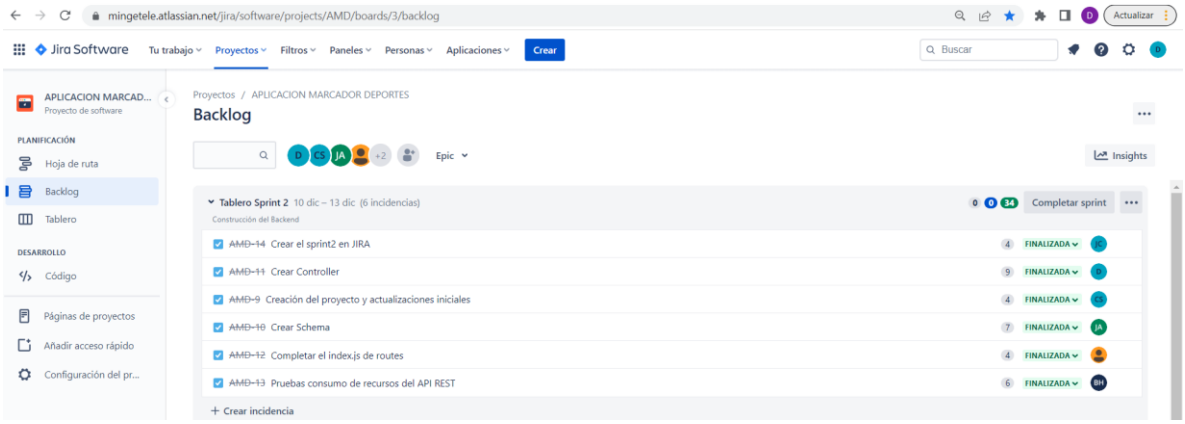
Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versión)

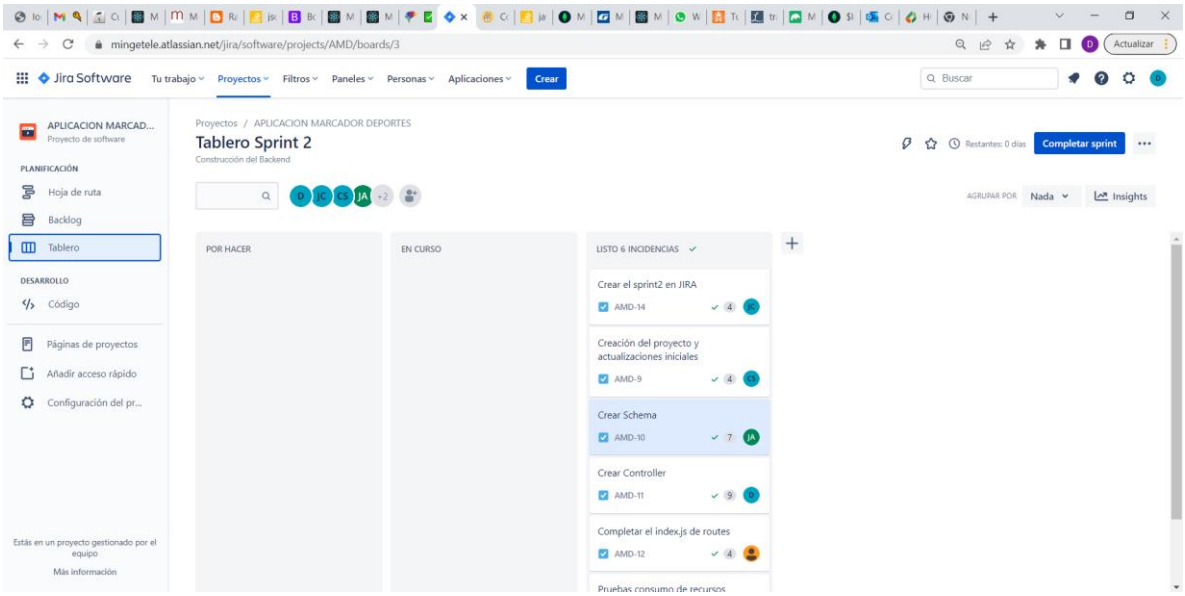
url del proyecto Github:
https://github.com/Daroca-2022/deportes.git



Evidencia JIRA (Seguimiento del proyecto)

Como evidencia del seguimiento del proyecto con la metodología ágil SCRUM, utilizando el software JIRA, se debe presentar capturas de pantalla donde se visualice la ejecución de los Sprint con las historias de usuario relacionadas con el desarrollo del Backend.

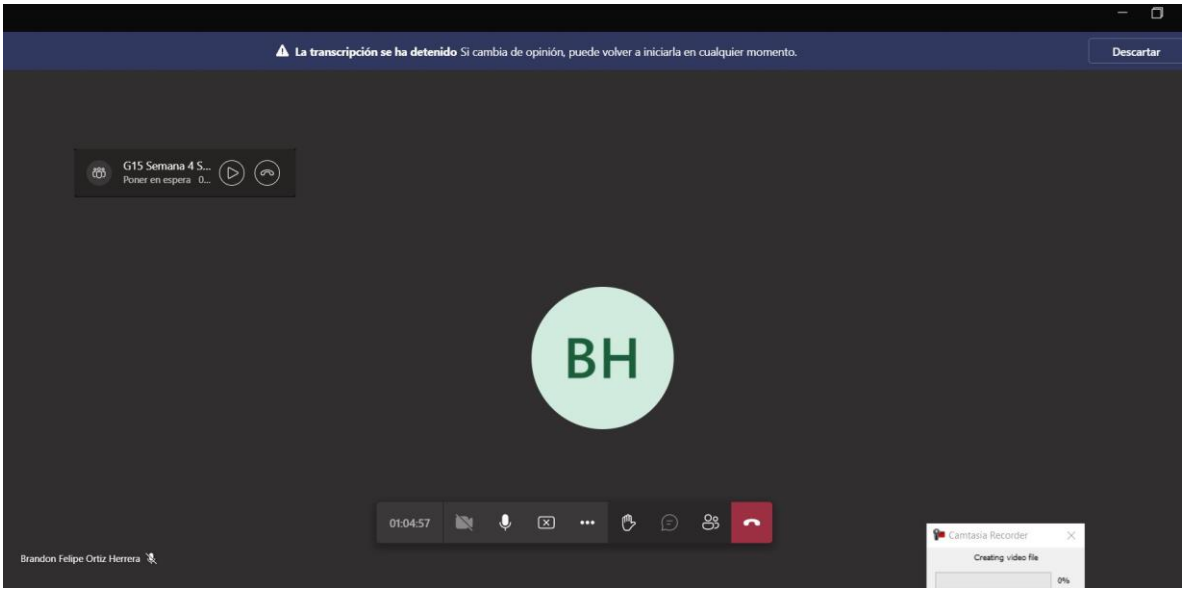




Evidencias de las Reuniones de Equipo

Como evidencia de las reuniones que efectúa el equipo del proyecto, presentar capturas de pantalla de las reuniones efectuadas y si lo consideran pertinente algunas actas de las reuniones.

Reunión 9 de diciembre de 2022



FECHA: 09 DE DICIEMBRE DE 2022

HORA: 10:22 P.M. A 11:26 P.M.

ASUNTO: Revisar el formato del Sprint 1 diligenciado para entrega

PARTICIPANTES:

- Juan Carlos Álvarez
- Doris Aroca
- Felipe Álvarez
- Brandon Ortiz Herrera

TEMAS TRATADOS:

- Se acuerda desarrollar el Backend en Mongo
- Nombre del proyecto: PROYECTOG15-G2
- Se definieron las tareas a ingresar para el sprint 2 en la aplicación JIRA:
  1- Creación del proyecto – subir servidor y dependencia
  2- Crear los esquemas: Juan Carlos Alvarez puntaje 7
  usuarios.js
  eventos.js
  deportes.js
  equipos.js –Crear controlares – Doris Aroca Tique puntaje 9
  usuariosControllers.js
  eventosControllers.js
  deportesControllers.js
  equiposControllers.js
  3- Completar el index.js de routes – Andres Felipe Alvarez puntaje 4.5
  4- Pruebas consumo de recursos del API REST - Brandon Felipe Ortiz - 6
  5- Crear el sprint en JIRA – Jorge Moreno puntaje 4
  Lanzamiento del Sprint2 sabado 10 de 2022 y termina lunes 12 de diciembre de 2022

- Evidencias Github: todos realizamos el proyecto completo y actualizados Github