```
// Actualizar un miembro
exports.updateMember = async (req, res) => {
  try {
    const { id } = req.params;
    const updatedMember = await Member.findByIdAndUpdate(id, req.body, { new: true });

    if (!updatedMember) {
      return res.status(404).json({ message: 'Miembro no encontrado' });
    }

    res.status(200).json({
      message: 'Miembro actualizado exitosamente',
      member: updatedMember,
    });
  } catch (error) {
    res.status(500).json({ message: 'Error al actualizar miembro', error: error.message });
  }
};

// Eliminar un miembro
exports.deleteMember = async (req, res) => {
  try {
    const { id } = req.params;
    const deletedMember = await Member.findByIdAndDelete(id);

    if (!deletedMember) {
      return res.status(404).json({ message: 'Miembro no encontrado' });
    }

    res.status(200).json({ message: 'Miembro eliminado correctamente' });
  } catch (error) {
    res.status(500).json({ message: 'Error al eliminar miembro', error: error.message });
  }
```

Team 4
Leader: Josue Marin
TName: Alpha Team
Project: Holistic CEnter
GitHub:

https://github.com/marinjosue/AWD22962024AlphaTeam
DB: https://railway.com/invite/16eq-kHtDvv
Inspector Team: TNT (3) Leader: Josue Guallichico

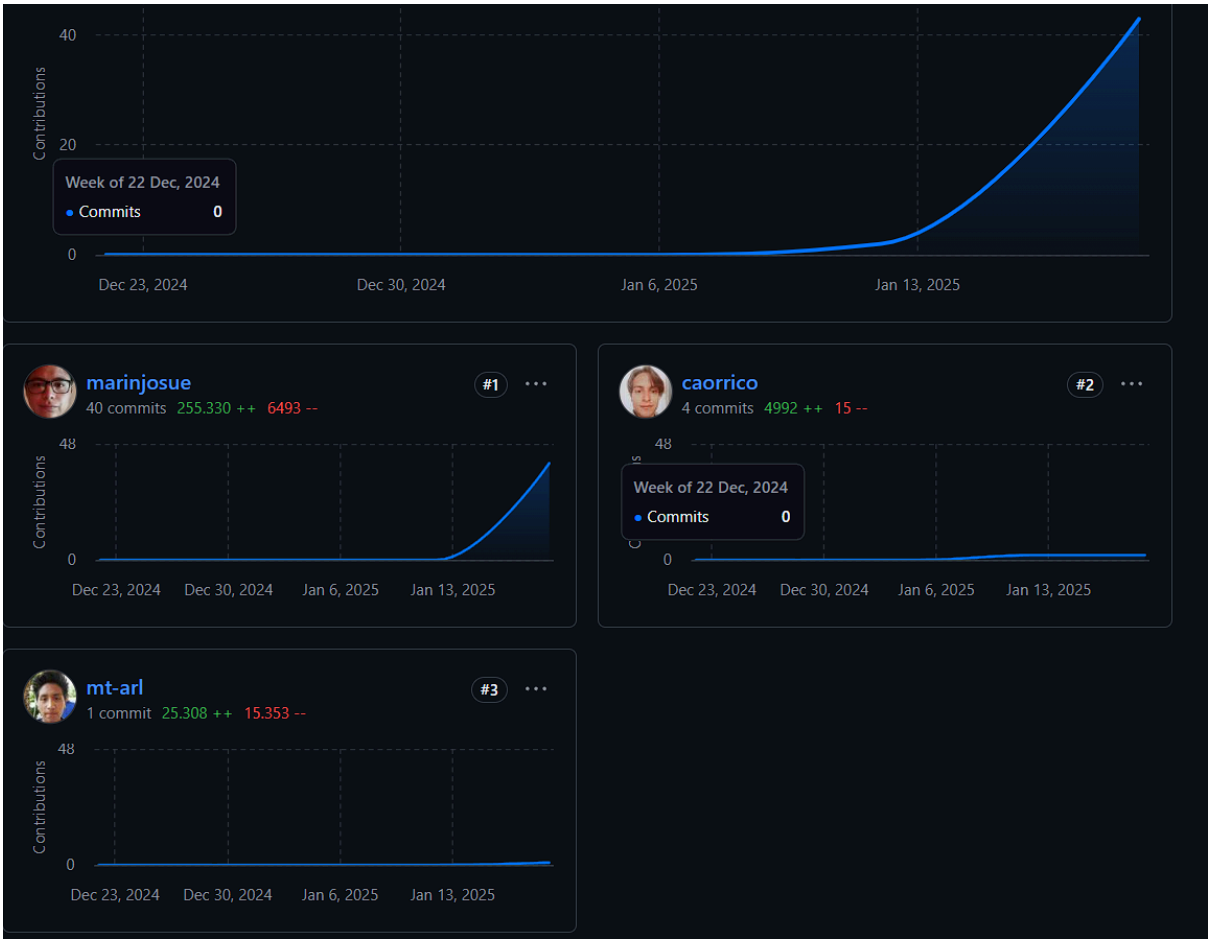
10. Lugmaña Matias
11. Marin Josue
12. Orrico Camilo

Item	Eval	Rubric	Comments
1. GitHub	80/10	commits and code	They should to write with clean code

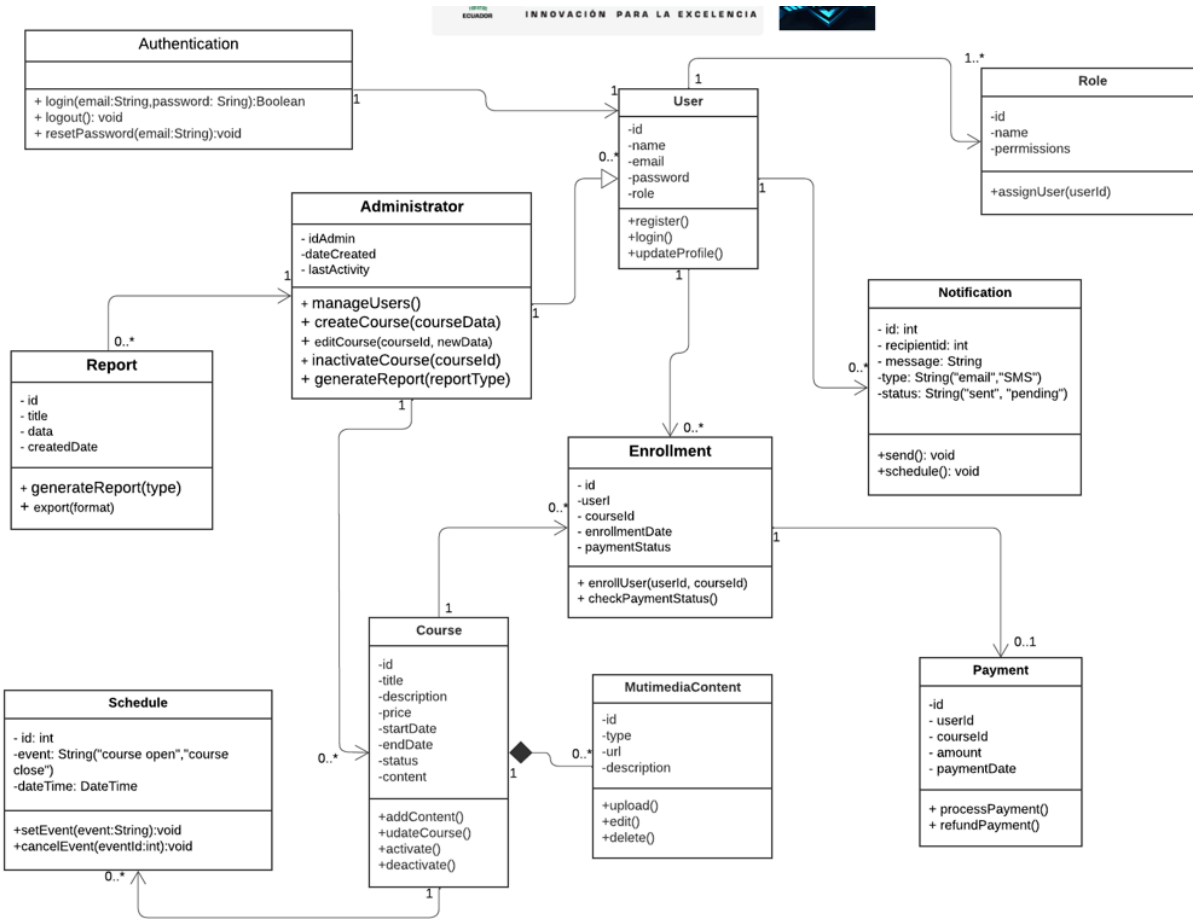
	0		
2. Class Diagram:	10/10	at least 10 classes, 0.5 points per class (attributes and method	The diagram correctly have good names and structure.
3. UC Diagram:	8.5/10	at least 10 classes, 0.5 point per class (attributes and method), relations. 0.5 per relation	There are cases that do not follow the format: verb+complement, some cases need to be more specific.
4. URIs (Doc)	25/30	At least 10 actions, 1 per use case	They not have 10 actions per use case.
5. CRUD Operations	8/10	50 URIs, 0.4 per URI: Method, URI, req, resp, Description	The URIs are incomplete
6. Functionality (business rules)	15/20	40 CRUD URIs working	They are working but there are not enough URIS
7. Clean code (MVC)	17/20	Run the project, every URI that works for CRUD is word .25 Every business rule → 1	Unclear names of variables unused commented code Wrong horizontal formatting
TOTAL	163.5		

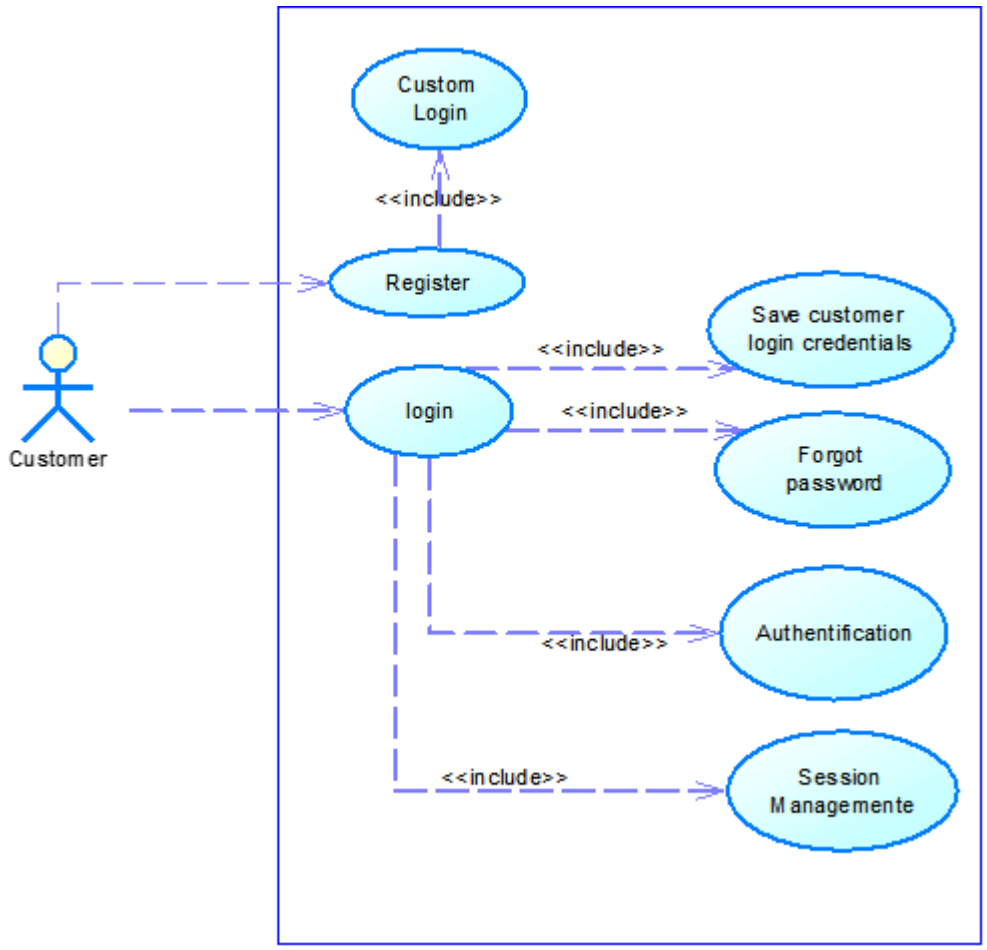
Evidence

Commits and Code



Class Diagram 10/10


Authentication
+ login(email:String,password: Sring):Boolean
+ logout(): void
+ resetPassword(email:String):void
User
-id
-name
-email
-password
-role
+register()
+login()
+updateProfile()
Role
-id
-name
-perrmissions
+assignUser(userId)
Administrator
- idAdmin
-dateCreated
- lastActivity
+ manageUsers()
+ createCourse(courseData)
+ editCourse(courseId, newData)
+ inactivateCourse(courseId)
+ generateReport(reportType)
Report
- id
- title
- data
- createdDate
+ generateReport(type)
+ export(format)
Notification
- id: int
- recipientid: int
- message: String
-type: String("email","SMS")
-status: String("sent", "pending")
+send(): void
+schedule(): void
Enrollment
- id
-userI
- courseId
- enrollmentDate
- paymentStatus
+ enrollUser(userId, courseId)
+ checkPaymentStatus()
Course
-id
-title
-description
-price
-startDate
-endDate
-status
-content
+addContent()
+udateCourse()
+activate()
+deactivate()
MutimediaContent
-id
-type
-url
-description
+upload()
+edit()
+delete()
Payment
-id
- userId
- courseId
- amount
- paymentDate
+ processPayment()
+ refundPayment()
Schedule
- id: int
-event: String("course open","course close")
-dateTime: DateTime
+setEvent(event:String):void
+cancelEvent(eventId:int):void

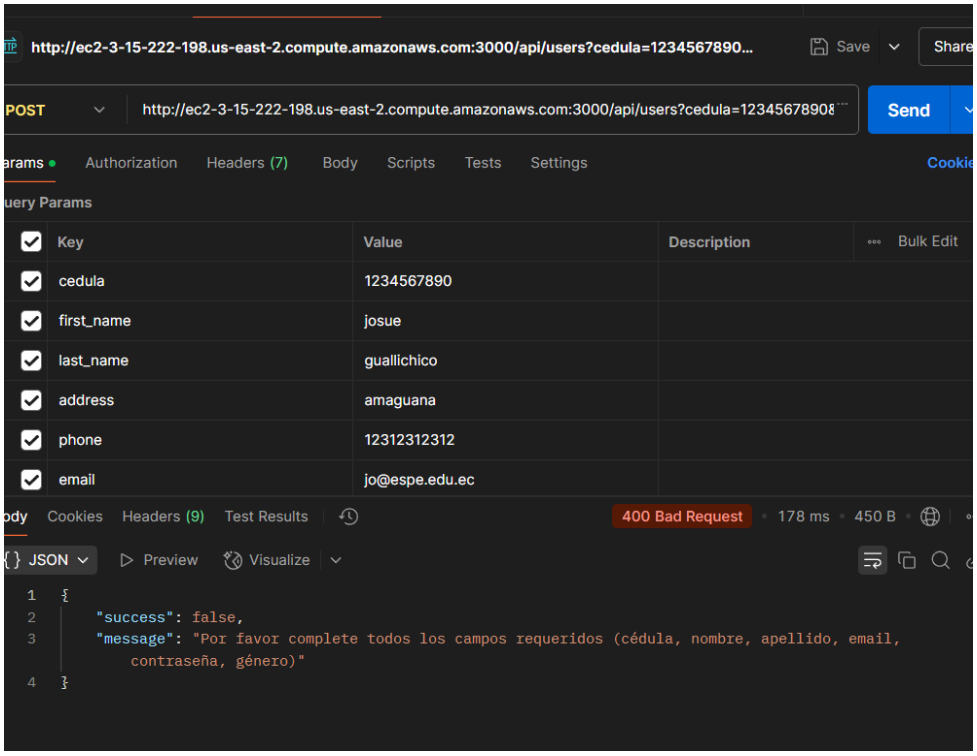


Logic Business Methods

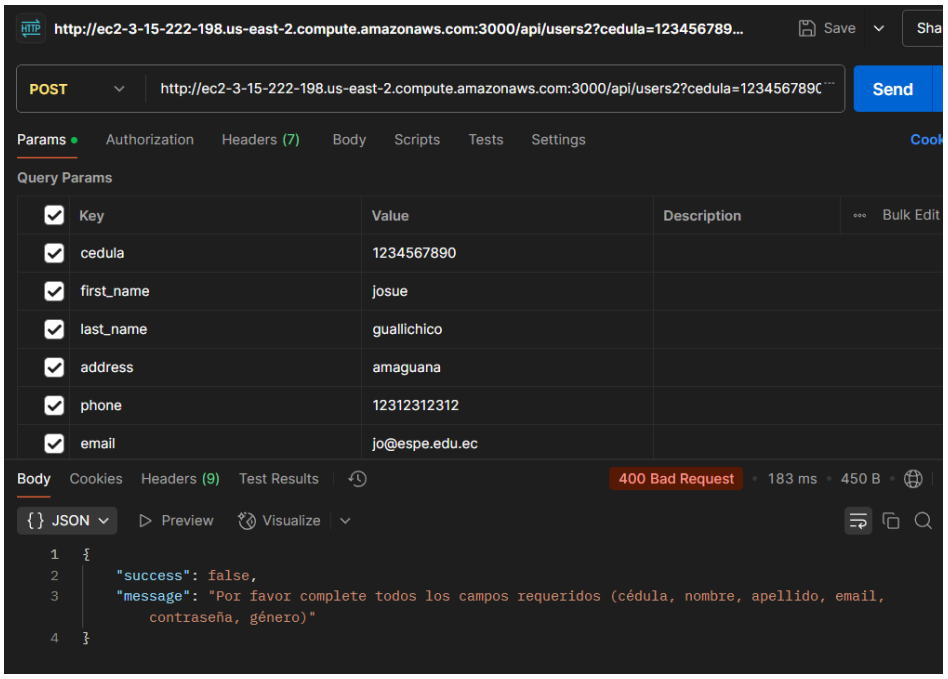
- PATCH http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users/{id}/password
- PATCH http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2/1/password
- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/auth/login
- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/auth/check-google-user
- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/auth/google-login
-

CRUD Methods

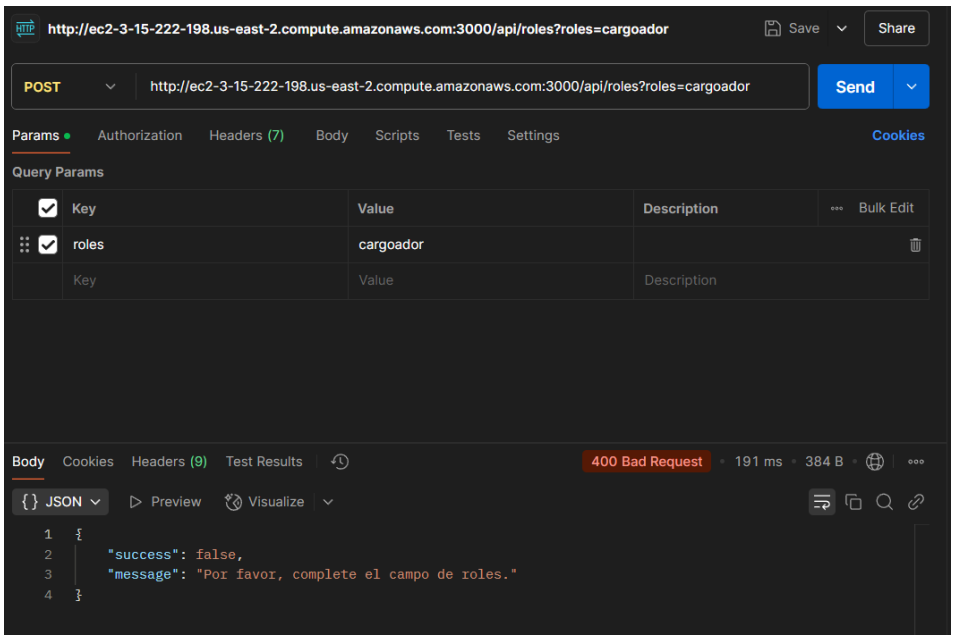
- **Create**
  - POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users

- 

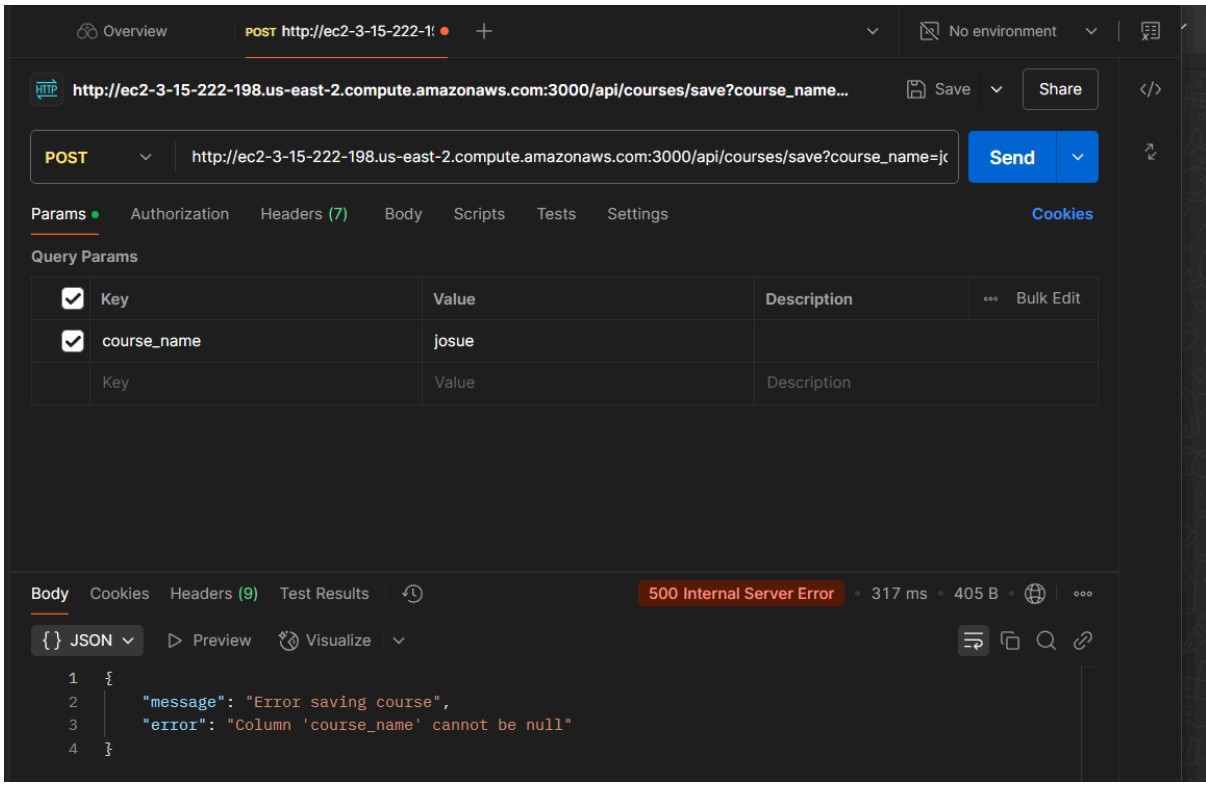
- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2



- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/roles/

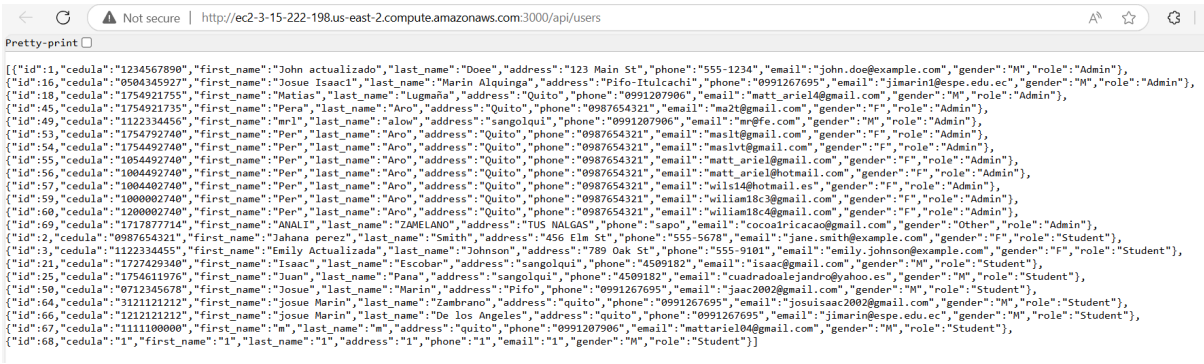


- POST http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/courses/save



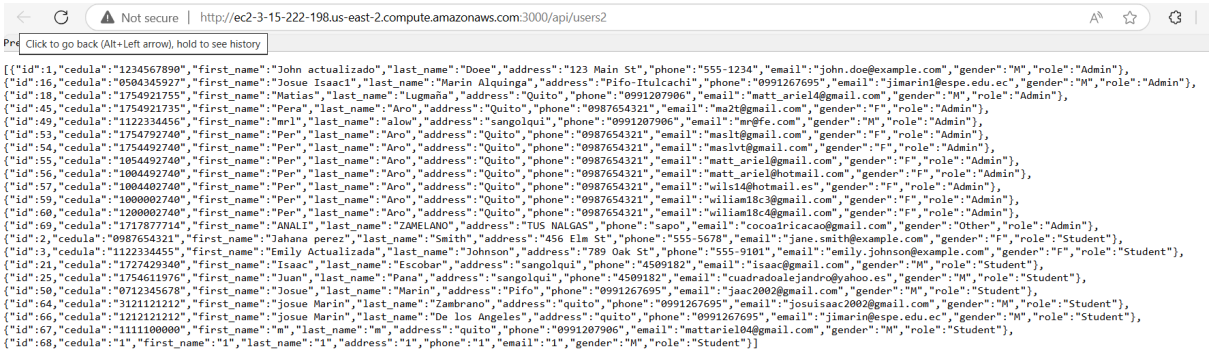
- **Read**

- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users

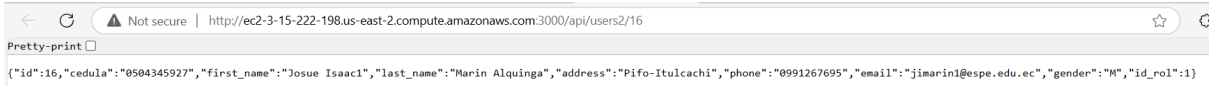


- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users/1

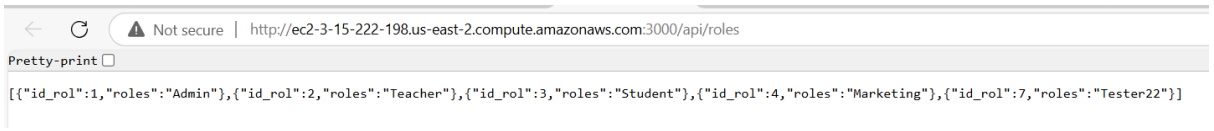
- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2



- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2/16

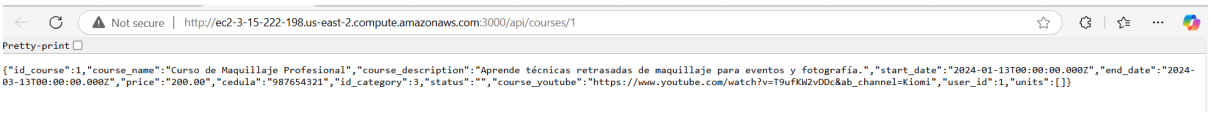


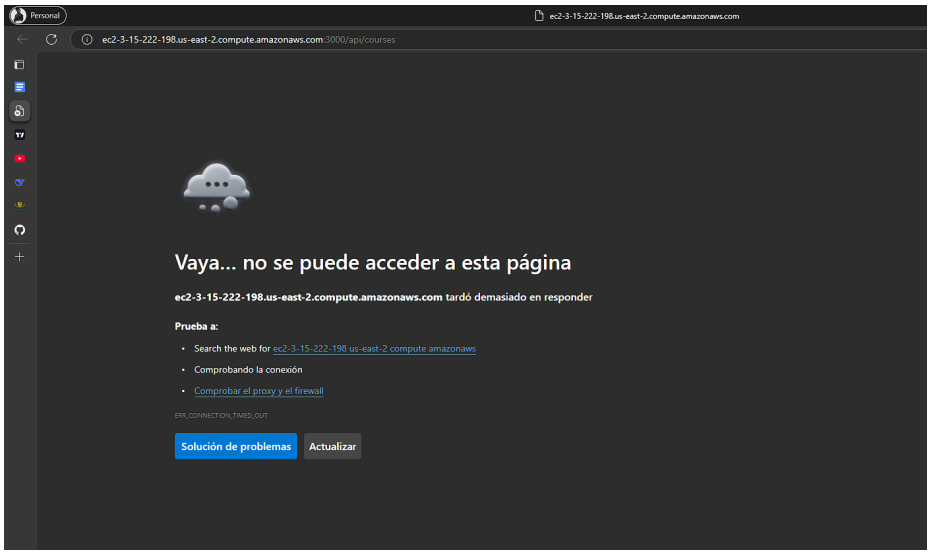
- GET
- http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/roles



- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/courses

- GET http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/courses/1



- **Update**
  - PUT http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users/1
  - 

  - PUT http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2/1
  - PUT http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/roles/:id
  - POST AND PUT http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/courses/save
- **Delete**
  - DELETE http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users/1
  - DELETE http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/users2/1

- DELETE http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/roles/1
- DELETE http://ec2-3-15-222-198.us-east-2.compute.amazonaws.com:3000/api/courses/{id}

Clean code:

```
const express = require('express');
const cors = require('cors');
const userRoutes = require('./routes/userRoutes');
const userRoutes2 = require('./routes/userRoutes2');
const courseRoutes = require('./routes/courseRoutes');
const authRoutes = require('./routes/authRoutes');
const rolesRoutes = require('./routes/rolesRoute');
```

Some variables are not clear enough

```
// Llamar la función (ejemplo)
// enviarCorreo(
//   "wmconya@espe.edu.ec",
//   "Hola desde Node.js",
//   "Este es un mensaje enviado desde Node.js usando SMTP y Nodemailer"
// );
```

Commented Unused code

```
document.getElementById('addUserForm').addEventListener('submit', async (event) => {
    event.preventDefault();
    const formData = new FormData(event.target);

    try {
        const response = await fetch('../controller/add_users.php', {
            method: 'POST',
            body: formData
        });
        const result = await response.json();

        if (result.success) {
            alert('Usuario agregado correctamente');
            cargarUsuarios();
            const modal = bootstrap.Modal.getInstance(document.getElementById('addUserModal'));
            modal.hide();
        } else {
            alert('Error al agregar usuario: ' + result.message);
        }
    } catch (error) {
        console.error('Error al agregar usuario:', error);
    }
});
```

Not horizontally formatted

Team 5
Leader: Proaño Jose

TName: Three Musketeers

Project: MADECOR
GitHub:
https://github.com/JoseProano/AWD22962024Team5
DB: https://railway.com/invite/qHxi9jiFGjc

Inspector Team:

13. Pabon Elkin
14. Darwin Panchez
15. Proaño Jose

Item	Eval	Rubric	Comments
1. GitHub	/100	commits and code	
2. Class Diagram:	8.5/10	at least 10 classes, 0.5 points per class (atributtes and methot	Improve class relations make the diagram in English.
3. UC Diagram:	9.5/10	at least 10 classes, 0.5 point per class (attributes and method), relations. 0.5 per relation	improve use case names: manager user, customer and productcs
4. URIs (Doc)	25/30	50 URIs, 0.6 per URI: Method, URI, req, resp, Description	URIS are not complete