



# Entrega - Habilitación Ciclo 3

Julian David Betancourt Cuervo

# Descripción del proceso:

Desafortunadamente el grupo asignado a mi no me ayudó mucho durante el proceso, y hoy, último día de entrega me hicieron saber que el FrontEnd no funcionaba. Yo hice el BackEnd, el prototipo, cree el board de Trello, las pocas tareas que ahí se metieron. Adjunto evidencia del backend funcionando y sinceramente el FrontEnd fué un intento por acomodar un proyecto ya hecho, pero no lo logré del todo.





# Evidencia:

Link al prototipo:

<https://www.figma.com/file/xMKaEsWkznp2scu4yQ03CS/Ciclo-3?node-id=108%3A81>

Link al Trello:

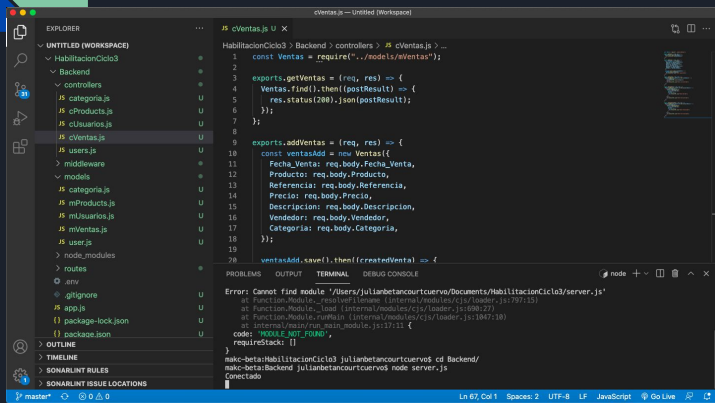
<https://trello.com/b/PIDB68IR/habilitaci%C3%B3n-ciclo-3>

Link al repositorio:

<https://github.com/JulianBeta/HabilitacionCiclo3/tree/master>

Acceso al los chats de whatsapp (Para evidenciar que se trató de organizar el equipo para hacer una sola presentación):

# Evidencia Gráfica:



```
const express = require('express');
const app = express();
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const User = mongoose.model('User');

// GET /usuarios
app.get('/usuarios', async (req, res) => {
  const usuarios = await User.find().exec();
  res.json(usuarios);
});

// POST /usuarios
app.post('/usuarios', async (req, res) => {
  const { nombre, apellido, correo, telefono } = req.body;
  const password = req.body.password;

  // Validar campos obligatorios
  if (!nombre || !apellido || !correo || !telefono || !password) {
    return res.status(400).json({ error: 'Todos los campos son obligatorios' });
  }

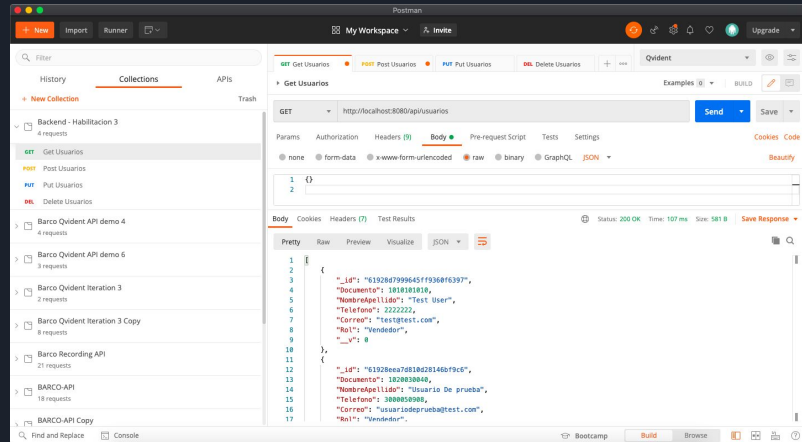
  // Verificar si el correo ya existe
  const usuarioExistente = await User.findOne({ correo });
  if (usuarioExistente) {
    return res.status(400).json({ error: 'El correo ya está registrado' });
  }

  // Hashear la contraseña
  const salt = bcrypt.genSalt(10);
  const hash = bcrypt.hashSync(password, salt);

  // Crear el usuario
  const nuevoUsuario = new User({
    nombre,
    apellido,
    correo,
    telefono,
    password: hash
  });

  await nuevoUsuario.save();
  res.status(201).json(nuevoUsuario);
});

// GET /usuarios/:id
app.get('/usuarios/:id', async (req, res) => {
  const { id } = req.params;
  const usuario = await User.findById(id).exec();
  if (!usuario) {
    return res.status(404).json({ error: 'Usuario no encontrado' });
  }
  res.json(usuario);
});
```



# Evidencia Gráfica:

