

SISTEMAS DIGITALES



INTEGRANTES:

PATRICIO BETANCOURT

JOAN RIOFRÍO

YOVIN URREGO

1. Índice

1.	Índice	2
2.	Nombre proyecto	3
3.	Integrantes	3
4.	Descripción del problema	3
5.	Ejecución de actividades	3
6.	Clasificación de requisitos por autor	4
7.	División de módulos	5
8.	Diagrama de Clases	6
9.	Diagrama de Despliegue	6
10.	Modelos de la base de datos.....	7
11.	Controlador ejemplo	¡Error! Marcador no definido.

2. Nombre proyecto

VLAB-UNL

3. Integrantes

- Yovin Urrego
- Joan Riofrío
- Diego Riofrío
- Patricio Betancourt

4. Descripción del problema

Se necesita crear una aplicación web, a la que puedan tener acceso estudiantes y profesores de la UNL (Universidad Nacional de Loja), donde los profesores tengan libertad de crear y modificar cursos y prácticas para los estudiantes, y los alumnos puedan ver, subir y presentar dichas prácticas.

5. Ejecución de actividades

Nro	Fecha	Actividad	Responsables	Resultados	Observaciones	Anexos
001	16/05/2023	Recolección de información	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío			
002	22/05/2023	Obtención de requerimientos	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío			
003	23/05/2023	Diagramado del sistema	Yovin Urrego Patricio Betancourt			
004	24/05/2023	Prototipado de vistas	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío			
005	03/06/2023	Revisión del modelado	Diego Riofrío Joan Riofrío			
006	17/06/2023	Modificación diagrama clases	Yovin Urrego Joan Riofrío	Se solucionaron problemas de relaciones		

				muchos - muchos		
007	20/06/2023	Modificación diagrama de despliegue	Patricio Betancourt Diego Riofrío	Se dividió el servidor en sus partes y cambio las conexiones de la placa	Falta aclaración del medio para la comunicación entre la placa y el server	https://app.diagrams.net/#G10TD-pOc11F0KEdxIG68gNHGzsE_vkmgj
008	30/06/2023	Modificación front-end	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío	Ventanas finales de la aplicación	Esperar modificaciones en los requerimientos	
009	01/07/2023	Avance Back-end	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío		Revisar los ORMs y las relaciones	
010	02/07/2023	Avance Back-end	Yovin Urrego Patricio Betancourt Diego Riofrío Joan Riofrío	Solución de problemas de relaciones		

6. Clasificación de requisitos por autor

Código	Autor	Descripción	Categoría
RF001	Docente	El sistema permitirá al docente registrar estudiantes en una materia	Visible
RF002		El sistema permitirá al docente crear desde 3 hasta 4 módulos para una materia	Inferido
RF003		El sistema permitirá al docente colocar una fecha límite de entrega para cada práctica	Visible
RF004		El sistema permitirá al docente visualizar el estado de las prácticas	Visible
RF005		El sistema permitirá al docente asignar calificaciones a cada práctica	Visible
RF006		El sistema permitirá al docente asignar prácticas para determinados estudiantes	Visible
RF007		El sistema permitirá al docente crear una materia	Visible
RF008	Estudiante	El sistema permitirá al estudiante guardar sus datos de la práctica	Visible
RF009		El sistema permitirá al estudiante realizar prácticas	Visible

RF010		El sistema permitirá al estudiante revisar su historial de prácticas	Visible
RF011	Usuario	El sistema permitirá al usuario modificar su perfil	Visible
RF012		El sistema permitirá al usuario iniciar sesión	Visible
RF013		El sistema permitirá al usuario cerrar sesión	Visible
RF014		El sistema permitirá al usuario modificar sus credenciales de acceso	Visible
		EL sistema permitirá al usuario agendar una fecha para poder realizar una práctica	

Código	Categoría	Descripción	Prioridad
RNF001	Seguridad	El sistema cifrará todas las claves de acceso	Alta
RNF002		El sistema admitirá correos de la institución UNL	Alta
RNF003		EL sistema dejará el router CISCO por defecto, una vez terminada la práctica.	
RNF004	Usabilidad	El sistema será amigable	Alta
RNF005		El sistema será fácil de usar	Alta
RNF006		El sistema tendrá un diseño responsivo	Alta
RNF007	Portabilidad	El sistema será desarrollado para ser usado desde navegadores web (Firefox, Chrome, Brave y Edge)	Alta

7. División de módulos

Front-end

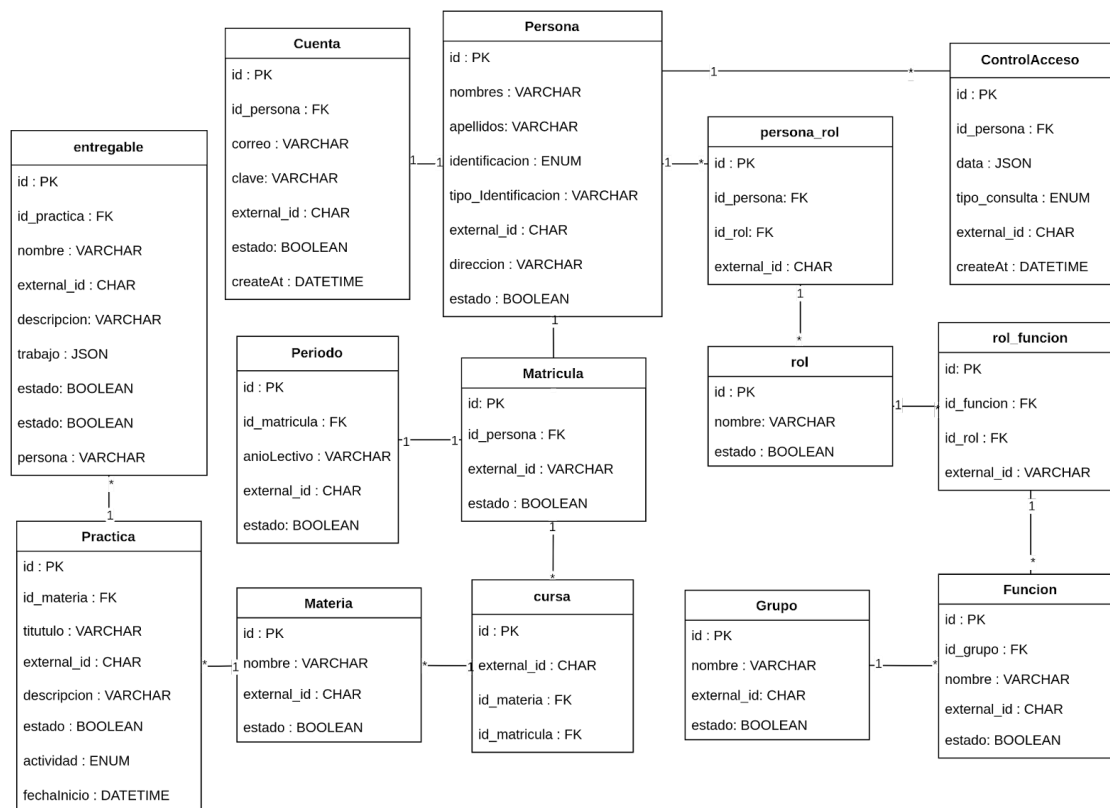
Encargado	Módulos		
	Inicio de sesión	Resultados	
Patricio Betancourt	X		
Joan Riofrío		x	
Diego Riofrío			
Yovin Urrego			

Back-end

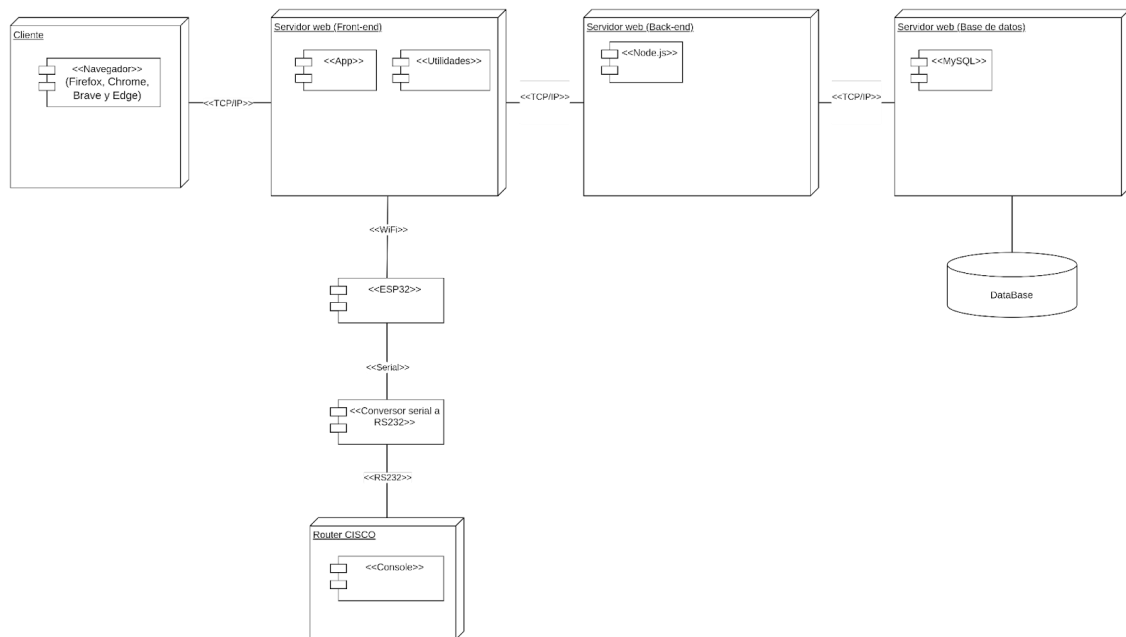
Módulo	Encargado			
	Patricio Betancourt	Joan Riofrío	Yovin Urrego	Diego Riofrío

ControlAcces	X	X	X	X
Cuenta	X			
Entregable		X	X	
materia		X		X
matricula				X
periodo				X
persona	X			
práctica		X	X	X
rol	X			

8. Diagrama de Clases



9. Diagrama de Despliegue



10. Models de la base de datos

Model de controlAcces

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const controlAcces = sequelize.define('controlAcces', {
    data: { type: DataTypes.JSON, allowNull: false },
    tipo_consulta: { type: DataTypes.ENUM('MODIFICAR', 'LEER', 'GUARDAR', 'ELIMINAR'), allowNull: false, defaultValue: 'LEER' },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
  }, {
    freezeTableName: true
  });

  controlAcces.associate = function (models){
    controlAcces.belongsTo(models.persona, {foreignKey: 'id_persona'});
  }
  return controlAcces;
};

```

Model de cuenta

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const cuenta = sequelize.define('cuenta', {
    correo: { type: DataTypes.STRING(50), allowNull: false },
    clave: { type: DataTypes.STRING(100), allowNull: false },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true },
  }, {
    freezeTableName: true
  });

  cuenta.associate = function (models){
    cuenta.belongsTo(models.persona, {foreignKey: 'id_persona'});
  }
  return cuenta;
};

```

Model de cursa

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const cursa = sequelize.define('cursa', {
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
  }, {
    freezeTableName: true
  });

  cursa.associate = function (models){
    cursa.belongsTo(models.materia, {foreignKey: 'id_materia'});
    cursa.belongsTo(models.matricula, {foreignKey: 'id_matricula'});
  }
  return cursa;
};

```

Model de entregable

```

'use strict'
module.exports = (sequelize, DataTypes) => {
  const entregable = sequelize.define('entregable', {
    nombre: {
      type: DataTypes.STRING,
      allowNull: true,
    },
    external_id: {
      type: DataTypes.UUID,
      defaultValue: DataTypes.UUIDV4,
      allowNull: false
    },
    descripcion: {
      type: DataTypes.STRING,
    },
    trabajo:{
      type: DataTypes.JSON,
    },
    estado: {
      type: DataTypes.BOOLEAN,
      allowNull: false,
      defaultValue: true
    },
    fechaEntrega:{
      type: DataTypes.DATE
    },
    persona:{
      type: DataTypes.STRING,
    }
  },{
    freezeTableName: true
  });
  entregable.associate = function(models) {
    entregable.belongsTo(models.practica, {foreignKey: 'id_practica'});
  }
  return entregable;
}

```

Model de funcion


```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const funcion = sequelize.define('funcion', {
    nombre: { type: DataTypes.STRING(50), allowNull: false },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true },
  }, {
    freezeTableName: true
  });

  funcion.associate = function (models){
    funcion.belongsTo(models.grupo, {foreignKey: 'id_grupo'});
    funcion.hasOne(models.rol_funcion, {foreignKey: 'id_funcion', as: 'funcion'});
  }
  return funcion;
};

```

Model de grupo

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const grupo = sequelize.define('grupo', {
    nombre: { type: DataTypes.STRING(50), allowNull: false },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true },
  }, {
    freezeTableName: true
  });

  grupo.associate = function (models){
    grupo.hasMany(models.funcion, {foreignKey: 'id_grupo' ,as: 'grupo'});
  }
  return grupo;
};

```

Model de materia

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const materia = sequelize.define('materia', {
    nombre: {type: DataTypes.STRING(30), allowNull: false},
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true }
  }, {
    freezeTableName: true
  });

  materia.associate = function (models){
    materia.hasMany(models.cursa, {foreignKey: 'id_materia', as: 'cursa'});
    materia.hasMany(models.practica,{foreignKey:'id_materia', as:'practica'});
  }

  return materia;
};

```

Model de matrícula

```

'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const matricula = sequelize.define('matricula', {
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true },
  }, {
    freezeTableName: true
  });

  matricula.associate = function (models){
    matricula.belongsTo(models.persona, {foreignKey: 'id_persona'});
    matricula.hasOne(models.periodo, { foreignKey: 'id_matricula', as: 'periodo'});
    matricula.hasMany(models.cursa, { foreignKey: 'id_matricula', as: 'cursa'});
  }

  return matricula;
};

```

Modelo periodo

```

s > JS periodo.js > ...
'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const periodo = sequelize.define('periodo', {
    anioLectivo: { type: DataTypes.STRING(50), defaultValue: "NO_DATA" },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
    estado: { type: DataTypes.BOOLEAN, defaultValue: true }
  }, {
    freezeTableName: true
  });

  periodo.associate = function (models){
    periodo.belongsTo(models.matricula, {foreignKey: 'id_matricula'});
  }

  return periodo;
};

```

Modelo persona_rol

```

s > JS persona_rol.js > ...
'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const persona_rol = sequelize.define('persona_rol', {
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
  }, {
    freezeTableName: true
  });

  persona_rol.associate = function (models){
    persona_rol.belongsTo(models.persona, {foreignKey: 'id_persona'});
    persona_rol.belongsTo(models.rol, {foreignKey: 'id_rol'});
  }

  return persona_rol;
};

```

Modelo persona

```

1 'use strict';
2 const { UUIDV4 } = require("sequelize");
3 module.exports = (sequelize, DataTypes) => {
4   const persona = sequelize.define('persona', {
5     nombres: { type: DataTypes.STRING(50), defaultValue: "NO_DATA" },
6     apellidos: { type: DataTypes.STRING(50), defaultValue: "NO_DATA" },
7     identificacion: { type: DataTypes.STRING(15), unique: true },
8     tipo_identificacion: { type: DataTypes.ENUM("CEDULA", "PASAPORTE", "RUC"), defaultValue: "CEDULA" },
9     external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
10    direccion: { type: DataTypes.STRING(255), defaultValue: "NO_DATA" },
11
12    estado: { type: DataTypes.BOOLEAN, defaultValue: true }
13  }, { (property) freezeTableName: boolean
14    freezeTableName: true
15  });
16  persona.associate = function (models){
17    persona.hasOne(models.cuenta, { foreignKey: 'id_persona', as: 'cuenta' });
18    persona.hasMany(models.controlAcces, { foreignKey: 'id_persona', as: 'controlAcces' });
19    persona.hasOne(models.persona_rol, { foreignKey: 'id_persona', as: 'persona_rol' });
20  }
21  return persona;
22 };

```

Modelo practica

```

1 'use strict'
2 module.exports = (sequelize, DataTypes) => {
3   const practica = sequelize.define('practica', {
4     titulo: {
5       type: DataTypes.STRING,
6       allowNull: true,
7     },
8     external_id: {
9       type: DataTypes.UUID,
10      defaultValue: DataTypes.UUIDV4,
11      allowNull: false
12    },
13    descripcion: {
14      type: DataTypes.STRING(256),
15      allowNull: true,
16    },
17    estado: {
18      type: DataTypes.BOOLEAN,
19      allowNull: false,
20      defaultValue: true
21    },
22    actividad: {
23      type: DataTypes.ENUM('PENDIENTE', 'ENTREGADO', 'REVISADO', 'OCULTO'),
24      allowNull: false,
25      defaultValue: 'PENDIENTE'
26    },
27    fechaInicio: {
28      type: DataTypes.DATE,
29      allowNull: true
30    },
31    fechaFin: {
32      type: DataTypes.DATE,
33      allowNull: true,
34    }
35  }, {
36    freezeTableName: true
37  });
38  practica.associate = function(models) {
39    practica.belongsTo(models.materia, { foreignKey: 'id_materia' });
40    practica.hasOne(models.entregable, { foreignKey: 'id_practica', as: 'entregable' });
41  }
42  return practica;
43 };

```

Modelo rol_funcion

```
'use strict';
const { UUIDV4 } = require("sequelize");
module.exports = (sequelize, DataTypes) => {
  const rol_funcion = sequelize.define('rol_funcion', {
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
  }, {
    freezeTableName: true
  });

  rol_funcion.associate = function (models) {
    rol_funcion.belongsTo(models.funcion, { foreignKey: 'id_funcion' });
    rol_funcion.belongsTo(models.rol, { foreignKey: 'id_rol' });
  }
  return rol_funcion;
};
```

Modelo Rol

```
'use strict';

module.exports = (sequelize, DataTypes) => {
  const rol = sequelize.define('rol', {
    nombre: DataTypes.STRING(20),
    estado: { type: DataTypes.BOOLEAN, defaultValue: true },
    external_id: { type: DataTypes.UUID, defaultValue: DataTypes.UUIDV4 },
  }, { freezeTableName: true });

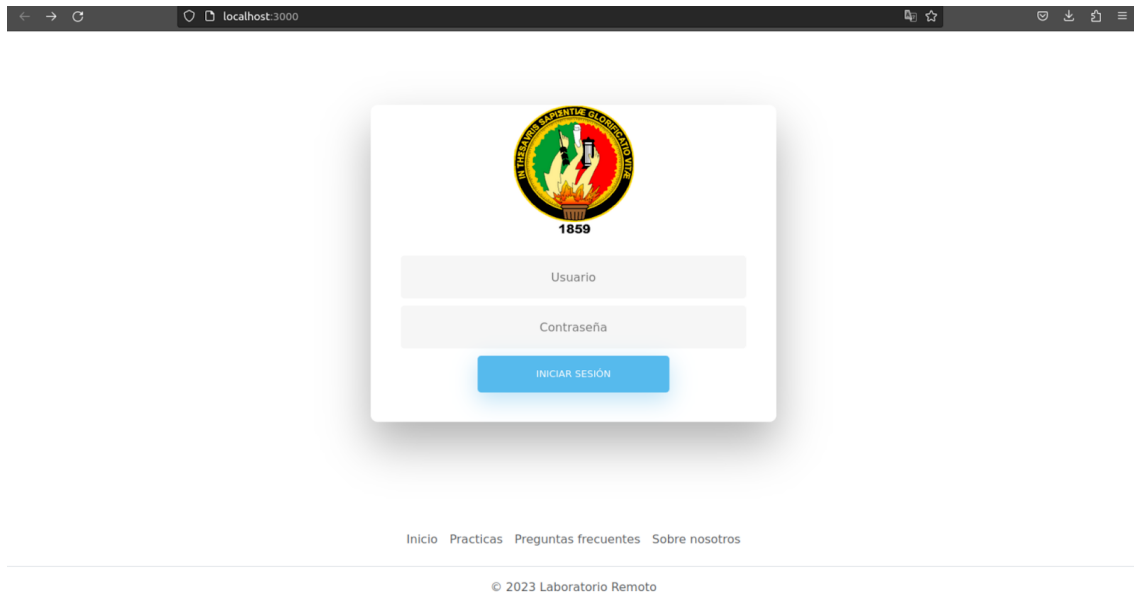
  rol.associate = function (models) {
    rol.hasMany(models.rol_funcion, { foreignKey: 'id_rol', as: 'rol_funcion' });
    rol.hasMany(models.persona_rol, { foreignKey: 'id_rol', as: 'persona_rol' });
  }

  return rol;
};
```

11. Vistas y controladores

Vista inicio de sesión:

```
};
return (
  <div className="p-6">
    <div className="contenedor_completo">
      <div className="wrapper fadeInDown">
        <div id="formContent">
          <div className="fadeIn first">
            </img>
          </div>
          <br />
          <form>
            <div>
              <input type="text" id="correo" className="fadeIn second" name="correo" placeholder="Usuario" />
              <br />
              <input type="password" id="clave" className="fadeIn third" name="clave" placeholder="Contraseña" />
              <br />
              <a type="submit" className="fadeIn fourth" onClick={onSubmit} >Iniciar Sesión </a>
            </div>
          </form>
        </div>
      </div>
    </div>
    <Footer />
  </div>
);
export default Login;
```



Controlador entregable:

```
bls > JS EntregableController.js > ...
'use strict';

var models = require('../models');

class EntregableController {
  async listar(req, res) {
    let listar = await rol.findAll({
      attributes: ['nombre', 'external_id', 'estado']
    });
    res.json({ msg: 'OK!', code: 200, info: listar });
  }

  async guardar(req, res) {
    const fechaActual = new Date();
    let rperAux = await models.persona.findOne({ where: { identificacion: req.body.identificacion } });
    let pract = await models.practica.findOne({ where: { external_id: req.body.external_id } });
    let transaction = await models.sequelize.transaction();
    if (rperAux) {
      try {
        const data = {
          "titulo": req.body.titulo,
          "descripcion": req.body.descripcion,
          "trabajo": req.body.trabajo,
          "fechaEntrega": fechaActual,
          "persona": req.body.identificacion,
          "id_practica": pract.id
        };
        await models.entregable.create(data, transaction);
        await transaction.commit();
        res.json({
          msg: "SE HAN REGISTRADO SUS DATOS",
          code: 200
        });
      } catch (error) {
        if (transaction) await transaction.rollback();
        if (error.errors && error.errors[0].message) {
          res.json({ msg: error.errors[0].message, code: 400 });
        } else {
          res.json({ msg: error.message, code: 400 });
        }
      }
    } else {
      res.json({ msg: 'Persona no encontrada', code: 400 });
    }
  }
}
```

Consola en react

```

import React, { useState, useRef, useEffect } from 'react';
import './css/style.css';
import Header from './Header';
import Footer from './Footer';
import { useForm } from 'react-hook-form';
import { saveCodigo } from './utilidades/ides';
const commands = {
  echo: {
    description: 'Echo a passed string.',
    usage: 'echo <string>',
    fn: (...args) => args.join(' ')
  }
};
const character = '>';
const Consola = () => {
  const [commandHistory, setCommandHistory] = useState([]);
  const { register, handleSubmit, formState: { errors }, reset } = useForm();
  const consoleRef = useRef(null);
  useEffect(() => {
    // Desplazarse al final de la consola cuando se actualiza el historial de comandos
    scrollToBottom();
  }, [commandHistory]);
  const scrollToBottom = () => {
    if (consoleRef.current) {
      consoleRef.current.scrollTop = consoleRef.current.scrollHeight;
    }
  };
  const onSubmit = (data) => {
    const command = data.tiituloText.trim();
    if (command !== '') {
      setCommandHistory((prevHistory) => [...prevHistory, command]);
      // executeCommand(command);
      console.log(commandHistory);
      saveCodigo(commandHistory);
      reset();
    }
  };
};

```

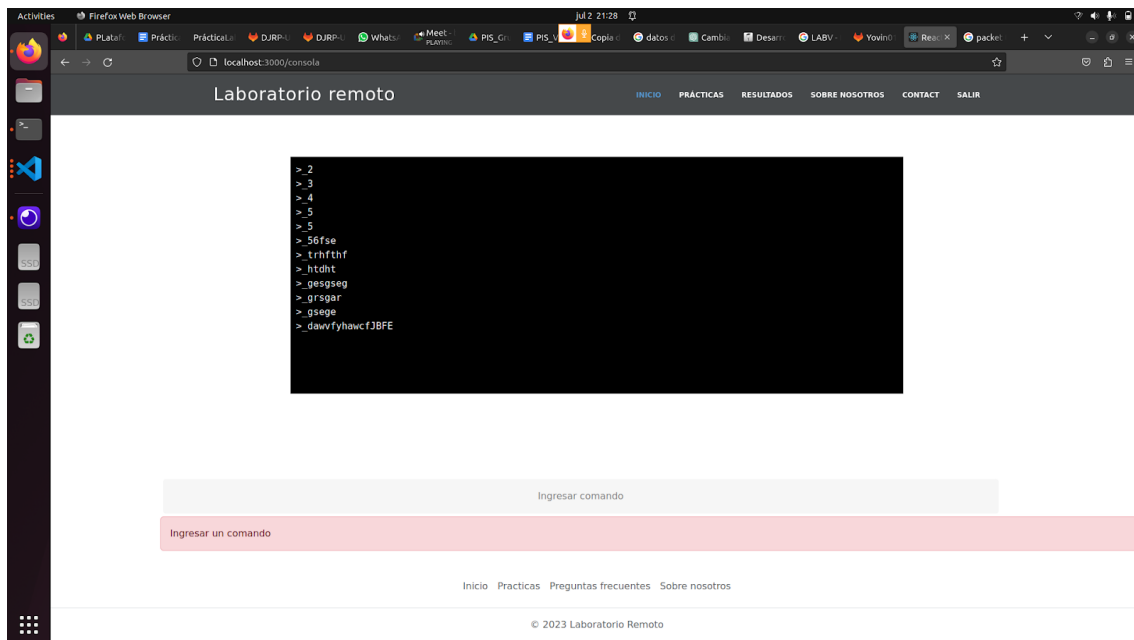
```

return (
  <div>
    <Header></Header>
    <div className='container'>
      <div className="bodyConsole">
        <div className='consoleYovin' ref={consoleRef}>
          {commandHistory.map((command, index) => (
            <div key={index} className="command"> {caracter}_{command}</div>
          ))}
        </div>
      </div>
      <div id="console"></div>
    </div>
    <div className='consoleInput'>
      <form onSubmit={handleSubmit(onSubmit)}>
        <input className='consoleInput' type="text"
          {...register('tiituloText', { required: true })}
          placeholder="Ingresar comando"
        />
        {errors.tiituloText && errors.tiituloText.type === 'required' && (
          <div className='alert alert-danger'>Ingresar un comando</div>
        )}
      </form>
    </div>
    <Footer></Footer>
  </div>
);
];

export default Consola;

```

Vista desde navegador



Guardado de comandos

```
Comando ejecutado: 2
2
Comando ejecutado: 3
3
Comando ejecutado: 4
4
Comando ejecutado: 5
5
Comando ejecutado: 5
5
▶ Array(4) [ "2", "3", "4", "5" ]
▶ Array(9) [ "2", "3", "4", "5", "5", "56fse", "trhfthf", "htdht", "gesgseg" ]
▶ Array(10) [ "2", "3", "4", "5", "5", "56fse", "trhfthf", "htdht", "gesgseg", "grsgar" ]
▶ Array(11) [ "2", "3", "4", "5", "5", "56fse", "trhfthf", "htdht", "gesgseg", "grsgar", ... ]
>>
```

Post para guardar un entregable, resultante de una práctica en consola

```
POST http://localhost:3006/api/entregable/guardar Send 200 OK 106 ms 48 B
JSON Auth Query Headers 1 Docs Preview Headers 8 Cookies Timeline
1 {
2   "titulo": "Practica n3",
3   "external_id": "bb890b32-1948-11ee-9497-489ebd1149bf",
4   "description": "datos de descripcion datos",
5   "trabajo": [ {"auto":{
6     "precio": 4000, "cantidad": 1, "producto": "dest-de"
7   }},{
8     "auto":{
9       "precio": 4000, "cantidad": 1, "producto": "dest-de"
10    } } ],
11   "identificacion": "134256789"
12 }
```

12. 10. Anexos

Front-end: <https://gitlab.com/DJRP-U/pis-grupo-c.git>

Back-end: <https://gitlab.com/DJRP-U/pis-grupo-c-backend.git>