




## BASE DE DATOS PARA LA FOMENTACIÓN Y APRENDIZAJE DEL LENGUAJE DE SEÑAS

<p>Nombres y apellidos Dayana Jesús Mejía Núñez</p> <p>Teléfono:314878793 Correo:Dmejianunez52@correo.unicordoba.edu.co</p>	
<p>Nombres y apellidos Yoryi Doria Petro</p> <p>Teléfono:3128551492 Correo:Ydoriapetro44@correo.unicordoba.edu.co</p>	
<p>Nombres y apellidos Glorimar Daza</p> <p>Teléfono: 3226903472</p>	

Grupo:



BASE DE DATOS PARA LA FOMENTACIÓN Y APRENDIZA DEL LENGUAJE DE SEÑAS



PROGRAMACIÓN WEB

PRESENTADO POR: Yoryi Doria Petro  
Dayana Jesus Mejia Núñez  
Glorimar Daza

PRESENTADO A: ALEXANDER ENRIQUE TOSCANO RICARDO

UNIVERSIDAD DE CÓRDOBA  
FACULTAD DE EDUCACIÓN Y CIENCIAS HUMANAS  
LICENCIATURA EN INFORMÁTICA Y MEDIOS AUDIOVISUALES  
MONTERIA – CÓRDOBA  
2019

## TABLA DE CONTENIDOS

### Tabla de contenido

1. PROPUESTA_____	Pagina 4
2. DISEÑO DE LA BASE DE DATOS_____	Pagina 5
3. MODELOS ENTIDAD-RELACIÓN, RELACIONAL_____	Pagina 7
ESQUEMAS.	
3.1. MODELO ENTIDAD-RELACIÓN_____	Pagina 7
3.2. MODELO RELACIONAL_____	Pagina 8
3.3. ESQUEMAS_____	Pagina 9
4. RUTAS AMIGABLES_____	Pagina 12
5. MAPA DE NAVEGACIÓN_____	Pagina 16
6. COMPONENTES Y SUS RUTAS_____	Pagina 17
7. ESQUEMAS_____	Pagina 18
8. COMPONENTES_____	Pagina 21

## 1. PROPUESTA

Cada vez está más extendida la práctica del lenguaje de signos. Hoy en día, no es raro encontrar a personas que sean capaces de hablar en este lenguaje. Además, contar con esta habilidad es de gran importancia para quienes trabajan en organismos o entidades de ayuda social, ONG'S, fundaciones; o ámbitos que tienen cierta vinculación con proyectos o instituciones de inclusión educativa, laboral o social.

### ¿A QUIÉN VA DIRIGIDO?

Dirigido a cualquier persona interesada en aprender lenguaje de signos, así como a profesionales y estudiantes de diferentes ámbitos: Educativo, Laboral, Administrativo, Psicología, Pedagogía, Trabajo Social, etc. En la base de datos del curso se almacenará la información de los registros, los cursos, niveles y módulos que en los cuales están definidos los contenidos, actividades y evaluaciones.

La importancia de realizar este proyecto se debe a la falta de comunicación con las personas que presentan discapacidad auditiva y discapacidad física (Sordomudos), por lo cual se plantea la creación de un curso para el aprendizaje de lenguaje de señas. Este curso facilitara y agilizara el aprendizaje a los estudiantes y personas con o sin conocimientos previos en este.

El curso cuenta con un plan de estudio que está clasificado por varios niveles en los cuales se encuentran el nivel básico, intermedio y avanzado, dentro los niveles se encuentran los módulos que abarcan los contenidos, recursos, actividades y evaluaciones; con el fin de interactuar en espacios cotidianos con los usuarios de la lengua y con sus pares de aprendizaje.

--

## 2. DISEÑO DE LA BASE DE DATOS

Para el diseño de la base de datos analizamos, e identificamos la información necesaria para la aplicación que deseamos crear, de la cual resultan todas las consultas que los usuarios pueden ejecutar. Luego se determinaron todas las entidades siguiendo el modelo de datos, de esta manera se organizaron los datos y atributos de todas las entidades las cuales fueron representadas y relacionadas en un diagrama entidad-relación.

El siguiente proceso es crear el modelo relacional donde se describió el tipo y las relaciones entre los datos. de la base de datos, ya se muestra en la base de datos a través, tablas en las cuales se insertó toda la información, además se realizó la normalización para evitar reiteración de datos. A continuación, se describen todas las entidades de la base de datos:

### 1. Login

En la entidad login se encuentra la manera de como ingresar al curso virtual con su usuario y su contraseña.

Usuario
usuarios
Contraseñas

### 2. Cursos

En esta entidad del curso se encuentran los nombres de los cursos y su respectivo contenido, de tal manera que en el plan de clases obtenga cada contenido.

Curso
Id
nombres
plan_clases

### 3. Módulos

En esta entidad de los módulos contara con el plan de estudio de cada nivel y sus respectivos contenidos que por cada nivel se clasifican los módulos.

modulos
Id
nombres
contenidos
id nivel

### 4. niveles

En esta entidad se clasifican 3 tipo de niveles, que van a salir en un botón, donde se va a mostrar la información en el momento de hacer clic en el nivel que desee

Después de dar clic en el botón respecto a los niveles, van a salir los módulos, en donde van a salir videos, imágenes, entre otros y la evaluación por cada módulo.

<b>niveles</b>
Id
Nivel
descripcion
Id_cursos

## 5. Actividades

En esta entidad contendrá todos los trabajos que se harán durante todo el nivel, se activara por etapas esta opción, las actividades contenidas son video conferencias, creación de infografías, sopas de letras, escoge la imagen correcta...

actividades
Id
nombres
descripción
id módulos

## 6. Recursos

Esta entidad va disponer de video tutoriales, infografías, presentaciones, pdf, blogs...

Recursos
Id
nombres
url

## 7. Evaluaciones

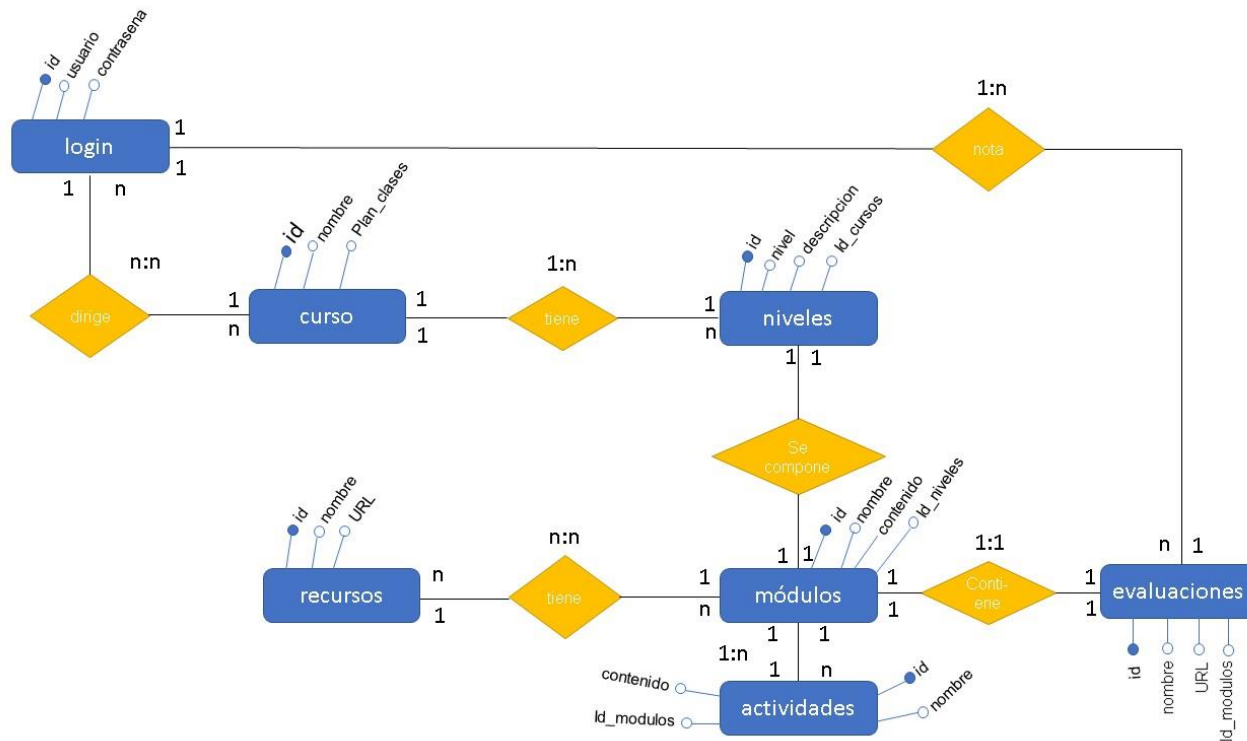
En esta entidad va disponer con los modelos de evaluación para cada módulo y nivel. La forma se evaluará cada nivel para poder hacer uno con más complejidad, además de eso cada evaluación tendrá su URL, las evaluaciones podrían ser de selección múltiple o respuesta libre

evaluaciones
Id
nombres
URL
Id_modulos

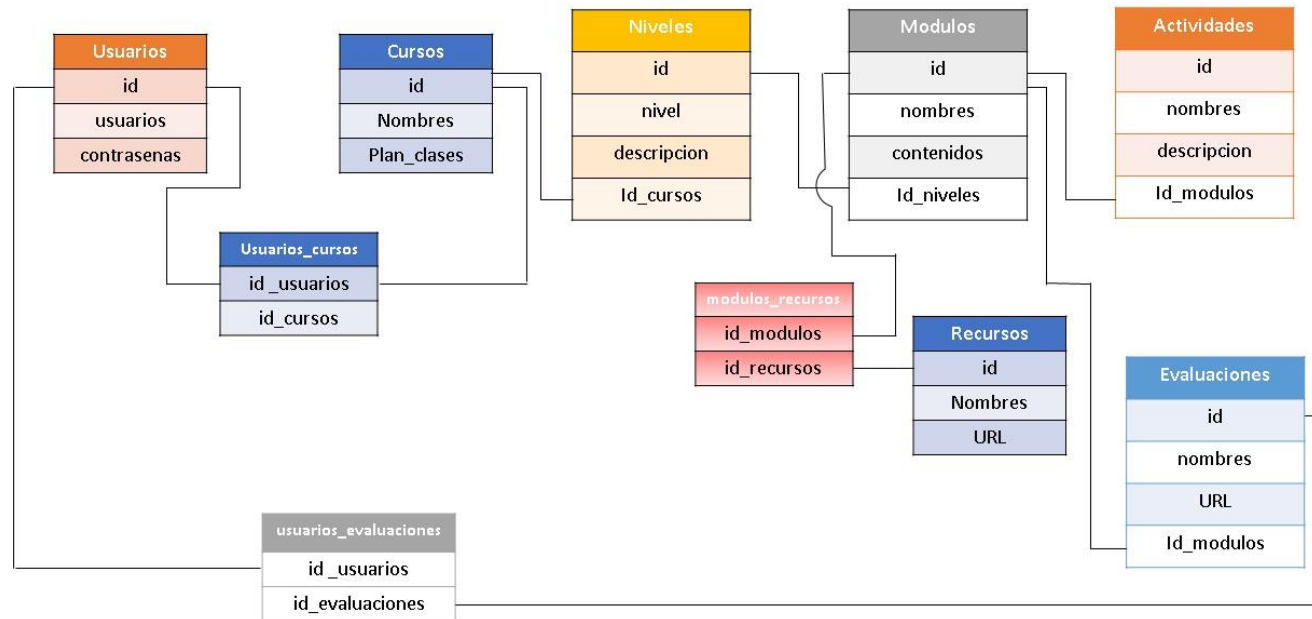


### 3. MODELOS ENTIDAD RELACION, RELACIONAL, ESQUEMAS

#### MODELO ENTIDAD-RELACIÓN



## MODELO RELACIONAL



## ESQUEMAS

### Login.js

```
{  
  
  id: object_id,  
  
  usuario: string,  
  contraseña: string,  
  
  id_curso: [  
    {  
      id: object_id  
      ref: "curso"  
    }  
  ],  
  
  id_evaluacion: [  
    {  
      id: object_id  
      ref: "evaluaciones"  
    }  
  ]  
}
```

### Cursos.js

```
{  
  
id: object_id,  
nombres: string,  
plan_clases: string,  
id_usuario: [  
    {  
  
        id: object_id  
  
        ref: "usuarios"  
    }  
]  
}
```

### **Niveles.js**

```
{  
id: object_id,  
nivel: string,  
descripción: string,  
id_curso:  
  
    {  
        id: object_id  
        ref: "curso"  
    }  
}
```

### **Modulos.js**

```
{  
id: object_id,
```

```
nombres: string,
contenidos: string,
id_nivel:

    {
    id:object_id
    ref: "niveles"
    },

    id_recurso: [

        {
        id: object_id
        ref: "recursos"
        }
    ]

}
```

#### **Actividades.js**

```
{
id: object_id,
nombre: string,
descripcion: string,
id_modulo:

    {
    id:object_id
    ref: "modulos"
    }

}
```

#### **Recursos.js**

```
{
id: object_id,
```

```

nombre: string,
URL: string,
id_modulo:
    {
        id:object_id
        ref: “modulos”
    }
]
}

```

### Evaluaciones.js

```

{
id: object_id,
nombres: string,
URL: string,
id_modulo:
    {
        id:object_id
        ref: “modulos”
    },

    id_usuario: [
        {
            id: object_id
            ref: “usuarios”
        }
    ]
}

```

## 4. RUTAS AMIGABLES

METODO	Post
FRONTEND	<a href="http://localhost:5000/login">http://localhost:5000/login</a>
BACKEND	/login

FORMATO DE ENTRADA	<pre>{   "login": {     "usuario": "dayo",     "contrasena": "1234m"   } }</pre>
FORMATO DE SALIDA	<pre>{   "status": 200,   "nU": {     "id_curso": [],     "id_evaluacion": [],     "_id": "5d84597823bbab0578221694",     "nombre1": "dayo",     "nombre2": "mejida",     "apellido1": "Rodriguez",     "apellido2": "Saladen",     "correo": "dayito@gmail.com",     "contrasena": "1234m",     "__v": 0   } }</pre>
METODO	Put
FRONTEND	FRONTEND
BACKEND	/actividad/:id
FORMATO DE ENTRADA	<pre>{   "usuario": {     "nombre1": "Miguelote"   } }</pre>
FORMATO DE SALIDA	<pre>{   "message": "Actividades Actualizadas",   "raw": {     "n": 1,     "nModified": 1,     "opTime": {       "ts": "6738770100779745281",       "t": 4     }   } }</pre>

	<pre>     },     "electionId": "7ffffffff00000000000000004",     "ok": 1,     "operationTime": "6738770100779745281",     "\$clusterTime": {       "clusterTime": "6738770100779745281",       "signature": {         "hash": "dHe7VcxUT++EYMwqMrtmYRp7F3o=",         "keyId": "6715138580831797249"       }     }   } } </pre>
--	---

METODO	Post
FRONTEND	<a href="https://3000-a544d16c-6dc6-48c5-95c7-8baa78490fb6.ws-us1.gitpod.io/modulos/:idNivel">https://3000-a544d16c-6dc6-48c5-95c7-8baa78490fb6.ws-us1.gitpod.io/modulos/:idNivel</a>
BACKEND	/modulos/:idNivel
FORMATO DE ENTRADA	<pre> {   "modulo": {     "nombres": "Modulo Y",     "contenido": "Contenido del modulo"   } } </pre>
FORMATO DE SALIDA	

METODO	Post
FRONTEND	
BACKEND	
FORMATO DE ENTRADA	<pre> {   "nivel": {     "nivel": 4,     "descripcion": "nivel 4"   } } </pre>
FORMATO DE SALIDA	



METODO	Post
FRONTEND	
BACKEND	/actividades
FORMATO DE ENTRADA	{ "actividad": { "nombre": "Actividad Postman", "descripcion": "Hecha en postman" } }
FORMATO DE SALIDA	
METODO	Post
FRONTEND	<a href="http://localhost:5000/login">http://localhost:5000/login</a>
BACKEND	/login
FORMATO DE ENTRADA	{ "usuario": { "nombre": "dayo", "contrasena": "1234m" } }
FORMATO DE SALIDA	

METODO	get
FRONTEND	https://3000-a544d16c-6dc6-48c5-95c7-8baa78490fb6.ws-us1.gitpod.io/niveles
BACKEND	/niveles
FORMATO DE ENTRADA	
FORMATO DE SALIDA	[ { "_id": "5d35cf081c9d4400003de79c", "nivel": 1, "descripcion": "basico", "id_curso": null }, { } ]

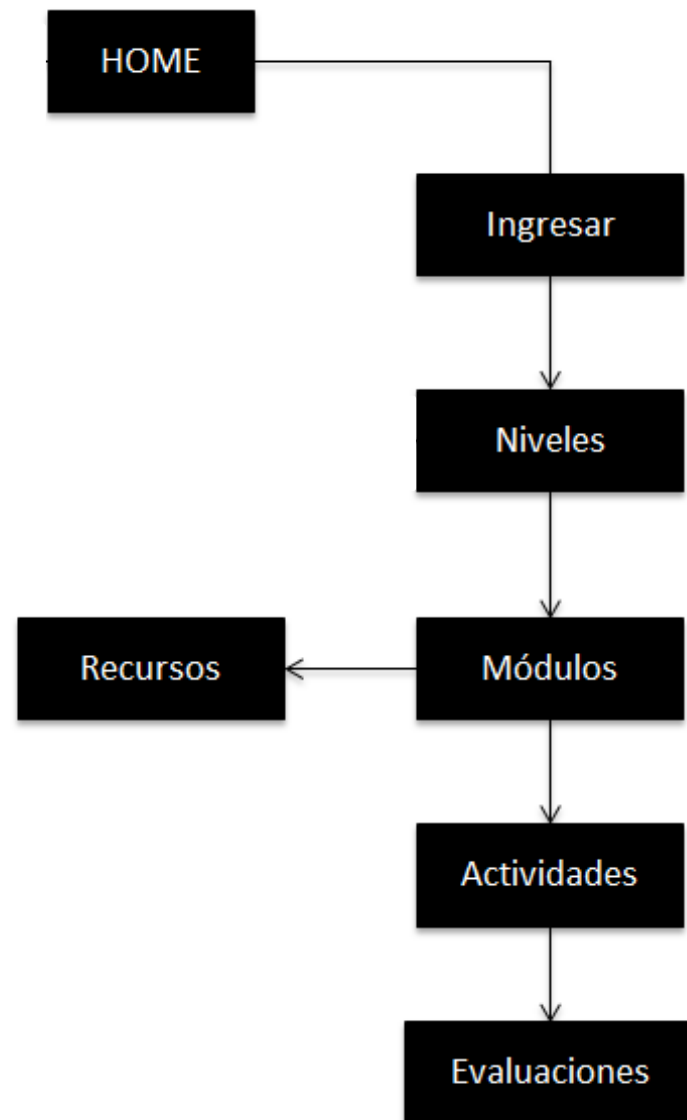
	<pre>       "_id": "5d49bbb9c2b24c2259ceba0a",       "nivel": 4,       "descripcion": "nivel 4",       "id_curso": {         "id_usuario": [           "5d43259ef6e7aa10c8825999"         ],         "_id": "5d35ccb71c9d4400003de799",         "nombres": "alfabeto",         "plan_clases": "numeros"       },       "__v": 0     }   ] </pre>
--	--

METODO	get
FRONTEND	https://3000-a544d16c-6dc6-48c5-95c7-8baa78490fb6.ws-us1.gitpod.io/recursos
BACKEND	/recursos
FORMATO DE ENTRADA	
FORMATO DE SALIDA	<pre> [   {     "id_modulo": [       {         "id": "5d35d00a6cb5430000fe6fad",         "ref": "modulos"       }     ],     "_id": "5d35cfd71c9d4400003de79f",     "nombre": "video",     "url": "youtube.com"   } ] </pre>

METODO	post
FRONTEND	http://localhost:5000/curso
BACKEND	/curso

FORMATO DE ENTRADA	<pre>{    "nombre": "alfabeto",   "plan_clases": "modulo 1-alfabeto y numeros "  }</pre>
FORMATO DE SALIDA	

## 5. MAPA DE NAVEGACION



## 6. COMPONENTES Y SUS RUTAS

- **Home** -> El componente principal de la página web, podemos encontrar en la parte superior el nombre de la página y con las opciones de **ingresar** (el usuario tiene una cuenta ya creada en el curso y podrá ingresar directamente) donde pedirá dos datos “usuario” y “contraseña”  
Contará con una breve introducción de que es el curso, su objetivo principal.

**Ruta:** /home

- **Nivel básico** -> El componente tendrá una introducción, donde se dará la bienvenida al curso, recomendaciones al momento realizar alguna activada del curso y mostrará el contenido con el que cuenta el nivel, recursos y actividades.

**Ruta:** /nivelesbasico

- **Niveles** -> Este componente se encargará de mostrar cuantos niveles tiene el curso y en que se enfatiza cada uno. cada nivel puede contar muchos módulos.

**Ruta:** /niveles

- **Módulos** -> El componente módulos contara con el plan de estudio de cada nivel.

**Ruta:** /modulos

- **Recursos** -> El componente va a disponer de las herramientas de apoyo que tiene el curso por medio de hipervínculos.

**Ruta:** /recursos

- **Actividades** ->Este componente contara las actividades y la descripción de cada una de estas que se encuentran por módulos y niveles.

**Rutas:** /actividades

- **Evaluaciones** -> El componente va disponer con los modelos de evaluación para cada módulo y nivel. La forma se evaluara cada nivel para poder hacer uno con mas complejidad.

**Rutas:** /evaluaciones

## 7. ESQUEMAS

### Login.js

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
```

```
const LoginsSchema = new Schema({
  usuario: String,
  contrasena: String,
  id_curso: Array,
  id_evaluacion: Array,
});
```

```
var Usuarios = mongoose.model("Usuarios", UsuariosSchema);
module.exports = Usuarios;
```

### Modulos.js

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
```

```
const ModulosSchema = new Schema({
  nombres: String,
  contenido: String,
```

```
    id_nivel: Array
  });

var Modulos = mongoose.model("Modulos", ModulosSchema);
module.exports = Modulos;
```

### **Cursos.js**

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const CursosSchema = new Schema({
  nombre: String,
  plan_clases: String,
  id_usuario: Array
});

var Cursos = mongoose.model("Cursos", CursosSchema);
module.exports = Cursos;
```

### **Evaluaciones.js**

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const EvaluacionesSchema = new Schema({
  nombre: String,
  url: String,
  id_modulo: Array,
  id_usuario: Array
});
```

```
});
```

```
var Evaluaciones = mongoose.model("Evaluaciones", EvaluacionesSchema);  
module.exports = Evaluaciones;
```

### **Niveles.js**

```
const mongoose = require("mongoose");  
const Schema = mongoose.Schema;
```

```
const NivelesSchema = new Schema({  
  nivel: Number,  
  descripcion: String,  
  id_curso: Array  
});
```

```
var Niveles = mongoose.model("Niveles", NivelesSchema);  
module.exports = Niveles;
```

### **Actividades.js**

```
const mongoose = require("mongoose");  
const Schema = mongoose.Schema;
```

```
const ActividadesSchema = new Schema({  
  nombre: String,  
  descripcion: String,  
  id_modulo: Array  
});
```

```
var Actividades = mongoose.model("Actividades", ActividadesSchema);  
module.exports = Actividades;
```



### **Recursos.js**

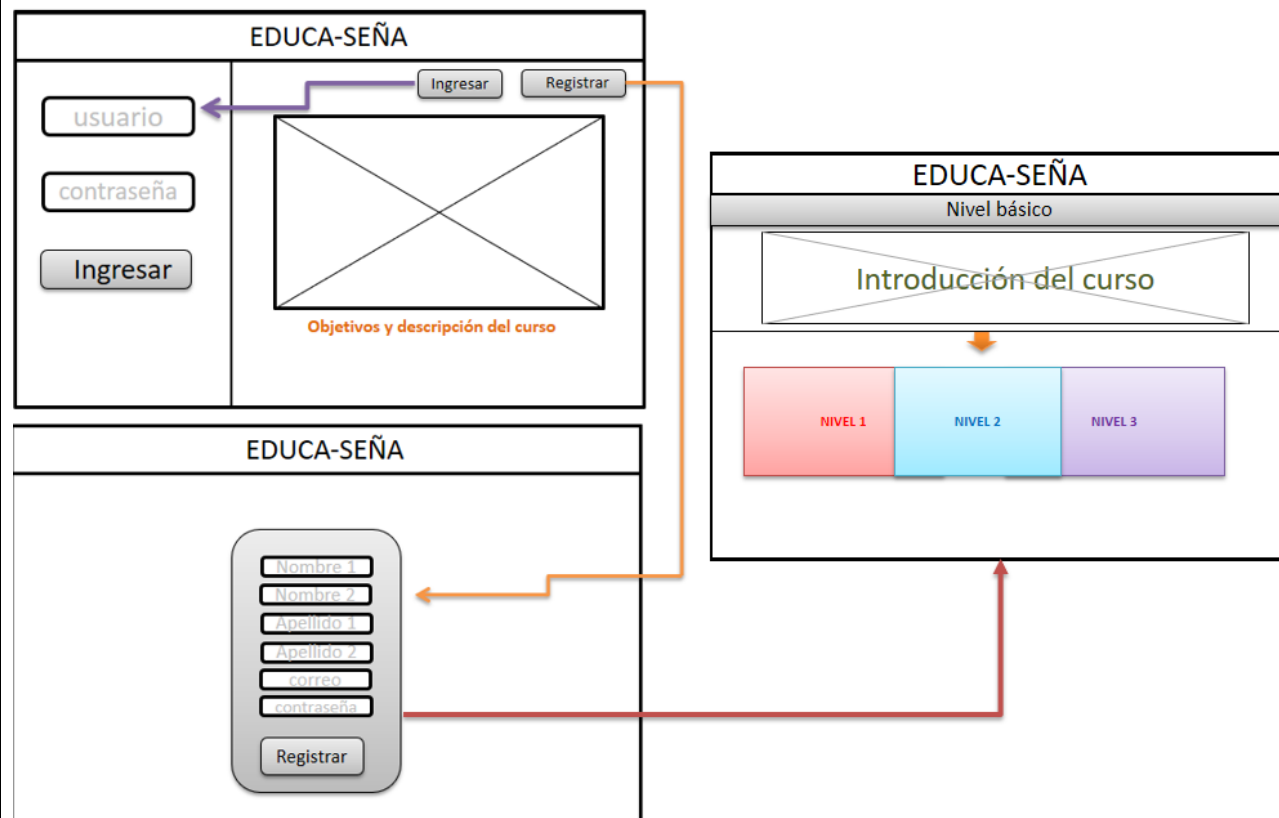
```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;

const RecursosSchema = new Schema({
  nombre: String,
  url: String,
  id_modulo: Array
});

var Recursos = mongoose.model("Recursos", RecursosSchema);
module.exports = Recursos;
```

## 8. COMPONENTES

### COMPONENTE [NOMBRE DEL COMPONENTE] INTERFASE



### RUTAS

POST: /USUARIO/

GET: /USUARIO/ID

UPDATE: /USUARIO/ID

PUT: /USUARIO/ID

**DELETE: /USUARIOS/ID**

**CONSULTAS**

**Usuarios.create({})**

**Usuarios.find({})**

**Usuarios.find({\_id: id})**

**Usuarios.find({\_id: id\_usuario, id\_usuario}, {\$push: {id\_nivele curso: id\_nivelcurso}})**

**Usuarios.updateoone({\_id:id\_usuario}, {\$ push: {id\_nivelecurso: id\_nivelcurso,  
id\_modulos:id\_modulos}})**

**Usuarios.delete({\_id: id})**

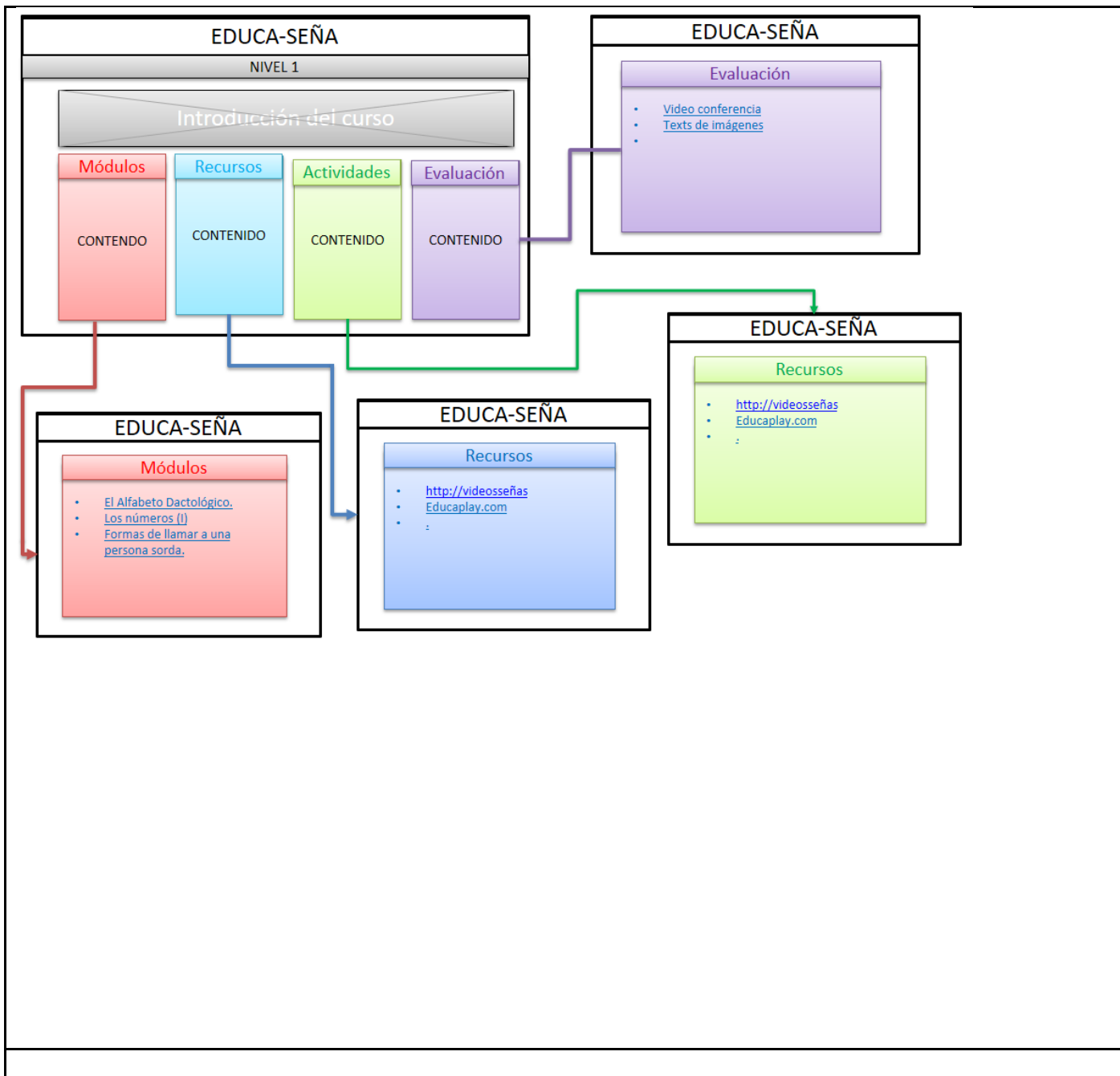


Tabla de revisiones por fecha									
Integrantes									
1	Dayana Jesus Mejia Núñez								
2	Yoryi Doria Petro								
3	Glorimar Daza								
Fecha							Actividad	Observaciones	
11/10/2019							Video del Backend y Frontend		


--

