TITULO DEL PROYECTO

|  |  |
| --- | --- |
| Yina Marcela Cogollo Conde |  |

Grupo:



|  |
| --- |
| PRESENTACIÓN |
| SHARE CODE , APP EDUCATIVA COMO HERRAMIENTA TECNOLÓGICA PARA LA SOLUCIÓN DE CÓDIGOS DE PROGRAMACIÓN EN LA MATERIA PROGRAMACIÓN WEB.  PROGRAMACIÓN WEB  PRESENTADO POR: YINA MARCELA COGOLLO CONDE  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 |
| Contenidos   1. PROPUESTA…………………………………………………………………. 2. DISEÑO DE LA BASE DE DATOS……………………………………….. 3. MODELO ENTIDAD-RELACIÓN, RELACIONAL, ESQUEMAS…………… 4. RUTAS AMIGABLES………………………………………………………. 5. MAPA DE NAVEGACIÓN……………………………………………….. 6. COMPONENTES Y SUS RUTAS……………………………………… 7. ESQUEMAS………………………………………………. 8. COMPONENTES………………………………………………. |

|  |
| --- |
| PROPUESTA |
| En la universidad de Córdoba se esta presentando una problemática en todas las materias relacionadas con la programación: fundamentos de algoritmia, fundamentos de programación, técnicas avanzadas de programación, base de datos y programación web, el problema radica en que muchos estudiantes se les presentan problemas cuando están realizando sus trabajos o proyectos, y el profesor encargado de la respectiva materia, no puede atender a todos, dato que tiene muchas responsabilidades, aun asi, destina parte de su tiempo a asesorar a sus estudiantes, pero una asesoria no es suficente para muchos… justo aquí radica el problema los estudiantes necesitan que docentes o estudiantes con un nivel mas avanzado en la programación le colaboren solucionando y explicando los errores que tienen en sus respectivos códigos. Por lo tanto se desea crear un sitio web parecido a stack overflow donde los estudiantes de Licenciatura en Informática y medios audiovisuales de la Universidad de Córdoba, puedan resolver sus dudas acerca de los errores en sus códigos. Cada estudiante puede crear un post con información relacionada de su error, cada post cuenta con un titulo, el codigo fuente del proyecto, una descripción y el id del estudiante que esta creando el post. Los estudiantes a su vez tambien se registran en sitio con la credenciales de power campus, esto es requisito para poder loguearse ya que el sitio web es solo para estudiantes de LIMAV de la universidad de Córdoba. El estudiante al registrarse debe ingresar su primer nombre, segundo nombre, apellido, segundo apellido, la identificación y la materia que esta cursando…  La lógica del sitio (Backend) se escribirá usando node.js y express y el Frontend con Vue.js, Bootstrap, Css y Sass. En cuanto a base de datos utilizaremos Mongodb una base de datos Nosql, muy popular en la comunidad de desarrolladores. |

|  |
| --- |
| DISEÑO DE LA BASE DE DATOS |
| Para diseñar la base de datos se tuvo en cuenta, cada una de las colecciones a utilizar :Estudiantes, Materias, Post y Comentarios, cada una de las colecciones tiene un identificador único.. Una vez se establecieron cada una de la colecciones se realizo el modelo entidad relación para identificar la relación existente entre las diferentes colecciones. A continuación se describen cada una de las colecciones previamente establecidas:  1. Estudiantes  En primer lugar hemos creado la colección estudiante. Cada uno de estos debe registrarse en el sitio web, para esto debe contar con los siguientes datos:  **nota: Pegas la tabla de estudiante y su relación con materia..**  **2.** Materias  En segundo lugar se encuentra la colección materia con los respectivas materias de programación que se ven en el programa Limav de la universidad de Córdoba…  **NOTA: PEGAS LA TABLA MATERIAS**  **3.** Post  La colección post juega un papel fundamental puesto que el el estudiante establecera el problema en el código, esta colección esta relacionada con la de estudiantes, puesto que se necesita el id del estudiante que esta creando el post y que necesita que le ayuden con su error.  **NOTA: PEGAS LA TABLA POST Y LA RELACIÓN QUE ESTABLECE CON EL ESTUDIANTE.**  **4.** Comentario  El la colección comentarios tenemos almacenamos la solución que el estudiante x le da al error que tiene su compañero. Esta colección esta relacionada con el estudiante que creo el post mediante su \_id y con el \_id del post..  **NOTA: PEGAS LA TABLA POST Y LA RELACIÓN QUE ESTABLECE CON LAS DEMAS TABLAS** |

|  |
| --- |
| MODELOS ENTIDAD RELACION, RELACIONAL, ESQUEMAS |
| **Nota: pegas el modelo entidad relación y el modelo relacional**  **ESQUEMAS:**  **estudiante** {  nombre:{type:String},  nombre2:{type:String},  apellido:{type:String},  apellido2:{type:String},  edad:{type:String},  materia:{type:Schema.Types.ObjectId,ref:'Materia'}  }    **materia** {  nombre:{type:String},  descripcion:{type:String}  }  **comentario** {  titulo:{type:String},  descripcion:{type:String},  estudiante:{type:Schema.Types.ObjectId, ref:'Estudiante'},  post:{type:Schema.Types.ObjectId, ref:'Post'}  }  **post** {  titulo:{type:String},  codigo\_fuente:{type:String},  descripcion:{type:String},  estudiante:{type:Schema.Types.ObjectId,ref:'Estudiante'}  } |

|  |
| --- |
| RUTAS AMIGABLES |
| |  |  | | --- | --- | | METODO | Post | | FRONTEND | Localhost:3000/estudiantes | | BACKEND | /estudiante | | FORMATO DE ENTRADA | {  “nombre”:”Yina”,  “nombre2”: “Marcela”,  “apellido”: “Cogollo”,  “apellido2”: “Conde”,  “edad”: 25,  “materia”: 5dd59bcc5bbeaa17a62dec58  } | | FORMATO DE SALIDA | {  "estudiantes": [  {  "\_id": "5dd59cdb5bbeaa17a62dec5a",  "nombre": "Yina",  "nombre2": "Marcela",  "apellido": "Cogollo",  "apellido2": "Conde",  "edad": "26",  "materia": {  "\_id": "5dd59bcc5bbeaa17a62dec58",  "nombre": "PROGRAMACIÓN WEB",  "descripcion": "Aprende tecnologías de desarrollo web modernas y aplicalas a tu proyecto"  }  }  ]  } |  |  |  | | --- | --- | | METODO | Get | | FRONTEND | Localhost:3000/estudiantes | | BACKEND | /estudiantes | | FORMATO DE ENTRADA |  | | FORMATO DE SALIDA | {  "estudiantes": [  {  "\_id": "5dd59cdb5bbeaa17a62dec5a",  "nombre": "Yina",  "nombre2": "Marcela",  "apellido": "Cogollo",  "apellido2": "Conde",  "edad": "26",  "materia": {  "\_id": "5dd59bcc5bbeaa17a62dec58",  "nombre": "PROGRAMACIÓN WEB",  "descripcion": "Aprende tecnologías de desarrollo web modernas y aplicalas a tu proyecto"  }  }  ]  } |  |  |  | | --- | --- | | METODO | Get | | FRONTEND | Localhost:3000/estudiante/5dd59c0c5bbeaa17a62dec59 | | BACKEND | */estudiante/:id* | | FORMATO DE ENTRADA |  | | FORMATO DE SALIDA | {  "estudiantes": [  {  "\_id": "5dd59c0c5bbeaa17a62dec59",  "nombre": "Francisco",  "nombre2": "Miguel",  "apellido": "Pereira",  "apellido2": "Torrez",  "edad": "22",  "materia": {  "\_id": "5dd59bcc5bbeaa17a62dec58",  "nombre": "PROGRAMACIÓN WEB",  "descripcion": "Aprende tecnologías de desarrollo web modernas y aplicalas a tu proyecto"  }  }  ]  } |  |  |  | | --- | --- | | METODO | Put | | FRONTEND | http://localhost:3000/estudiante/5dd59cdb5bbeaa17a62dec5a | | BACKEND | */ estudiante/:id* | | FORMATO DE ENTRADA | {  “nombre”: “Camila”,  “edad”: 34  } | | FORMATO DE SALIDA | {  "estudiante": {  "\_id": "5dd59cdb5bbeaa17a62dec5a",  "nombre": "Camila",  "nombre2": "Marcela",  "apellido": "Cogollo",  "apellido2": "Conde",  "edad": "34",  "materia": "5dd59bcc5bbeaa17a62dec58"  }  } |  |  |  | | --- | --- | | METODO | Delete | | FRONTEND | http://localhost:3000/estudiante/5dd59cdb5bbeaa17a62dec5a | | BACKEND | */ estudiante/:id* | | FORMATO DE ENTRADA |  | | FORMATO DE SALIDA | {  "message": "Estudiante eliminado"  } |   **Nota: Tienes que agregar las demás colecciones** |

|  |
| --- |
| MAPA DE NAVEGACION |
|  |

|  |
| --- |
| COMPONENTES Y SUS RUTAS |
| NOTA:  **Agrega cada uno de los componentes y los describes, ejemplo**  **1. about**  Es el componente principal de nuestra app, en esta sección podemos visualizar un buen diseño de bienvenida etc.…  **2. Ruta/** estudiantes  **3. Ruta /** materias  **4. Ruta /** posts  **5. Ruta /** comentarios |

|  |
| --- |
| ESQUEMAS |
| **1. Estudiantes**  const EstudianteSchema=new Schema({  nombre:{type:String},  nombre2:{type:String},  apellido:{type:String},  apellido2:{type:String},  edad:{type:String},  materia:{type:Schema.Types.ObjectId,ref:'Materia'}  },{versionKey:false});  **2. Materias**  const MateriaSchema=new Schema({  nombre:{type:String},  descripcion:{type:String}  },{versionKey:false})  **3. Posts**  const PostSchema=new Schema({  titulo:{type:String},  codigo\_fuente:{type:String},  descripcion:{type:String},  estudiante:{type:Schema.Types.ObjectId,ref:'Estudiante'}  },{versionKey:false});  **4. Comentarios**  const ComentarioSchema=new Schema({  titulo:{type:String},  descripcion:{type:String},  estudiante:{  type:Schema.Types.ObjectId,  ref:'Estudiante'  },  post:{  type:Schema.Types.ObjectId,  ref:'Post'  }  },{versionKey:false}); |

|  |
| --- |
| **COMPONENTES** |
| **COMPONENTE [NOMBRE DEL COMPONENTE]**  **INTERFASE**    **RUTAS**  **POST: /USUARIO/**  **GET: /USUARIO/ID**  **UPDATE: /USUARIO/ID**  **DELETE: /USUARIO/ID**  **CONSULTAS**  **Usuarios.save({\_\_\_\_:\_\_\_\_})**  **Usuarios.find({\_\_\_\_:\_\_\_\_})**  **Usuarios.update({\_\_\_\_:\_\_\_\_})**  **Usuarios.remove({\_\_\_\_:\_\_\_\_})** |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |