

TITULO DEL PROYECTO: PROYECTO PARA LA AGILIZACIÓN DE LA COMPRA Y VENTA DE FICHOS DE LOS
ALMUERZOS SUBSIDADOS EN LA UNIVERSIDAD DE CÓRDOBA.

<p>Nombres y apellidos: Jorge Carlos Martínez Pastrana.</p> <p>Celular: +57 314 883 2879</p> <p>Email: JMartinezpastrana@correo.unicordba.edu.co</p>	
<p>Nombres y apellidos: Ricardo Avelino José Madera Berrocal.</p> <p>Celular: +57 304 354 6328</p> <p>Email: ricardomabec@gmail.com</p>	

Grupo:



**PROYECTO PARA LA AGILIZACIÓN DE LA COMPRA Y VENTA DE FICHOS DE
LOS ALMUERZOS SUBSIDADOS EN LA UNIVERSIDAD DE CÓRDOBA**



Programación Web

Presentado por: Jorge Martínez Pastrana y Ricardo Madera Berrocal

Presentado a: Alexander Toscano Ricardo

Licenciatura en Informática y Medios Audiovisuales
Facultad de Educación y Ciencias Humanas
Universidad de Córdoba
2019

TABLA DE CONTENIDOS

Pag. 1: Título del proyecto

Pag. 2: Presentación

Pag. 4: Propuesta

Pag 5: Diseño de la base de datos
--

Pag 9: Rutas amigables

Pag 12: Mapa de navegación

Pag 13: Componentes y sus rutas
--

Pag 14: Esquemas

Pag 15: Componentes

PROPUESTA

Para la Universidad de Córdoba y todos los que sean usuarios del programa de almuerzos subsidiados se pretende desarrollar una base de datos acompañada de una aplicación y de una mejora en el carnet estudiantil para agilizar el proceso de compra y venta de los fichos para los almuerzos y llevar control de los fichos comprados, dicha base de datos debe contener: estudiante, carnet, ficho y vendedor.

Paso 1: Generar un código QR para cada estudiante mediante su número de identificación usando un generador: (<https://play.google.com/store/apps/details?id=com.ykart.tool.qrcodegen>)

Paso 2: Crear una base de datos que contenga los datos de los estudiantes asociados a su código QR.

Paso 3: Crear un sistema que permite hacer la compra y venta del ficho por medio del código usando el código QR, la base de datos y un teléfono móvil.

El vendedor tendrá acceso privilegiado a la aplicación para hacer el debido registro de los estudiantes y sus compras, teniendo una sección dentro de la aplicación donde puede ingresar los datos, y también tiene acceso a los datos realizando una pequeña consulta por entidad.

Procedimiento:

- El estudiante se acerca a punto de compra y usa el código de su carnet para identificarse ante el vendedor.
- La persona encargada de la venta recibe el pago para luego confirmar y registrar la compra en la base de datos
- El estudiante recoge su tarjeta y recibe el ficho con su compra registrada.

//Este método ayuda a contrarrestar el problema de la reventa de los tickets, ya que solo se puede comprar uno al día.

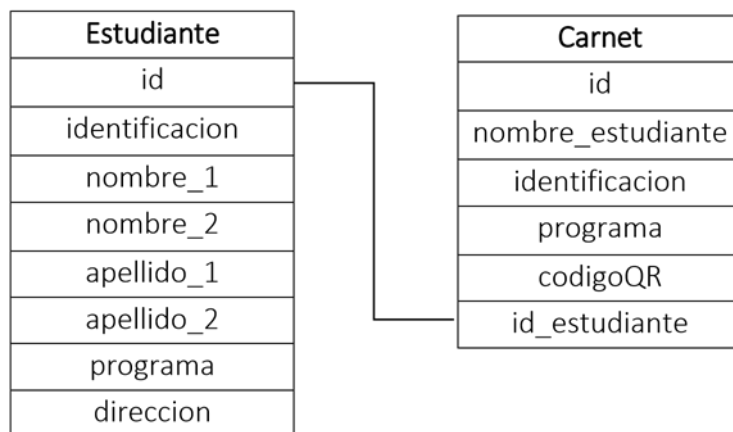
//Al ser más rápida la compra, los estudiantes pueden prescindir de largas filas y dedicar más tiempo a la realización de actividades académicas.

DISEÑO DE LA BASE DE DATOS

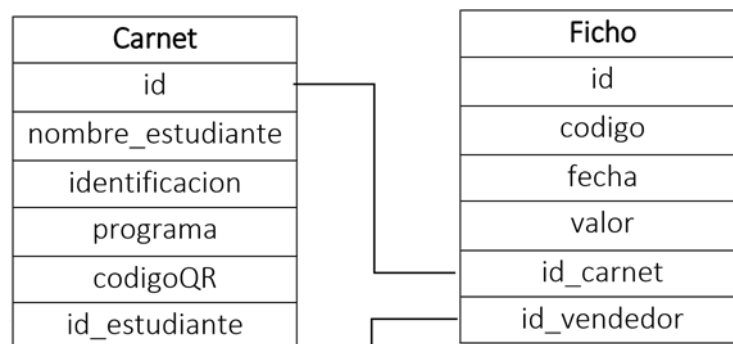
Para trabajar de manera ordenada y eficiente con la base de datos es necesario establecer cuáles son las partes más importantes (entidades) dentro de la misma para luego organizarlas, listar sus componentes (atributos) y relacionarlas, todo esto con el fin de agilizar su utilización y navegación con el uso de consultas entre las entidades y sus partes.

Se realizan una serie de esquemas para dar a conocer de forma gráfica las partes de la base de datos y sus relaciones entre sí, también pasa por un proceso llamado: “normalización” para eliminar relaciones que no sean necesarias y hacer conexiones que faciliten aún más la navegación por la base de datos y la aplicación.

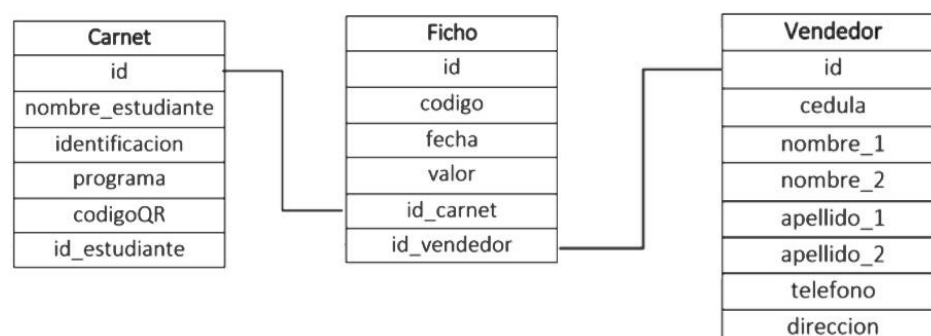
#1: Estudiantes: En esta parte se almacenan los datos de los estudiantes que pertenezcan a la Universidad de Córdoba y que deseen hacer la compra de un ficho. Esta entidad está unida a la entidad *Carnet* mediante una relación 1:1



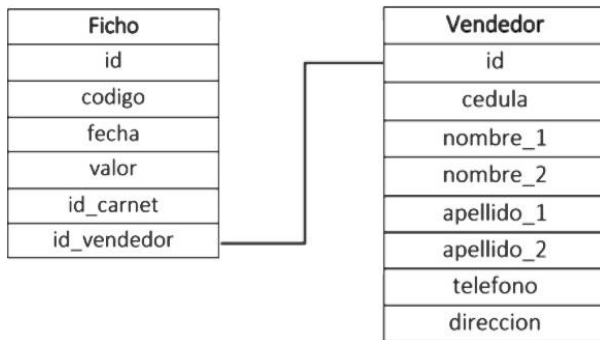
#2: Carnet: Aquí se guardan los datos de los carnets de los estudiantes para tener un registro y no permitir que un estudiante compre más de un ficho. Esta entidad esta unida a la entidad *Ficho* mediante una relación 1:1



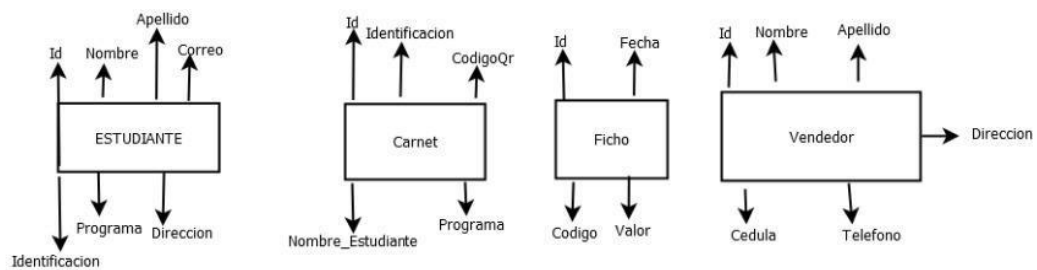
#3: Ficho: En esta parte se almacena todo los relacionado con los fichos para la venta. Esta entidad está relacionada con la entidad *Carnet* mediante una relación 1:1 y con la entidad *Vendedor* mediante una relación 1:N



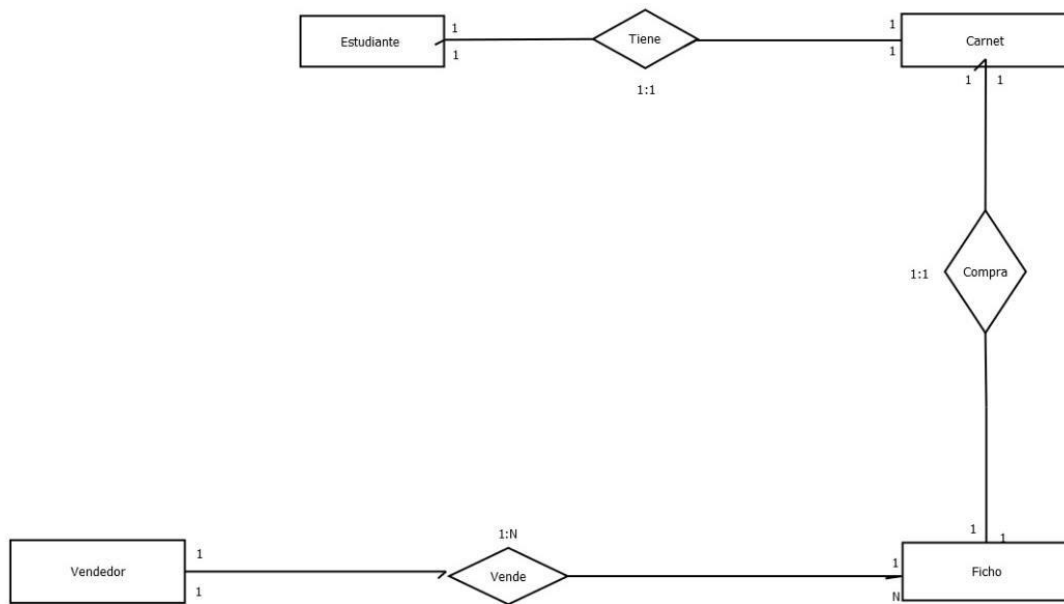
#4: Vendedor: Por último, en esta sección se almacenan los datos de los vendedores que están a cargo de la comercialización de los fichos. Esta entidad está relacionada con la entidad *Ficho* mediante una relación 1:N



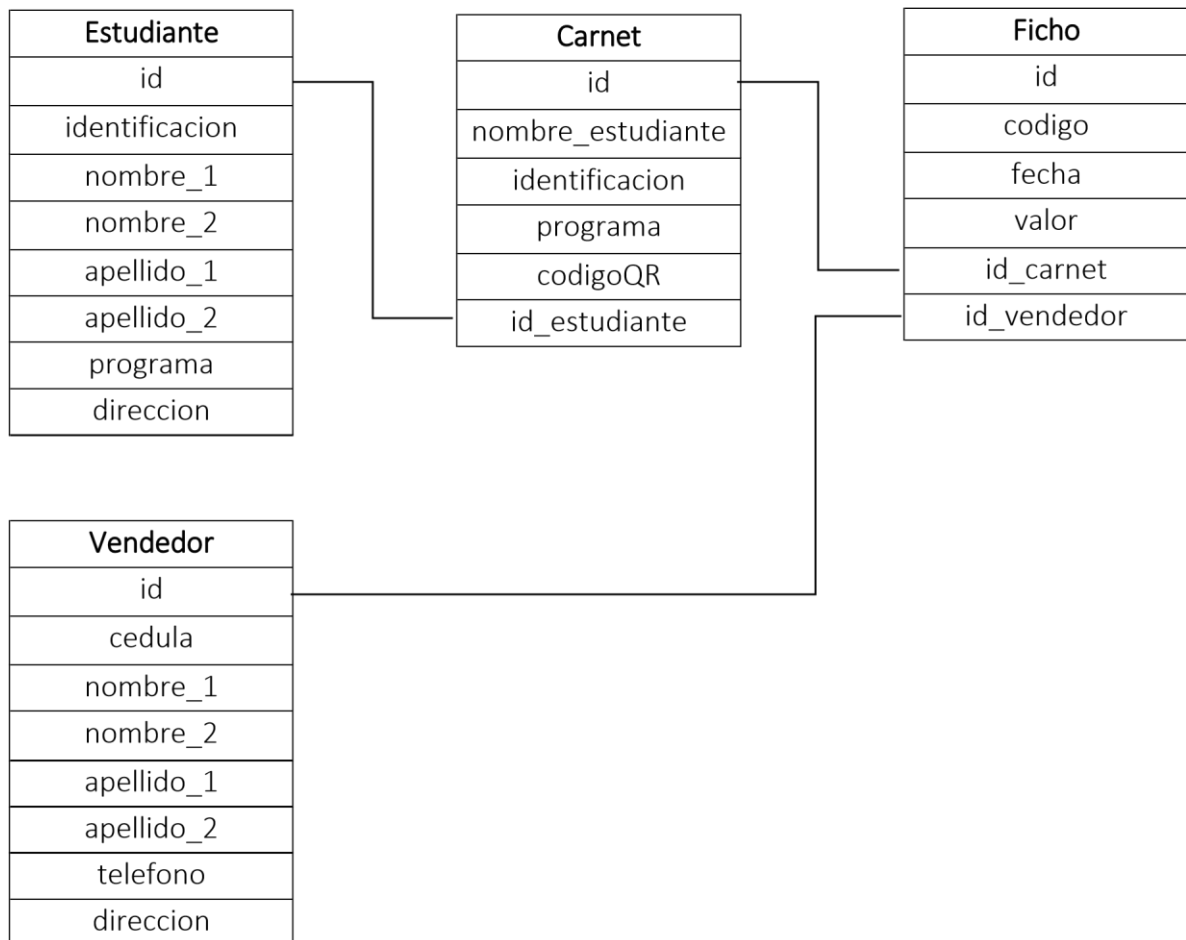
ENTIDADES:



MODELO ENTIDAD-RELACIÓN:



MODELO RELACIONAL:



ESQUEMAS:

Estudiantes

```
EstudiantesSchema = new Schema({
  nombre_1: String,
  nombre_2: String,
  apellido_1: String,
  apellido_2: String,
  identificacion: String,
  programa: String,
  direccion: String,
  correo: String,
});
```

Carnet

```
carnetSchema = new Schema({
  codigo_qr: String,
  id_estudiante: {
    type: Schema.Types.ObjectId,
    ref: 'Estudiante'
  }
});
```

Ficho:

```
fichoSchema = new Schema({
  codigo: String,
  fecha: String,
  valor: Number,
  id_carnet: {
    type: Schema.Types.ObjectId,
    ref: "Carnet"
  },
  id_vendedor: {
    type: Schema.Types.ObjectId,
    ref: "Vendedor"
  }
});
```

Vendedor:

```
vendedorSchema = new Schema({
  nombre_1: String,
  nombre_2: String,
  apellido_1: String,
  apellido_2: String,
  cedula: String,
  telefono: String,
  direccion: String
});
```


RUTAS AMIGABLES

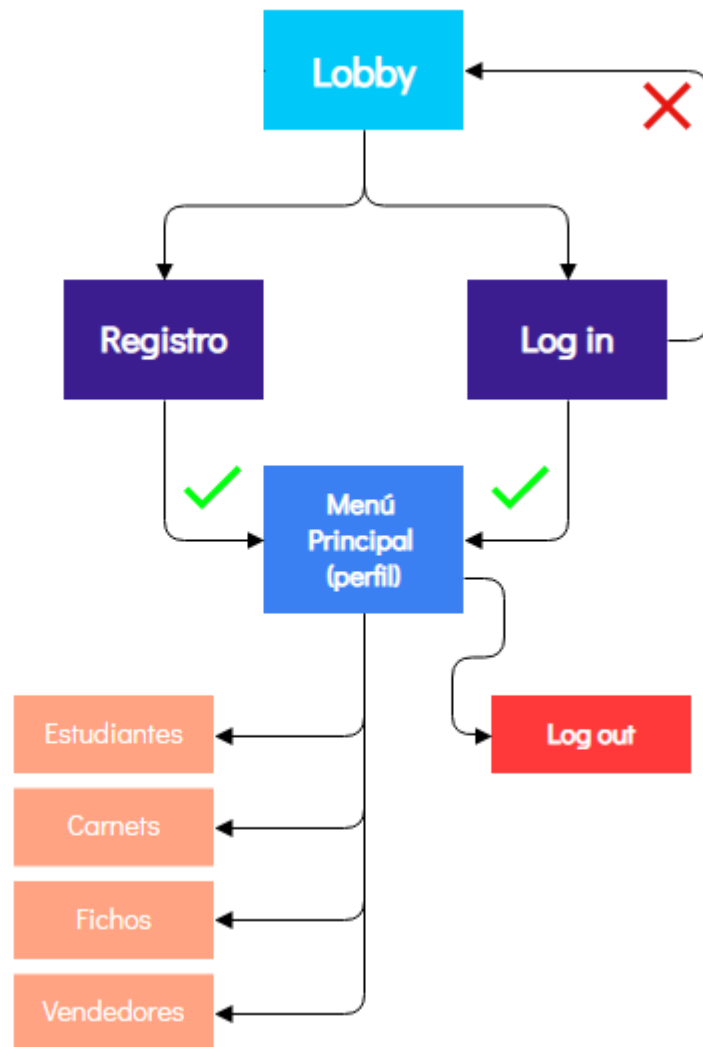
METODO	GET
FRONTEND	http://localhost:3000/estudiantes
BACKEND	/estudiantes
FORMATO DE ENTRADA	
FORMATO DE SALIDA	<pre>{ "_id": "5d47a7f50d09b409d8fa39d7", "nombre_1": "Julieta", "nombre_2": "Carolina", "apellido_1": "Martinez", "apellido_2": "Martinez", "identificacion": "1201372849", "programa": "Licenciatura en Informatica", "direccion": "Cra.9 #23-21", "correo": "julieta@gmail.com", "_v": 0 },</pre>

METODO	GET
FRONTEND	http://localhost:3000/carnets
BACKEND	/carnets
FORMATO DE ENTRADA	
FORMATO DE SALIDA	<pre>{ "_id": "5d47b3b4bd966629b8d9daef", "codigo_qr": "567898654", "id_estudiante": { "_id": "5d47aa040d09b409d8fa39d8", "nombre_1": "Miguel", "nombre_2": "Jose", "apellido_1": "Vega", "apellido_2": "Jimenez", "identificacion": "1067968008", "programa": "Licenciatura en Informatica" } },</pre>

METODO	GET
FRONTEND	http://localhost:3000/vendedor
BACKEND	/vendedor
FORMATO DE ENTRADA	
FORMATO DE SALIDA	<pre>{ "_id": "5d47c1b9be742b2f5ccd8048", "nombre_1": "Ricardo", "nombre_2": "Milos", "apellido_1": "Madeira", "apellido_2": "Dos Santos", "cedula": "4567890232", "telefono": "3009875678", "direccion": "La Gallera #18", "_v": 0 },</pre>

	},
METODO	GET
FRONTEND	http://localhost:3000/fichos/5d47c5c3f44e8c1f387c9030
BACKEND	/fichos/:id
FORMATO DE ENTRADA	
FORMATO DE SALIDA	<pre>{ "_id": "5d47c5c3f44e8c1f387c9030", "codigo": "5697", "fecha": "31-07-2019", "valor": 1000, "id_carnet": { "_id": "5d47c5a3f44e8c1f387c902f", "codigo_qr": "567898654", "id_estudiante": { "_id": "5d47abd80d09b409d8fa39d9", "nombre_1": "Alejandro", "nombre_2": "Martin", "apellido_1": "Hernandez", "apellido_2": "Espitia", "identificacion": "1023298374", "programa": "Licenciatura en Informatica" } } }, "id_vendedor": "5d47c1b9be742b2f5ccd8048", "_v": 0 }</pre>
METODO	POST
FRONTEND	http://localhost:3000/estudiantes/5d47abd80d09b409d8fa39d9/generar-carnet
BACKEND	/estudiates/:id/:generar-carnet
FORMATO DE ENTRADA	<pre>{ "carnet": { "codigo_qr": "567898654" } }</pre>
FORMATO DE SALIDA	<pre>{ "status": 200, "nC": { "_id": "5d8446289ae17c110847030c", "codigo_qr": "567898654", "id_estudiante": "5d47abd80d09b409d8fa39d9", "_v": 0 } }</pre>
METODO	POST
FRONTEND	http://localhost:3000/vendedor
BACKEND	/vendedor
FORMATO DE ENTRADA	<pre>{ "vendedor": {</pre>

	<pre> "nombre_1": "Jose", "nombre_2": "Antonio", "apellido_1": "Riquelme", "apellido_2": "Montenegro", "cedula": "1220818262", "telefono": "3029872674", "direccion": "Cra. 8 #18-19" } </pre>
FORMATO DE SALIDA	<pre> { "msg": "Vendedor creado", "nV": { "_id": "5d843ffe9ae17c110847030b", "nombre_1": "Jose", "nombre_2": "Antonio", "apellido_1": "Riquelme", "apellido_2": "Montenegro", "cedula": "1220818262", "telefono": "3029872674", "direccion": "Cra. 8 #18-19", "_v": 0 } } </pre>
METODO	POST
FRONTEND	http://localhost:3000/vendedor/5d47c1b9be742b2f5ccd8048/carnets/5d47c5a3f44e8c1f387c902f/generar-ficho
BACKEND	/vendedor/:id/:carnets/:id/:generar-ficho
FORMATO DE ENTRADA	<pre> { "ficho": { "codigo": "5697", "fecha": "31-07-2019", "valor": 1000 } } </pre>
FORMATO DE SALIDA	<pre> { "msg": "Ok", "nF": { "_id": "5d8447469ae17c110847030d", "codigo": "5697", "fecha": "31-07-2019", "valor": 1000, "id_carnet": "5d47c5a3f44e8c1f387c902f", "id_vendedor": "5d47c1b9be742b2f5ccd8048", "_v": 0 } } </pre>



- **Lobby:** el lobby es la pantalla inicial que los usuarios pueden ver al acceder al sitio, el cual permite acceder a muchos sitios por medio de botones e hipervínculos, el primer botón que salta a las vista es el botón de *Log in*, cuya función es llevarnos a la sección de inicio de sesión para aquellos usuarios que y estén registrados; el segundo botón que podemos ver es el botón de *Registro*, el cual da acceso a la pestaña de registro, para aquellos que no tengan una cuenta y deseen poseer una; en la parte inferior tenemos un texto que dice: “Universidad de Córdoba”, este está sujeto a un hipervínculo el cual nos lleva al sitio web oficial de dicha universidad; por último, en la parte superior derecha se encuentra una enlace que nos lleva a una pestaña de *Preguntas frecuentes*, a la cual es muy fácil acceder.

Ruta: /lobby

- **Log in:** es donde los usuarios ya registrados ingresan sus datos para acceder a la base de datos, esta pestaña está compuesta por dos barras para rellenar con texto donde se piden; el número de teléfono y la contraseña, si los campos diligenciados no corresponden a ningún usuario registrado, la aplicación regresará automáticamente al lobby.

Ruta: /lobby/:login

- **Registrarse:** esta sección está hecha para todos aquellos usuarios que no posean una cuenta, en esta pestaña encontrarán cuadros de texto parecidos a los del *log in*, pero deberán rellenar más datos tales como: nombre, programa, contraseña, entre otras cosas necesarias, cabe aclarar es que es obligatorio leer y aceptar los términos y condiciones del sitio.

Ruta: /lobby/:registro

- **Menú principal:** en este menú se muestran las partes a las que el usuario puede acceder y conocer, las cuales son: estudiantes, carnets, fichos, vendedores.

Ruta: /lobby/:cellphone/:mainmenu

- **Estudiantes:** en esta pestaña se almacenan y muestran todos los *estudiantes* que se encuentran registrados en la base de datos.

Ruta: /lobby/:cellphone/:mainmenu/:estudiantes

- **Carnets:** en esta pestaña se almacenan y muestran todos los *carnets* que se encuentran registrados en la base de datos.

Ruta: /lobby/:cellphone/:mainmenu/:carnets

- **Fichos:** en esta pestaña se almacenan y muestran todos los *fichos* que se encuentran registrados en la base de datos.

Ruta: /lobby/:cellphone/:mainmenu/:fichos

- **Vendedor:** en esta pestaña se almacenan y muestran todos los *vendedores* que se encuentran registrados en la base de datos.

Ruta: /lobby/:cellphone/:mainmenu/:vendedores

ESQUEMAS

Estudiantes

```
EstudiantesSchema = new Schema({
  nombre_1: String,
  nombre_2: String,
  apellido_1: String,
  apellido_2: String,
  identificacion: String,
  programa: String,
  direccion: String,
  correo: String,
});
```

Carnet

```
carnetSchema = new Schema({
  codigo_qr: String,
  id_estudiante: {
    type: Schema.Types.ObjectId,
    ref: 'Estudiante'
  }
});
```

Ficho:

```
fichoSchema = new Schema({
  codigo: String,
  fecha: String,
  valor: Number,
  id_carnet: {
    type: Schema.Types.ObjectId,
    ref: "Carnet"
  },
  id_vendedor: {
    type: Schema.Types.ObjectId,
    ref: "Vendedor"
  }
});
```

Vendedor:

```
vendedorSchema = new Schema({
  nombre_1: String,
  nombre_2: String,
  apellido_1: String,
  apellido_2: String,
  cedula: String,
  telefono: String,
  direccion: String
});
```

COMPONENTES

COMPONENTE [LOBBY] INTERFASE

Preguntas frecuentes

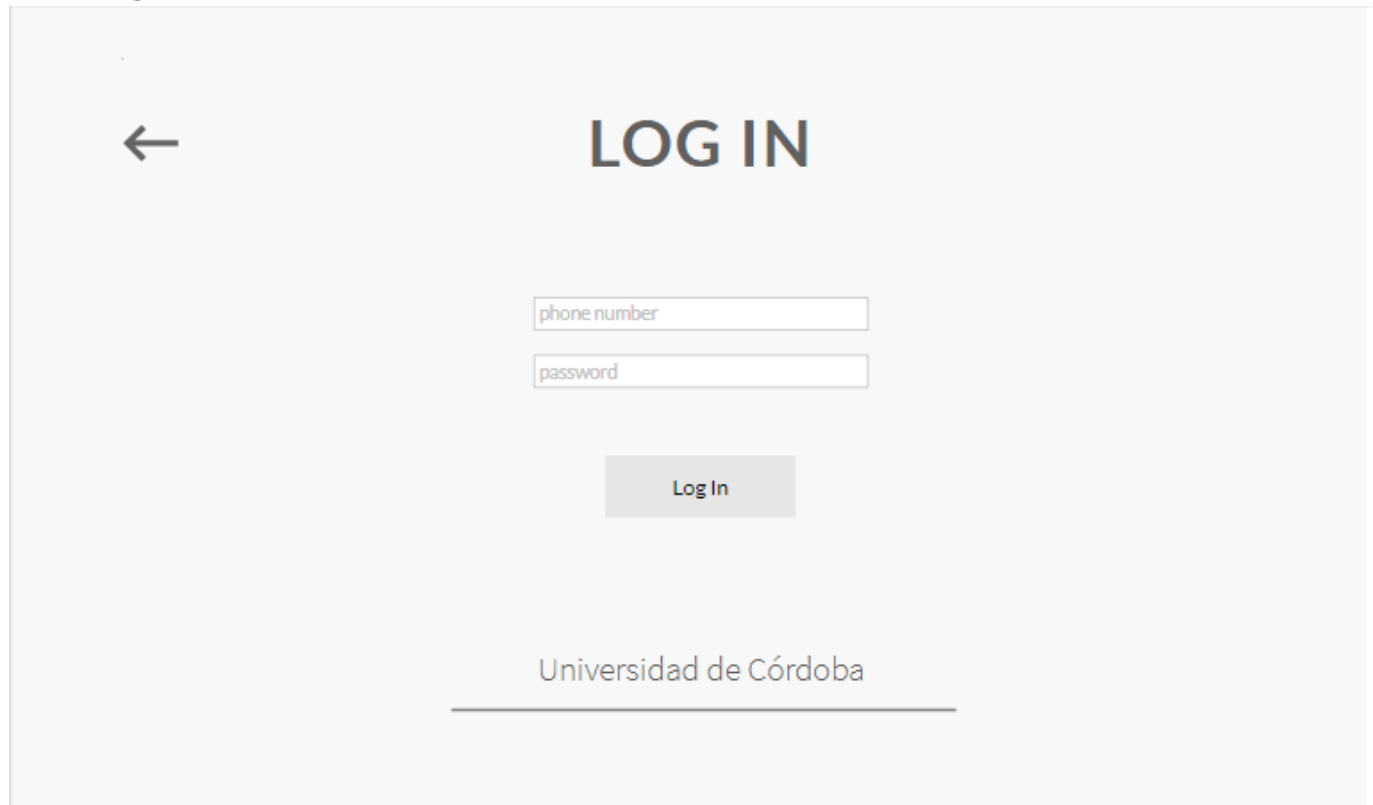
FICHOS

Log In

Registrate

Universidad de Córdoba

COMPONENTE [LOG IN] INTERFASE



A login interface mockup on a light gray background. In the top left corner, there is a dark gray left-pointing arrow. To its right, the text "LOG IN" is displayed in a large, bold, dark blue font. Below this, there are two white input fields with thin gray borders. The first field is labeled "phone number" in a small, gray font, and the second field is labeled "password" in a similar font. Centered below these fields is a gray rectangular button with the text "Log In" in a white font. At the bottom of the interface, the text "Universidad de Córdoba" is centered in a gray font, with a thin gray horizontal line positioned directly beneath it.

RUTAS

GET: /vendedores

CONSULTAS

```
getUser(res, data) {  
  Vendedor.findOne({  
    $and: [  
      { telefono: data.user},  
      { cedula: data.password }  
    ]  
  }, (err, data) => {  
    if (err) throw err;  
    if (!data) {  
      res.status(400).json({  
        status: 400,  
        message: 'Este usuario no está registrado'  
      });  
    } else {  
      res.status(200).json(data);  
    }  
  })  
}
```


COMPONENTE [REGISTRARSE] INTERFASE



The image shows a registration form titled "REGISTRO". On the left, there is a back arrow icon. The form contains four input fields: "name", "programa", "password", and "confirm password". To the right of these fields is a placeholder for a profile picture, showing a silhouette of mountains and a sun, with the text "Añade una imagen" below it. Below the input fields is a checkbox labeled "Acepto los [Terminos y condiciones](#)". At the bottom of the form is a "Registrarse" button. The footer of the form area says "Universidad de Córdoba" with a horizontal line underneath.

RUTAS

POST: /vendedor

CONSULTAS

```
Vendedor.create(vendedor, (err, newV) => {  
  if(err) throw err;  
  res.send(  
    {  
      msg: "Created",  
      nV: newV  
    }  
  )  
})
```



MENÚ

ESTUDIANTES

CARNETS

FICHOS

VENDEDORES

Universidad de Córdoba

Log Out

COMPONENTE [ESTUDIANTES]
INTERFASE

←

ESTUDIANTES

Buscar

Universidad de Córdoba

Log Out

RUTAS

GET: /estudiantes

GET: /estudiantes/:id

CONSULTAS

```
Estudiantes.find({}, (err, estudiantes) => {  
  if (err) throw err;  
  res.send(estudiantes)  
})
```

```
Estudiantes.find({  
  _id: id  
}, (err, estudiante) => {  
  if (err) throw err;  
  res.send(estudiante);  
});
```

COMPONENTE [CARNETS] INTERFASE



RUTAS
GET; /carnets

CONSULTAS

```
Carnet.find({})  
  .populate({  
    path: "id_estudiante",  
    select: "nombre_1 nombre_2 apellido_1 apellido_2 identificacion programa"  
  })  
  .exec(function (err, carnet) {  
    if (err) throw err;  
    res.send(carnet)  
  })
```

COMPONENTE [FICHO]
INTERFASE



RUTAS

GET: /fichos

CONSULTAS

```
Ficho.find({_id: id})  
  .populate({  
    path: "id_carnet",  
    select: "codigo_qr id_estudiante",  
    populate: {  
      path: "id_estudiante",  
      select: "nombre_1 nombre_2 apellido_1 apellido_2 identificacion programa"  
    }  
  })
```

COMPONENTE [VENDEDOR] INTERFASE

←

VENDEDOR

Añadir estudiante

Añadir ficho

Buscar

Añadir carnet

Añadir vendedor

Universidad de Córdoba

Log Out

RUTAS

POST: /estudiantes/5d47abd80d09b409d8fa39d9/generar-carnet

POST: /vendedor

POST: /vendedor/5d47c1b9be742b2f5ccd8048/carnets/5d47c5a3f44e8c1f387c902f/generar-ficho

GET: /vendeor

CONSULTAS

```
Vendedor.create(vendedor, (err, newV) => {  
  if(err) throw err;  
  res.send(  
    {  
      msg: "Vendedor creado",  
      nV: newV  
    }  
  )  
})
```

```
Ficho.create(ficho, (err, newFicho) => {  
  if(err) throw err;  
  res.send({  
    msg: "Ok",  
    nF: newFicho  
  })  
})
```

```
Carnet.create(data, (err, newCarnet) => {  
  if (err) throw err;  
  res.send({  
    status: 200,  
    nC: newCarnet  
  });  
})
```

```
setEstudiante(res, data) {  
  Estudiantes.create(data, (err, newEstudiante) => {  
    if (err) throw err;  
    res.send({  
      status: 200,  
      nE: newEstudiante  
    });  
  })  
}
```



