

OBJECT-RELATIONAL MAPPING

TECNOLOGÍAS

- » SQLite3
- » Node.js
- » Sequelize.js
- » Express
- » Socket.io
- » Angular.js
- » Bootstrap

Medicamentos

Buscar:



	ID_MEDICAMENTO	NOMBRE_MEDICAMENTO	CANTIDAD_DISPONIBLE	LABORATORIO
<input type="radio"/>	57	ALBOCRESIL	57	7
<input type="radio"/>	240	ADEROGL	240	5
<input type="radio"/>	494	ANAFRANIL	494	6
<input type="radio"/>	552	ANUSOL DUO S	552	8
<input type="radio"/>	554	ANUSOL HC	554	8
<input type="radio"/>	555	ANUSOL HC	555	7
<input type="radio"/>	601	ARTRILASE	601	7
<input type="radio"/>	602	ARTRILASE	602	8
<input type="radio"/>	603	ARTRILASE	603	8
<input type="radio"/>	820	BELMALEN PLUS	820	8

Panel de edición

ID_MEDICAMENTO

NOMBRE_MEDICAMENTO

CANTIDAD_DISPONIBLE

LABORATORIO

Salir

Insertar

Modificar

Borrar

Limpiar

SQLITE3

Sistema Gestor de Base de Datos (SGDB)

1. Transacciones ACID
2. Funciona sin servidor
3. Multiplataforma
4. Zero-configuración

SQLite

NODE.JS

Framework asincrono dirigido por eventos

1. JavaScript nativo

2. API nativa

3. Basado en el motor V8 de Chrome

4. Gestor de paquetes (NPM)



SEQUELIZE.JS

Mapeo Objeto-Relacional (ORM)

1. Basado en promesas
2. Diversos dialectos
3. Transacciones sólidas
4. Node.js & io.js

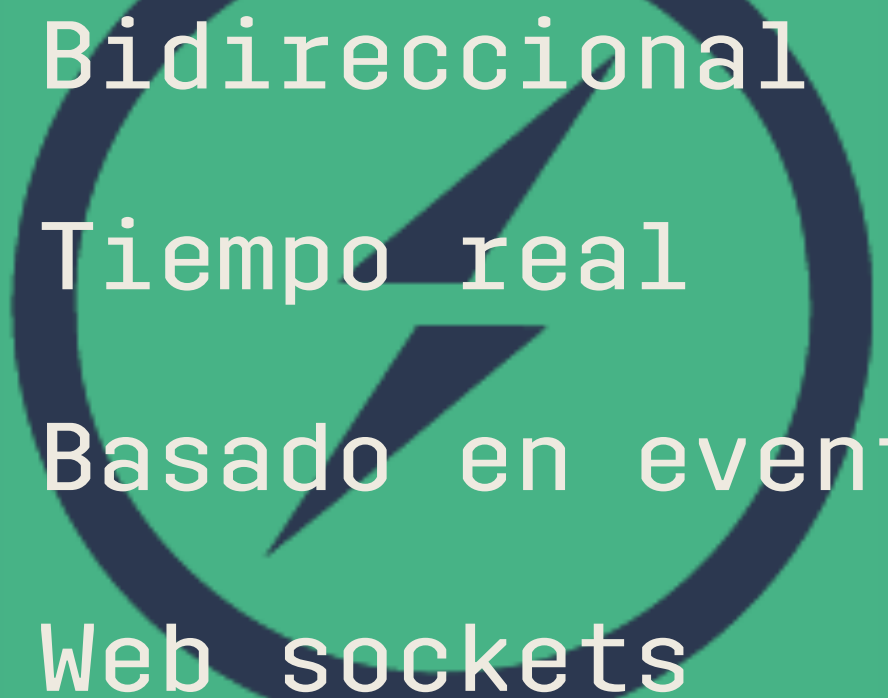
EXPRESS

Web framework

1. Mínimo y flexible
2. Web y móvil
3. Métodos HTTP (REST)
4. No aísla Node.js

SOCKET.IO

Networking framework

- 
1. Bidireccional
 2. Tiempo real
 3. Basado en eventos
 4. Web sockets

socket.io

ANGULAR.JS

MVC framework

1. Aplicaciones dinámicas
2. Web semántica
3. Extensible
4. Data-binding

ANGULARJS
by Google

BOOTSTRAP

UI web framework

1. Responsivo
2. Orientado a móvil
3. Sass & Less
4. jQuery plugins

SEQUELIZE.JS

INICIALIZACIÓN

```
var Sequelize = require('sequelize');  
  
var sequelize = new Sequelize('database.db', null, null, {  
    dialect: 'sqlite',  
    storage: 'data/database.db'  
});
```

MODELLO

PANTALLA

```
var Pantalla = sequelize.define('tPantalla', {  
  pantalla: {  
    type: Sequelize.STRING,  
    primaryKey: true  
  }  
}, {  
  freezeTableName: true,  
  timestamps: false  
});
```

ROL

```
var Rol = sequelize.define('tRol', {
  rolName: {
    type: Sequelize.STRING,
    primaryKey: true
  },
  rolDes: {
    type: Sequelize.STRING
  },
  admin: {
    type: Sequelize.BOOLEAN
  }
}, {
  freezeTableName: true,
  timestamps: false
});
```

PERMISO

```
var Permiso = sequelize.define('tPermiso', {
  rolName: {
    type: Sequelize.STRING,
    primaryKey: true,
    references: {
      model: Rol,
      key: 'rolName'
    }
  },
  pantalla: {
    type: Sequelize.STRING,
    primaryKey: true,
    references: {
      model: Pantalla,
      key: 'pantalla'
    }
  },
  acceso: {
    type: Sequelize.BOOLEAN
  },
  modificacion: {
    type: Sequelize.BOOLEAN
  }
}, {
  freezeTableName: true,
  timestamps: false
});
```

USUARIO

```
var Usuario = sequelize.define('tUsuario', {
  nombre: {
    type: Sequelize.STRING,
    primaryKey: true
  },
  password: {
    type: Sequelize.STRING
  },
  rolName: {
    type: Sequelize.STRING,
    references: {
      model: Rol,
      key: 'rolName'
    }
  }
}, {
  freezeTableName: true,
  timestamps: false
});
```


LABORATORIO

```
var Laboratorio = sequelize.define('tLaboratorio', {  
  ID_LABORATORIO: {  
    type: Sequelize.INTEGER,  
    primaryKey: true  
  },  
  NOMBRE_LABORATORIO: {  
    type: Sequelize.STRING  
  }  
}, {  
  freezeTableName: true,  
  timestamps: false  
});
```

MEDICAMENTO

```
var Medicamento = sequelize.define('tMedicamento', {
  ID_MEDICAMENTO: {
    type: Sequelize.INTEGER,
    primaryKey: true
  },
  NOMBRE_MEDICAMENTO: {
    type: Sequelize.STRING
  },
  CANTIDAD_DISPONIBLE: {
    type: Sequelize.INTEGER
  },
  LABORATORIO: {
    type: Sequelize.INTEGER,
    references: {
      model: Laboratorio,
      key: 'ID_LABORATORIO'
    }
  }
}, {
  freezeTableName: true,
  timestamps: false
});
```

QUERIES

SELECT *

```
Medicamento.findAll().then(function(result) {  
    console.log(result);  
}).catch(function(error) {  
    console.log(error);  
});
```

SELECT <ATTR>

```
Usuario.findAll({  
  where: {  
    nombre: 'phil-fish'  
  }  
}).then(function(result) {  
  console.log(result);  
}).catch(function(error) {  
  console.log(error);  
});
```

MANIPULATION

INSERT

```
Medicamento.create({  
  ID_MEDICAMENTO: 42,  
  NOMBRE_MEDICAMENTO: 'vicodin',  
  CANTIDAD_DISPONIBLE: ∞,  
  LABORATORIO: 'heisenberg-labs'  
}).then(function(result) {  
  console.log(result);  
}).catch(function(error) {  
  console.log(error);  
});
```

UPDATE

```
Medicamento.update({
  ID_MEDICAMENTO:  $\pi$ ,
  NOMBRE_MEDICAMENTO: 'morphine',
  CANTIDAD_DISPONIBLE:  $\phi$ ,
  LABORATORIO: 'burns-biohazard-labs'
}, {
  where: {
    ID_MEDICAMENTO: 42
  }
}).then(function(result) {
  console.log(result);
}).catch(function(error) {
  console.log(error);
});
```


DELETE

```
Medicamento.destroy({  
  where: {  
    ID_MEDICAMENTO: 1337  
  }  
}).then(function(result) {  
  console.log(result);  
}).catch(function(error) {  
  console.log(error);  
});
```