UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Centro de e-Learning
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Diplomatura en programación web full stack con React JS

UTN.BA | Centro de e-Learning
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Secretaría de Cultura y Extensión Universitaria

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Módulo 5:

Base de datos, integración con Node.js

Unidad 4:

Integración de Node.js con Base de datos (parte 2)

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Presentación

A partir de esta unidad comenzamos con la creación del panel de administración del sitio que armamos previamente. Este proyecto nos permitirá vincular todos los conceptos que fuimos incorporando en las unidades anteriores. Para este fin vamos a comenzar programando las funcionalidades de login y logout, así como también incorporaremos el concepto de páginas o rutas privadas.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Objetivos

Que los participantes logren...

- Crear la base de datos y las tablas necesarias para manejar usuarios.
- Comprender los procesos de autenticación de usuarios y la destrucción de sesiones.
- Implementar mecanismos para autorizar o no el acceso a ciertas rutas de la aplicación

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bloques temáticos

1. Login.
2. Logout
3. Rutas privadas.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

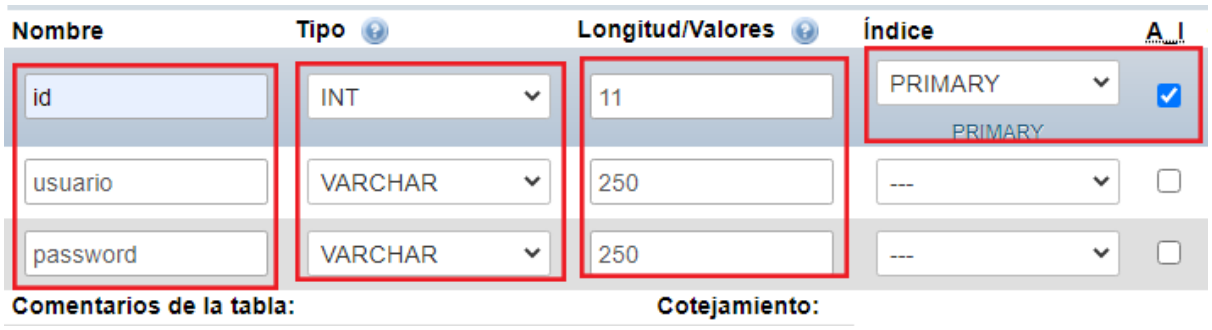
1. Login

Creamos la base de datos: **cervecería.** Para el charset o juego de caracteres seleccionamos **utf8_unicode_ci** que nos permitirá guardar acentos y caracteres especiales sin dificultad.

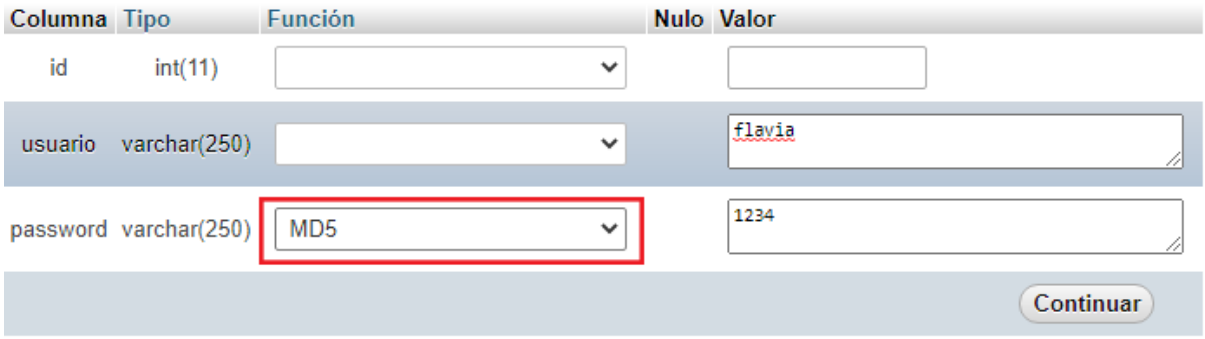
Con nuestra base de datos ya definida vamos a crear nuestra primera tabla, la cual va a almacenar a los usuarios que vamos a permitir ingresar a nuestro panel de administración.

La tabla se llamará **usuarios** y contará con 3 campos:

- **id:** un entero de 11 dígitos que será nuestra clave primaria.
- **usuario:** un varchar de máximo 250 caracteres.
- **password:** otro varchar de máximo 250 caracteres



Cuando aceptemos estos valores tendremos acceso a la opción de **menú insertar,** que nos permitirá agregar registros a nuestra tabla.



Es importante que en el campo password seleccionamos **la función MD5**. Esta función es un algoritmo de encriptación de una sola vía, lo que significa que una vez encriptado el valor no puede ser revertido a un valor legible. Esto lo hacemos por cuestiones de seguridad, ya que si alguna persona llegara a tener acceso a nuestra base de datos, si no encriptan las contraseñas, tendría acceso a las claves de todos los usuarios de nuestra aplicación.

Una vez finalizada la creación de nuestra base de datos instalaremos las dependencias necesarias en nuestro proyecto para poder comenzar con la programación del login.

Instalaremos **express-session, mysql, útil y md5**. A excepción de md5, el resto ya las hemos utilizado en ejemplos previos y conocemos su funcionamiento. md5 cumple la función de encriptar cadenas de texto usando el mismo algoritmo que usamos para encriptar la contraseña de nuestro usuario. En este caso la usaremos para encriptar lo que el usuario ingrese en el formulario de login y de esa manera comparar las 2 cadenas encriptadas para asegurarnos que el usuario sea quien dice ser.

Completados todos los pasos de preparación estamos listos para comenzar con la programación del login de nuestro panel de administración.

Paso 1

Crearemos el archivo **routes/admin/login.js** donde iremos incorporando las rutas relacionadas al proceso. Por el momento solo incluiremos este código.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
var express = require('express');
var router = express.Router();


router.get('/', function (req, res, next) {
  res.render('admin/login', {
    layout: 'admin/layout',
  });
});

module.exports = router;
```

Paso 2

Importamos nuestro manejador de rutas en **app.js** y lo configuramos para que se encargue de las rutas que comienzan con **/admin/login**

```
var loginRouter = require('./routes/admin/login');


app.use('/admin/login', loginRouter);
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Paso 3

Creamos el archivo **views/admin/layout.hbs** que será el nuevo layout base para nuestro administrador.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.
css"
        integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link href="https://fonts.googleapis.com/css?
family=Alfa+Slab+One|Open+Sans:400,700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="/stylesheets/font-awesome.min.css">
    <link rel="stylesheet" href="/stylesheets/style.css">
    <title>Cerveceria X</title>
</head>
<body>
    <nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="/">BEER X</a>
    </nav>

    {{{body}}}
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
    <footer>
        <p class="text-center">Diseñado por: Flavia Ursino año</p>
    </footer>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
        integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
        crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js
"
        integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
        crossorigin="anonymous"></script>
    <script
        src="https://cdn.jsdelivr.net/gh/cferdinandi/smooth-
scroll@15.0/dist/smooth-scroll.polyfills.min.js"></script>
</body>
</html>
```

Paso 4

Creamos el archivo **views/admin/login.hbs** que albergará el formulario de login para acceder al panel.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
<div class="container">
  <div class="row" style="margin:200px 0">
    <div class="col">
      <form action="/admin/login" method="post">
        <div class="form-group">
          <input type="text" class="form-control" placeholder="Usuario"
                 name="usuario">
        </div>
        <div class="form-group">
          <input type="password" class="form-control"
                 placeholder="Password" name="password">
          </div>
        <button type="submit" class="btn btn-primary">Entrar</button>
      </form>
    </div>
  </div>
</div>
```

Paso 5

Para este paso es importante que tengamos configurados los archivos **.env** y **bd.js** con la configuración correcta para conectarnos a nuestro servidor de MySQL local.

Debido a que vamos a empezar a trabajar con más archivos y a realizar más operaciones involucrando a la base de datos, y dichas operaciones se utilizarán en varios archivos, vamos a emplear el concepto de modelos de la arquitectura MVC.

Los modelos serán archivos donde crearemos funciones que nos permitan consultar, crear, actualizar o eliminar un tipo de dato en particular. De esta forma, al importar o requerir estas funciones en otros archivos como nuestros controladores tendremos acceso a la información sin necesidad de tipear varias veces lo mismo.

El primer archivo que crearemos con esta finalidad es **models/usuariosModel.js,** cuyo contenido será el siguiente:

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

```
var pool = require('./bd');
var md5 = require('md5');


async function getUserByUsernameAndPassword(user, password) {
    try {
        var query = "select * from usuarios where usuario = ? and password = ? limit
1";
        var rows = await pool.query(query, [user, md5(password)]);
        return rows[0];
    } catch (error) {
        throw error;
    }
}


module.exports = { getUserByUsernameAndPassword }
```

En este archivo vemos empleado el uso de la estructura **try/catch** que sirve para interceptar excepciones y que estas no lleguen al usuario. Las excepciones son errores que nuestra aplicación arroja y que podemos ir interceptando con distintos bloques try/catch y poder actuar en consecuencia.

La función **getUserByUsernameAndPassword** recibe como parámetros el nombre de usuario y la contraseña y devuelve, en caso de encontrar una coincidencia en la tabla usuarios de nuestra base de datos, la fila correspondiente como un objeto.

Paso 6

En el archivo **routes/admin/login.js** agregaremos el controlador necesario para capturar los datos enviados por el formulario. También vamos a importar el archivo usuariosModel para poder utilizar la función que creamos.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

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


router.post('/', async (req, res, next) => {
  try {
    var usuario = req.body.usuario;
    var password = req.body.password;

    var data = await
usuariosModel.getUserByUsernameAndPassword(usuario, password);

    if (data != undefined) {
      req.session.id_usuario = data.id;
      req.session.nombre = data.usuario;
      res.redirect('/admin/novedades');
    } else {
      res.render('admin/login', {
        layout: 'admin/layout',
        error: true
      });
    }

  } catch (error) {
    console.log(error);

  }
})
```

Como podemos ver, estamos capturando los valores enviados por el formulario y los almacenamos en variables que después pasamos como argumentos del método que creamos en el modelo de usuarios. En caso de que la respuesta sea diferente a undefined (la combinación de usuario y contraseña existe) vamos a crear las variables de sesión id_usuario y nombre y asignaremos su valor a los devueltos por la consulta a la base de datos. Por último redirigimos al usuario a hacia la ruta **/admin/novedades.**

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Si no llega a haber coincidencia, ya sea porque el usuario escribio mal algún dato o porque directamente no existe esa combinación, haremos un render del template de **admin/login.hbs** especificando el nuevo layout de admin/layout.hbs y le enviaremos la variable error seteada en true para poder notificar al usuario de que hubo un error en el proceso de autenticación, para lo cual incluiremos el siguiente código en el template:

```
{{#if error}}
     <p>Usuario o clave incorrecto.</p>
{{/if}}
```

Paso 7

Creamos el archivo **routes/admin/novedades.js** con el siguiente contenido para mostrar el template correspondiente:

```
var express = require('express');
var router = express.Router();

router.get('/', function (req, res, next) {
   res.render('admin/novedades', {
    layout: 'admin/layout',
    usuario: req.session.nombre,
    });
});

module.exports = router;
```

En este archivo pasamos a la vista el nombre de usuario registrado en la sesión que acabamos de iniciar.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Paso 8

Creamos la vista para novedades en **views/admin/novedades.hbs** e ingresamos este contenido.

```
<div class="container" style="margin:50px auto">
    <div class="row">
        <div class="col">
            <p class="text-right">Hola {{ usuario }} ! </p>
        </div>
    </div>
</div>
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

2. Logout

Paso 1

En el archivo **routes/admin/login.js** creamos la función que se encargará de terminar la sesión del usuario.

```
router.get('/logout', function (req, res, next) {
  req.session.destroy();
  res.render('admin/login', {
    layout: 'admin/layout'
  });
});
```

El método **destroy()** del objeto sesión es el encargado de eliminar por completo la sesión del usuario y todos los valores almacenados en ella.

Paso 2

Modificamos nuestro archivo **views/admin/novedades.hbs** e incluimos un link a nuestra nueva función de logout.

```
<div class="row">
    <div class="col">
        <p class="text-right">Hola {{ usuario }} ! <a href="/admin/login/logout"
     class="btn btn-sm btn-danger">Cerrar sesión<i class="fa fa-sign-out"></i>
        </a></p>
    </div>
</div>
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

3. Rutas privadas

El proceso de autorización es el cual en el que se verifica no solo que el usuario haya iniciado correctamente (proceso de autenticación) sino que además tendremos la posibilidad de verificar (en casos más avanzados) que el usuario logueado tenga los permisos necesarios para realizar la acción que esté solicitando.

En nuestro ejercicio simplemente verificaremos que el usuario haya pasado por el proceso de autenticación, el cual deja como resultado algunas variables de sesion que podemos, cuya existencia podemos chequear en nuestras rutas para determinar si está o no logueado.

Paso 1

En el archivo **app.js** vamos a crear una nueva función y vamos a asignarla a una variable llamada secured.

```
secured = async(req,res,next) => {
    try{
        console.log(req.session.id_usuario);
        if(req.session.id_usuario){
            next();
        } else {
            res.redirect('/admin/login');
        }

    }catch(error){
        console.log(error);

    }
}
```

Esta función, que utilizaremos como **middleware** en todas las rutas que deseemos proteger, será la encargada de verificar que exista la variable de sesion id_usuario.

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

En caso de encontrarla ejecutará la función next() que pasa el control de la petición a la siguiente función.

Si un usuario ingresa a una ruta protegida sin haber iniciado sesión correctamente interrumpimos el proceso de la petición y lo redirigimos hacia la pantalla de login.

Para implementar esta nueva función en nuestras rutas simplemente debemos ponerla antes del manejador de ruta al momento de declararla.

```
app.use('/admin/novedades',secured, adminNovedadesRouter);
```

Centro de e-Learning SCEU UTN - BA. Medrano 951 2do piso
(1179) // Tel. +54 11 7078- 8073 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Bibliografía utilizada y sugerida

Artículos de revista en formato electrónico:

Google Cloud. Disponible desde la URL: https://cloud.google.com/functions/docs/writing/specifying-dependencies-nodejs?hl=es-419

PhpMyAdmin. Disponible desde la URL: https://www.phpmyadmin.net/

npmjs. Disponible desde la URL: https://www.npmjs.com/