Manual Técnico de la Aplicación "Cursos Virtuales Enfermeras"

Versión: 1.1

Requisitos del Sistema

Hardware:

Procesador: Intel i5 o superior

RAM: 8GB mínimo

Espacio en disco: 50GB mínimo

Software:

Sistema Operativo: Windows 10 o superior

Visual Studio con soporte para ASP.NET

SQL Server (versión reciente)

.NET Framework 4.8

Instalación del Proyecto

1. Descargar el Repositorio:

Clona el repositorio con el siguiente comando en la terminal:

git clone <URL-del-repositorio>

cd <nombre-del-repositorio>

2. Abrir el Proyecto en Visual Studio:

Abre Visual Studio y selecciona la opción "Abrir un proyecto o solución".

Navega al directorio clonado y selecciona el archivo .sln correspondiente.

3. Configurar la Cadena de Conexión:

En el archivo DbContext del proyecto, localiza la sección donde se define la cadena de conexión.

Sustituye la cadena existente con la siguiente:

"Server=localhost;Database=DBNurseCourse;User Id=sa;Password=tu\_contraseña\_sql;"

Guarda los cambios.

4. Preparar la Base de Datos:

Inicia SQL Server y asegúrate de que el servicio esté activo.

Crea una nueva base de datos llamada DBNurseCourse.

Ejecuta los scripts SQL incluidos en el repositorio para crear las tablas necesarias.

5. Ejecutar el Proyecto:

Configura el proyecto como inicio predeterminado en Visual Studio.

Pulsa F5 o haz clic en "Ejecutar" para iniciar la aplicación.

-

Credenciales para APIs

## 🔐 **1. Crear y configurar tu cuenta en Auth0**

1. Ve a [https://auth0.com](https://auth0.com/) y crea una cuenta (o inicia sesión).
2. En el **Dashboard**:
   * Ve a **Applications > Applications**.
   * Crea una nueva aplicación:
     + **Name**: Tu nombre de proyecto.
     + **Type**: Selecciona **Regular Web Application**.
     + Haz clic en **Create**.

## ⚙️ **2. Configura la aplicación en Auth0**

En tu nueva app:

1. En la pestaña **Settings**:
   * **Allowed Callback URLs**: https://localhost:5001/callback
   * **Allowed Logout URLs**: https://localhost:5001/
   * **Allowed Web Origins**: https://localhost:5001
   * Guarda los cambios.
2. Anota:
   * Domain
   * Client ID
   * Client Secret

## 🧠 **3. Instalar paquetes necesarios en tu proyecto C# (ASP.NET Core)**

En el proyecto, abre la consola del administrador de paquetes o terminal:

dotnet add package Auth0.AspNetCore.Authentication

## 🛠️ **4. Configurar Auth0 en** Program.cs **(ASP.NET Core 6/7/8)**

var builder = WebApplication.CreateBuilder(args);

// Configura Auth0

builder.Services.AddAuth0WebAppAuthentication(options => {

options.Domain = builder.Configuration["Auth0:Domain"];

options.ClientId = builder.Configuration["Auth0:ClientId"];

options.ClientSecret = builder.Configuration["Auth0:ClientSecret"];

});

builder.Services.AddControllersWithViews();

var app = builder.Build();

app.UseAuthentication();

app.UseAuthorization();

app.MapDefaultControllerRoute();

app.Run();

## 📄 **5. Añadir configuración en** appsettings.json

"Auth0": {

"Domain": "tu-dominio.auth0.com",

"ClientId": "tu-client-id",

"ClientSecret": "tu-client-secret"

}

## 🔑 **6. Crear las rutas para Login y Logout**

### En un controlador (AccountController.cs):

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.AspNetCore.Mvc;

public class AccountController : Controller

{

public IActionResult Login(string returnUrl = "/")

{

return Challenge(new AuthenticationProperties { RedirectUri = returnUrl });

}

public IActionResult Logout()

{

return SignOut(new AuthenticationProperties

{

RedirectUri = "/"

},

CookieAuthenticationDefaults.AuthenticationScheme,

"Auth0");

}

}

## 🧾 **7. Proteger rutas y acceder a información del usuario**

### Para proteger controladores:

[Authorize]

public IActionResult Dashboard()

{

return View();

}

### Obtener datos del usuario en un controlador:

var name = User.Identity?.Name;

var email = User.Claims.FirstOrDefault(c => c.Type == "email")?.Value;

## 👥 **8. Gestión de usuarios y roles en Auth0**

### Roles:

1. En el dashboard Auth0 ve a **User Management > Roles**
2. Crea roles como admin, editor, etc.
3. Asocia roles a usuarios desde **Users > (selecciona un usuario) > Roles**

### Permitir que roles lleguen como claims:

1. Ve a **Actions > Flows > Login** y añade una acción llamada **"Add user roles to ID token"** con este código:

exports.onExecutePostLogin = async (event, api) => {

const namespace = 'https://schemas.yoursite.com/';

const assignedRoles = event.authorization.roles;

api.idToken.setCustomClaim(`${namespace}roles`, assignedRoles);

};

1. En tu código C# puedes acceder al claim:

var roles = User.Claims.Where(c => c.Type == "https://schemas.yoursite.com/roles").Select(c => c.Value);

## ✅ **Resumen**

| **Paso** | **Acción** | **Detalles** |
| --- | --- | --- |
| 1 | Crear App en Auth0 | Regular Web App |
| 2 | Configurar URLs permitidas | Callback y logout |
| 3 | Instalar paquete | Auth0.AspNetCore.Authentication |
| 4 | Configurar Program.cs | Middleware de autenticación |
| 5 | Configurar appsettings.json | Credenciales de Auth0 |
| 6 | Crear rutas de login/logout | En controlador |
| 7 | Proteger rutas | [Authorize] |
| 8 | Roles y claims | Action y acceso a roles |

-Ingresa al proyecto

-Lograte preferiblemente con Google

-tendras acceo por defecto como estudiante si necesitas administrador entra a la api al lado de users y habilitateb el rol

# Guía de Uso del Sistema NurseCourse

## Acceso al Sistema

1. Ingrese a la dirección web del sistema (por ejemplo, https://localhost:5001).
2. Haga clic en la opción "Iniciar sesión".
3. Será redirigido a la plataforma de autenticación de Auth0.
4. Introduzca su correo electrónico y contraseña.
5. Tras un inicio de sesión exitoso, regresará automáticamente al sistema y podrá utilizar las funcionalidades permitidas según su tipo de usuario.

## Funcionalidades para Usuarios Comunes (Estudiantes o Usuarios sin rol administrativo)

Una vez autenticado, un usuario común tiene acceso a:

* Ver el contenido publicado (videos, audios, artículos, documentos PDF).
* Consultar los cursos disponibles y sus módulos.
* Acceder al material de cada módulo (videos, documentos PDF).
* Realizar las evaluaciones asociadas a los módulos.
* Obtener certificados al completar cursos (si están habilitados).
* Cerrar sesión desde cualquier parte del sistema usando la opción correspondiente.

## Funcionalidades para Administradores

Además de todas las funciones disponibles para los usuarios comunes, el administrador tiene acceso al **panel de administración**, donde puede realizar lo siguiente:

### Gestión de Usuarios

* Consultar la lista de usuarios registrados.
* Promover usuarios comunes al rol de administrador.
* Eliminar cuentas de usuarios si es necesario.

### Gestión de Contenidos

* Crear nuevos contenidos: videos, audios, artículos o libros digitales.
* Ingresar metadatos como título, descripción, tipo, etiquetas, imagen y archivo adjunto.
* Editar contenidos previamente cargados.
* Eliminar contenidos que ya no sean necesarios.

### Gestión Académica

* Crear y administrar cursos.
* Crear módulos asociados a cursos.
* Subir material en PDF o videos a cada módulo.
* Crear y gestionar evaluaciones (preguntas, respuestas, opciones).
* Asociar certificados a los cursos que lo requieran.

## Estructura Académica del Sistema

El sistema organiza el contenido educativo en dos niveles principales: **Cursos** y **Módulos**.

### Curso

Un **curso** es una unidad de formación general que agrupa varios módulos. Incluye:

* Nombre del curso.
* Descripción.
* Lista de módulos que lo componen.
* Lista de certificados entregables al finalizar el curso (si corresponde).

**Ejemplo:**  
Curso: “Técnicas de Afrontamiento del Estrés”  
Incluye: Introducción, Técnicas de Relajación, Evaluación Final.

### Módulo

Un **módulo** es una sección específica dentro de un curso. Incluye:

* Nombre y descripción.
* Relación directa con un curso.
* Archivo PDF como guía o material de lectura.
* Video explicativo (opcional).
* Evaluación con preguntas y opciones (opcional).

**Ejemplo:**  
Módulo: “Respiración Consciente”  
Contenido: Video explicativo + Guía PDF + Preguntas evaluativas.

## Roles de Usuario y Seguridad

* Los roles (usuario común o administrador) se definen en la plataforma Auth0.
* Solo los usuarios con rol de **administrador** tienen acceso al panel de gestión.
* Si un usuario no puede acceder a ciertas funciones, debe comunicarse con el administrador para verificar que su cuenta esté correctamente configurada.

## Cierre de Sesión

1. Seleccione la opción "Cerrar sesión" disponible en el menú o parte superior de la interfaz.
2. El sistema lo redirigirá fuera de la plataforma.

## Recomendaciones

* Verifique siempre estar autenticado con el rol correcto.
* Asegúrese de cerrar sesión al finalizar el uso.
* Si presenta problemas de acceso o errores, contacte al administrador o responsable técnico del sistema.
* Para cargar contenidos correctamente, utilice los formatos requeridos y revise que los datos ingresados sean coherentes.