**Manual Técnico de la Aplicación "Cursos Virtuales Enfermeras"**

**1. Portada**

* **Título**: Manual Técnico de la Aplicación "Cursos Virtuales Enfermeras"
* **Fecha**: 02/12/2024
* **Versión**: 2.0
* **Autores**: Noelia Gonzales,Lucas Espinoza

14. Cursos Modulares  
15. Creación de Exámenes  
16. Utilización de Videos  
17. Generación de Títulos en PDF  
18. Función para Subir PDFs

**2. Índice**

* Introducción
* Requisitos del Sistema
* Instalación
* Arquitectura de la Aplicación
* Componentes de la Aplicación
* Bases de Datos
* Funcionalidades Clave
* Mantenimiento y Actualizaciones
* Solución de Problemas Comunes
* Seguridad
* Anexos

**3. Introducción**

La aplicación "Cursos Virtuales Enfermeras" pretende proporcionar una plataforma educativa para enfermeras, permitiendo el acceso a cursos online diseñados para mejorar sus habilidades y conocimientos. Este manual técnico proporciona una guía completa sobre la instalación, configuración, y mantenimiento de la aplicación, además de ofrecer detalles sobre su arquitectura y componentes clave. La aplicación cuenta con dos roles principales: Administrador y Estudiante.

**4. Requisitos del Sistema**

* **Requisitos de hardware**:
  + Procesador: Intel i5 o superior
  + RAM: 8GB mínimo
  + Espacio en disco: 50GB mínimo
* **Requisitos de software**:
  + Sistema operativo: Windows 10 o superior
  + .NET Framework 4.8
  + Docker
  + DBEaver para la administración de bases de datos

**5. Instalación**

* **Instalación del entorno de desarrollo**:
  + **Visual Studio**:
    - Descarga e instala [Visual Studio](https://visualstudio.microsoft.com/) con soporte para ASP.NET.
    - Selecciona las cargas de trabajo necesarias durante la instalación, incluyendo "Desarrollo de aplicaciones web y ASP.NET".
  + **Docker**:
    - Descarga e instala Docker Desktop.
    - Asegúrate de que Docker esté funcionando correctamente ejecutando docker --version en una terminal.
  + **DBEaver**:
    - Descarga e instala DBEaver.
    - Configura la conexión a la base de datos utilizando la URL de conexión: jdbc:mysql://localhost:3306/DBNurseCourse?user=root&password=jose&allowPublicKeyRetrieval=true&useSSL=false.
* **Instalación de la aplicación**:
  + **Clonar el repositorio del proyecto**:

bash

Copiar código

git clone <URL-del-repositorio>

cd <nombre-del-repositorio>

* + **Configurar los contenedores Docker**:
    - Ejecuta el siguiente comando para iniciar el contenedor de MySQL:

bash

Copiar código

docker run --name mysql-container -e MYSQL\_ROOT\_PASSWORD=root -p 3306:3306 -d mysql

* + - Accede al contenedor de MySQL:

bash

Copiar código

docker exec -it mysql-container mysql -u root -p

* + - Crea la base de datos:

sql

Copiar código

CREATE DATABASE DBNurseCourse;

* + - Ejecuta el siguiente comando para iniciar el contenedor de Minio:

bash

Copiar código

docker run -p 9000:9000 -p 9001:9001 --name minio1 -e "MINIO\_ROOT\_USER=minio" -e "MINIO\_ROOT\_PASSWORD=minio1234" -v "C:/Users/VICTOR123/OneDrive/Documentos/Falso\_Color/minio:/data" -d minio/minio server /data --console-address ":9001"

* **Desplegar la aplicación en el servidor**:
  + Configura el proyecto en Visual Studio.
  + Publica el proyecto en el servidor utilizando las herramientas de despliegue de Visual Studio.

**6. Arquitectura de la Aplicación**

* **Descripción general**:
  + La arquitectura está basada en ASP.NET.
  + Utiliza contenedores Docker para la base de datos y el almacenamiento de archivos.

**7. Componentes de la Aplicación**

* **Frontend**:
  + Interfaz de usuario desarrollada en ASP.NET con componentes de Windows Forms para evaluaciones.
* **Backend**:
  + Lógica del servidor y gestión de usuarios, cursos, y material educativo.
* **API**:
  + Endpoints para la gestión de usuarios, cursos, evaluaciones, y generación de certificados.

**8. Bases de Datos**

* **Modelo de datos**:
  + Diagramas de la base de datos con tablas para usuarios, cursos, evaluaciones, y materiales educativos.
* **Consultas principales**:
  + Ejemplos de consultas SQL para la gestión de datos.
  + **Esquema de la base de datos**:

sql

Copiar código

CREATE TABLE Roles (

RolID int NOT NULL AUTO\_INCREMENT,

NombreRol varchar(50) NOT NULL,

PRIMARY KEY (RolID)

);

CREATE TABLE Cursos (

CursoID int NOT NULL AUTO\_INCREMENT,

Titulo varchar(150) NOT NULL,

Descripcion varchar(350) NOT NULL,

PRIMARY KEY (CursoID)

);

CREATE TABLE Modulos (

ModuloID int NOT NULL AUTO\_INCREMENT,

Nombre varchar(150) NOT NULL,

Orden int NOT NULL,

CursoID int NOT NULL,

PRIMARY KEY (ModuloID),

FOREIGN KEY (CursoID) REFERENCES Cursos(CursoID)

);

CREATE TABLE Usuarios (

UsuarioID int NOT NULL AUTO\_INCREMENT,

Nombre varchar(50) NOT NULL,

CorreoElectronico varchar(50) NOT NULL,

Contraseña varchar(150) NOT NULL,

Edad int NOT NULL,

Cargo varchar(50) NOT NULL,

RolID int NOT NULL,

PRIMARY KEY (UsuarioID),

FOREIGN KEY (RolID) REFERENCES Roles(RolID)

);

CREATE TABLE Examenes (

ExamenID int NOT NULL AUTO\_INCREMENT,

Titulo varchar(150) NOT NULL,

LinkExame varchar(550) NOT NULL,

CursoID int NOT NULL,

PRIMARY KEY (ExamenID),

FOREIGN KEY (CursoID) REFERENCES Cursos(CursoID)

);

CREATE TABLE Contenidos (

ContenidoID int NOT NULL AUTO\_INCREMENT,

Tipo varchar(50) NOT NULL,

URL varchar(550) NOT NULL,

Texto varchar(550) NOT NULL,

ModuloID int NOT NULL,

PRIMARY KEY (ContenidoID),

FOREIGN KEY (ModuloID) REFERENCES Modulos(ModuloID)

);

CREATE TABLE Progreso (

ProgresoID int NOT NULL AUTO\_INCREMENT,

ModuloActual int NOT NULL,

Completo bool NOT NULL,

ContenidoID int NOT NULL,

UsuarioID int NOT NULL,

PRIMARY KEY (ProgresoID),

FOREIGN KEY (ContenidoID) REFERENCES Contenidos(ContenidoID),

FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID)

);

CREATE TABLE NotaExamen (

NotaExamenId int NOT NULL AUTO\_INCREMENT,

UsuarioID int NOT NULL,

ExamenID int NOT NULL,

Calificacion double NOT NULL,

PRIMARY KEY (NotaExamenId),

FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID),

FOREIGN KEY (ExamenID) REFERENCES Examenes(ExamenID)

);

INSERT INTO Roles (NombreRol) VALUES

('Usuario'),

('Administrador');

**9. Funcionalidades Clave**

* **Registro y autenticación de usuarios**:
  + Registro de médicos e internistas con control de tiempo y acceso limitado.
  + Validación mediante el código del Ministerio de Salud.
* **Gestión de cursos**:
  + Creación y actualización de cursos por parte del administrador.
  + Acceso secuencial a módulos y evaluaciones.
  + Evaluaciones al inicio y fin del curso con control de tiempo.
* **Generación de certificados**:
  + Certificado en PDF al aprobar el curso, controlado por el administrador.
  + CRUD de videos y PDFs.

**10. Mantenimiento y Actualizaciones**

* **Procedimientos de mantenimiento**:
  + Limpieza de datos, monitorización del rendimiento.
* **Proceso de actualización**:
  + Despliegue de nuevas versiones de la aplicación y actualización de contenedores Docker.

**11. Solución de Problemas Comunes**

* **Errores comunes y soluciones**:
  + Problemas de conexión a la base de datos y Docker.
  + Fallos en la carga de materiales educativos.
* **Logs y monitoreo**:
  + Acceso y análisis de logs para la solución de problemas.

**12. Seguridad**

* **Medidas de seguridad implementadas**:
  + Autenticación y autorización de usuarios.
  + Seguridad en la transmisión de datos.
* **Pruebas de seguridad**:
  + Análisis de vulnerabilidades y pruebas de penetración.

**13. Anexos**

* **Glosario de términos**:
  + **ASP.NET**: Un framework para aplicaciones web desarrollado por Microsoft que permite a los desarrolladores construir aplicaciones dinámicas y servicios web.
  + **Docker**: Una plataforma que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores, asegurando que las aplicaciones se ejecuten de manera consistente en diferentes entornos.
  + **DBEaver**: Una herramienta de administración de bases de datos de código abierto que permite a los usuarios interactuar con bases de datos a través de una interfaz
  + **Minio**: Un sistema de almacenamiento de objetos de alta eficiencia compatible con la API de Amazon S3.
  + **CRUD**: Acrónimo de Create, Read, Update, Delete; operaciones básicas utilizadas en la gestión de datos.
* **Referencias y recursos adicionales**:
* [Documentación de Visual Studio](https://docs.microsoft.com/en-us/visualstudio/)
* Guía de instalación de Docker Desktop
* Página de descarga de DBEaver
* Documentación de Minio

# 14. Cursos Modulares

#### Descripción

La aplicación permite estructurar los cursos en módulos para facilitar el aprendizaje progresivo. Cada curso puede contener múltiples módulos organizados de manera secuencial.

#### Funcionalidades

- \*\*Creación de Módulos\*\*: Los administradores pueden dividir un curso en módulos temáticos o por niveles.

- \*\*Seguimiento del Progreso\*\*: El sistema registra el avance del usuario en cada módulo, permitiendo reanudar desde el último punto alcanzado.

- \*\*Gestión de Contenidos\*\*: Dentro de cada módulo, se pueden añadir videos, PDFs y evaluaciones específicas.

#### Implementación Técnica

Los módulos se almacenan en la tabla `Modulos` de la base de datos y están relacionados con los cursos a través de claves foráneas.

# 15. Creación de Exámenes

#### Descripción

Los exámenes permiten evaluar el conocimiento adquirido por los estudiantes al final de cada módulo o curso.

#### Funcionalidades

- \*\*Editor de Exámenes\*\*: Los administradores pueden crear exámenes personalizados desde la plataforma, incluyendo preguntas de opción múltiple, verdadero/falso y respuestas abiertas.

- \*\*Evaluación Automática\*\*: Las preguntas de opción múltiple y verdadero/falso son calificadas automáticamente.

- \*\*Resultados y Retroalimentación\*\*: Los usuarios pueden visualizar sus calificaciones y recibir comentarios sobre sus respuestas.

#### Implementación Técnica

Los exámenes se gestionan en la tabla `Examenes` de la base de datos. Las calificaciones se registran en la tabla `NotaExamen`.

# 16. Utilización de Videos

#### Descripción

Los cursos pueden integrar videos como recurso educativo principal o complementario.

#### Funcionalidades

- \*\*Carga de Videos\*\*: Los administradores pueden subir videos a la plataforma.

- \*\*Reproducción Integrada\*\*: Los estudiantes pueden visualizar los videos directamente desde la aplicación.

- \*\*Seguimiento de Visualización\*\*: El sistema registra qué videos han sido vistos y por cuánto tiempo.

#### Implementación Técnica

Los videos se almacenan en un servidor de objetos como Minio, y sus metadatos se registran en la tabla `Contenidos` de la base de datos.

# 17. Generación de Títulos en PDF

#### Descripción

La aplicación permite generar certificados o títulos en formato PDF tras la finalización de un curso.

#### Funcionalidades

- \*\*Diseño Personalizable\*\*: Los administradores pueden configurar plantillas para los certificados.

- \*\*Generación Automática\*\*: Al aprobar el curso, el sistema genera automáticamente el PDF con los datos del usuario y el curso.

- \*\*Descarga Directa\*\*: Los usuarios pueden descargar el certificado desde su perfil.

#### Implementación Técnica

La generación de PDFs se realiza mediante bibliotecas de manejo de PDF como iText o PDFsharp.

# 18. Función para Subir PDFs

#### Descripción

La plataforma admite la carga de archivos PDF como material educativo adicional.

#### Funcionalidades

- \*\*Subida de Archivos\*\*: Los administradores pueden cargar PDFs asociados a módulos o cursos específicos.

- \*\*Organización de Recursos\*\*: Los PDFs se clasifican por curso y módulo, facilitando su acceso.

- \*\*Visualización y Descarga\*\*: Los estudiantes pueden visualizar los PDFs directamente en la aplicación o descargarlos.

#### Implementación Técnica

Los archivos PDF se almacenan en el servidor de objetos y sus referencias se guardan en la tabla `Contenidos`.