



Examen Psicométrico de la Universidad del Norte

Documentación

Equipo 3

44534 – Abraham Moreno Pérez
43722 – Mario Alberto Loredó Cantú
44498 – Rogelio Samuel Ambriz Castro
42410 – Ricardo Jhair Sifuentes Ontiveros



Índice

123



Requisitos del Sistema

Requisitos de Software

Componente	Versión mínima recomendada	Descripción
Sistema Operativo	Windows 10 (64 bits) o superior	Compatible con .NET SDK y Visual Studio
.NET SDK	.NET 8.0	Requerido para compilar y ejecutar el proyecto MVC
Visual Studio / VS Code	Visual Studio 2022+ / VS Code + Ext.	IDE recomendado para desarrollo y depuración
SQL Server	SQL Server 2019 o superior	Base de datos relacional utilizada
SQL Server Management Studio (SSMS)	SSMS 18+	Administración y pruebas de base de datos
Navegador Web	Google Chrome / Edge / Firefox	Para acceder a la aplicación web
Git (opcional)	Git 2.x	Para control de versiones si se usa integración con GitHub
Node.js y npm (opcional)	Última estable	Solo si se incluye soporte para frontend moderno o Bootstrap



Requisitos del Sistema

Requisitos de Hardware

Componente	Mínimo	Recomendado
Procesador	Intel Core i3 (o equivalente)	Intel Core i5/i7 de 10 ^a gen o superior
Memoria RAM	4 GB	8 GB o más
Espacio en disco	2 GB libres para proyecto y entorno	10 GB para entorno completo (VS, SQL, etc.)
Resolución de pantalla	1366x768	1920x1080 (Full HD)
Conectividad	Acceso a internet (para login con Google)	Conexión estable



Estructura del MVC

Resumen del Sistema

El sistema desarrollado es un portal web en ASP.NET Core MVC orientado a la gestión y aplicación de exámenes psicométricos para estudiantes. Está diseñado con una arquitectura modular, donde se distinguen tres roles principales de usuario: Administrador, Docente y Estudiante. Permite autenticación mediante Google, asignación de exámenes, visualización de resultados e interpretación automática del nivel de ansiedad.

Tecnologías Utilizadas

- **Frontend:** HTML, CSS, y Razor Pages
- **Backend:** ASP.NET Core MVC (.NET 8)
- **Base de Datos:** SQL Server
- **ORM:** Entity Framework Core
- **Autenticación:** Login con Google
- **Control de Versiones:** Git
- **IDE:** Visual Studio Code

Estructura del MVC

Arquitectura MVC

- **Modelos:** Representan la lógica de negocio y entidades como Usuario, Pregunta, ResultadoExamen, etc.
- **Vistas:** Interfaces para los diferentes roles. Usan Razor (.cshtml) para lógica de presentación.
- **Controladores:** Manejan las peticiones de HTTP, como por ejemplo, LoginController

Roles del Sistema

Rol	Funciones Principales
Administrador	Registrar usuarios, asignar exámenes, modificar preguntas, ver resultados
Docente	Visualizar usuarios y resultados
Estudiante	Realizar el examen psicométrico, visualizar resultado personal

Estructura del MVC

Flujo de Autenticación

- El usuario accede mediante Login con Google.
- Se verifica si el correo está registrado en la base de datos (USUARIO).
- Si está autorizado, se crea sesión con su Usuariold y Rol.
- Redirección a la vista correspondiente al rol.
- Si no está autorizado, se redirige a la vista NoAutorizado.

Estructura de Base de Datos

- **USUARIO:** Id, Nombre, Correo, Matricula, Rol
- **PREGUNTA:** Id, Texto, Activa
- **RESULTADOEXAMEN:** Id, Usuariold (FK), PuntajeTotal, Interpretación, Fecha
- **RESPUESTA:** Id, Preguntald (FK), Usuariold (FK), Valor (0–3)

Estructura del MVC

Casos de Uso

- **Administrador**
 - Crear usuarios (matrícula, nombre, correo, rol)
 - Agregar/modificar/eliminar preguntas del examen
 - Ver resultados de todos los estudiantes
- **Docente**
 - Consultar lista de usuarios
 - Consultar resultados
- **Estudiante**
 - Presentar examen psicométrico (una pregunta a la vez, con temporizador de 20 segundos)
 - Ver resultado al finalizar

Descripción de Controladores

- **LoginController.cs**
 - Maneja login con Google
 - Verifica correo contra DB
- **UsuarioController.cs**
 - RegisterUser: formulario de registro de usuarios (solo admin)
 - ListUsers: lista de usuarios registrados (solo admin/docente)
- **ExamenPsicoController.cs**
 - Muestra preguntas una a una con temporizador
 - Guarda respuestas en RESPUESTA
 - Calcula y guarda RESULTADOEXAMEN
- **PreguntaController.cs**
 - Gestión CRUD de preguntas (solo admin)
 - Usa PreguntaViewModel para vista combinada

Estructura del MVC

Vistas Implementadas

- **Views/Usuario/RegisterUser.cshtml**
formulario de registro
- **Views/Usuario/ListUsers.cshtml**
tabla de usuarios
- **Views/ExamenPsico/Examen.cshtml**
ejecución del examen (pregunta por pregunta, temporizador)
- **Views/ExamenPsico/Resultado.cshtml**
muestra resultado e interpretación
- **Views/Pregunta/Index.cshtml**
gestión de preguntas (alta, edición, activar/desactivar)

Vista del Examen

- Cada pregunta se muestra individualmente
- Temporizador de 20 segundos
- Botones: Siguiente, Finalizar
- Al final se calcula el puntaje y se interpreta el resultado



Base de Datos

Código

```
CREATE DATABASE Psicometrico1
go

CREATE TABLE Pregunta (
    PreguntaId INT PRIMARY KEY IDENTITY(1,1),
    Texto NVARCHAR(500) NOT NULL
);

CREATE TABLE Usuario (
    UsuarioId INT PRIMARY KEY IDENTITY(1,1),
    Nombre NVARCHAR(100),
    Correo NVARCHAR(100)
    -- Agrega más campos si lo necesitas
);

ALTER TABLE Usuario ADD Matricula NVARCHAR(50) NULL;
ALTER TABLE Usuario
ADD Rol NVARCHAR(20) NOT NULL DEFAULT 'Estudiante';
select * from Usuario

INSERT INTO Usuario (Nombre, Correo, Matricula, Rol)
VALUES ('Rogelio', 'al44498@un.edu.mx', '44498', 'Administrador'),
('Administrador Principal', 'equipoinfo650@gmail.com', 'ADM001', 'Administrador');

select * from Usuario

-- Tabla de respuestas del examen por usuario
CREATE TABLE RespuestaExamen (
    RespuestaId INT PRIMARY KEY IDENTITY(1,1),
    UsuarioId INT NOT NULL,
    PreguntaId INT NOT NULL,
    Valor INT NOT NULL, -- 0: En absoluto, 1: Levemente, 2: Moderadamente, 3: Severamente
    FechaRegistro DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (PreguntaId) REFERENCES Pregunta(PreguntaId),
    FOREIGN KEY (UsuarioId) REFERENCES Usuario(UsuarioId) -- Asegúrate de tener esta tabla
);
INSERT INTO RespuestaExamen (UsuarioId, PreguntaId, Valor)
VALUES (1, 3, 2);
ALTER TABLE RespuestaExamen
ADD ValorRespuesta INT NULL;

ALTER TABLE RespuestaExamen
ADD FechaRespuesta DATETIME NULL;

INSERT INTO Pregunta (Texto) VALUES
('Torpe o entumecido.'),
('Acalorado.'),
('Con temblor en las piernas.'),
('Incapaz de relajarse.'),
('Con temor a que ocurra lo peor.'),
('Mareado, o que se le va la cabeza.'),
('Con latidos del corazón fuertes y acelerados.'),
('Inestable.'),
('AltamORIZADO o asustado.'),
('Nervioso.'),
('Con sensación de bloqueo.'),
('Con temblores en las manos.'),
('Inquieto, inseguro.'),
('Con miedo a perder el control.'),
('Con sensación de ahogo.'),
('Con temor a morir.'),
('Con miedo.'),
('Con problemas digestivos.'),
('Con desvanecimientos.'),
('Con rubor facial.'),
('Con sudores, frios o calientes.');

INSERT INTO RespuestaExamen (UsuarioId, PreguntaId, Valor)
VALUES (1, 3, 2); -- Usuario 1 respondió "Moderadamente" a la pregunta 3

SELECT
    re.UsuarioId,
    p.Texto AS Pregunta,
    CASE re.Valor
        WHEN 0 THEN 'En absoluto'
        WHEN 1 THEN 'Levemente'
        WHEN 2 THEN 'Moderadamente'
        WHEN 3 THEN 'Severamente'
    END AS Respuesta
FROM RespuestaExamen re
JOIN Pregunta p ON re.PreguntaId = p.PreguntaId;

SELECT UsuarioId, SUM(Valor) AS TotalPuntos
FROM RespuestaExamen
GROUP BY UsuarioId;

select * from RespuestaExamen

CREATE TABLE ResultadoExamen (
    ResultadoId INT PRIMARY KEY IDENTITY(1,1),
    UsuarioId INT,
    PuntajeTotal INT,
    Diagnostico NVARCHAR(100),
    Fecha DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (UsuarioId) REFERENCES Usuario(UsuarioId)
);

select * from ResultadoExamen

select * from Pregunta

ALTER TABLE Pregunta ADD Activa BIT NOT NULL DEFAULT 1;

select * from Usuario

INSERT INTO Usuario (Nombre, Correo, Matricula, Rol)
VALUES ('Abraham', 'al44534@un.edu.mx', '44534', 'Administrador');
```



Base de Datos

Nombre de la Base de Datos

Psicometrico1

Tablas Principales

Usuario

Campo	Tipo de Dato	Descripción
Usuariold	INT (PK)	Identificador único, autoincremental
Nombre	NVARCHAR(100)	Nombre completo del usuario
Correo	NVARCHAR(100)	Correo electrónico
Matricula	NVARCHAR(50)	Matrícula del estudiante (opcional)
Rol	NVARCHAR(20)	Rol del usuario

Pregunta

Campo	Tipo de Dato	Descripción
Preguntald	INT (PK)	Identificador único de la pregunta
Texto	NVARCHAR(500)	Enunciado de la pregunta
Activa	BIT	Indica si la pregunta está activa (1 = sí)



Base de Datos

Tablas Principales

RespuestaExamen		
Campo	Tipo de Dato	Descripción
Respuestald	INT (PK)	Identificador único de la respuesta
Usuariold	INT (FK)	Usuario que respondió
Preguntald	INT (FK)	Pregunta respondida
Valor	INT	Valor de la respuesta
FechaRegistro	DATETIME	Fecha en la que se registró la respuesta
ValorRespuesta	INT (nullable)	Campo adicional para almacenar otros valores si es necesario
FechaRespuesta	DATETIME	Fecha precisa en que el usuario respondió

Relaciones

Usuariold con Usuario(Usuariold)
Preguntald con Pregunta(Preguntald)



Base de Datos

Tablas Principales

ResultadoExamen		
Campo	Tipo de Dato	Descripción
Resultadold	INT (PK)	Identificador único del resultado
Usuariold	INT (FK)	Usuario al que le pertenece el resultado
PuntajeTotal	INT	Suma total de los valores del sistema
Diagnostico	NVARCHAR(100)	Diagnóstico basado en el puntaje total
Fecha	DATETIME	Fecha de generación del resultado

Relaciones

Usuariold con Usuario(Usuariold))



Instalación del Programa

ANTES DE INSTALAR EL PROGRAMA, DEBES
ASEGURARTE DE CONTAR CON LOS
REQUISITOS MÍNIMOS

Para conocer los requisitos mínimos, favor de
referir a la página ##

Objetivo del Programa

El objetivo principal del sistema es automatizar la aplicación, registro y evaluación de exámenes psicométricos en un entorno institucional, permitiendo a los estudiantes presentar sus pruebas en línea, y a los docentes y administradores gestionar eficazmente tanto el contenido del examen como los resultados obtenidos.

Este portal web, desarrollado bajo la arquitectura ASP.NET Core MVC, busca ofrecer una herramienta confiable, segura y accesible para:

- Evaluar el nivel de ansiedad de los estudiantes mediante un cuestionario estructurado
- Generar resultados automáticos con interpretación basada en rangos establecidos
- Facilitar la administración de preguntas, usuarios y reportes por parte del personal académico
- Garantizar la integridad de la información mediante roles diferenciados y autenticación con Google

En resumen, el programa tiene como finalidad mejorar el proceso de evaluación psicológica, agilizar el manejo de datos psicométricos y ofrecer una plataforma tecnológica adecuada para la toma de decisiones educativas.

Sistema de Autorización y Autenticación

Autenticación (Login con Google)

El sistema implementa un mecanismo de autenticación externa mediante Google, utilizando el protocolo OAuth 2.0. Este permite que los usuarios inicien sesión de forma segura con su cuenta de Google, sin necesidad de crear contraseñas locales.

Funcionamiento

1. El usuario da clic en “Iniciar sesión con Google”
2. Se redirige al portal de Google para autenticar al usuario
3. Si el inicio de sesión es exitoso, Google devuelve al sistema los siguientes datos:
 1. Nombre
 2. Correo electrónico
 3. Token de acceso
4. El sistema busca ese correo en la tabla USUARIO
 1. Si existe, se permite el acceso y se guarda el Usuarioid, Nombre, Rol y Correo en la sesión
 2. Si no existe, se redirige al usuario a una vista llamada NoAutorizado.cshtml

Autorización basada en roles

- Una vez que un usuario ha iniciado sesión correctamente con su cuenta de Google, el sistema determina qué nivel de acceso debe tener dentro de la plataforma. Esto se hace mediante un esquema de roles. Cada usuario pertenece a uno de los siguientes tres roles:
 - Administrador
 - Docente
 - Estudiante
- El rol define qué puede hacer y qué secciones del sistema puede ver. Por ejemplo:
 - Solo los administradores pueden acceder a la sección para registrar usuarios y gestionar las preguntas del examen psicométrico.
 - Los docentes pueden visualizar listas de estudiantes y los resultados obtenidos por ellos.
 - Los estudiantes solo pueden presentar el examen asignado y ver sus resultados personales.
- El sistema verifica constantemente el rol del usuario para permitir o negar el acceso a ciertas partes del sistema. Esta validación se realiza automáticamente en segundo plano cada vez que el usuario intenta abrir una sección protegida.

Gestión de sesión de usuario

- Después de que el usuario inicia sesión correctamente, el sistema guarda su información básica (como su identificador, rol y nombre) de forma temporal durante toda la sesión. Esta información no es visible para el usuario, pero permite que el sistema:
 - Sepa quién está navegando actualmente
 - Asocie correctamente sus respuestas al examen
 - Muestre solo las opciones que le corresponden según su rol
- Por ejemplo, si un estudiante inicia sesión, el sistema le mostrará únicamente el botón para realizar el examen, mientras que si un administrador ingresa, verá las opciones de gestión y administración del sistema.
- Esta sesión se mantiene activa hasta que el usuario cierre el navegador o se cierre la sesión manualmente.



Vista para usuarios no autorizados

Si una persona intenta ingresar al sistema con su cuenta de Google, pero esa cuenta no ha sido registrada previamente por el administrador, el sistema la detecta como “no autorizada”.

En ese caso, en lugar de permitirle entrar, el sistema la redirige automáticamente a una página especial que le informa que su acceso ha sido denegado por no estar registrada. Esta medida evita que personas ajenas puedan usar el sistema aunque tengan una cuenta de Google válida.

Esta vista de “no autorizado” es una capa adicional de seguridad que garantiza que solo los usuarios autorizados por la institución puedan interactuar con el sistema.

Tipos de Pruebas Realizadas

1. Ejecución prolongada del sistema
 - El sistema fue ejecutado de forma continua en un solo equipo de desarrollo, permitiendo simular un uso extendido por parte de un usuario. Durante estas sesiones se exploraron todas las rutas disponibles (registro, inicio de sesión, examen, resultados, gestión de preguntas, etc.) de manera repetitiva.
 - Resultado: Se identificaron y corrigieron errores menores en validaciones, visualización de datos y redirecciones incorrectas.
2. Pruebas de navegación rápida entre vistas
 - Se realizaron pruebas navegando rápidamente entre secciones del sistema (por ejemplo, pasando del examen a la página de inicio y de vuelta al examen), lo cual permitió detectar posibles fallos en el control de sesión y rutas protegidas por rol.
 - Resultado: Se reforzaron mecanismos de verificación de sesión y autorización para evitar accesos no permitidos.
3. Ingreso y registro de datos múltiples
 - Se probó el formulario de registro de usuarios con distintos tipos de datos y entradas masivas, lo que permitió validar el manejo de excepciones, los límites del modelo y la persistencia de datos en la base de datos.
 - Resultado: Se ajustaron controles de campos obligatorios, validaciones de rol, y mensajes de retroalimentación al usuario.
4. Simulación de comportamiento del examen
 - Se probó exhaustivamente el módulo de examen psicométrico, incluyendo el temporizador de 20 segundos por pregunta, navegación pregunta a pregunta y el cálculo final de resultados.
 - Resultado: Se optimizó el control del flujo de preguntas y el registro preciso de cada respuesta con su puntaje correspondiente.