SISTEMA AKDMIA

Documentación Técnica - Pantalla de Login

Versión:	1.0
Fecha:	22/11/2025
Sistema:	AKDMIA - Sistema de Gestión Académica
Módulo:	Autenticación

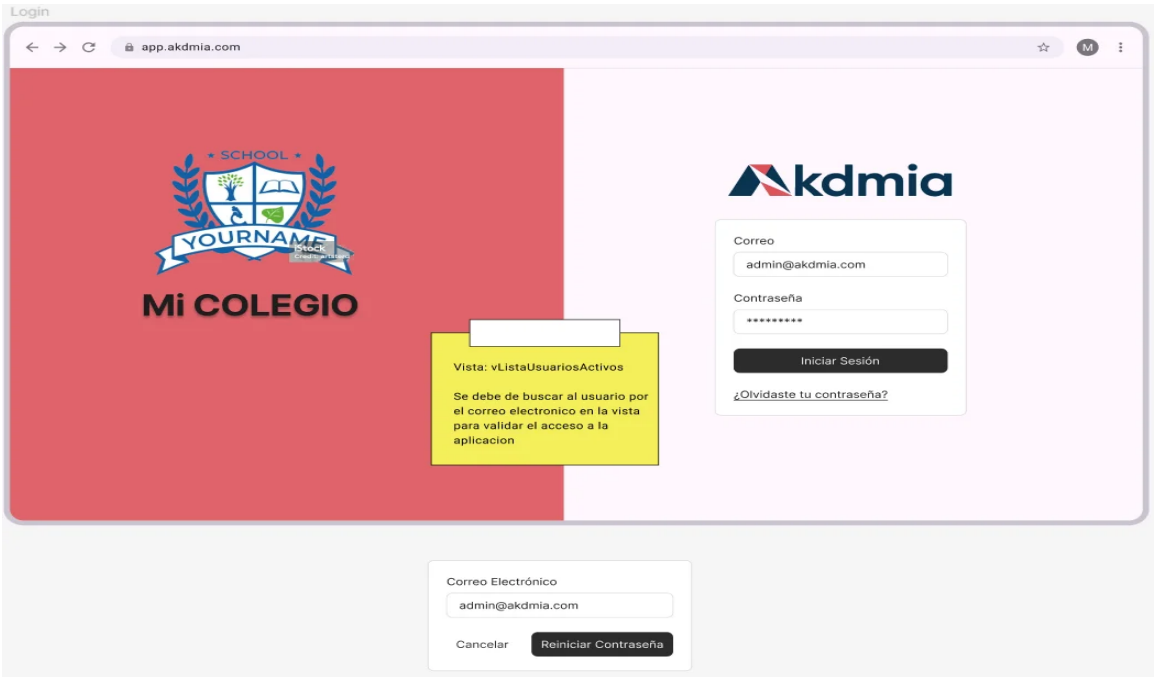

Figura 1: Pantalla de Login del Sistema AKDMIA

Tabla de Contenido

1. Descripción General

Esta es la **pantalla de inicio de la aplicación AKDMIA**. Es el punto de entrada al sistema donde los usuarios deben autenticarse para acceder a las funcionalidades de la plataforma. La pantalla presenta el logo del colegio, el nombre de la institución y un formulario de inicio de sesión.

2. Elementos Visuales de la Interfaz

2.1 Lado Izquierdo (Información del Colegio)

• **Logo del Colegio:** Se muestra el logo de la institución educativa

• **Nombre del Colegio:** Debe mostrarse el nombre real del colegio obtenido de la base de datos

2.2 Lado Derecho (Formulario de Login)

• **Logo de AKDMIA:** Logo de la aplicación

• **Campo 'Correo':** Input tipo email para el correo electrónico del usuario

• **Campo 'Contraseña':** Input tipo password (se muestra con asteriscos)

• **Botón 'Iniciar Sesión':** Botón principal para autenticarse

• **Enlace '¿Olvidaste tu contraseña?':** Link para recuperación de contraseña

3. Flujo de Autenticación

3.1 Proceso de Login

1. El usuario ingresa su correo electrónico en el campo correspondiente
2. El usuario ingresa su contraseña en el campo correspondiente
3. El usuario hace clic en el botón 'Iniciar Sesión'
4. El sistema valida las credenciales contra la base de datos
5. Si las credenciales son correctas, redirige a la página de módulos del sistema
6. Si las credenciales son incorrectas, muestra un mensaje de error

3.2 Vista Utilizada para Validación

Vista: `AKDMIA_MASTER.dbo.vListaUsuariosActivos`

Esta vista contiene la información de todos los usuarios activos del sistema. Los campos más relevantes son:

Campo	Tipo	Descripción
idUsuario	int	Identificador único del usuario
sUsuario	nvarchar(50)	Nombre de usuario único para login
sCorreoElectronico	nvarchar(100)	Correo electrónico del usuario
sContrasena	nvarchar(500)	Contraseña encriptada (hash)
guidColegio	uniqueidentifier	GUID del colegio al que pertenece
bActivo	bit	Indica si el usuario está activo (debe ser 1)
bUsuarioConfirmado	bit	Indica si el usuario ha confirmado su cuenta
bBloqueado	bit	Indica si la cuenta está bloqueada
iIntentosLogin	tinyint	Número de intentos fallidos consecutivos
bCambiarContrasena	bit	Si debe cambiar contraseña en próximo login
sServidor	nvarchar(255)	Servidor donde reside la BD del colegio
sBaseDatos	nvarchar(128)	Nombre de la base de datos del colegio

4. Validaciones a Implementar

4.1 Validaciones de Campos

Campo Correo Electrónico:

• No puede estar vacío

• Debe tener formato válido de email (contener @ y dominio)

• Máximo 100 caracteres

Campo Contraseña:

• No puede estar vacío

• Máximo 500 caracteres

4.2 Validaciones de Autenticación

El sistema debe validar lo siguiente **EN ESTE ORDEN**:

1. Verificar que el usuario existe

Buscar en vListaUsuariosActivos filtrando por sCorreoElectronico = [correo ingresado] y guidColegio = [GUID del colegio]

2. Verificar que la cuenta está activa

bActivo = 1. Si no está activo: ERROR - 'Cuenta desactivada. Contacte al administrador'

3. Verificar que la cuenta no está bloqueada

bBloqueado = 0. Si está bloqueado: ERROR - 'Cuenta bloqueada por múltiples intentos fallidos. Contacte al administrador'

4. Verificar que el usuario ha confirmado su cuenta

bUsuarioConfirmado = 1. Si no está confirmado: ERROR - 'Cuenta no confirmada. Revise su correo electrónico'

5. Validar contraseña

Comparar el hash de la contraseña ingresada con el campo sContrasena. Si no coincide: Incrementar contador de intentos fallidos

4.3 Control de Intentos Fallidos

Cuando la contraseña es incorrecta:

1. Incrementar el campo `iIntentosLogin` en 1

2. Obtener el servidor y base de datos del colegio desde `vListaUsuariosActivos` (campos sServidor y sBaseDatos) y consultar la tabla PoliticasContrasenas en la base de datos del cliente para

obtener `iIntentosMaxLogin`

3. Si `iIntentosLogin >= iIntentosMaxLogin`:
    - Establecer `bBloqueado = 1`
    - Establecer `dFechaBloqueo = GETDATE()`
    - ERROR - 'Cuenta bloqueada por exceder el número máximo de intentos fallidos'
4. Si aún no alcanza el máximo:
    - ERROR - 'Correo o contraseña incorrectos. Intentos restantes: [X]'

4.4 Login Exitoso

Cuando el login es exitoso:

1. Actualizar `dUltimoLogin = GETDATE()`
2. Resetear `iIntentosLogin = 0`
3. Verificar si `bCambiarContrasena = 1`:
    - Si es 1: Redirigir a pantalla de cambio de contraseña obligatorio
    - Si es 0: Continuar al siguiente paso
4. Guardar información de sesión:
    - idUsuario, guidColegio, sServidor, sBaseDatos
    - Tokens/cookies según el framework utilizado
5. Redirigir a la **Página de Módulos del Sistema**

5. Recuperación de Contraseña

5.1 Flujo de '¿Olvidaste tu contraseña?'

Cuando el usuario hace clic en el enlace '¿Olvidaste tu contraseña?':

1. Se abre un modal/pantalla solicitando el correo electrónico
2. El usuario ingresa su correo electrónico
3. El sistema valida que el correo existe en vListaUsuariosActivos con el guidColegio correspondiente

5.2 Proceso de Recuperación

Si el correo existe:

1. Generar un token único de recuperación (GUID o cadena aleatoria segura)
2. Actualizar en tabla AKDMIA_MASTER.dbo.Usuarios:
    - sTokenRecuperacion = [token generado]
    - dExpiracionToken = DATEADD(hour, 1, GETDATE()) // token válido por 1 hora
3. Enviar email al usuario con un enlace que contenga el token
    - Ejemplo: https://app.akdmia.com/reset-password?token=[token]
4. Mostrar mensaje: 'Se ha enviado un enlace de recuperación a tu correo electrónico'

Si el correo NO existe:

Por seguridad, mostrar el mismo mensaje que cuando existe para no revelar qué correos están registrados.

Mensaje: 'Si el correo existe en nuestro sistema, recibirás instrucciones para recuperar tu contraseña'

5.3 Pantalla de Restablecimiento de Contraseña

Cuando el usuario accede al enlace del email:

1. Validar que el token existe y no ha expirado:
    - sTokenRecuperacion = [token de la URL]
    - dExpiracionToken > GETDATE()
2. Si es válido: Mostrar formulario para establecer nueva contraseña
3. Si no es válido: Mostrar error 'Token inválido o expirado'

Validaciones de Nueva Contraseña:

Obtener el servidor y base de datos del colegio y consultar la tabla PoliticasContrasenas en la base de datos del cliente para obtener los requisitos de contraseña:

Campo	Validación
iLongitudMinima	Longitud mínima de caracteres
bMayuscula	Debe contener al menos una letra mayúscula
bMinuscula	Debe contener al menos una letra minúscula
bNumeros	Debe contener al menos un número
bCaracterEsp	Debe contener al menos un carácter especial (!@#$%^&*)
iHistorialContrasenas	Número de contraseñas anteriores que no se pueden reutilizar

Validación de Historial:

1. Antes de guardar la nueva contraseña, consultar la tabla UsuariosHistCont en la base de datos del cliente
2. Obtener las últimas iHistorialContrasenas contraseñas del usuario
3. Comparar el hash de la nueva contraseña con el historial
4. Si coincide con alguna: ERROR - 'No puedes reutilizar una de tus últimas [X] contraseñas'

Al Guardar Nueva Contraseña:

1. Encriptar la nueva contraseña (hash)
2. Actualizar en AKDMIA_MASTER.dbo.Usuarios:
    - sContrasena = [nuevo hash]
    - dFechaUltCambioCont = GETDATE()
    - sTokenRecuperacion = NULL
    - dExpiracionToken = NULL
    - bCambiarContrasena = 0
    - bBloqueado = 0 (desbloquear si estaba bloqueado)
    - iIntentosLogin = 0 (resetear intentos)
3. Mostrar mensaje: 'Contraseña actualizada exitosamente'
4. Redirigir a pantalla de login

Nota: El registro de la contraseña anterior en el historial (tabla UsuariosHistCont) y la limpieza de contraseñas antiguas que excedan iHistorialContrasenas se realiza automáticamente mediante un trigger de base de datos al actualizar el campo sContrasena. No es necesario realizar estas operaciones manualmente.

6. Identificación del Colegio (guidColegio)

6.1 ¿Cómo se determina el guidColegio?

El guidColegio se determina según el dominio o subdominio desde el cual se accede a la aplicación. Existen varias estrategias:

Opción 1: Por Subdominio

- `URL: https://[nombre-colegio].akdmia.com`

• Extraer el subdominio y buscar en AKDMIA_MASTER.dbo.ColegiosDB por sServidor o una tabla de configuración que mapee subdominios a GUIDs

Opción 2: Por Dominio Personalizado

- `URL: https://colegio.edu.do (dominio propio del colegio)`

• Tener una tabla de mapeo de dominios a GUIDs en AKDMIA_MASTER

Opción 3: Por Parámetro en URL

- `URL: https://app.akdmia.com?colegio=[guid]`

• No recomendado por seguridad, pero viable para desarrollo

6.2 Obtención de Datos del Colegio

Una vez identificado el guidColegio, obtener información del colegio desde AKDMIA_MASTER.dbo.ColegiosDB:

```
SELECT c.idColegiosDB, c.guidColegio, c.sServidor, c.sBaseDatos
FROM AKDMIA_MASTER.dbo.ColegiosDB c
WHERE c.guidColegio = '[GUID identificado]'
```

Esta información se utiliza para:

• Conectarse a la base de datos específica del colegio

• Mostrar el logo y nombre del colegio en la pantalla

• Filtrar usuarios en el login

Nota Importante: El logo y nombre del colegio se obtienen de la tabla Colegios en la base de datos del cliente específico usando el sServidor y sBaseDatos obtenidos de ColegiosDB:

```
SELECT sColegio, sColegioLogo
FROM [sServidor].[sBaseDatos].dbo.Colegios
WHERE guidColegio = '[GUID identificado]'
```

7. Mensajes de Error

Listado completo de mensajes de error a mostrar según cada situación:

Situación	Mensaje
Correo vacío	Por favor ingrese su correo electrónico
Correo inválido	Por favor ingrese un correo electrónico válido
Contraseña vacía	Por favor ingrese su contraseña
Usuario no existe	Correo o contraseña incorrectos
Cuenta inactiva	Cuenta desactivada. Contacte al administrador
Cuenta bloqueada	Cuenta bloqueada. Contacte al administrador
Cuenta no confirmada	Cuenta no confirmada. Revise su correo electrónico
Contraseña incorrecta	Correo o contraseña incorrectos. Intentos restantes: [X]
Máximo intentos	Cuenta bloqueada por exceder intentos permitidos
Error de conexión	Error al conectar con el servidor. Intente nuevamente
Error general	Ha ocurrido un error. Por favor intente nuevamente

8. Consideraciones de Seguridad

8.1 Encriptación de Contraseñas

• Las contraseñas se almacenan encriptadas en formato hash (nunca en texto plano)
• Utilizar algoritmo seguro: bcrypt, PBKDF2 o Argon2
• El hash se compara con la contraseña ingresada encriptada

8.2 Protección contra Ataques

• **Bloqueo por intentos fallidos:** Implementado mediante iIntentosLogin e iIntentosMaxLogin
• **Mensajes genéricos:** No revelar si el correo existe (decir 'correo o contraseña incorrectos')
• **Rate limiting:** Limitar número de intentos por IP en un período de tiempo
• **CAPTCHA:** Considerar implementar después de X intentos fallidos

8.3 Sesiones

• Generar token de sesión seguro tras login exitoso

• Establecer tiempo de expiración de sesión (ej: 30 minutos de inactividad)
• Implementar renovación de token antes de expiración

8.4 HTTPS

• La aplicación DEBE usar HTTPS (visible en la barra de direcciones: candado)
• Nunca transmitir credenciales por HTTP sin encriptar

9. Redirección Post-Login

9.1 Destino según Estado del Usuario

Condición	Destino
bCambiarContrasena = 1	Pantalla de cambio de contraseña obligatorio
Contraseña expirada (según iMesesExpira)	Pantalla de cambio de contraseña
Login exitoso normal	Página de Módulos del Sistema

9.2 Verificación de Expiración de Contraseña

Al hacer login exitoso, verificar si la contraseña ha expirado consultando:

• La fecha del último cambio de contraseña (dFechaUltCambioCont) del usuario

• El campo iMesesExpira de la tabla PoliticasContrasenas en la base de datos del cliente

• Si DATEADD(MONTH, iMesesExpira, dFechaUltCambioCont) < GETDATE(): Forzar cambio de contraseña

10. Notas para el Desarrollador

10.1 Archivos/Componentes a Crear

• **Página de Login:** Interfaz principal

• **Componente de Formulario:** Campos de correo y contraseña

• **Modal de Recuperación:** Para solicitar correo

• **Página de Restablecimiento:** Para nueva contraseña

• **Servicio de Autenticación:** Lógica de validación

• **Servicio de Email:** Para envío de enlaces de recuperación

10.2 Estados a Manejar en el Frontend

• Estado de loading durante autenticación

• Mensajes de error

• Validaciones de formulario en tiempo real

• Contador de intentos restantes

• Estado de modal de recuperación de contraseña

10.3 Información a Almacenar en Sesión

• idUsuario

• guidColegio

• sServidor (servidor de la base de datos del cliente)

• sBaseDatos (nombre de la base de datos del cliente)

• Nombre y apellido del usuario

• Token de sesión

• Permisos/roles del usuario (para uso posterior)

10.4 Flujo de Conexión a Base de Datos del Cliente

IMPORTANTE: El sistema utiliza una arquitectura multi-tenant donde cada colegio tiene su propia base de datos. El proceso es:

1. Al iniciar la aplicación, identificar el guidColegio según el dominio/subdominio de acceso

2. Consultar AKDMIA_MASTER.dbo.ColegiosDB para obtener sServidor y sBaseDatos del colegio

3. Establecer una conexión dinámica a la base de datos del cliente usando estos parámetros

4. Todas las consultas que requieran datos específicos del colegio (PoliticasContrasenas, UsuariosHistCont, Colegios, etc.) se ejecutan contra [sServidor].[sBaseDatos].dbo.[Tabla]

5. Las consultas generales de usuarios y autenticación se ejecutan contra AKDMIA_MASTER

Fin del documento