FUNDACIÓN ESCUELA TECNOLÓGICA DE NEIVA
JESÚS OVIEDO PÉREZ

Sistema De Control De Acceso Universitario: Fet Access

Alex Orlando Triana Hernández, Jeisson Javier Cortes Lozano

Universidad Fundación Escuela Tecnológica Jesús Oviedo Pérez

Tecnología En Desarrollo De Sistemas De Información Y Redes

Rivera, Huila

14 de noviembre 2024

FUNDACIÓN ESCUELA TECNOLÓGICA DE NEIVA
JESÚS OVIEDO PÉREZ

TABLA DE CONTENIDO

Fet Access

1. Propuesta del proyecto

La propuesta del proyecto **FET Access** se centra en el desarrollo de una plataforma digital que mejore el control de acceso y la gestión de la información personal dentro de la universidad, este sistema permitirá a estudiantes, guardias de seguridad y administradores ingresar al campus de forma segura y rápida a través de un código QR único para cada usuario, evitando el ingreso no autorizado. La aplicación también ofrecerá un espacio donde cada usuario podrá acceder a sus datos personales para poder cambiar su contraseña, el número telefónico y la foto de perfil.

El proyecto busca no solo reforzar la seguridad en el acceso al campus, sino también simplificar y optimizar el proceso de verificación, reduciendo los tiempos de espera y mejorando la experiencia de los usuarios y guardias de seguridad.

2. Objetivos

2.1 Objetivo General

Desarrollar una aplicación móvil que permita gestionar el control de acceso al campus universitario mediante el uso de códigos QR, mejorando la seguridad y eficiencia del sistema de verificación de ingreso.

2.2 Objetivos específicos

- Diseñar e implementar una base de datos segura en MySQL y un backend robusto utilizando Laravel, que gestione las funcionalidades del sistema, incluyendo la generación y validación de códigos QR.
- Implementar las funcionalidades clave del sistema, priorizando la generación y validación de códigos QR, junto con la gestión de usuarios y visitantes.
- Diseñar una interfaz intuitiva con Ionic, además de configurar y desplegar la aplicación en un entorno de producción seguro para su uso efectivo.

3. Fases del proyecto

3.1 Planificación inicial

Recopilación de requisitos, análisis de viabilidad y definición de objetivos a corto y largo plazo.

3.2 Diseño

Creación de arquitectura de sistemas, diseño de bases de datos, creación de prototipos de interfaces de usuario y diagramas de flujo.

3.3 Desarrollo

Implementación del backend, frontend, integración de bases de datos y desarrollo de funcionalidades principales.

3.4 Pruebas

Realización de pruebas unitarias, de integración, de usuario y de seguridad para garantizar la funcionalidad y confiabilidad del sistema.

3.5 Despliegue

Configuración y despliegue de la aplicación en un entorno de producción seguro.

3.6 Mantenimiento y mejoras

Seguimiento, soporte técnico y actualización continua de la aplicación según las necesidades de la universidad y el feedback de los usuarios.

4. Metodología

4.1 Fase 1: Implementación de la base de datos y backend

Actividades principales:

- Diseñar y crear la estructura de la base de datos en MySQL.
- Configurar el backend utilizando Laravel para manejar la lógica del sistema y la integración con la base de datos.
- Implementar medidas de seguridad en la base de datos y las APIs del backend (cifrado, validación de datos y protección contra ataques).

4.2 Fase 2: Desarrollo de las funcionalidades principales

Actividades principales:

- Crear las funcionalidades centrales del sistema, como la generación y validación de códigos QR.
- Implementar módulos para la gestión de usuarios, visitantes, vigilantes y administrador, incluyendo sus funcionalidades principales.
- Garantizar que las funcionalidades se integren correctamente entre el backend y la base de datos.

4.3 Fase 3: Creación del frontend y despliegue del sistema

Actividades principales:

- Diseñar y desarrollar una interfaz de usuario amigable y accesible con Ionic

- Conectar el frontend con el backend para garantizar una experiencia fluida y eficiente.
- Configurar el entorno de producción y desplegar la aplicación, asegurando su estabilidad y seguridad.

5. Diagramas

5.1 Historia de usuarios

Captura las necesidades de cada tipo de usuario (estudiantes, vigilantes, administradores), se desarrollarán historias de usuarios. Estas historias definirán cómo interactúa cada usuario con la aplicación y qué funcionalidad requiere. Ejemplos de historias de usuarios incluyen:

Iniciar sesión en la aplicación

Como: Usuario registrado
Quiero: Iniciar sesión en la aplicación
Para: Acceder a mis funcionalidades personalizadas y autorizadas.

Criterios de aceptación

- Debo ingresar un correo electrónico y contraseña válidos para iniciar sesión.
- Si las credenciales son correctas, recibiré un token de autenticación.
- Si las credenciales son incorrectas, se mostrará un mensaje de error indicando "Credenciales inválidas".
- No debería poder acceder a rutas protegidas sin haber iniciado sesión.

Cerrar sesión de la aplicación

Como: Usuario autenticado
Quiero: Cerrar sesión de mi cuenta
Para: Asegurar que nadie más use mi sesión abierta en el dispositivo.

Criterios de aceptación

- Al solicitar el cierre de sesión, mi token debe ser invalidado.
- Debería recibir una confirmación de que la sesión ha sido cerrada exitosamente.
- Después del cierre de sesión, no debería poder acceder a rutas protegidas.
- Si no estoy autenticado y solicito esta acción, debería recibir un mensaje de error "No autenticado".

Registrar visitantes al ingresar al campus

Como: Visitante
Quiero: Registrar mi ingreso desde un formulario público
Para: Hacer el proceso accesible desde cualquier dispositivo sin necesidad de iniciar sesión.

Criterios de aceptación:

- El formulario debe estar disponible públicamente para cualquier visitante.
- Debo poder ingresar mi información y confirmar el registro.
- Si el registro es exitoso, debería recibir una confirmación visual o por correo (si aplica).
- Si los datos son incompletos, debería recibir un mensaje de error indicando qué campos corregir.

Consultar la lista de usuarios

Como: Administrador
Quiero: Consultar la lista de todos los usuarios registrados
Para: Gestionar la información y permisos de los usuarios en el sistema.

Criterios de aceptación:

- Debería poder acceder a una lista con la información de todos los usuarios registrados.
- La lista debe incluir datos relevantes como nombre, correo electrónico y rol.
- Si no hay usuarios registrados, debería mostrarse un mensaje indicando "No hay usuarios registrados".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Consultar los detalles de un usuario

Como: Administrador
Quiero: Ver los detalles de un usuario específico
Para: Gestionar la información y actividades de ese usuario.

Criterios de aceptación:

- Al acceder a los detalles de un usuario existente, debería poder visualizar su información completa, como nombre, correo y rol.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Actualizar información de un usuario

Como: Administrador
Quiero: Actualizar la información de un usuario específico
Para: Corregir datos o cambiar permisos cuando sea necesario.

Criterios de aceptación:

- Debería poder modificar datos de un usuario existente como el nombre, correo electrónico, teléfono o rol del usuario.
- Si los datos proporcionados son inválidos o incompletos, debería recibir un mensaje de error indicando qué corregir.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Eliminar un usuario

Como: Administrador
Quiero: Eliminar un usuario específico del sistema
Para: Gestionar usuarios inactivos o no autorizados.

Criterios de aceptación

- Debería poder eliminar a un usuario proporcionando su ID.
- Si el usuario no existe, debería recibir un mensaje indicando "Usuario no encontrado".
- Si la eliminación es exitosa, debería recibir una confirmación de "Usuario eliminado exitosamente".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Registrar un nuevo usuario

Como: Administrador
Quiero: Registrar nuevos usuarios en el sistema
Para: Proveer acceso a nuevos miembros con roles específicos.

Criterios de aceptación

- Debería poder registrar un nuevo usuario proporcionando información como nombre, correo electrónico, código, teléfono y rol.
- Si los datos proporcionados son inválidos o incompletos, debería recibir un mensaje de error indicando qué corregir.
- Si el registro es exitoso, debería recibir una confirmación con los datos del nuevo usuario registrado.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Consultar la lista de visitantes

Como: Administrador
Quiero: Consultar la lista de todos los visitantes registrados
Para: Gestionar y supervisar las visitas al campus.

Criterios de aceptación

- Debería poder acceder a una lista con información de todos los visitantes registrados, incluyendo nombre, motivo de la visita y estado.
- Si no hay visitantes registrados, debería mostrarse un mensaje indicando "No hay visitantes registrados".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Consultar los detalles de un visitante

Como: Administrador
Quiero: Ver los detalles de un visitante específico
Para: Supervisar la información de la visita al campus.

Criterios de aceptación:

- Al acceder a los detalles de un visitante existente, debería poder visualizar información como nombre, motivo de la visita, fecha y hora de acceso.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Eliminar un visitante

Como: Administrador
Quiero: Eliminar un visitante específico del sistema
Para: Gestionar visitantes que ya no requieren estar en el sistema.

Criterios de aceptación:

- Debería poder eliminar a un visitante existente.
- Si la eliminación es exitosa, debería recibir una confirmación de "Visitante eliminado exitosamente".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Consultar los registros de acceso

Como: Administrador
Quiero: Consultar los registros de acceso al campus
Para: Supervisar quiénes han ingresado y a qué hora.

Criterios de aceptación:

- Debería poder acceder a una lista con los registros de acceso, incluyendo información como nombre del usuario o visitante, hora de ingreso y salida.
- Si no hay registros de acceso, debería mostrarse un mensaje indicando "No hay registros de acceso disponibles".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Consultar los detalles de un registro de acceso

Como: Administrador
Quiero: Ver los detalles de un registro de acceso específico
Para: Analizar la actividad de ingreso de un usuario o visitante.

Criterios de aceptación:

- Al acceder a los detalles de un registro existente, debería visualizar información como el nombre del usuario/visitante, fecha, hora de entrada y salida.
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Eliminar un registro de acceso

Como: Administrador
Quiero: Eliminar un registro de acceso específico
Para: Depurar o corregir información incorrecta en los registros.

Criterios de aceptación:

- Debería poder eliminar un registro de acceso existente.
- Si la eliminación es exitosa, debería recibir una confirmación de "Registro de acceso eliminado exitosamente".
- Esta funcionalidad debe estar restringida solo para el rol de administrador.

Generar código QR

Como: Estudiante
Quiero: Generar un código QR personalizado
Para: Identificarme fácilmente al ingresar al campus

Criterios de aceptación:

- Debería poder generar un código QR que incluya mi información básica, como nombre, tipo de vehículo, matricula, correo.
- El código QR debería ser único y asociado a mi perfil de estudiante.
- Si el QR no puede generarse, debería recibir un mensaje de error claro.
- Esta funcionalidad debe estar restringida solo para estudiantes registrados.

Consultar mi perfil

Como: Estudiante
Quiero: Consultar la información de mi perfil
Para: Revisar y asegurarme de que mis datos estén actualizados.

Criterios de aceptación:

- Al acceder a mi perfil, debería poder visualizar información como mi nombre, correo electrónico, teléfono y código de estudiante.
- Si mi perfil no está completo o hay errores en la información, debería recibir una notificación para corregirlos.
- Esta funcionalidad debe estar restringida solo para estudiantes autenticados.

Actualizar mi perfil

Como: Estudiante
Quiero: Actualizar mi información personal
Para: Corregir o completar datos como mi número de teléfono y foto de perfil

Criterios de aceptación:

- Debería poder modificar información específica, como teléfono y foto de perfil.
- Si los datos proporcionados son inválidos o incompletos, debería recibir un mensaje de error indicando qué corregir.
- Después de la actualización, debería recibir una confirmación de que los datos han sido guardados correctamente.
- Esta funcionalidad debe estar restringida solo para estudiantes autenticados.

Actualizar mi contraseña

Como: Estudiante
Quiero: Cambiar mi contraseña de acceso
Para: Mantener segura mi cuenta y proteger mi información personal.

Criterios de aceptación:

- Debería poder proporcionar mi contraseña actual, junto con la nueva contraseña.
- Si la contraseña actual es incorrecta, debería recibir un mensaje indicando "Contraseña actual incorrecta".
- Si la nueva contraseña no cumple con los requisitos de seguridad, debería recibir un mensaje claro para corregirla.
- Después de un cambio exitoso, debería recibir una confirmación de que la contraseña ha sido actualizada.
- Esta funcionalidad debe estar restringida solo para estudiantes autenticados.

Validar códigos QR

Como: Vigilante
Quiero: Validar los códigos QR presentados por estudiantes
Para: Autorizar o denegar el acceso al campus de manera rápida y segura.

Criterios de aceptación:

- Debería poder escanear un código QR y verificar su validez.
- Si el código QR es válido, debería recibir un mensaje indicando "Acceso autorizado", junto con información relevante del usuario, como nombre, tipo vehículo, matrícula y código estudiante.
- Si el código QR es inválido, debería recibir un mensaje indicando "Acceso denegado", con la razón correspondiente (por ejemplo, "Código expirado" o "No registrado").
- Esta funcionalidad debe estar restringida solo para vigilantes autenticados.

5.1.1 Tabla de Priorización de Historias de Usuario

ID	Rol	Historia de Usuario	Prioridad	Estado
1	Administrador	Consultar la lista de usuarios	Alta	Completado
2	Administrador	Consultar los detalles de un usuario	Alta	Completado
3	Administrador	Actualizar información de un usuario	Alta	Completado
4	Administrador	Eliminar un usuario	Alta	Completado
5	Administrador	Registrar un nuevo usuario	Alta	Completado
6	Administrador	Consultar la lista de visitantes	Media	Completado
7	Administrador	Consultar los detalles de un visitante	Media	Completado
8	Administrador	Eliminar un visitante	Media	Completado
9	Administrador	Consultar los registros de acceso	Media	Completado
10	Administrador	Consultar los detalles de un registro de acceso	Media	Completado
11	Administrador	Eliminar un registro de acceso	Baja	Completado
12	Estudiante	Generar código QR	Alta	Completado
13	Estudiante	Consultar mi perfil	Alta	Completado
14	Estudiante	Actualizar mi perfil	Alta	Completado
15	Estudiante	Actualizar mi contraseña	Alta	Completado
16	Vigilante	Validar códigos QR	Alta	Completado

5.2 Casos de uso

Los casos de uso definen las principales interacciones entre los usuarios y el sistema. Algunos de los casos de uso identificados incluyen:

5.2.1 Caso de Uso: Iniciar Sesión

Actor Principal: Usuario registrado

Propósito: Permitir que un usuario autenticado acceda al sistema para utilizar las funcionalidades autorizadas según su rol.

Precondiciones:

1. El usuario debe estar registrado en el sistema.

2. Debe proporcionar credenciales válidas (correo electrónico y contraseña).

Flujo Principal:

1. El usuario accede a la pantalla de inicio de sesión.

2. El sistema solicita el correo electrónico y la contraseña.

3. El usuario introduce las credenciales y las envía.

4. El sistema valida las credenciales con la base de datos.

5. Si las credenciales son correctas:

- El sistema genera un token de autenticación.

- El usuario es redirigido a la página principal correspondiente a su rol.

Flujo Alternativo:

- Credenciales incorrectas:

- El sistema muestra un mensaje: "Credenciales inválidas".

Postcondiciones:

1. El usuario obtiene acceso a las funcionalidades personalizadas según su rol.

2. El token de autenticación se guarda de forma segura.

5.2.2 Caso de Uso: Cerrar Sesión

Actor Principal: Usuario autenticado

Propósito: Permitir que el usuario finalice su sesión.

Precondiciones:

1. El usuario debe estar autenticado con un token válido.

Flujo Principal:

1. El usuario selecciona la opción "Cerrar Sesión".

2. El sistema invalida el token de autenticación.

3. El usuario es redirigido a la página de inicio de sesión.

4. El sistema registra el cierre de sesión para auditoría.

Postcondiciones:

1. El usuario no tiene acceso a funcionalidades protegidas.

2. El token de autenticación queda invalidado.

5.2.3 Caso de Uso: Consultar la Lista de Usuarios

Actor Principal: Administrador

Propósito: Permitir al administrador visualizar todos los usuarios registrados en el sistema.

Precondiciones:

El administrador debe estar autenticado con un token válido.

Debe tener permisos de administrador.

Flujo Principal:

El administrador selecciona la opción "Lista de Usuarios".

El sistema verifica que el token es válido y que el usuario tiene permisos de administrador.

El sistema muestra la lista de usuarios con datos relevantes como nombre, correo electrónico y rol.

Flujo Alternativo:

No hay usuarios registrados:

El sistema muestra el mensaje "No hay usuarios registrados".

Postcondiciones:

El administrador puede visualizar la información de los usuarios registrados.

5.2.4 Caso de Uso: Registrar un Nuevo Usuario

Actor Principal: Administrador

Propósito: Permitir al administrador crear un nuevo usuario con un rol asignado.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para registrar usuarios.

Flujo Principal:

El administrador selecciona la opción "Registrar Usuario".

El sistema solicita los datos necesarios (nombre, correo electrónico, contraseña, rol).

El administrador completa el formulario y lo envía.

El sistema valida los datos e inserta el nuevo usuario en la base de datos.

El sistema confirma el registro exitoso.

Flujo Alternativo:

Datos inválidos o incompletos:

El sistema muestra un mensaje indicando los errores.

Postcondiciones:

El nuevo usuario queda registrado en el sistema con el rol asignado.

5.2.5 Caso de Uso: Actualizar Información de un Usuario

Actor Principal: Administrador

Propósito: Permitir al administrador actualizar la información de un usuario existente.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para actualizar usuarios.

Flujo Principal:

El administrador selecciona un usuario de la lista.

El sistema muestra los datos actuales del usuario.

El administrador modifica los campos necesarios (nombre, correo, rol, etc.) y guarda los cambios.

El sistema valida y actualiza la información en la base de datos.

El sistema confirma la actualización exitosa.

Flujo Alternativo:

El usuario no existe:

El sistema muestra "Usuario no encontrado".

Postcondiciones:

La información del usuario queda actualizada en la base de datos.

5.2.6 Caso de Uso: Eliminar un Usuario

Actor Principal: Administrador

Propósito: Permitir al administrador eliminar un usuario del sistema.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para eliminar usuarios.

Flujo Principal:

El administrador selecciona un usuario de la lista.

El sistema solicita confirmación antes de eliminar.

El administrador confirma la acción.

El sistema elimina al usuario de la base de datos.

El sistema confirma la eliminación exitosa.

Flujo Alternativo:

El usuario no existe:

El sistema muestra "Usuario no encontrado".

Postcondiciones:

El usuario queda eliminado del sistema.

5.2.7 Caso de Uso: Consultar la Lista de Visitantes

Actor Principal: Administrador

Propósito: Permitir al administrador visualizar la lista completa de visitantes registrados en el sistema.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos de administrador.

Flujo Principal:

El administrador selecciona la opción "Visitantes".

El sistema verifica que el administrador está autenticado y autorizado.

El sistema muestra la lista de visitantes con información relevante como nombre, motivo de visita, fecha y hora de entrada.

Flujo Alternativo:

No hay visitantes registrados:

El sistema muestra el mensaje "No hay visitantes registrados".

Postcondiciones:

El administrador puede visualizar la lista de todos los visitantes registrados.

5.2.8 Caso de Uso: Consultar los Detalles de un Visitante

Actor Principal: Administrador

Propósito: Permitir al administrador consultar la información detallada de un visitante específico.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para consultar visitantes.

Flujo Principal:

El administrador selecciona un visitante de la lista.

El sistema verifica que el visitante exista

El sistema muestra los detalles del visitante, incluyendo nombre, identificación, motivo de visita y horarios registrados.

Flujo Alternativo:

El visitante no existe:

El sistema muestra "Visitante no encontrado".

Postcondiciones:

El administrador visualiza la información detallada del visitante.

5.2.9 Caso de Uso: Eliminar un Visitante

Actor Principal: Administrador

Propósito: Permitir al administrador eliminar a un visitante del sistema.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para eliminar visitantes.

Flujo Principal:

El administrador selecciona un visitante de la lista.

El sistema solicita confirmación para eliminar al visitante.

El administrador confirma la acción.

El sistema elimina al visitante de la base de datos.

El sistema confirma que la eliminación fue exitosa.

Flujo Alternativo:

El visitante no existe:

El sistema muestra "Visitante no encontrado".

Postcondiciones:

El visitante queda eliminado del sistema.

5.2.10 Caso de Uso: Consultar los Registros de Acceso

Actor Principal: Administrador

Propósito: Permitir al administrador consultar la lista de registros de acceso al campus.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para consultar registros de acceso.

Flujo Principal:

El administrador selecciona la opción "Registros de Acceso".

El sistema verifica que el administrador está autenticado.

El sistema muestra la lista de registros, incluyendo nombre, rol (usuario o visitante), fecha, hora de entrada y salida.

Flujo Alternativo:

No hay registros de acceso:

El sistema muestra "No hay registros de acceso disponibles".

Postcondiciones:

El administrador puede visualizar todos los registros de acceso.

5.2.11 Caso de Uso: Consultar los Detalles de un Registro de Acceso

Actor Principal: Administrador

Propósito: Permitir al administrador consultar la información detallada de un registro de acceso específico.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para consultar registros.

Flujo Principal:

El administrador selecciona un registro de la lista.

El sistema muestra los detalles del registro, incluyendo nombre, rol, motivo de acceso (si aplica), fecha, hora de entrada y salida.

Flujo Alternativo:

El registro no existe:

El sistema muestra "Registro de acceso no encontrado".

Postcondiciones:

El administrador visualiza los detalles del registro de acceso.

5.2.12 Caso de Uso: Eliminar un Registro de Acceso

Actor Principal: Administrador

Propósito: Permitir al administrador eliminar un registro de acceso del sistema.

Precondiciones:

El administrador debe estar autenticado.

Debe tener permisos para eliminar registros.

Flujo Principal:

El administrador selecciona un registro de la lista.

El sistema solicita confirmación para eliminarlo.

El administrador confirma la acción.

El sistema elimina el registro de la base de datos.

El sistema confirma que la eliminación fue exitosa.

Flujo Alternativo:

El registro no existe:

El sistema muestra "Registro de acceso no encontrado".

Postcondiciones:

El registro de acceso queda eliminado del sistema.

5.2.13 Caso de Uso: Generar Código QR

Actor Principal: Estudiante

Propósito: Permitir al estudiante generar un código QR único para identificarse al ingresar al campus.

Precondiciones:

El estudiante debe estar autenticado con un token válido.

Debe tener el rol de estudiante.

Flujo Principal:

El estudiante accede a la opción "Generar Código QR".

El sistema valida que el usuario está autenticado y autorizado.

El sistema genera un código QR único asociado al perfil del estudiante.

El sistema muestra el código QR al estudiante.

Flujo Alternativo:

Error al generar el QR:

El sistema muestra el mensaje "No se pudo generar el código QR. Inténtelo más tarde".

Postcondiciones:

El estudiante obtiene un código QR único y válido.

El sistema registra el evento de generación del QR.

5.2.14 Caso de Uso: Consultar Perfil

Actor Principal: Estudiante

Propósito: Permitir al estudiante consultar la información registrada en su perfil.

Precondiciones:

El estudiante debe estar autenticado con un token válido.

Debe tener el rol de estudiante.

Flujo Principal:

El estudiante accede a la opción "Mi Perfil".

El sistema valida que el usuario está autenticado y autorizado.

El sistema muestra la información del perfil del estudiante, incluyendo nombre, correo electrónico, teléfono, código.

Flujo Alternativo:

Perfil no disponible:

El sistema muestra "Error al cargar la información del perfil. Inténtelo más tarde".

Postcondiciones:

El estudiante puede visualizar su información actualizada.

5.2.15 Caso de Uso: Actualizar Mi Perfil

Actor Principal: Estudiante

Propósito: Permitir al estudiante modificar su información personal registrada en el sistema.

Precondiciones:

El estudiante debe estar autenticado con un token válido.

Debe tener el rol de estudiante.

Flujo Principal:

El estudiante accede a la opción "Actualizar Perfil".

El sistema muestra un formulario editable con la información actual del perfil.

El estudiante realiza los cambios necesarios (nombre, teléfono, dirección, etc.) y los envía.

El sistema valida los datos proporcionados.

El sistema actualiza la información en la base de datos.

El sistema confirma que los cambios se han guardado exitosamente.

Flujo Alternativo:

Datos inválidos o incompletos:

El sistema muestra un mensaje indicando los errores específicos.

Postcondiciones:

La información del perfil del estudiante queda actualizada en el sistema.

5.2.16 Caso de Uso: Actualizar Contraseña

Actor Principal: Estudiante

Propósito: Permitir al estudiante cambiar su contraseña de acceso al sistema para mantener su cuenta segura.

Precondiciones:

El estudiante debe estar autenticado con un token válido.

Debe tener el rol de estudiante.

Flujo Principal:

El estudiante accede a la opción "Actualizar Contraseña".

El sistema solicita la contraseña actual y la nueva contraseña.

El estudiante ingresa las contraseñas requeridas y las envía.

El sistema verifica que la contraseña actual es correcta.

El sistema valida que la nueva contraseña cumple con los requisitos de seguridad.

El sistema actualiza la contraseña en la base de datos.

El sistema confirma que la contraseña se ha cambiado exitosamente.

Flujo Alternativo:

Contraseña actual incorrecta:

El sistema muestra "Contraseña actual inválida".

Nueva contraseña no válida:

El sistema muestra un mensaje indicando los errores específicos.

Postcondiciones:

La contraseña del estudiante queda actualizada en el sistema.

El sistema registra el cambio de contraseña para auditoría.

5.2.17 Caso de Uso: Validar Código QR

Actor Principal: Vigilante

Propósito: Permitir al vigilante validar códigos QR presentados por estudiantes para autorizar el ingreso al campus.

Precondiciones:

El vigilante debe estar autenticado con un token válido.

Debe tener el rol de vigilante.

El código QR debe ser presentado por el estudiante.

Flujo Principal:

El vigilante accede a la opción "Validar Código QR".

El sistema solicita al vigilante escanear

El sistema valida el código QR en la base de datos.

Si el código QR es válido:

El sistema muestra el mensaje "Acceso autorizado".

Se muestra información relevante del usuario asociado, como nombre, código, tipo vehículo y matricula.

El sistema registra el intento de validación (ya sea exitoso o fallido).

Flujo Alternativo:

Código QR Inválido:

El sistema muestra "Acceso denegado" junto con la razón del fallo ("Código expirado" o "Código no registrado").

Postcondiciones:

El intento de validación del código QR queda registrado en el sistema para auditoría.

Si es exitoso, el acceso es autorizado.

Excepciones:

El sistema no está disponible: El sistema muestra "Servicio no disponible. Inténtelo más tarde".

Fallo al escanear el QR: El sistema muestra "Error técnico al leer el código QR. Intente nuevamente".

5.2.18 Caso de Uso: Registrar Visitante al Ingresar al Campus

Actor Principal: Visitante

Propósito: Permitir que los visitantes registren su ingreso al campus mediante un formulario público, sin necesidad de autenticarse.

Precondiciones:

El formulario público debe estar disponible en el sistema.

El visitante debe tener acceso a un dispositivo con conexión a internet.

Flujo Principal:

El visitante accede al formulario público de registro a través de un código QR proporcionado por la universidad.

El sistema muestra el formulario de registro con campos obligatorios (nombre, identificación, motivo de visita).

El visitante completa el formulario y envía los datos.

El sistema valida que todos los campos requeridos estén completos y sean correctos.

Si la validación es exitosa:

El sistema registra la información y el acceso del visitante en la base de datos.

El sistema muestra un mensaje de confirmación visual: "Registro y acceso exitoso".

Flujo Alternativo:

Datos incompletos o inválidos:

El sistema muestra un mensaje de error indicando qué campos necesitan corrección.

El visitante puede corregir los datos y volver a enviar el formulario.

Postcondiciones:

El visitante queda registrado en el sistema con su información personal y el motivo de ingreso.

El administrador puede visualizar el registro en tiempo real.

Excepciones:

El sistema no está disponible:

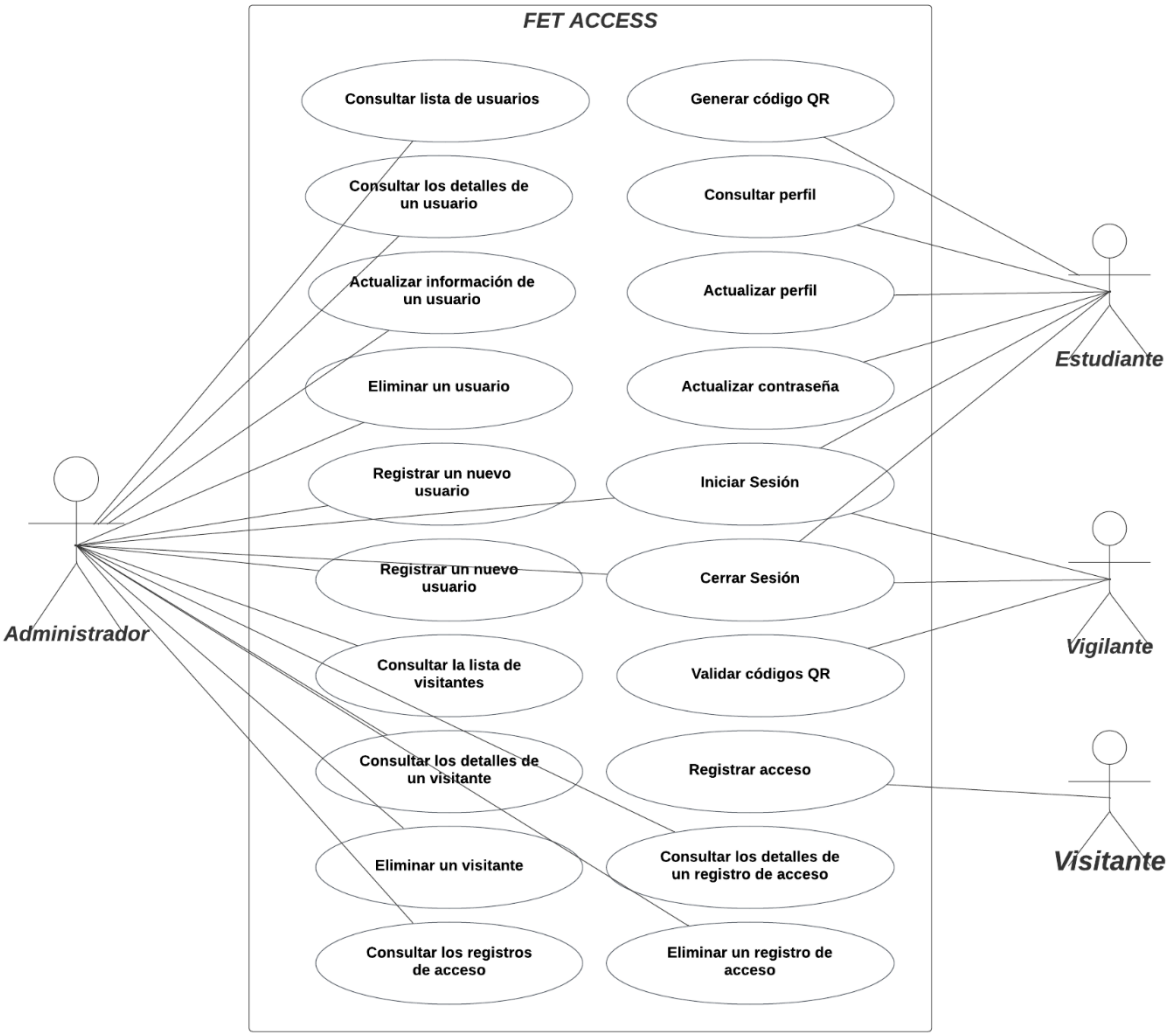
El visitante recibe un mensaje indicando: "El servicio no está disponible en este momento. Intente más tarde".

Conexión a internet fallida:

El visitante no puede acceder al formulario público y recibe un mensaje del navegador indicando que no hay conexión.

5.3 Diseño de casos de uso

Se desarrollarán unos diagramas detallados que muestren el flujo de interacción del usuario con la aplicación para cada uno de los casos de uso. Estos diagramas facilitarán la comprensión del funcionamiento del sistema y permitirán detectar posibles problemas en el flujo de trabajo.



6. Implementación

6.1 Backend

Desarrollado con el framework **Laravel**, el backend se encargará de la lógica del sistema y la gestión de la base de datos. Laravel ofrece un entorno seguro para el desarrollo de aplicaciones web, incluyendo herramientas para la autenticación, validación de datos y protección contra ataques comunes como la inyección SQL y el XSS. El backend manejará las solicitudes de autenticación, la generación y verificación de los códigos QR y la administración de permisos de usuario.

6.2 Frontend

Se utilizará **Ionic** junto con **Angular** para el desarrollo de la aplicación móvil. Ionic permite crear aplicaciones multiplataforma, facilitando su despliegue en dispositivos Android e iOS. La interfaz de usuario será intuitiva y estará diseñada para ofrecer una experiencia accesible, con navegación clara y opciones de fácil acceso tanto para estudiantes como para el personal de seguridad.

6.3 Base de datos

La aplicación utilizará **MySQL** como sistema de gestión de bases de datos, asegurando un almacenamiento estructurado y seguro de los datos de los usuarios. Para proteger la privacidad de la información personal, se implementarán técnicas de cifrado en los datos sensibles y se aplicarán normas de protección de datos en línea con la Ley 1581 de 2012.

6.4 Seguridad

La aplicación implementará medidas de seguridad avanzadas para proteger la información de los usuarios, tales como:

- **Cifrado de datos**: Cifrado de datos personales y credenciales de usuario.
- **Autenticación**: Uso de tokens de autenticación para garantizar que solo usuarios autorizados accedan al sistema.
- **Prevención de ataques de inyección SQL y XSS**: Uso de validaciones y filtros para proteger el sistema contra posibles vulnerabilidades.

7. Pruebas

7.1 Objetivo

Realizar pruebas exhaustivas de la aplicación para garantizar que cumpla con los requisitos de funcionalidad, seguridad y usabilidad.

7.2 Objetivos específicos

- Pruebas unitarias: asegurarse de que cada componente de la aplicación funcione correctamente de forma aislada.
- Pruebas de integración: comprobar si los diferentes componentes de la aplicación funcionan juntos sin errores.
- Pruebas de seguridad: evaluación de la resiliencia del sistema ante posibles vulnerabilidades, especialmente en la autenticación y el manejo de datos personales.
- Pruebas de usuario: recopilación de comentarios de usuarios reales para mejorar la experiencia y la interfaz del usuario.

7.3 Descripción del sistema

La aplicación FET Access está diseñada para gestionar y verificar el acceso de las personas mediante autenticación con códigos QR únicos. Los usuarios autorizados (estudiantes, personal de seguridad y personal administrativo) dispondrán de un código QR personal que será escaneado por el personal de seguridad al acceder al campus.

8. Bases teóricas

8.1 Antecedentes

La seguridad en el acceso a instituciones educativas es un área de creciente importancia en todo el mundo, muchos sistemas actuales no ofrecen un nivel adecuado de seguridad o eficiencia, especialmente en entornos con grandes poblaciones, como universidades, este proyecto, **FET Access** busca cubrir esta necesidad y ser única.

Estas son algunas aplicaciones similares:

- Swiftlane ofrece un sistema de control de acceso basado en códigos QR fácil de instalar y usar que reduce el contacto físico y permite una gestión de acceso eficiente. Sin embargo, presenta algunos desafíos de seguridad, como la posibilidad de compartir códigos QR. (Swiftlane, 2024).
- Link-to-QR proporciona soluciones de acceso y gestión para campus educativos, utilizando códigos QR para mejorar la accesibilidad a los recursos y para el seguimiento de la asistencia, entre otros usos. (Enlace a QR, 2024).

8.2 Marco teórico

Este proyecto se basa en los siguientes conceptos teóricos:

- Gestión de Identidad y Acceso: Este concepto involucra el control y la administración de quién tiene acceso a recursos específicos dentro de un sistema. La gestión de identidad y acceso (IAM) utiliza tecnologías para asegurar que solo las personas autorizadas puedan ingresar a lugares o sistemas, lo que es crucial

para proteger la seguridad en entornos como universidades o empresas. La adopción de herramientas IAM ayuda a reducir riesgos, mejorar la eficiencia y garantizar el cumplimiento normativo (Okta, 2024)

- Códigos QR: Los códigos QR (Quick Response) son una tecnología que permite almacenar información que puede ser leída rápidamente mediante un escaneo. En el ámbito de la gestión de acceso, los códigos QR ofrecen un método seguro y ágil para permitir el acceso a usuarios autorizados. Además, la flexibilidad y facilidad de uso de los QR los hacen ideales para aplicaciones móviles, donde la autenticación rápida y precisa es fundamental (Swiftlane, 2024)
- Seguridad de la Información: La seguridad de la información implica un conjunto de políticas, procedimientos y tecnologías utilizadas para proteger la confidencialidad, integridad y disponibilidad de los datos. En el contexto de sistemas de acceso, la protección de la información personal de los usuarios es crucial. Se implementan métodos como el cifrado de datos y las políticas de autenticación multifactor para evitar accesos no autorizados y proteger la privacidad (NIST, 2024)

8.3 Maco conceptual

El Marco Conceptual de FET Access define los conceptos técnicos y teóricos fundamentales que guían el diseño y desarrollo del sistema de control de acceso, dichos estos conceptos son esenciales para comprender las bases sobre las cuales se construye el sistema y las tecnologías que utiliza.

- Autenticación de usuario: La autenticación es el proceso mediante el cual se verifica la identidad de un usuario antes de permitirle el acceso al sistema. Según Mowbray y Zahavi (2017), la autenticación se puede realizar de varias formas, siendo una de las más comunes el uso de factores como contraseñas, biometría o, en este caso, códigos QR. Este tipo de autenticación es rápida y eficiente, lo que mejora la seguridad en entornos como FET Access.
- Códigos QR: Un código QR (Respuesta Rápida) es un tipo de código de barras bidimensional que almacena información que puede ser leída por dispositivos electrónicos. Según Ittelson (2014), los códigos QR son ideales para aplicaciones móviles debido a su capacidad para almacenar datos de manera eficiente y escanearse rápidamente. En el contexto de FET Access, los códigos QR permiten a los usuarios identificarse de forma única y segura.
- Gestión de identidad y acceso (IAM): La gestión de identidad y acceso (IAM) es un conjunto de políticas y herramientas que garantizan que solo los usuarios autorizados tengan acceso a los recursos de un sistema. Según Harris (2018), IAM permite a las organizaciones gestionar eficientemente el acceso a sus recursos, protegiendo la integridad y confidencialidad de la información. En FET Access, se implementa un sistema IAM para garantizar la verificación de acceso y la gestión de usuarios adecuadas.
- Base de datos relacional: las bases de datos relacionales, como MySQL, organizan los datos en tablas que pueden relacionarse entre sí, facilitando el acceso y la gestión de grandes volúmenes de información. Según Elmasri y

Navathe (2015), los sistemas de bases de datos relacionales son esenciales para almacenar y gestionar información estructurada de manera eficiente y segura.

- Seguridad informática: La seguridad informática se refiere a la protección de los sistemas de información contra accesos no autorizados y otros riesgos. Según Stallings (2017), la seguridad informática abarca varios aspectos, como la confidencialidad, integridad y disponibilidad de la información. El sistema FET Access implementa varias medidas de seguridad, incluido el cifrado de datos y la protección contra vulnerabilidades como las inyecciones SQL.
- Protección de datos personales: la protección de datos personales busca garantizar que la información sobre los usuarios sea tratada de forma que se respete su privacidad. Ley 1581 de 2012 en Colombia establece lineamientos para el tratamiento de datos personales. Según González (2013), la ley establece las medidas que deben seguir las entidades para garantizar la protección y confidencialidad de los datos personales. FET Access sigue estas regulaciones para garantizar que la información del usuario esté protegida adecuadamente.

8.4 Marco legal

La aplicación FET Access debe cumplir con los estándares legales relacionados con la protección de datos personales y la seguridad informática, garantizando así la privacidad del usuario y la integridad de los datos almacenados, las principales leyes que regulan el tratamiento de información personal en Colombia son:

- Ley 1.581 de 2012 – Protección de Datos Personales: Esta ley establece el marco regulatorio para la protección de datos personales en Colombia. Regula el

tratamiento de datos personales por parte de entidades públicas y privadas, exigiendo que se adopten medidas para garantizar la seguridad y confidencialidad de la información. Según González (2013), esta ley exige que las entidades obtengan el consentimiento explícito de los usuarios antes de recopilar y procesar sus datos personales. Además, la ley establece derechos de los usuarios, como el derecho a acceder, rectificar y suprimir sus datos.

- Ley 1.273 de 2009 – Delitos Informáticos: Esta ley sanciona el uso indebido de los sistemas informáticos y el acceso no autorizado a datos o sistemas. En el contexto de Fet Access, la ley garantiza que el sistema de control de acceso se implemente de manera que prevenga el uso fraudulento de información personal y proteja contra ataques cibernéticos. Ramírez (2015) explica que la ley establece sanciones para quienes accedan a la información sin la debida autorización y para quienes implementen mecanismos de seguridad inadecuados.

9. Sprints

Fecha de inicio del proyecto: 5 de agosto

Fecha de finalización del proyecto: 7 de noviembre

9.1 Sprint 1: Planificación y recopilación de requisitos

Duración: 5 de agosto al 18 de agosto

Objetivo: Identificar los requisitos del proyecto, definir objetivos, analizar la viabilidad y planificar las fases del proyecto.

Actividades:

- Reuniones de lanzamiento del proyecto.
- Identificación de requerimientos con los usuarios (estudiantes, personal de seguridad, administración).
- Propuesta de proyecto y redacción de objetivos.

9.2 Sprint 2: Diseño del sistema

Duración: del 19 de agosto al 1 de septiembre

Objetivo: Crear la arquitectura del sistema, el diseño de la base de datos y los primeros prototipos de interfaz.

Actividades:

- Diseño de bases de datos en MySQL.
- Prototipado de interfaz de usuario en Ionic.
- Creación de diagramas de casos de uso y diagramas de flujo.
- Revisión y aprobación del diseño por parte de los stakeholders.

9.3 Sprint 3: Desarrollo backend (Parte 1)

Duración: 2 de septiembre al 15 de septiembre

Objetivo: Implementar la estructura básica de backend con Laravel y la integración de bases de datos.

Actividades:

- Configuración inicial del proyecto en Laravel.
- Implementación de base de datos y creación de modelos.
- Desarrollo de las primeras APIs para el sistema de autenticación y registro de usuarios.

9.4 Sprint 4: Desarrollo frontend (Parte 1)

Duración: del 16 de septiembre al 29 de septiembre

Objetivo: Crear la estructura básica del frontend en Ionic e integración con las API del backend.

Actividades:

- Implementación de la interfaz de inicio de sesión y perfil de usuario.
- Conexión del frontend con APIs para autenticación y consulta de datos.
- Pruebas de integración entre el frontend y backend.

9.5 Sprint 5: Desarrollo backend (Parte 2)

Duración: 30 de septiembre al 13 de octubre

Objetivo: funcionalidades completas de backend, incluida la generación de códigos QR y la gestión de permisos.

Actividades:

- Implementación del sistema de generación y verificación de códigos QR.
- Desarrollo de lógica para la gestión de permisos de usuarios.
- Implantación de medidas de seguridad para proteger los datos.

9. 6 Sprint 6: Desarrollo frontend (Parte 2)

Duración: del 14 de octubre al 27 de octubre

Objetivo: Completar las funcionalidades del frontend, optimizando la interfaz de usuario y agregando las últimas pantallas.

Actividades:

- Creación de la interfaz de lectura de códigos QR para el personal de seguridad.
- Implementación de notificaciones y alertas en la aplicación.
- Optimización de la experiencia del usuario.

9.7 Sprint 7: pruebas y correcciones

Duración: 28 de octubre al 3 de noviembre

Objetivo: Realizar pruebas exhaustivas de todas las funcionalidades y corregir errores identificados.

Actividades:

- Testing unitarios de cada módulo (frontend y backend).
- Pruebas de integración entre el frontend y backend.

- Pruebas de seguridad para verificar la protección de datos sensibles.
- Revisión y corrección de errores identificados durante las pruebas.

9.8 Sprint 8: documentación y preparación para la implementación

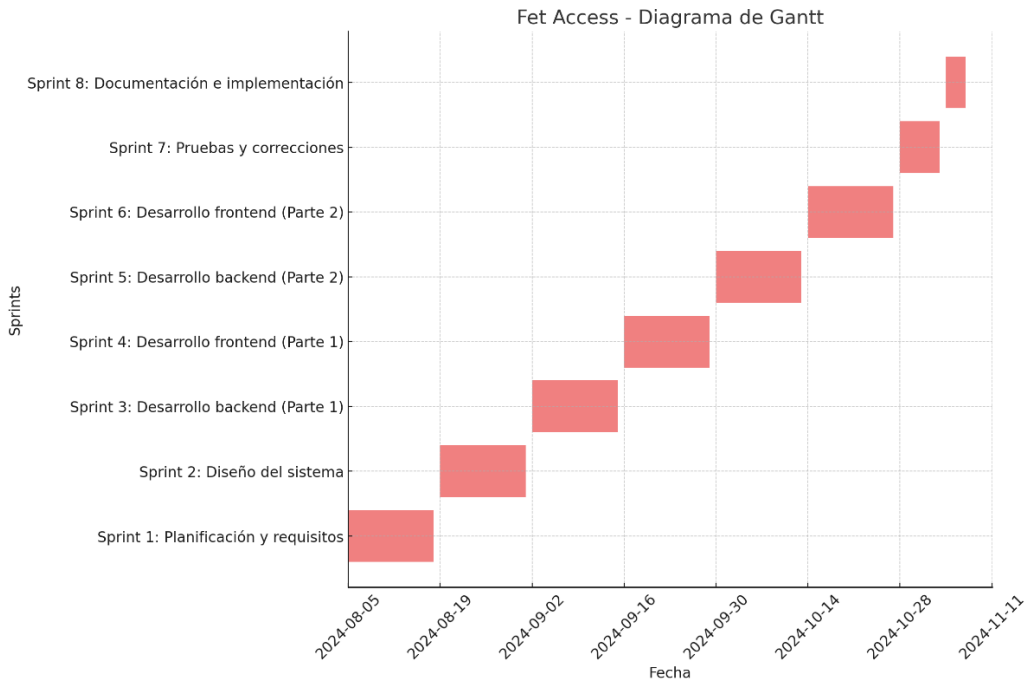
Duración: 4 de noviembre al 7 de noviembre

Objetivo: Documentar el proyecto y realizar configuraciones finales para la implementación en producción.

Actividades:

- Elaborar documentación técnica y manual de usuario.
- Configurar el entorno de producción y desplegar la aplicación.
- Revisión final con los interesados y ajuste de detalles finales.

9.9 Diagrama de Gantt



10. Repositorios

10.1 Backend

https://github.com/JJCL1997/laravel_fet_access.git

10.2 Ionic

https://github.com/JJCL1997/ionic_fet_access.git

10.3 Documentación

https://github.com/DesignsByAlex/Fet-Access---AlexTriana-y-JeissonCortes.git

11. Referencias

Swiftlane, 2024 https://www.swiftlane.com

QR, 2024 https://link-to-qr.com

NIST, 2024 https://ninesmart.io/smart-access/

Mowbray, M. y Zahavi, R. (2017). Autenticación y control de acceso. Wiley. https://www.wiley.com/en-us

Ittelson, T. (2014). Códigos QR: una guía para especialistas en marketing. Revista de tecnología de marketing, 10(2), 102-110. https://www.researchgate.net/

Harris, S. (2018). Manual de gestión de seguridad de la información. Prensa CRC. https://www.crcpress.com/

Elmasri, R. y Navathe, SB (2015). Fundamentos de los sistemas de bases de datos (7ª ed.). Educación Pearson. https://www.pearson.com/

Stallings, W. (2017). Criptografía y seguridad de redes: principios y prácticas. Pearson. https://www.pearson.com/

González, M. (2013). La protección de datos personales en Colombia. Editorial Jurídica. http://editorialjuridica.com/

González, M. (2013). Protección de datos personales en Colombia. Editorial Jurídica. http://editorialjuridica.com/

Ramírez, J. (2015). La ley sobre delitos informáticos y su impacto en la protección de datos. Publicación legislativa, https://www.legis.com.co/