**Manual Técnico**

1. **Roles / integrantes**

* **Team Líder:** Bricely Gutierrez Cadima
* **Git Master:** Ebvam Brandon Zambrana Via
* **Developer:** Fernando Santos Davalos Lovera
* **Developer:** Luis Fernando Kellner Zerda
* **DB:** Nelson Daniel Garcia Copa

1. **Introducción:**

El presente proyecto consiste en una aplicación diseñada para optimizar la gestión de recursos en entornos de salud, proporcionando herramientas eficientes para la administración y seguimiento de procesos clave. La aplicación, desarrollada bajo el patrón MVC y conectada a una base de datos en SQL Server, cuenta con las siguientes.

Principales Funcionalidades:

* Gestión de Establecimientos: Permite registrar establecimientos de salud y asignarles recursos necesarios como botiquines.
* Administración de Botiquines: Facilita la asignación de botiquines a los establecimientos, así como la gestión de su contenido mediante el registro de medicamentos.
* Gestión de Enfermeros: Incluye el registro y administración de enfermeros, quienes a su vez tienen la capacidad de registrar estudiantes en el sistema, acceder a su histórico médico y generar prescripciones personalizadas.
* Reportes Dinámicos: Proporciona informes detallados sobre el número de enfermos registrados en cada establecimiento, permitiendo un análisis preciso de la situación de salud en diferentes localidades.

El sistema está diseñado para garantizar una interfaz intuitiva, una gestión eficiente de datos y una experiencia de usuario optimizada, apoyando la toma de decisiones en el ámbito de la salud.

1. **Descripción del proyecto:**

Este proyecto es una aplicación web desarrollada con el patrón arquitectónico MVC y utilizando SQL Server como base de datos. Su objetivo principal es mejorar la gestión de recursos y procesos en entornos de salud, permitiendo un control eficiente y organizado de establecimientos, personal de salud, estudiantes y medicamentos.

La aplicación está estructurada en módulos funcionales que abarcan las siguientes áreas:

* Registro de Establecimientos:  
  Permite almacenar información de los establecimientos de salud, centralizando datos relevantes para su administración.
* Gestión de Botiquines y Medicamentos:
  + Asignación de botiquines a establecimientos.
  + Registro y actualización del inventario de medicamentos asociados a cada botiquín.
* Administración de Enfermeros:
  + Registro de enfermeros en el sistema.
  + Asignación de enfermeros a establecimientos específicos.
* Atención Estudiantil:
  + Registro de estudiantes en el sistema por parte de los enfermeros.
  + Generación de históricos médicos para cada estudiante.
  + Emisión de prescripciones médicas personalizadas.
* Reportes y Análisis:
  + Generación de reportes sobre estudiantes enfermos organizados por establecimiento.
  + Visualización de estadísticas para apoyar la toma de decisiones.

La aplicación es escalable y está diseñada para garantizar la seguridad y confidencialidad de los datos, brindando herramientas que optimizan las operaciones en el sector salud.

1. **Link al Video demostrativo YouTube (5 minutos máximo)**

[**https://youtu.be/aKCcukNjAvY**](https://youtu.be/aKCcukNjAvY)

1. **Listado de los Requisitos Funcionales del Sistema**

Gestión de Establecimientos:

* RF01: Registrar, editar y eliminar establecimientos en el sistema.
* RF02: Visualizar la lista de establecimientos registrados.

Administración de Botiquines:

* RF03: Registrar y asignar botiquines a establecimientos.
* RF04: Añadir, editar y eliminar medicamentos de un botiquín.
* RF05: Visualizar el inventario de medicamentos por botiquín.

Gestión de Enfermeros:

* RF06: Registrar, editar y eliminar información de enfermeros.
* RF07: Asignar enfermeros a establecimientos específicos.

Atención a Estudiantes:

* RF08: Registrar estudiantes en el sistema.
* RF09: Consultar el historial médico de un estudiante.
* RF10: Registrar diagnósticos y emitir prescripciones médicas.

Generación de Reportes:

* RF11: Generar reportes sobre el número de estudiantes enfermos por establecimiento.
* RF12: Filtrar reportes por fechas, establecimiento o tipo de diagnóstico.

Seguridad del Sistema:

* RF13: Implementar autenticación de usuarios con roles definidos (administrador, enfermero).
* RF14: Restringir el acceso a módulos específicos según el rol del usuario.

Interfaz de Usuario:

* RF15: Proveer una interfaz intuitiva para la gestión de datos.
* RF16: Mostrar notificaciones o alertas al completar operaciones críticas (ej.: asignación de botiquines, eliminación de registros).

Auditoría:

* RF17: Registrar automáticamente la fecha y hora de creación, modificación y eliminación de registros.
* RF18: Permitir la consulta de logs de actividad por usuario.

1. **Arquitectura del software:**

El sistema está desarrollado bajo la arquitectura MVC (Model-View-Controller), que divide la aplicación en tres componentes principales para facilitar la organización, mantenimiento y escalabilidad del software. A continuación, se describen los elementos principales y sus interacciones:

Componentes Principales

* Modelo (Model):
  + Representa la lógica de negocio y la interacción con la base de datos.
  + Incluye las clases que definen las entidades principales del sistema (por ejemplo, *Establishment*, *Nurse*, *Student*, *Medication*, *FirstAidKit*).
  + Contiene las operaciones CRUD (Create, Read, Update, Delete) y la lógica relacionada con los datos, utilizando el patrón DAO (Data Access Object) para acceder a la base de datos.
* Vista (View):
  + Representa la interfaz de usuario.
  + Contiene las páginas web desarrolladas con tecnologías como Razor Pages, HTML, CSS y JavaScript para mostrar información y capturar datos del usuario.
  + Las vistas están vinculadas directamente al controlador correspondiente.
* Controlador (Controller):
  + Actúa como intermediario entre el modelo y la vista.
  + Recibe las solicitudes del usuario, interactúa con el modelo para procesar los datos y selecciona la vista adecuada para presentar los resultados.
  + Define las reglas de negocio y controla el flujo de la aplicación.

**Interacciones entre Componentes**

* Usuario → Controlador:
  + El usuario envía solicitudes desde la interfaz (por ejemplo, registrar un establecimiento).
  + El controlador recibe la solicitud y la procesa.
* Controlador → Modelo:
  + El controlador solicita al modelo que realice operaciones con los datos (consultar o actualizar registros en la base de datos).
* Modelo → Controlador:
  + El modelo devuelve los datos procesados al controlador.
* Controlador → Vista:
  + El controlador selecciona la vista adecuada y envía los datos para ser presentados al usuario.
* Vista → Usuario:
  + La vista muestra la información procesada o notifica el resultado de las operaciones realizadas.

1. **Base de datos**
   1. **Diagrama completo y actual** **Diagrama, Dibujo de ingeniería

      Descripción generada automáticamente**
   2. **En el GIT una carpeta con la base de datos con script de generación e inserción de datos de ejemplo utilizados**
   3. **Script simple (copiado y pegado en este documento)**
2. **Listado de Roles más sus credenciales de todos los Admin / Users del sistema**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rol | UserName | Contraseña | Email | Estado |
| Administrador | **Admin** | **password123** | **123@ejemplo** | **Activo** |
| Enfermero |  |  |  |  |
|  |  |  |  |  |

1. **Requisitos del sistema:**

**Requerimientos de Hardware (Cliente)**

* **Procesador:** Intel Core i3 o equivalente.
* **Memoria RAM:** 4 GB.
* **Almacenamiento:** 500 MB de espacio disponible.
* **Pantalla:** Resolución mínima de 1366x768 píxeles.
* **Conexión a Internet:** Ancho de banda mínimo de 2 Mbps para acceso a funcionalidades en línea.

**Requerimientos de Software (Cliente)**

* **Sistema Operativo:** Windows 10/11, macOS Catalina o superior.
* **Navegador Compatible:**
  + Google Chrome (versión 90 o superior).
  + Mozilla Firefox (versión 88 o superior).
  + Microsoft Edge (versión 91 o superior).
* **Frameworks/Plugins:** .NET Framework 4.8 o superior (si la aplicación incluye módulos locales).

**Requerimientos de Hardware (Servidor/Hosting/BD)**

* **Procesador:** Intel Xeon o equivalente (2 GHz o superior, 4 núcleos).
* **Memoria RAM:** 16 GB (mínimo para servidores pequeños).
* **Almacenamiento:**
  + 250 GB de SSD para almacenamiento del sistema y la base de datos.
  + Backup adicional con al menos 500 GB de espacio.
* **Conexión de Red:** Conexión dedicada de al menos 10 Mbps.

**Requerimientos de Software (Servidor/Hosting/BD)**

* **Sistema Operativo:**
  + Windows Server 2019/2022 o Linux (Ubuntu 20.04 LTS o superior).
* **Servidor Web:**
  + IIS (Internet Information Services) 10 para Windows.
  + Nginx o Apache (si se usa Linux).
* **Base de Datos:** Microsoft SQL Server 2019 o superior.
* **Frameworks/Entornos:**
  + .NET Core 3.1 o superior / .NET 6 para la aplicación web.
  + Entity Framework para manejo de base de datos.
* **Seguridad:**
  + Certificado SSL/TLS configurado en el servidor.
  + Firewall y herramientas de monitoreo como Fail2Ban o Azure Security Center.

1. **Instalación y configuración:**

**Requisitos Previos:**

* Instalar un navegador compatible (Google Chrome, Mozilla Firefox o Microsoft Edge).
* Verificar que el sistema operativo cumpla con los requisitos mínimos especificados.

**Pasos de Instalación:**

1. Acceda al enlace de descarga proporcionado por el administrador del sistema.
2. Descargue el instalador del cliente, si es necesario.
3. Ejecute el instalador y siga las instrucciones en pantalla para completar la instalación.
4. Una vez instalado, acceda al sistema a través del navegador ingresando la URL del servidor.

**2. Configuración del Servidor y Base de Datos**

**Requisitos Previos:**

* Un servidor con los requerimientos de hardware y software especificados.
* Instalación de SQL Server y un servidor web como IIS o Nginx.

**Pasos de Configuración:**

**A. Configuración del Servidor Web:**

1. **Instalación de IIS (Windows):**
   * Abra el Administrador de Servidores y active el rol de **IIS**.
   * Habilite las características necesarias como **ASP.NET Core Hosting Bundle**.
2. **Configuración del Sitio Web:**
   * Copie los archivos del proyecto al directorio del servidor web.
   * Configure el archivo web.config o appsettings.json para especificar la URL base y otras configuraciones.

**B. Instalación y Configuración de SQL Server:**

1. Instale Microsoft SQL Server (versión 2019 o superior) con herramientas como SQL Server Management Studio (SSMS).
2. Cree una base de datos nueva con el nombre requerido (ej.: bdHealthSystem).
3. Importe las tablas iniciales utilizando el script SQL proporcionado.

sql

Copiar código

-- Script para crear tablas y datos iniciales

USE bdHealthSystem;

**C. Configuración del Archivo appsettings.json:**

1. Localice el archivo appsettings.json en el proyecto.
2. Actualice la cadena de conexión para reflejar la configuración del servidor SQL:

{

"ConnectionStrings": {

"DefaultConnection": "Server=SERVER\_NAME;Database=bdHealthSystem;User=Usuario;Password=Contraseña;Trusted\_Connection=True;Encrypt=False;"

}

}

**3. Configuración de Roles y Usuarios Iniciales**

1. Acceda al sistema con credenciales predefinidas de administrador
2. Configure los roles y usuarios adicionales desde el módulo de administración de usuarios.
3. Verifique que los permisos de acceso estén correctamente asignados.

**4. Verificación del Sistema**

**Pruebas Básicas:**

1. Inicie sesión como administrador y acceda a todos los módulos principales.
2. Registre un establecimiento, un botiquín y asigne medicamentos.
3. Genere un reporte para confirmar que la conexión con la base de datos es funcional.

**Pruebas de Cliente:**

1. Desde un equipo cliente, acceda al sistema ingresando la URL del servidor.
2. Realice una operación básica como consultar el inventario o ver un historial médico.
3. **PROCEDIMIENTO DE HOSTEADO / HOSTING (configuración)**

* **Sitio Web.**
* **B.D.**
* **API / servicios Web**
* **Otros** *(firebase, etc.)*

*Detalle DETALLADO paso a paso de la puesta en marcha en hosting, tanto para el sitio Web, API, B.D., etc.etc. (incluir scripts BD, Credenciales de acceso server, root BD, Admin, users clientes etc.)*

1. **GIT :**

* Versión final entregada del proyecto.
* Entrega compilados ejecutables

1. **Dockerizado** *Del Sitio WEB, de la Base de Datos*
   1. **Proceso de dokerizado, Configuración**
   2. **Como hacer Correr, Acceso credenciales:** 
      1. **base datos**
      2. **Roles Admin, User, etc**
      3. **Base de datos con datos válidos y legibles.**
2. **Personalización y configuración:**

*Parámetros Personalizables:*

* 1. *Cambiar colores y logos desde la carpeta /wwwroot/css.*
  2. *Configurar idioma en appsettings.json.*

1. **Seguridad**:
   1. *Permisos:*
   2. *Restringir acceso a carpetas sensibles del servidor.*

*• Autenticación:*

* 1. *Implementar roles para acceso limitado.*

*• Cifrado:*

* 1. *Usar HTTPS y cifrar contraseñas con SHA256*

1. **Depuración y solución de problemas:**

* Conexión a la base de datos: Revisar cadena en appsettings.json.
* API no responde: Verificar puertos abiertos.

1. **Glosario de términos:**
   1. CRUD: Operaciones Crear, Leer, Actualizar, Eliminar.
   2. MVC: Modelo-Vista-Controlador, patrón de diseño.
2. **Referencias y recursos adicionales:**
   1. Microsoft Docs: Documentación oficial de ASP.NET Core
   2. Docker: Guía oficial
3. **Herramientas de Implementación:**

* Lenguajes de Programación: C#, SQL, HTML, JS, CSS.
* Frameworks: ASP.NET Core, Entity Framework.
* APIs de Terceros: Firebase para notificaciones.

1. **Bibliografía**

**Libros**

1. **Martin Fowler (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.**
   * **Una referencia clave para comprender patrones arquitectónicos como MVC y su aplicación en sistemas empresariales.**
2. **Robert C. Martin (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.**
   * **Proporciona principios fundamentales para escribir código limpio y mantenible.**
3. **Eric Evans (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.**
   * **Guía sobre cómo estructurar sistemas complejos mediante un diseño orientado al dominio.**
4. **Adam Freeman (2020). *Pro ASP.NET Core 5*. Apress.**
   * **Una guía integral sobre desarrollo con ASP.NET Core, incluyendo configuraciones avanzadas y patrones de implementación.**
5. **Itzik Ben-Gan (2019). *T-SQL Fundamentals*. Microsoft Press.**
   * **Introducción y fundamentos de T-SQL, con ejemplos prácticos aplicables a proyectos en SQL Server.**

**Documentación Oficial**

1. **Microsoft Documentation**
   * **ASP.NET Core:** [**https://learn.microsoft.com/en-us/aspnet/core/**](https://learn.microsoft.com/en-us/aspnet/core/)
   * **Entity Framework Core:** [**https://learn.microsoft.com/en-us/ef/core/**](https://learn.microsoft.com/en-us/ef/core/)
   * **SQL Server:** [**https://learn.microsoft.com/en-us/sql/sql-server/**](https://learn.microsoft.com/en-us/sql/sql-server/)
2. **Docker Documentation**
   * **https://docs.docker.com/**
3. **Firebase Documentation**
   * **https://firebase.google.com/docs**

**Artículos y Blogs Técnicos**

1. **CodeProject**
   * **Artículos detallados sobre implementación de MVC en ASP.NET Core:** [**https://www.codeproject.com/**](https://www.codeproject.com/)
2. **Stack Overflow**
   * **Preguntas frecuentes y soluciones prácticas:** [**https://stackoverflow.com/**](https://stackoverflow.com/)
3. **Medium**
   * **Artículos sobre desarrollo con ASP.NET y bases de datos SQL:** [**https://medium.com/**](https://medium.com/)
4. **SQL Shack**
   * **Recursos avanzados sobre SQL Server:** [**https://www.sqlshack.com/**](https://www.sqlshack.com/)

**Recursos Online**

1. **Pluralsight**
   * **Cursos en línea sobre ASP.NET Core, Docker y SQL Server:** [**https://www.pluralsight.com/**](https://www.pluralsight.com/)
2. **Udemy**
   * **Tutoriales prácticos y proyectos sobre ASP.NET Core:** [**https://www.udemy.com/**](https://www.udemy.com/)
3. **GitHub Repositories**
   * **Repositorios abiertos para proyectos similares:** [**https://github.com/**](https://github.com/)
4. **W3Schools**
   * **Introducción a HTML, CSS y JS:** [**https://www.w3schools.com/**](https://www.w3schools.com/)
5. **Reddit: r/programming**
   * **Discusiones y recursos útiles para desarrolladores:** [**https://www.reddit.com/r/programming/**](https://www.reddit.com/r/programming/)