**AEGCM: Agendacion Eficiente para Gestión de Citas Médicas.**

**FRANKLIN PARRALES-BRAVO1, (Senior Member, IEEE), TOMALA GONZALES FEELIX 1, MALAVE BRIONES JOSE 1, (Senior Member, IEEE), ROMERO SERANO JEAN1, FALQUEZ PICO STEVEN1, AND FALCONI VILLAMAR MARCOS 1**

**1Grupo de Investigación en Ingeniería en Software, Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil 090514, Ecuador.**

**Corresponding author: Franklin Parrales-Bravo (franklin.parralesb@ug.edu.ec**

**I. ABSTRACT:** Este documento ha sido elaborado por estudiantes de sexto semestre de la carrera de Software de la Universidad de Guayaquil, bajo la guía de la asignatura de Construcción de Software, del paralelo SOFT-VE-6-2.

En la actualidad, la gestión eficiente de citas médicas es esencial para proporcionar una atención médica de calidad y mejorar la experiencia del paciente. En este contexto, presentamos el manual técnico para la implementación del sistema de Agendación de Citas Médicas. Este sistema tiene como objetivo principal optimizar la gestión de citas médicas, mejorar la experiencia de los pacientes y facilitar la administración de horarios y recursos médicos.

**II. INDEX TERMS:** La creciente demanda por servicios de salud ha generado una necesidad apremiante por mejorar la eficiencia de los procesos administrativos en los centros médicos. La gestión de citas médicas, en particular, juega un papel fundamental en la organización de la atención al paciente. Sin embargo, los sistemas tradicionales suelen presentar limitaciones que afectan tanto a los pacientes como al personal médico.

Este artículo presenta una propuesta de sistema integrado para la optimización de la gestión de citas médicas, diseñado para abordar las deficiencias de los sistemas actuales. El sistema propuesto combina tecnologías de la información y comunicación para automatizar los procesos de solicitud, confirmación y gestión de citas, ofreciendo una interfaz intuitiva y fácil de usar tanto para los pacientes como para el personal médico.

**Las principales contribuciones de este trabajo son:**

El manuscrito introduce un sistema abierto y flexible, disponible en GitHub, que permite personalizar y adaptar.

* **Desarrollo de un sistema de agendamiento flexible y escalable:** Se presenta una arquitectura modular y abierta que permite una fácil personalización y adaptación a diferentes entornos clínicos, además el sistema ha sido diseñado para integrarse con diversos sistemas de historia clínica.
* **Interfaz de usuario intuitiva y amigable:** Se ha desarrollado una interfaz de usuario intuitiva y fácil de utilizar, tanto para los pacientes como para el personal administrativo. La interfaz proporciona una experiencia de usuario fluida y eficiente.
* **Optimización de la gestión de citas:** El sistema implementa una optimización para asignar citas de manera eficiente, teniendo en cuenta la disponibilidad de los médicos, las preferencias de los pacientes y la urgencia de las consultas. Esto permite reducir los tiempos de espera y mejorar la utilización de los recursos.
* **Evaluación exhaustiva del sistema:** Se ha realizado una evaluación rigurosa del sistema, tanto desde una perspectiva cuantitativa como cualitativa. Los resultados de la evaluación muestran que el sistema es capaz de mejorar significativamente la eficiencia y la calidad de la gestión de citas médicas.

**III. SYSTEM DESCRIPTION:** El desarrollo de este sistema se llevará a cabo utilizando la metodología Scrum, una estrategia ágil que se basa en la colaboración en equipo, la entrega incremental y la retroalimentación constante de los usuarios. La elección de Scrum se fundamenta en las siguientes características que se alinean con los objetivos del proyecto:

* **Enfoque en el Trabajo Colaborativo:** Scrum promueve un entorno de trabajo colaborativo en el que los miembros del equipo se reúnen regularmente para planificar, ejecutar y evaluar el progreso del proyecto. Esta colaboración fomenta la comunicación abierta, la resolución de problemas conjunta y la toma de decisiones compartida.
* **Obtención de Mejores Resultados a través de Entregas Parciales:** Scrum divide el desarrollo del proyecto en ciclos cortos, cada uno con un conjunto definido de objetivos y entregables. Esta metodología permite obtener resultados tangibles de forma regular, adaptándose a las necesidades cambiantes del proyecto y proporcionando retroalimentación temprana a los usuarios.
* **Priorización Basada en el Valor para el Usuario:** Las funcionalidades del sistema se priorizan en función de su valor para los usuarios finales. Esto asegura que se dediquen más recursos a las características que tienen un mayor impacto en la experiencia del usuario y la satisfacción general.
* **Gestión Ágil:** Scrum se basa en los principios de la gestión ágil, que enfatizan la adaptación a los cambios y la respuesta rápida a la retroalimentación. Esto permite que el proyecto se ajuste a las necesidades cambiantes del mercado y de los usuarios, asegurando su éxito a largo plazo.

**IV. SYSTEM MODULE**

**Módulos del Sistema**

El sistema se compone de dos módulos principales:

* **Módulo de Gestión**: Se encarga de la administración de los recursos humanos y materiales, incluyendo la información de pacientes, médicos, y citas.
* **Módulo de Operación:** Maneja la coordinación y el seguimiento de las citas, desde la solicitud hasta la finalización de la cita.

Ventajas del Sistema

**El sistema de Agendación de Citas Médicas ofrece varias ventajas:**

* Optimización de la gestión de citas, permitiendo una programación eficiente.
* Mejora en la experiencia del paciente, brindando una atención rápida y personalizada.
* Generación de informes con indicadores clave como el número de citas programadas, canceladas y la satisfacción del paciente.

Este sistema se presenta como una herramienta innovadora que mejora la eficiencia y efectividad en la gestión de citas médicas, lo que contribuye al crecimiento y competitividad de las instituciones médicas.

**Descripción de Bases de Datos**

La base de datos del sistema de Agendación de Citas Médicas ha sido diseñada para soportar eficientemente los procesos definidos en el software. A continuación, se describen las tablas y se incluyen diagramas para una mejor comprensión de la estructura y relaciones.

# 

# **Ambiente de pruebas**

El programa fue testeado en Windows 11 en conjunto con, Mstest, Visual Studio y SQL SERVER con su debido servidor de conexión

## **Requerimientos base de hardware**

Los siguientes elementos sirven y bastan para ejecutar las pruebas unitarias que se requieran

|  |  |  |
| --- | --- | --- |
| **Recurso** | **Cantidad** | **Nombre y tipo** |
| Servidor de conexión | 1 | SQL SERVER – base de datos |
| Equipo Electrónico | 1 | PC o laptop |

## **Requerimientos base de software en el ambiente de pruebas**

La siguiente tabla define los elementos de software requeridos en el ambiente de pruebas.

|  |  |  |
| --- | --- | --- |
| **Elemento de software** | **Versión** | **Tipo** |
| Windows 11 | 11 | S. O. |
| Visual Studio Community 2022 | 11 | IDE |
| SQL Server | 2022 | SSMS |
| Mstest | 8 | Framework |

Este artículo presenta una guía detallada sobre las mejores prácticas para optimizar y mantener el software de agendamiento de citas médicas, garantizando su correcto funcionamiento y seguridad.

**1. Actualización del Sistema Operativo:** Mantener el sistema operativo actualizado a la última versión de Windows 10 es crucial para garantizar la compatibilidad con el software de agendamiento y asegurar un rendimiento óptimo. Las actualizaciones regulares corrigen errores, mejoran la seguridad y optimizan el funcionamiento general del sistema.

**2. Actualización de Controladores:** Los controladores son los encargados de la comunicación entre el sistema operativo. Es fundamental verificar que los controladores de la computadora donde se instala el software de agendamiento estén actualizados. Las actualizaciones de controladores pueden mejorar el rendimiento, la estabilidad y la compatibilidad del software.

**3. Actualización de Software Relevante:** Es importante mantener actualizado todo el software relacionado con el sistema de agendamiento, incluyendo el software médico, el software de gestión de pacientes y cualquier otro software complementario. Las actualizaciones regulares corrigen errores, mejoran la seguridad y garantizan la compatibilidad con las últimas versiones del sistema operativo y del software de agendamiento.

**4. Eliminación de Software Innecesario:** Un sistema operativo sobrecargado con software innecesario puede afectar el rendimiento del software de agendamiento. Se recomienda realizar un análisis periódico para identificar y eliminar programas que no se utilizan. Esto liberará recursos del sistema y mejorará el rendimiento general del software de agendamiento.

**5. Mantenimiento del Antivirus Actualizado:** La seguridad de los datos de los pacientes es primordial. Es fundamental mantener un antivirus actualizado para proteger el sistema de agendamiento contra virus, malware y otras amenazas cibernéticas. Las actualizaciones regulares del antivirus garantizan la máxima protección contra las últimas amenazas.

**6. Revisión de Eventos Planificados:** Es importante revisar periódicamente los eventos programados en el calendario del sistema de agendamiento para asegurarse de que sean precisos y no generen conflictos o errores. Esto ayudará a prevenir problemas de programación y garantizará una experiencia fluida para los pacientes.

**7. Monitoreo del Rendimiento del Sistema:** La implementación de herramientas de monitoreo del rendimiento del sistema permite detectar posibles cuellos de botella, problemas de rendimiento o fallas antes de que afecten a los usuarios. El monitoreo regular del rendimiento del sistema ayuda a identificar y resolver problemas de manera proactiva, garantizando un funcionamiento óptimo del software de agendamiento.

**8. Gestión de la Configuración del Software:** Establecer un proceso de gestión de la configuración para controlar y administrar los cambios en el sistema de software es crucial para mantener la estabilidad y la integridad del sistema. Se recomienda utilizar herramientas de control de versiones y mantener un registro detallado de las modificaciones realizadas. Esto facilita la identificación y resolución de problemas en caso de que surjan.

**9. Pruebas y Control de Calidad:** Realizar pruebas regulares del sistema de agendamiento es fundamental para verificar su correcto funcionamiento y detectar posibles errores o fallos. Las pruebas deben incluir pruebas de integración, pruebas de rendimiento, pruebas de seguridad y pruebas funcionales.

# **V. Metodología Scrum**

El desarrollo de este sistema se llevará a cabo utilizando la metodología Scrum, una estrategia ágil que se basa en la colaboración en equipo, la entrega incremental y la retroalimentación constante de los usuarios. La elección de Scrum se fundamenta en las siguientes características que se alinean con los objetivos del proyecto: **Enfoque en el Trabajo Colaborativo:** Scrum promueve un entorno de trabajo colaborativo en el que los miembros del equipo se reúnen regularmente para planificar, ejecutar y evaluar el progreso del proyecto. Esta colaboración fomenta la comunicación abierta, la resolución de problemas conjunta y la toma de decisiones compartida. **Obtención de Mejores Resultados a través de Entregas Parciales:** Scrum divide el desarrollo del proyecto en ciclos cortos denominados "sprints", cada uno con un conjunto definido de objetivos y entregables. Esta metodología permite obtener resultados tangibles de forma regular, adaptándose a las necesidades cambiantes del proyecto y proporcionando retroalimentación temprana a los usuarios. **Priorización Basada en el Valor para el Usuario:** Las funcionalidades del sistema se priorizan en función de su valor para los usuarios finales. Esto asegura que se dediquen más recursos a las características que tienen un mayor impacto en la experiencia del usuario y la satisfacción general. **Gestión Ágil:** Scrum se basa en los principios de la gestión ágil, que enfatizan la adaptación a los cambios y la respuesta rápida a la retroalimentación. Esto permite que el proyecto se ajuste a las necesidades cambiantes del mercado y de los usuarios, asegurando su éxito a largo plazo.

**VI. SYSTEM DESCRIPTION:**

**Funcionalidades del Sistema:**

* **Registro de pacientes:** Los pacientes pueden registrarse fácilmente en el sistema proporcionando su información personal y médica básica.
* **Búsqueda y selección de citas:** Los pacientes pueden buscar citas disponibles por médico, especialidad, fecha y hora a través de una interfaz de usuario intuitiva.
* **Solicitud de citas:** Los pacientes pueden solicitar citas en línea con unos pocos clics, eliminando la necesidad de llamadas telefónicas.
* **Confirmación y recordatorios de citas:** El sistema envía automáticamente confirmaciones de citas por correo electrónico o SMS a los pacientes. Además, se pueden configurar recordatorios previos a la cita para evitar olvidos.
* **Gestión de agenda médica:** El personal médico puede visualizar su agenda en tiempo real, gestionar las citas solicitadas por los pacientes y programar nuevas citas.
* **Cancelación de citas**: Los pacientes pueden cancelar citas fácilmente a través del sistema, liberando espacio en la agenda del médico para otros pacientes.

**Beneficios del Sistema:**

* Para los pacientes:
  + Reduce los tiempos de espera para programar citas.
  + Aumenta la comodidad y flexibilidad para solicitar citas en línea.
  + Disminuye el riesgo de olvidar citas gracias a los recordatorios automáticos.
  + Proporciona una experiencia de usuario más positiva.

**VII. Análisis del Diagrama:**

El diagrama representa un sistema llamado "ConsultorioPrivado". Dentro de este sistema se pueden identificar varios paquetes o módulos, cada uno con una función específica:

* **ConsultorioPrivado:** Este es el paquete principal que engloba a todos los demás. Representa el sistema completo.
* **Datos**: Este paquete parece encargarse de la gestión de datos. Contiene:
* **DbOperaciones:** Probablemente se encarga de realizar operaciones directamente con la base de datos, como consultas, inserciones, actualizaciones y eliminaciones.
* **Interface:** Podría ser una interfaz que define las operaciones básicas que se pueden realizar con los datos, sirviendo como contrato para otras clases.
* **Modelo:** Este paquete probablemente contiene las clases que representan los objetos del dominio del problema, es decir, las entidades del mundo real que se quieren modelar en el sistema (por ejemplo, pacientes, citas, etc.).
* **Controlador:** Este paquete suele contener las clases que se encargan de coordinar la interacción entre la vista y el modelo. Recibe las entradas del usuario (a través de la vista) y las traduce en operaciones sobre el modelo.
* **Vista:** Este paquete contiene las clases que se encargan de presentar la información al usuario, como las interfaces gráficas o los reportes.
* **Utilidad:** Este paquete podría contener clases que ofrecen funcionalidades auxiliares o de utilidad general para el sistema, como herramientas de validación, utilidades de fecha y hora, etc. Dentro de este paquete se encuentran:
* **Datos:** Podría contener clases relacionadas con datos que no son persistentes en la base de datos, como datos de configuración o datos temporales.
* **Forms:** Probablemente contiene clases que representan los formularios que el usuario utiliza para interactuar con el sistema.
* **Resources:** Podría contener recursos como imágenes, archivos de configuración, etc.

Diagrama

Descripción generada automáticamente

**VIII. Diagrama De Componentes:**

El diagrama representa un sistema llamado "ConsultorioPrivado". Dentro de este sistema se pueden identificar varios paquetes o módulos, cada uno con una función específica:

* **ConsultorioPrivado:** Este es el paquete principal que engloba a todos los demás. Representa el sistema completo.
* **Datos:** Este paquete se encarga de la gestión de datos:
* **DbOperaciones:** Probablemente se encarga de realizar operaciones directamente con la base de datos, como consultas, inserciones, actualizaciones y eliminaciones.
* **Interface:** Podría ser una interfaz que define las operaciones básicas que se pueden realizar con los datos, sirviendo como contrato para otras clases.
* **Modelo:** Este paquete probablemente contiene las clases que representan los objetos del dominio del problema, es decir, las entidades del mundo real que se quieren modelar en el sistema (por ejemplo, pacientes, citas, etc.).
* **Controlador:** Este paquete suele contener las clases que se encargan de coordinar la interacción entre la vista y el modelo. Recibe las entradas del usuario (a través de la vista) y las traduce en operaciones sobre el modelo.
* **Vista:** Este paquete contiene las clases que se encargan de presentar la información al usuario, como las interfaces gráficas o los reportes.
* **Utilidad:** Este paquete podría contener clases que ofrecen funcionalidades auxiliares o de utilidad general para el sistema, como herramientas de validación. Diagrama

  Descripción generada automáticamenteDentro de este paquete se encuentran:
* **Datos:** Podría contener clases relacionadas con datos que no son persistentes en la base de datos, como datos de configuración o datos temporales.
* **Forms:** Probablemente contiene clases que representan los formularios que el usuario utiliza para interactuar con el sistema.
* **Resources:** Podría contener recursos como imágenes, archivos de configuración, etc.