

DOCUMENTO DE PLAN DE PRUEBAS

**PLAN DE PRUEBAS PROYECTO**

**AGENDACION DE CITAS MEDICAS.**

**INTEGRANTES:**

* [**TOMALA GONZÁLEZ FÉLIX JAVIER**](https://campusvirtual2.ug.edu.ec/user/view.php?id=17915&course=776). **(LIDER)**
* [**MALAVE BRIONES JOSE ROBERTO**](https://campusvirtual2.ug.edu.ec/user/view.php?id=16834&course=776).
* [**FALCONI VILLAMAR MARCOS ISAAC**](https://campusvirtual2.ug.edu.ec/user/view.php?id=15035&course=776).
* [**FALQUEZ PICO STEVEN DAVID**](https://campusvirtual2.ug.edu.ec/user/view.php?id=15156&course=776).
* [**ROMERO SERRANO JEAN MARCO**](https://campusvirtual2.ug.edu.ec/user/view.php?id=15151&course=776)

Tabla de contenido

[1. Introducción 3](#_Toc150201557)

[1.1. propósito 3](#_Toc150201558)

[1.2. objetivos 3](#_Toc150201559)

[2. Alcance 4](#_Toc150201560)

[2.1. alcance de pruebas 4](#_Toc150201561) [2.2 modulos de conexión 4](#_Toc150201563) 2.3 [modulos de datos 4](#_Toc150201563)

# 3. DEFINICIONES, SIGLAS, Y ABREVIATURAS………………………………………………4

[4. Documentos de referencia 5](#_Toc150201564)

[5. Estrategia de pruebas 5](#_Toc150201565)

[5.1 pruebas funcionales 5](#_Toc150201566)

[6. Criterios de entrada y de salida 9](#_Toc150201567)

[6.1. criterio de entrada del plan de pruebas 9](#_Toc150201568)

[6.2. criterio de aceptación del plan de pruebas 9](#_Toc150201569)

[6.3. criterio de suspensión y reanudación 10](#_Toc150201570)

[7. Entregables 10](#_Toc150201571)

[7.1. protocolo de pruebas 10](#_Toc150201572)

[7.2. informe de pruebas 10](#_Toc150201573)

[7.3. reporte de observaciones 10](#_Toc150201574)

[8. Ambiente de pruebas 10](#_Toc150201575)

[8.1. requerimientos base de hardware 11](#_Toc150201576)

[8.2. requerimientos base de software en el ambiente de pruebas 11](#_Toc150201577)

[8.3. herramientas de apoyo para la ejecución de pruebas 11](#_Toc150201578)

[9. Cronograma de trabajo 12](#_Toc150201579)

[10. Riesgos, dependencias, suposiciones y restricciones 13](#_Toc150201580)

[10.1. riesgos 13](#_Toc150201581)

[10.2. dependencias 13](#_Toc150201582)

[10.3. suposiciones 13](#_Toc150201583)

[10.4. restricciones 13](#_Toc150201584)

# Introducción

## Propósito

El propósito de las pruebas es el de gestionar las conexiones con la base de datos, también el de comprobar si algunos de nuestras consultas tienen errores del lado de las interacciones con la propia base de datos, como lo es caso de la intercomunicación y gestión de estos datos.

## Objetivos

El documento informa como está diseñados las pruebas que integra nuestro sistema, cada uno de los diferentes puntos de las integraciones de lado de loa base de datos y como nuestra aplicación interactúa con estas partes para que se desarrollen de forma adecuada.

* Validar la calidad de las integraciones, tanto las funciones que receptan datos como los que reciben parámetros
* Gestionar como estas trasferencias se ejecutan adecuadamente y como estos datos entran en el sistema
* Buscar esclarecer los sistemas funcionales de insertar, datos y eliminar para comprobar su correcto funcionamiento con sus respectivos datos de origen

# Alcance

## Alcance de pruebas

|  |  |
| --- | --- |
| Tipo de pruebas | Aplica según el proyecto |
| Pruebas funcionales | Si |
| Pruebas de regresión | No |
| Pruebas de interfaz de usuario | Si |
| Prueba de rendimiento | Si |
| Pruebas de carga | Si |
| Pruebas de seguridad y control de acceso | Si |
| Pruebas de instalación | No |
| Pruebas de integración | Si |
| Pruebas de control de bitácoras | SI |

* **Módulo de Conexión**

En este módulo lo que se busca es que todas las conexiones con la base de datos que usamos en nuestra aplicación estén correctas, sin una debida y adecuada conexión no podemos gestionar la entrada de datos y las peticiones de cierto conjunto de atributos que se necesitan para que nuestra aplicación funcione de forma adecuada

* **Módulo de Datos**

En esta parte de aplicación se gestiona la entrada de los datos, estos datos le pertenecen a una entidad en concreto, los mismos datos que deben ser ingresados dentro de los formularios, pero en este caso solo se comprobaran los parámetros de interconectividad con la aplicación y la base de datos de acuerpo a un procedimiento estipulado en los requisitos.

También se toma encuentra la parte de la recolección de datos, en esto se requerirá que nuestra base de datos sea capaz de proveer cada uno de los registros que esta tenga y a su vez que sean correctamente adecuados dentro del sistema y recolectados de forma adecuada

# Definiciones, siglas, y abreviaturas

|  |  |
| --- | --- |
| **Siglas** | **Definición** |
| *M* | Página dentro de la web y que se puede acceder desde un navegador y que tendrá las siguientes características:   * Catálogo de productos que se ofrecen al usuario * Descripción de productos * Presentación de ofertas |
| *Pa* | Entidad paciente dentro del sistema |
| *Me* | Entidad medico dentro del sistema |
| *Catálogo* | Es donde se va a presentar los productos a vender, teniendo un orden dentro de la web mucho más agradable a la vista del usuario y de manera ordenada. |
| *FR* | Framework |
| *MSTESTER* | Framework que se usa para realizar las pruebas |

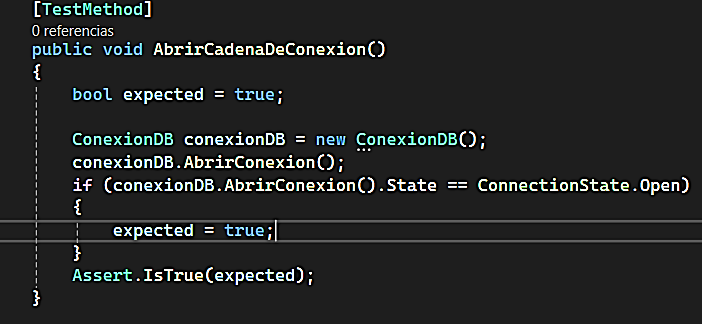
# Documentos de referencia

|  |
| --- |
| DOCUMENTO VERSIÓN |
| Documento de sistema |
| Integración de los módulos |
| Documento de plan de pruebas |
| Especificación de requisitos |

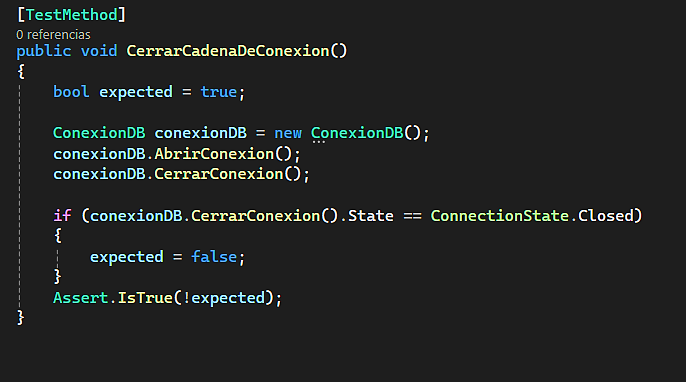
# Estrategia de pruebas

## Pruebas funcionales

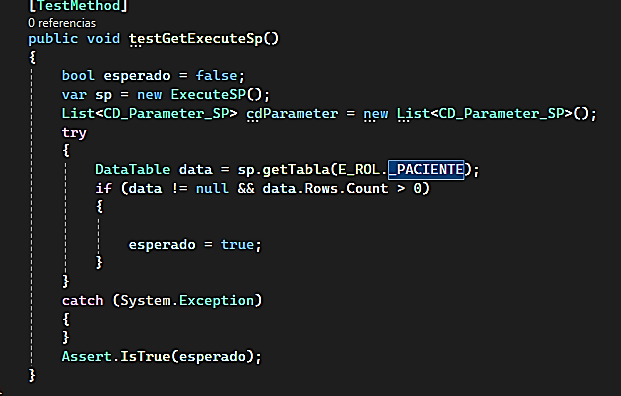
|  |  |
| --- | --- |
| Código | CP-001 |
| Caso de prueba | **Testeo de la puerta de conexión (Abierto)** |
| Responsable | **Tester** |
| Descripción de la prueba | Ejecutar la conexión con la base de datos encargada de gestionar los datos del sistema |
| Requisito previo | Tener configurado los datos de entradas en el sistema |
| Resultado esperado | Conexión abierta y lista para la entrada de más datos en caso de mantenerse |
| Resultado obtenido | Se ha abierto la conexión |
| Estado | Exitoso |
| Observaciones | Ninguna |

**

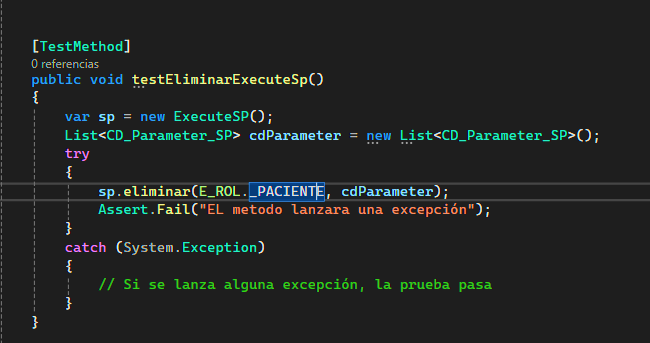
|  |  |
| --- | --- |
| *Código* | *CP-002* |
| Caso de prueba | **Testeo de la puerta de conexión (cerrada)** |
| Responsable | **Tester** |
| Descripción de la prueba | Esta prueba tiene el rol de autentificar el cierre correcto de la cadena de conexión, sin esto los recursos del sistema están expuestos y no se gestionan adecuadamente |
| Requisito previo | Tener correctamente establecida la cadena de conexión  Mantener la conexión previa abierta |
| Resultado esperado | Cierre correcto de la conexión |
| Resultado obtenido | Cierre de la conexión |
| Estado | Exitoso |
| Observaciones | Ninguna |

**

|  |  |
| --- | --- |
| *Código* | *CP-003* |
| Caso de prueba | **Recuperar datos de la base de datos** |
| Responsable | **Tester** |
| Descripción de la prueba | Ser requiere obtener los datos de la tabla prueba, todos los datos tienen que llegar desde la base de datos |
| Requisito previo | Tener registros en la tabla paciente |
| Resultado esperado | Datos de la tabla paciente |
| Resultado obtenido | Datos de la tabla paciente |
| Estado | Exitoso |
| Observaciones | Ninguna |

**

|  |  |
| --- | --- |
| *Código* | *CP-004* |
| Caso de prueba | **Falla al eliminar paciente** |
| Responsable | **Tester** |
| Descripción de la prueba | Al tener pacientes registrados, se tienen que cumplir una serie de parámetros localizados para que estos sean eliminados correctamente  Si los parámetros no se cumplen, tiene que arrojar una excepción |
| Requisito previo | Tener registros en la tabla paciente |
| Resultado esperado | Una excepción en la eliminación de los pacientes |
| Resultado obtenido | Una excepción en la eliminación de los pacientes |
| Estado | Exitoso |
| Observaciones | Ninguna |

**

# Criterios de entrada y de salida

Los criterios de entrada y salidas están gestionados en los datos y los que estos receptan para ser considerado de valor.

## Criterio de entrada del plan de pruebas

Es importante verificar que todas las herramientas necesarias para las pruebas estén instaladas y configuradas correctamente. Esto incluye no solo herramientas de desarrollo, sino también herramientas de automatización de pruebas, seguimiento de defectos y cualquier otro software necesario para el proceso de prueba configurado para garantizar un proceso de prueba fluido y facilitar la identificación y el seguimiento de problemas en el sistema de gestión de citas médicas.

Además, un entorno de pruebas adecuado es fundamental para realizar pruebas eficaces. En nuestro caso utilizamos herramientas básicas como un entorno de, una librería de pruebas, una biblioteca y un gestor de bases de datos. Estas herramientas son cruciales para gestionar eficazmente el repositorio y realizar pruebas. Los entornos de desarrollo como Visual Studio facilitan la creación y depuración de código, mientras que SQL Client y Microsoft SQL Server le permiten administrar de manera eficiente la base de datos de su sistema de citas médicas. Estas herramientas garantizan pruebas en condiciones realistas y con las mismas herramientas utilizadas en el entorno de producción.

Este programa de pruebas es fundamental para descubrir errores, omisiones y problemas potenciales en nuestro sistema de gestión de citas médicas. Al revisar el código, nuestro equipo puede identificar áreas potenciales de mejora, garantizar que cumpla con los estándares de codificación y confirmar que cumpla con los requisitos del proyecto, las salidas de los datos tienen un papel fundamental, tanto como la entrada ya que con estos se verifica la totalidad del funcionamiento de una integración del sistema o de un subsistema.

## Criterio de aceptación del plan de pruebas

Todas las pruebas tienen que tener un 100% de efectividad, tanto las de control de excepciones como las que se encargan de realizar las validaciones de que los datos hayan sigo entregados y gestionados adecuadamente, estos parámetros se basan en las diferentes respuestas que los diferentes test pueden integrar y dar.

## Criterio de suspensión y reanudación

Dentro del sistema se presentaron diferentes inconvenientes tanto para los programadores que gestionaban el sistema como para los usuarios encargados de la gestión del uso de la aplicación y sus derivados.

Está en nuestro compromiso gestionar cada usa de estas excepciones y reanudar las operaciones que respectan al manejo de la aplicación de forma organizada para que esto no afecte a las experiencias de los usuarios y a los diferentes sistemas integrados dentro de los módulos realizados por los programadores

# Entregables

Los entregables es el proyecto que está en el un repositorio público en GitHud

## Protocolo de pruebas

Las pruebas se realizan de forma secuencial y sistematizada en el entorno correspondiente

## Informe de pruebas

El informe de pruebas contendrá la evaluación de resultados con los siguientes puntos:

* Se ejecutaron 3 pruebas
* Acciones sugeridas: Ninguna.
* Análisis de Resultados: Todas las pruebas fueron satisfactorias.

## Reporte de observaciones

Se presentaron diferentes problemas durante la ejecución de las diferentes pruebas, algunos casos fueron exitosos, pero se necesitó mucha refactorización y conocimientos del framework y de nuestro sistema para llevarlas acabo

# 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 |

## Herramientas de apoyo para la ejecución de pruebas

En la siguiente tabla describimos los sistemas y software que se usaron para crear y ejecutar las pruebas unitarias

|  |  |  |
| --- | --- | --- |
| **Elemento de software** | **Versión** | **Tipo** |
| C# | 8 | Lenguaje de programación para escribir las pruebas unitarias |
| Word | 16 | Herramienta para guiarse en el entorno de las pruebas unitarias |
| Notion | Actual | Gestión de las pruebas unitarias |

# Riesgos, dependencias, suposiciones y restricciones

## Riesgos

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Estrategia de mitigación** | **Contingencia** |
| Falta de personal | Mejorar la distribución del equipo y repartir los asentamientos de otra forma | Compartir la gestión del proyecto entre desarrolladores |
| Retraso en las secciones de sistematización | Corregir los tiempos de asignación para que todos los colaboradores gestionen el proyecto de forma adecuada | Gestionar mejor el personal de desarrollo |
| Falta del tiempo correspondido | Tratar de gestionar el tiempo de forma adecuada y elaborar un plan de estimación de tiempo correctas | Asignar recursos de forma equivalente |

## Dependencias

|  |  |
| --- | --- |
| **Dependencia con otras áreas** | **Impacto de la dependencia** |
| Interdependencia entre proyectos. | bajo |
| Accesos (Permisos) a otros sistemas. | Medio |
| Comunicación con sistemas externos. | Medio |
| Comunicación con sistemas internos. | Alto |

## Suposiciones

|  |  |
| --- | --- |
| **Suposición para probar** | **Impacto de suposición incorrecta** |
| Las pruebas terminan de ejecutarse con éxito en su debido contexto | * Retraso en el cronograma de las pruebas * Mala integración con el sistema |
| Cumplimento de los requerimientos en las partes esenciales eficiente | * Se tienen que refactorizar las integraciones de forma completa * Retraso en el cronograma de pruebas |
| Datos correctos en el sistema | * No se logra la correcta integración de los datos en sistema teniendo impacto en la base de datos |

## Restricciones

|  |  |
| --- | --- |
| **Restricciones** | **Impacto de la restricción** |
| Las pruebas tienen que ejecutar en el tiempo correspondientes | * Iniciar a tiempo el paso a producción de la solución * Incumplimiento en la entrega del proyecto. |
| La realización de las pruebas tiene que darse en entornos controlados, su número será de 4 | * Pruebas insuficientes para el sistema |
| Uso de elementos del .NET para integrar las pruebas | * Mayor tiempo de espera para realizar las pruebas |

1. **Aprobación**

|  |  |  |
| --- | --- | --- |
| Integrantes | Contactos | Cargos |
| Tomalá González Félix. | [felix.tomalag@ug.edu.ec](mailto:felix.tomalag@ug.edu.ec) | Coordinador |
| Malavé Briones José. | [jose.malavebri@ug.edu.ec](mailto:jo%73e%2e%6d%61la%76%65br%69@%75g%2e%65%64%75.%65%63) | Administrador de BD |
| Falconi Villamar Marcos. | [marcos.falconivil@ug.edu.ec](mailto:%6da%72c%6f%73%2e%66%61%6cco%6e%69%76%69l@u%67%2e%65d%75.e%63) | Programador |
| Falquez Pico Steven. | [steven.falquezp@ug.edu.ec](mailto:%73%74%65v%65n.%66a%6c%71u%65%7a%70@%75%67%2e%65%64%75%2e%65%63) | Programador |
| Romero Serrano Jean. | [jean.romeros@ug.edu.ec](mailto:%6a%65%61%6e%2e%72%6fme%72%6f%73@%75g%2e%65%64%75%2e%65%63) | Programador |

|  |  |
| --- | --- |
| Elaborado por: | Revisado por: |
| MALAVE BRIONES JOSE  ADMISTRADOR BASE DATOS | FALCONI VILLAMAR MARCOS  FALQUEZ PICO STEVEN  ROMERO SERRANO JEAN  **PROGRAMADORES** |
| Aprobado por:  FELIX GONZALES TOMALA  Líder del Proyecto | **Aprobado por:**  ING. FRANKLIN PARRALES  **DOCENTE - TUTOR** |