

 FPUNA	 CNC Centro Nacional de Computación	 UNIVERSIDAD NACIONAL DE ASUNCIÓN	Plan de Pruebas	Producto: SIGH
				Versión: 1.0
				Página 1 de 14
Cliente: FCM-UNA				

PLAN DE PRUEBAS **SISTEMA DE GESTIÓN HOSPITALARIA**

Tabla de contenido

1. INTRODUCCIÓN
 - 1.1. PROPÓSITO
 - 1.2. ALCANCE
 - 1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS
 - 1.4. REFERENCIAS
2. ESTRATEGIA DE PRUEBAS
 - 2.1. PLANIFICACIÓN
 - 2.2. DISEÑO DE LAS PRUEBAS
 - 2.3. TIPOS DE PRUEBAS
 - 2.3.1. PRUEBAS UNITARIAS
 - 2.3.2. PRUEBAS DE INTEGRACIÓN
 - 2.3.3. PRUEBAS DE INTEGRIDAD DE LOS DATOS Y LA BASE DE DATOS
 - 2.3.4. PRUEBAS FUNCIONALES
 - 2.3.5. PRUEBAS DE USABILIDAD
 - 2.3.6. PRUEBAS DE SEGURIDAD
3. REPORTE Y SEGUIMIENTO DE LAS PRUEBAS
 - 3.2. HERRAMIENTAS DE REPORTES Y CONTROL DE INCIDENCIAS
 - 3.3. ADMINISTRACIÓN DE VERSIONES
4. ANEXOS
 - 4.1. CASOS DE PRUEBAS
 - 4.1.1. DEFINICIÓN DE CRITERIOS PARA LOS CASOS DE PRUEBAS
 - 4.1.2. LISTA DE CHEQUEO CASOS DE PRUEBAS FUNCIONALES
 - 5.4. INFORME DE PRUEBAS
 - 5.5. PROCEDIMIENTO PARA REPORTE DE INCIDENCIAS DE PRUEBAS

	Plan de Pruebas	Producto: SIGH
		Versión: 1.0
		Cliente: FCM-UNA
		Página 3 de 14

INTRODUCCIÓN

1.1. PROPÓSITO

Este documento tiene como propósito establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación del plan de pruebas; incluye responsabilidades de cada una de las tareas, los recursos y los pre-requisitos que deben ser considerados en el esfuerzo de las pruebas, en la búsqueda de garantizar el cumplimiento de los requerimientos planteados en el marco del desarrollo del proyecto SIGH.


El documento específicamente:

- Identifica información del proyecto y componentes de software que pueden ser probados
- Los tipos de pruebas a ser implementados
- Recomendaciones y descripciones de estrategias de prueba a ser empleadas.

1.2. ALCANCE

Este documento de PLAN DE PRUEBAS, se convierte en una guía para desarrollar de una forma organizada las diferentes actividades que se realizarán en el proceso del plan de pruebas en el desarrollo del proyecto SIGH.

La metodología de pruebas y este documento de plan de pruebas permitirán al equipo evaluar aspectos como: la funcionalidad, la seguridad, la usabilidad, y realizar un seguimiento de las pruebas realizadas, así como el control de los resultados.

	Plan de Pruebas	Producto: SIGH
		Versión: 1.0
	Cliente: FCM-UNA	Página 4 de 14

1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

El plan de prueba: describe todos los métodos que se utilizarán para verificar que el software satisface la especificación del producto y las necesidades del cliente. Incluye los objetivos de calidad, necesidades de recursos, cronograma, asignaciones, métodos, etc.

Casos de prueba: lista los ítems específicos que serán probados y describe los pasos detallados que serán seguidos para verificar el software.

Reporte de pruebas: describen los problemas encontrados al ejecutar los casos de prueba.

Herramientas de pruebas y automatización: documentación de las herramientas empleadas en el proceso de pruebas.

SIGH: Sistema de Gestión Hospitalaria

TIPO DE PRUEBA	DEFINICIONES	FASE DEL PROCESO DE DESARROLLO
UNITARIAS	Unitarias: Permite verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.	DESARROLLO (CONSTRUCCIÓN)
INTEGRACIÓN	Integración: Permite verificar el correcto ensamblaje entre los distintos módulos que componen el sistema desarrollado.	
SISTEMA: Estress Concurrencia, Interfaz de Usuario Rendimiento Seguridad Integridad de las BD Interoperabilidad Desempeño Configuración	<p>Sistema: Estas pruebas buscan diferencias entre la solución desarrollada y los requerimientos, con el fin de identificar errores que se puedan generar entre la especificación funcional y el diseño del sistema.</p> <p>Volumen: Esta prueba somete el software a grandes cantidades de datos para determinar si se alcanzan límites que causen la falla del software</p> <p>Estrés: Valida aquellos volúmenes de datos máximos que resiste el sistema antes de comenzar con errores.</p> <p>Concurrencia: Valida la capacidad del sistema de atender múltiples solicitudes de parte de los usuarios que acceden a un mismo recurso.</p> <p>Interfaz de usuario: Permite verificar que la navegación a través de los elementos que se están probando, reflejen las funciones del negocio y los requerimientos funcionales.</p> <p>Rendimiento: Permite validar si la aplicación cumple los criterios de tiempos de respuesta establecidos.</p> <p>Seguridad: Verifica el cumplimiento de las políticas de seguridad acordadas para el sistema.</p> <p>Integridad de las bases de datos: Consiste en asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.</p> <p>Interoperabilidad: Esta prueba permite verificar todos los artefactos de la solución desarrollada, su arquitectura base, los protocolos de la solución, las interfaces y los módulos del sistema, funcionando en forma conjunta.</p> <p>Desempeño: Este tipo de prueba es un aspecto</p>	DESARROLLO

	<p>fundamental en una aplicación, ya que si ésta no responde en el debido tiempo, se pueden perder clientes, o dañar la imagen ante los usuarios.</p> <p>Configuración: Establece y mantiene la integridad de los productos de software a través del ciclo de vida del proceso del mismo.</p>	
FUNCIONALES	<p>Funcional: La prueba funcional es un proceso que busca encontrar discrepancias entre el programa y la especificación funcional. Se toma como objeto de prueba cada Caso de Uso.</p> <p>Usabilidad: Esta prueba permite encontrar problemas de factores humanos, en relación a la facilidad de uso, la navegación, diseño amigable, aplicando un conjunto de reglas básicas de usabilidad.</p>	<p>SOBRE EL PRIMER RELEASE.</p> <p>CON LAS ITERACIONES NECESARIAS</p>
ACEPTACIÓN	<p>Es la prueba final basada en las especificaciones del usuario o basada en el uso del programa por el usuario final luego de un periodo de tiempo</p>	IMPLANTACIÓN

1.4. REFERENCIAS

Algunos documentos del proyecto SISTEMA DE GESTION HOSPITALARIA, son la base de este documento inicial de plan de pruebas, que a continuación se especifican:

- Alcances del Proyecto
- Documentos de Especificación de Requerimientos.
- Lista de Casos de Usos
- Casos de Usos
- Diagramas de clases

	Plan de Pruebas	Producto: SIGH
		Versión: 1.0
		Cliente: FCM-UNA
		Página 7 de 14

- Documento de la Especificación del Proceso de Desarrollo

2. ESTRATEGIA DE PRUEBAS

2.1. PLANIFICACIÓN


Se planificarán las pruebas específicas a ser aplicadas, para lo cual se incluye la definición de las pruebas, las estrategias, los recursos y las estimaciones de tiempo. Esta planificación deberá realizarse para cada sub-sistema en forma separada.

- Se definen las pruebas a aplicar.
- Se especifican las técnicas a utilizar.
- Se establece el tiempo para la ejecución de cada una de las pruebas.
- Uso de herramientas.
- Criterios de aceptación.
- Recursos involucrados.

Resultado de la planificación:

Se deberán definir los siguientes puntos:

- Cronograma detallado de la ejecución de las pruebas; donde se especifica qué prueba se realiza, cuánto tiempo se estima para su ejecución, recursos a utilizar (humanos y tecnológicos).
- Formatos a utilizar para el diseño de las pruebas.
- Formatos a utilizar para el registro y análisis de los resultados de las pruebas.
- Herramientas a utilizar para la gestión de incidencias.
- Procedimientos para el control de cambios.

	Plan de Pruebas	Producto: SIGH
		Versión: 1.0
	Cliente: FCM-UNA	Página 8 de 14

- Herramientas a utilizar para la ejecución de las pruebas.

2.2. DISEÑO DE LAS PRUEBAS

A. Definición de Casos de Pruebas: En esta sección se especifica el inventario de las pruebas, el cual permitirá:

- Definir y asignar prioridades como; alta, media o baja.
- Establecer un orden de trabajo.
- Estimación del tiempo en probar cada funcionalidad.

B. Resultados de la ejecución de las Pruebas: En este punto se especifican los resultados de las pruebas, identificando cada prueba y la estimación de esfuerzo

- Inventario de pruebas priorizado.
- Estimación de esfuerzo de cada funcionalidad.
- Plazos previstos para la solución
- Criterios de Terminación.

2.3. TIPOS DE PRUEBAS

2.3.1. PRUEBAS UNITARIAS

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado.

Objetivo de la Prueba:	Validar las piezas individuales del software como una unidad independiente.
Estrategia:	Automática
Herramientas requeridas:	JUNIT
Criterios de Terminación:	Ausencia de error

2.3.2. PRUEBAS DE INTEGRACIÓN

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos módulos que componen la solución una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces internas y externas, que cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

En esta prueba se comprueba la compatibilidad y funcionalidad de los interfaces entre las distintas ‘partes’ que componen el desarrollo de la solución. Estas partes pueden ser módulos, aplicaciones individuales, es decir esta prueba válida la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta, teniendo en cuenta los siguientes temas técnicos:

- El funcionamiento integrado de módulos inter-dependientes debe estar libre de errores
- Probar todas las dependencias entre módulos

Objetivo de la Prueba:	Validar la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta
Estrategia:	Pruebas de Integración Incremental Ascendente Automática
Herramientas requeridas:	JUNIT ?, JMETER
Criterios de Terminación:	Ausencia de error

2.3.3. PRUEBAS DE OPTIMIZACIÓN DE CÓDIGO

	Plan de Pruebas		Producto: SIGH
			Versión: 1.0
	Cliente: FCM-UNA	Página 10 de 14	


Objetivo de Prueba:	Controlar en forma automatizada la calidad del código desarrollado, en base a una serie de reglas predefinidas, de forma a optimizar las tareas de control de calidad manual
Técnica:	<ul style="list-style-type: none"> - Examinar el código del proyecto con la herramienta definida, para cuantificar los errores debidamente tipificados. - Cada desarrollador se ocupará de examinar su código, de modo a que se integre al proyecto, libre de errores.
Herramientas requeridas	SONAR
Criterios de Terminación:	Cantidad de errores minimizados, según los criterios definidos.

2.3.4. PRUEBAS DE INTEGRIDAD DE LOS DATOS Y LA BASE DE DATOS

Objetivo de Prueba:	Verificar la base de datos y la integridad de los datos cargados desde la aplicación, así como asegurar que la estructura sea la correcta, según el modelo.
Técnica:	Examinar la base de datos para asegurar que los datos han sido actualizados según lo previsto, todos los acontecimientos de la base de datos ocurridos correctamente, o repase los datos devueltos para asegurarse de que los datos correctos fueron recuperados por las razones correctas
Herramientas requeridas	<p>La técnica requiere las siguientes herramientas:</p> <ul style="list-style-type: none"> • Herramienta de automatización del script de prueba • Herramientas de recuperación y respaldo • Herramientas y utilidades SQL de base de datos • Herramientas de generación de datos.
Criterios de Terminación:	Todos los métodos y procesos de acceso de base de datos funcionan según lo diseñado y sin ninguna corrupción de los datos.

2.3.5. PRUEBAS FUNCIONALES

Objetivo de Prueba:	Asegurar el trabajo apropiado de los requisitos funcionales, tomando como objetos de prueba cada uno de los casos de uso.
---------------------	---

	Plan de Pruebas	Producto: SIGH
		Versión: 1.0
	Cliente: FCM-UNA	Página 11 de 14

	Incluyendo la navegación, entrada de datos, edición, el borrado, procesamiento y obtención de resultados.
Estrategia:	Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente: <ul style="list-style-type: none"> Los resultados esperados ocurren cuando se usan datos validos. Se despliegan mensajes de error cuando se usan datos inválidos. Cada regla de negocio es propiamente aplicada. En el anexo 1. se especifica un check-list para los casos de prueba
Herramientas requeridas	Formato de casos de prueba funcionales
Criterios de Terminación:	Ausencia de errores
Observaciones:	Para el reporte de incidencias se utilizará el Redmine, para lo cual se ha creado un proyecto de Reporte de Errores para cada sub-sistema.

2.3.6. PRUEBAS DE INTERFAZ DE USUARIO

La Prueba de interfaz del Usuario (UI) verifica la interacción de un usuario con el software. Además, estas pruebas deben asegurar, que los objetos dentro de la interfaz funcionan según lo esperado y se conforma con los estándares relacionados a la administración de la salud y a las reglas definidas de Usabilidad

Objetivo de Prueba:	El objetivo de la prueba de UI es asegurarse de que la interfaz del usuario provee al usuario el acceso y la navegación apropiados con las funciones del objetivo de prueba
Estrategia:	Aplicar el Plan de Prueba de Aceptación: Utilización del sistema por parte de los usuarios, a partir de un servidor de Homologación que tiene la función de que los usuarios a partir del uso del sistema; nos permita evaluar la utilidad y usabilidad a partir.
Herramientas	Aplicación instalada en el servidor de Homologación

requeridas	Software RT, para gestión de ticket, que permitirá a los usuarios enviar incidencias en relación al uso del sistema, que requiera una mejora o corrección.
Criterios de Terminación:	Evaluaciones iteradas que permita concluir que el sistema está listo para producción como ser: - Ausencia de incidencias de errores por parte del usuario
Observaciones:	Los tickets generados por los usuarios serán atendidos por los analistas responsables del sistema, generando incidencias de corrección de errores.

COMPORTAMIENTO DE VENTANAS y GRILLAS

- Se abren apropiadamente basados en los botones, o desde el menú
- Todos los contenidos dentro de la ventana o grilla están correctos
- Se regenera apropiadamente cuando es sobreescrita y después de re-llamada
- Están todas las funciones habilitadas correctamente y operacionales
- Están disponibles y se despliegan apropiadamente los menús, barras, diálogos, botones, mensajes, íconos y otros controles
- Cuando se despliegan múltiples ventanas, se representan apropiadamente sus nombre, de forma que permite identificarlo fácilmente

2.3.7. PRUEBAS DE SEGURIDAD Y AUDITORÍA

La prueba de seguridad y de control de acceso se enfoca en dos áreas clave de seguridad:

- Seguridad a nivel aplicación, incluyendo el acceso a los datos o a las funcionalidades del negocio
- Seguridad a nivel sistema, incluyendo el registro en el acceso remoto al sistema.

La seguridad a nivel aplicación se asegura de que, basada en la seguridad deseada, restringe a los actores de las funciones específicas o casos de uso, o los limita en los datos que están disponibles para ellos. Por ejemplo, para cada uno se puede permitir incorporar datos y para crear nuevas cuentas, pero solamente los administradores pueden borrarlos.

La seguridad a nivel sistema se asegura de que solamente esos usuarios que se les concede acceso al sistema sean capaces de tener acceso a las aplicaciones

Objetivo de Prueba:	<ul style="list-style-type: none"> ▪ Seguridad a nivel aplicación: Verifique que un actor pueda tener acceso solamente a esas funciones o datos para los cuales su tipo de usuario tenga permisos proporcionados. ▪ Seguridad a nivel sistema: Verifique que solamente permitan a esos actores el acceso al sistema y a las aplicaciones que estén permitidos para cada rol
Técnica:	<ul style="list-style-type: none"> ▪ Seguridad a nivel aplicación: <ul style="list-style-type: none"> - Identifique y enumere cada tipo de usuario y los permisos - Crear las pruebas para cada tipo del usuario y verificar comportamiento según los permisos definidos para ese rol
Criterios de Terminación:	Comportamiento correcto acorde a roles y permisos

4 CASOS DE PRUEBA

4.1 LISTA DE CHEQUEO CASOS DE PRUEBAS FUNCIONALES

La estrategia utilizada está basada en pruebas de caja negra, que se realizan a nivel de la interfaz, sin revisar los detalles internos ni la estructura lógica. Con el fin de garantizar que los casos de prueba contemplen todos los escenarios a probar para cada caso de uso; en su construcción deberá tenerse en cuenta la siguiente lista de chequeo.

Cada conjunto de casos de prueba para cada caso de uso deberá contemplar:

ELEMENTO DEL CASO DE USO	CASO DE PRUEBA
Datos de entrada	1. Verificar que los datos de entrada cumplan con: <ul style="list-style-type: none"> ▪ Obligatoriedad ▪ Tipo de datos ▪ Longitud ▪ Estructura 2. La entrada de datos alfanumérica se despliega e ingresa al sistema apropiadamente 3. Pruebas de valores frontera. Los datos numéricos,

	de tipo fecha aceptan el rango correcto de datos 4. Se reconoce apropiadamente los datos inválidos? 5. El orden en que se ordenan los atributos es el apropiado 6. <i>Clicks</i> del <i>mouse</i> múltiples o incorrectos dentro de la ventana causan efectos no esperados?
Reglas de Negocio	Validar reglas de negocio que afecten los datos de entrada (Dependencia de datos).
	Validar reglas de negocio que afecten los flujos.
Flujos Alternos	Verificar la ejecución de todos los flujos alternos.
Flujos de Excepción	Verificar la ejecución de todos los flujos de Excepción.
Flujo Básico	Verificar la ejecución del flujo básico.

5. HISTORIA DE CAMBIOS

Fecha	Versión Lógica	Descripción	Autora
11/02/13	1.0	Generación del documento	Gabriela Ruiz Diaz
18/04/13	1.1	Se agregaron las definiciones de las pruebas, tipos de pruebas y herramientas, el check list de pruebas y casos de pruebas	Gabriela Ruiz Diaz
07/10/13	1.2	Se ajustó el ítem correspondiente a las pruebas de aceptación	Gabriela Ruiz Diaz