Software para la interpretación de señales de Electrocardiograma orientado para estudiantes  
Plan de pruebas

Versión <1.0>

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| <dd/mmm/yyyy> | <x.x> | <detalles> | <nombre> |
| 12-9-2015 | 1.0 | Documento de visión. | William Ramírez Ruiz  Jhon Jairo Castañeda. |
| 4-10-2015 | 1.0 | Especificación de requerimientos (E.R) | William Ramírez Ruiz  Jhon Jairo Castañeda. |
| 26-10-2015 | 1.1 | Correcciones documento E.R. | William Ramírez Ruiz  Jhon Jairo Castañeda. |
| 28-10-2015 | 1.0 | Especificación de casos de uso. | William Ramírez Ruiz  Jhon Jairo Castañeda. |
| 20-11-2015 | 1.0 | Documento de arquitectura de software. | William Ramírez Ruiz  Jhon Jairo Castañeda |
| 24-11-2015 | 1.0 | Plan de pruebas | William Ramírez Ruiz  Jhon Jairo Castañeda |

Tabla de contenido

1. Introducción 3

1.1 Propósito 3

1.2 Alcance 3

1.3 Audiencia de este documento 3

1.4 Definiciones y acrónimos 3

1.5 Referencias 3

1.6 Estructura del documento 3

2. Misión de la evaluación y motivación de las pruebas 3

2.1 Antecedentes 3

2.2 Misión de la evaluación 3

2.3 Motivadores de las pruebas 3

3. Elementos objetivo de la prueba 3

4. Lista de pruebas planeadas 3

4.1 Descripción de lo que incluyen las pruebas 3

4.2 Descripción de otros candidatos a ser incluidos potencialmente 3

4.3 Descripción de exclusiones de las pruebas 3

5. Enfoque de las pruebas 3

5.1 Catálogos de ideas iniciales y otras fuentes de referencia 3

5.2 Tipos y técnicas de pruebas 3

5.2.1 Pruebas de integridad de datos y base de datos 3

5.2.2 Pruebas de funciones 3

5.2.3 Pruebas de negocio cíclicas 3

5.2.4 Pruebas de interfaz de usuario 3

5.2.5 Perfiles de desempeño 3

5.2.6 Pruebas de carga 3

5.2.7 Pruebas de stress 3

5.2.8 Prueba de volumen 3

5.2.9 Pruebas de Seguridad y control de acceso 3

5.2.10 Pruebas de fallos y recuperación 3

5.2.11 Pruebas de configuración 3

5.2.12 Pruebas de instalación 3

6. Criterios de entrada y de éxito 3

6.1 Plan de pruebas 3

6.1.1 Criterio de entrada al plan de pruebas 3

6.1.2 Criterio de salida del plan de pruebas 3

6.1.3 Criterio de suspensión y reinicio 3

6.2 Ciclos de pruebas 3

6.2.1 Criterio de entrada al ciclo de pruebas 3

6.2.2 Criterio de salida del ciclo de pruebas 3

6.2.3 Terminación anormal del ciclo de pruebas 3

7. Entregables 3

7.1 Resumen de evaluación de las pruebas 3

7.2 Reporte de cubrimiento de las pruebas 3

7.3 Reportes de la calidad percibida 3

7.4 Log de incidentes y peticiones de cambios 3

7.5 Conjunto de pruebas de humo y scripts de Soporte a las pruebas Scripts 3

7.6 Productos de trabajo adicionales 3

7.6.1 Resultados detallados de las pruebas 3

7.6.2 Scripts de pruebas automáticos adicionales 3

7.6.3 Guías de pruebas 3

7.6.4 Matrices de trazabilidad 3

8. Flujo de pruebas 3

9. Necesidades de ambiente 3

9.1 Hardware base del sistema 3

9.2 Elementos de software base del ambiente de pruebas 3

9.3 Herramientas de productividad y de soporte 3

9.4 Configuraciones del ambiente de pruebas 3

10. Responsabilidades, Recursos y necesidades de entrenamiento 3

10.1 Personas y roles 3

10.2 Necesidades de personal y entrenamiento 3

11. Hitos de iteración 3

12. Riesgos, dependencias, suposiciones y restricciones 3

13. Procedimientos y procesos de gestión 3

13.1 Medición y evaluación de la extension de las pruebas 3

13.2 Evaluación de los entregables de este pan de pruebas 3

13.3 Reporte de problemas, Escalamiento, y resolución de incidentes 3

13.4 Gestión de los ciclos de prueba 3

13.5 Estrategias de trazabilidad 3

13.6 Aprobación y terminación 3

Plan de pruebas

# Introducción

## Propósito

## El propósito del plan de pruebas iteración es recopilar toda la información necesaria para planificar y controlar el esfuerzo de la prueba para una iteración dada. En él se describe el enfoque de las pruebas del software, y es el plan de alto nivel generada y utilizada por los gerentes para dirigir el esfuerzo de la prueba.

## Este plan de pruebas para el < Software para la interpretación de señales de Electrocardiograma orientado para estudiantes> compatible con los siguientes objetivos:

## • Tomar datos mediante la comunicación datos, verificándose que los datos sean acordes con lo esperado.

## • Identificar si los datos están llegando correctamente a la variable generado, esto con el fin de interpretar y analizar el tipo de datos que se están recibiendo y como estos varían en el tiempo.

## • El enfoque de esta prueba ira relacionado con interfaces y contenidos que se generan en la comunicación.

## • Para la prueba será necesario el tener una tarjeta de adquisición o un microcontrolador el cual envié datos por el puerto serial.

## • Botón inicio (se manifiesta error si no se tiene comunicación serial establecida en el puerto COM3), menú desplegable (generación de información visual al presionar una opción del menú).

## Audiencia de este documento

Se espera que este documento tenga como audiencia cualquier persona la cual tenga interés en el proceso de pruebas, de tal manera que su lenguaje pueda ser entendido tanto para el “tester” y para el usuario. Por lo cual el lenguaje a usar sera bastante sencillo y se explicara con cierto grado de detalle algunos aspectos como lógica de funcionalidad y lógica de prueba.

## Definiciones y acrónimos

**Tester:** persona u software encargado del proceso referente a demostrar la ausencia de errores en un software y en su efecto detectarlos y reportarlos.

**Tipo de pruebas:** Al momento de ejecutar una prueba estas se pueden dividir en dos grandes ramificaciones, pruebas funcionales y no funcionales. Las funcionales hacen inferencia a aquellas pruebas que relacionan funcionalidades directas del software tales como el correcto funcionamiento de una clase “toma de datos”. Las no funcionales hacen inferencias a aspectos del software tales como desempeño, seguridad, mantenibilidad, etc.

* **Revisión de la documentación**: Consiste en revisar la calidad y completitud de los documentos insumo y casos de uso para la ejecución de las pruebas.
* ***Pruebas Unitarias:***Se validan las piezas individuales del software como una unidad independiente, bucles, condicionales, etc.
* ***Pruebas de integración:*** Se validara la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta.

## Referencias

Se tomaron como referencia los documentos ubicados en las siguientes direcciones:

* <https://s3.amazonaws.com/piazza-resources/idenwhs5uyd3ds/ih5fs1t836z6dt/Ch8.pdf?AWSAccessKeyId=AKIAJKOQYKAYOBKKVTKQ&Expires=1448168446&Signature=7oW3LvKgnHwtNLaCLxfhKXfQGdE%3D>
* <http://es.slideshare.net/choselin/plan-de-pruebas-15563690>

## Estructura del documento

Inicialmente tendremos que la plataforma a usar para las correspondientes preubas sera Windows 7 u en su efecto versiones superiores haciendo uso de servipacks, de igual manera haciendo uso de eclipse mars debido a que su estructuración se ha hecho sobre esta plataforma.

La realización del testing ira orientado a pruebas unitarias, en donde se tendrán en consideración al ejecutar esta la funcionalidad y resultado segmentadose por clases, esto debido a que sera de mayor practicidad y eficencia al momento de evidenciar un fallo durante la ejecución del software.

Con el fin de aumentar la productividad, se tendrá un diseño para la ejecución de la prueba, generándose una prueba patrón, esta suministrara al encargado de hacer el test con el fin de que tenga una guía con un orden lógico para la ejecución de la prueba. De igual manera se debe tener en consideración el hecho de que un usuario tendrá diversos caminos al momento de ejecutar el software, por lo cual se debe analizar el comportamiento y resultado al tomar dichos caminos.

# Misión de la evaluación y motivación de las pruebas

Al seguir un patrón establecido para la ejecicio de pruebas se lograr analizar de forma acertada como es el funcionamiento lógico del software, permitiendo al tester mayor agilidad y compresión de este al ejecutar la prueba. No obstante la ejecución de la prueba de distintas maneras sera crucial, debido a que deberá tenerse como objetivo en la prueba el lograr analizar y encontrar de manera unitaria los multiples errores u problemas que se encuentre al momento de ejecutar el software.

## Antecedentes

Como se ha evidenciado en documentos anteriores el software tendrá una estructura basada en un control centralizado, en donde apartir de un centro se regresan subsistemas o clases al generarse variables de estado desde dicho centro. Con ello se podrá hacer una relacion de la generación de la variable de estado con la ejecución del proceso que se relaciona con esta, haciéndose más fácil el análisis en caso de presentarse un error, ya que se localizara específicamente el origen del error.

Lo referente a los patrones para realizar la prueba , se tiene que el lenguaje usado para la construcción del software sera plenamente establecido (java) y sera ejecutado en la plataforma oracle mars con un sistema operativo Windows 7 . Por tal razón por tal razón al momento de ejecutar la prueba deberá hacerse bajo esta plataforma y enfatizándose a dicha lógica.

Con respecto a la prueba patrón se deberá considerar algunas condicionales para la ejecución de los procesos, por ejemplo algunos comportamientos, en el caso de este software a evaluar sera el hecho de que se deberá respetar un cierto tiempo de espera al momento de ejecutar la comunicación serial y tomar los datos ya que solo se tomaran muestras de dichos datos. Por ello el establecimiento y definición de una prueba patrón sera crucial para el tester ya que este requerirá de dicha información. De igual manera se espera evitar la generación de errores incesarios al mencionar las condiciones para el funcionamiento del software.

## Misión de la evaluación

Con la ejecución de la prueba y más específicamente la prueba al nivel unitario, es encontrar:

* Errores al ejecutar algún proceso e identificarlo plenamente a nivel unitario.
* Encontrar problemas importantes que involucren o arriesgen la integridad del software.
* Verificar cada requerimiento y caso de uso al ejecutar la prueba estableciendo falencias.
* Encontrar posibles riegos que el software puede manifiestar hacia el usuario final.
* Analizar características no funcionales tales como rendimiento, confiabilidad y seguridad.

## Motivadores de las pruebas

**Errores u problemas técnicos:** Al momento de ejecutar la prueba, se debe tener en consideración que al iniciar la comunicación serial, se debe tener el microprocesador conectado y programado para enviar datos a través de esta comunicación, en caso contrario se presentaran errores al ejecutar la función de comunicación serial en el software.

**Riesgos del proyecto:** Desde cierta perspectiva el proyecto no representara un riesgo desde la perspectiva de software, en con respecto al hardware se deberán tener en consideración ciertos factores tales como la correcta ubicación de electrodos y las condiciones ambientales en la cual se usa este.

**Requisitos funcionales:** lograr una comunicación serial entre el microcontrolador y java, verificándose una correcta transmisión de datos para posterioremente operar estos y manejarlos, por ejemplo para la graficacion. De igual manera proporcionar una interfaz en donde al usuario se le logre suministrar la información planteada para la compresión del software.

**Requisitos no funcionales:** lograr que el software no implique lags al momento de usarse, teniendo en cuenta que anteriormente se ha establecido que en esta versión se tomara una muestra de la señal lo cual requiere un tiempo de espera de 7 seg para el usuario, en donde el software se concentrara meramente en la comunicación y transmisión de datos seriales. De igual manera entre estos requisitos se debe tener un nivel de mantenibilidad, en donde este se valla actualizando continuamente para la correcion de errores. Por lo cual sera misión del tester verificar estos aspectos meticulosamente ya que en cierta medida puede implicar errores en los requisitos funcionales.

**Solucitudes de corrección:** Como se estableció anteriormente las pruebas se realizaran a nivel unitario, por lo cual al manifestarse un error sera necesario el especificar claramente el origen del error y la provenencia de este de una forma especifica, detectando en su efecto directamente el subsistema o caso de uso en el cual se manifestó el error. Por ello sera acertado que el tester tenga a su disposición el diagrama proporcionado en el documento de casos de uso en donde se explican secuencialmente la dependencia entre cada caso de uso.

# Elementos objetivo de la prueba

La siguiente lista manifiesta los “objetos” que serán puestos a pruebas desde la perspectiva de software y hardware, y la forma en que estos serán puestos a prueba:

* Funcionalidad comunicación serial: Se evaluara la forma en que se esta realizando la comunicación serial en java, verificándose la calidad de esta y la correcta transmisión de datos.
* Funcionalidad y lógica de la interfaz: La interfaz en cierta medida controlara la lógica secuencial para la ejecución del software por tanto esta deberá ser objeto de prueba, verificándose que esta haga lo que se menciona en la interfaz y lo ejecute correctamente.
* Verificar la funcionalidad del hardware y los posibles errores que se manifiesten durante la complementancion con el software, debido a que este actuara como un sistema de adquision, este aspecto tendrá un gran nivel de importancia ya que sin ello el software no cumplirá con su función principal.

# Lista de pruebas planeadas

La interfaz realizada contendrá dos menus desplegables “¿Cómo usar? Y”ayuda””, en donde mostrara ciertas opciones, en el caso de como usar, se proporcionara una guía acerca del correcto uso del software y en el caso menú ayuda se suministrara cierta información bien espicificada y segmentada. Para el caso de las pruebas se tendrá tener en consideración el hecho de que al tomar una opciones se suminstre la información que es y que exista una correcta transision entre cada cambio. En este caso el teste tendrá que tomar multiples caminos evidenciando que la información que se suministre corresponda a lo indicado en la interfaz. Se observara según lo anteriore como al generarse una interaccion con estos “menus” se devuelve una acción contenida en una clase.

Dentro de esta interfaz también se tendrá un botón inicio, este se encargara de iniciar mendiante el uso de una librería la comunicación serial entre la trajeta de adquisición (microcontrolador) y java. Este al igual que caso anterior tendrá una lógica de llamamiento de clases a través de interacciones. En este caso el tester deberá observar como al presionar ese botón se inicia la comunicación, como se indico anteriormente este proceso tomara en su totalidad los recursos del software mientras se toma la muestra, de tal modo que el tester deberá corroborar ese tiempo y que ocurre durante el transcurso de ese tiempo y después.

## Descripción de lo que incluyen las pruebas

Esta prueba deberá incluir el generar una señal de prueba en la entrada del microcontrolador, esta señal deberá se de tipo análoga que varie en el tiempo, definiéndose claramente los rangos de variación de esta señal y los datos que envía a través del puerto serial, con ello se pondrá interactura con el botón inicio, evidenciando como se reciben los datos y se trabajan con ellos. La prueba entonces incluirá la necesidad de tener una señal análoga en la entrada del microcontrolador a usar y el generar a través de este una escritura en el puerto serial. Se debe tener en cuenta que al iniciar la comunicación se debe tener conectado el microcontrolador en el puerto, en caso contrario se generara error.

Ademas de ello la prueba incluirá un recorrido por la interfaz, mirando e interactuando con las multiples opciones que este nos ofrece, tales como los menus desplegables para la suministración de información.

## Descripción de otros candidatos a ser incluidos potencialmente

Las areas que se deben considerar con mayor dedicación es la correspondiente a la información que se suministra en la interfaz, ya que debido a las multiples opciones puedan que se generen errores no esperado o establecidos en el plan de pruebas. Tambien sera conveniente el probar lo referente a la comunicación serial debido a las multiples variantes que pueden surgir y que en cierta medida no se han tomado en cuenta.

## Descripción de exclusiones de las pruebas

Durante la planeación de las pruebas se llegaron a puntos de quiebre en donde el sistema evidentemente no funcionara de la forma esperada, debido a su estructuración y funcionamiento, por lo cual se han definido como exclusiones para las pruebas que se le realizara.

Este punto critico se encuentra en el momento en que se inicia la comunicación serial y al mismo tiempo se desea acceder al menú de ayuda, por ello se ha implantado un tiempo de espera para la toma de la muestra.

# Enfoque de las pruebas

Tendremos que el enfoque de las pruebas a realizar son de tipo unitario. Se pondrá a prueba principalmente dos aspectos, por un lado la funcionalidad con respecto a la suministración de información y la transmisión de datos a través del puerto serial.

Tendremos entonces en el primer apartado, que las pruebas a realizara consistirán en la exploración de la interfaz y el accionamiento de menus, los cuales deberán arrojar la información indicada. La prueba a realizar en este caso consistirá básicamente en hacer un recorrido a través de las multiples opciones que se muestran en el menú, esto debido a que en este aspecto la funcionalidad de cada apartado tendrá una lógica muy similar. Por ello sera acertado hacer un recorrido aleatoriamente verificando específicamente como es el efecto de transision entre cada opción y si esta información es acorde a lo que se indica en el menú.

Lo referente al inicio de la comunicación, se evaluara el hecho de que la comunicación sea adecuada y que esta transmita información acorde a lo esperado. En este apartado se ha indicado un caso posible de error, en donde al iniciar la comunicación en java no exista ninguna transmisión de datos en el puerto serial, este sera causante de error ya que si ello no iniciara esta funcionalidad, además del hecho de que al indicar la comunicación se deberá tener en consideración un tiempo minimo de espera, en donde el software se concentrara solamente en la transmisión de datos.

## Catálogos de ideas iniciales y otras fuentes de referencia

Al iniciar el análisis en pruebas se generan una serie de ideas iniciales referentes a este proceso, una de estas es un uso paralelo del proceso iniciar comunicación y consultar información u intentar hacer una transición rápida entre la toma de datos y el menú. Por ello se deberá tener en consideración que durante las pruebas se han generado fuentes referencia donde se especfica meticulosamente casos generadores de errores. Como fuente de referencia se tendrán los códigos generados en las versiones, ubicados en los repositorios, en donde se ubicaran una pequeña especificación con respecto a la lógica de cada clase y constructor, con esto se hara mas fácil la ubicación del error en caso de tenerse que hacer alguna corrección.

## Tipos y técnicas de pruebas

### Pruebas de integridad de datos y base de datos

|  |  |
| --- | --- |
| Objetivo de la técncia: | Identificar claramente en donde se esta generando los errores al hacer el tipo de prueba especificado, esto permitirá mayor facilidad al programador para correguir el error. |
| Técnica: | Como de momento este tipo de software no inlcuira o no sera necesario como tal el uso de una base de datos, tendremos que la técnica a usar se centrara en verificar la integridad de datos, para ello se hara uso de pruebas e tipo unitario en donde se verificaran dos aspectos principalmente, la comunicación de datos y que tan veraz es esta información transmitida. |
| Oracles: | Para lograr que la prueba sea lo suficientemente eficaz, definiremos que durante la ejecución de la prueba es muy probable la generación de ciertos errores en especifico, por ejemplo que al iniciar a comunicación no se pueda ejecutar otra función durante cierto tiempo u que al iniciar la comunicación se genere error por no tener nada en el puerto serial. |
| Herramientas requeridas: | Las herramientas que se usaran serán principalmente los repositorios, el disponer de una tarjeta de adquision que transmita por el puerto serial, disponer de una persona para el testing u en su efecto hacer uso de herramientas tales como JUnit, TestNG u alguna de este tipo. Para este caso se hara uso de test manual en donde una persona verificara los casos de uso en el software, apoyándose en los documentos anteriormente generados para el diseño de este mismo. |
| Criterios de exito: | Esta prueba abarcara todos los métodos y procesos establecidos en el diseño del software, para llegar a este apartado se deberá tener en consideración la lógica estructural generada en la prueba patrón, en donde se espera el software funcione correctamente. |
| Consideraciones especiales: | Al presentarse error se deberá recurrir a los repositorios y analizar si en una versión anterior el caso de uso a analizar estaba funcionando de una forma correcta u fue hasta esta versión que se genero este caso de usos como un añadido. |

### Pruebas de funciones

|  |  |
| --- | --- |
| Technique Objective: | Verificar que cada una de las funcionlidades se este ejecutando según lo acordado y que paralelamente no este generando error en si misma o este influyendo a la generación de error en otras funcionalidades. |
| Technique: | Se explorara de forma meticulosa y desde una persectiva unitaria la generación de errores. Por ello sera necesario el explorar cada funcionalidad y verificar que se este ejecutando de la manera indicada en anteriores documentos, esto viéndose desde la perspectiva de que se este ejucutandose en donde es y de la forma como debería ser. |
| Oracles: | Tener en consideración que al realizar las pruebas de forma unitaria, se debe seguir un orden lógico, especificado en los diagramas de secuencia, ya que en algunos casos algunas funcionalidades dependerán de un proceso anterior. |
| Required Tools: | En este caso sera necesario disponer de diagrama de secuencia para cada caso de uso y tener establecido la lógica que usa el software. Este se realizara de manera manual por lo cual requerirá de un tester y de los documentos en donde se especifica cada caso de uso. |
| Success Criteria: | El los diagramas de secuencia se ha generado un escenario ideal, este ira estrechamente ligado a los criterios de éxito, ya que en este se asume y se verifica que el sistema funcióna correctamente. |
| Special Considerations: | Identificar plenamente los escenarios error y de éxito ya que estos denmarcaran si se debe o no ejecutar una corrección y en donde se debe aplicar esta. |

### Pruebas de negocio cíclicas

|  |  |
| --- | --- |
| Technique Objective: | Verificar el estado del proyecto. |
| Technique: | La técnica consistirá en hacer un análisis de un posible estado del proyecto en cierto tiempo futuro, para error se asumen ciertas posibles acciones, en donde el proyecto pueda cambiar u no. Por ejemplo el hecho de que el software no llegue a ser tan utilizado en cierto tiempo y este deba cerrarse a la inversión y al cambio o en caso contrario, si este llega a ser altamente utilizado , el rango de tiempo en el cual se deban generar actualizaciones del software y la calidad minima que se debe tener al momento de generar versiones nueva de este mismo. |
| Oracles: | Se debe ser lo mas critico posible al generar estados futuros, por lo cual se debería hacer un estudio previo acerca del nivel de aceptación que actualmente posee el software y como este podría incrementar o estancarse en el tiempo. |
| Required Tools: | -Estudios estadísticos de comportamiento.  -Analisis de estados actuales y futuros.  -Registro de las condiciones del software, como se deberá hacer una prueba de negocio se deberá tener en cuenta el estado del proyecto durante cierto periodo de tiempo para preveer un comportamiento futuro. |
| Success Criteria: | Los criterios de éxito en este caso serán asumido para un periodo de tiempo futuro, por lo cual esto ira muy ligado al registro y análisis que se le realizara al comportamiento de software durante cierto periodo de tiempo. |
| Special Considerations: | Se deberá tener en consideración algunos aspectos tales como estudios en el tiempo y análisis de información, lo cual se deberá documentar en el proceso de actualización y desarrollo del software. |

### Pruebas de interfaz de usuario

|  |  |
| --- | --- |
| Technique Objective: | Verificar como sera la lógica y funcionalidad de la estructura final que se le mostrara al usuario (interfaz). |
| Technique: | En este apartado la prueba se tornara a un efasis integrado. Esto es debido al hecho de que en la interfaz se tendrá unas relaciones entre casos de usos, ya que uno puede estar muy ligado en la ejecución de otro. |
| Oracles: | Se debe tener en cuenta que se debe generar una ruta lógica y secuencial para la ejecución de procesos desde la interfaz, por lo cual se debe tener en cuenta como el usuario puede tomar varias rutas y como estas rutas desde cierta perspectivas llegaran a algo plenamente definido en los casos de uso |
| Required Tools: | Est técnica se puede ejecutar mediante el uso de software especializado para el testeo el cual seguirá una lógica preestablecida por los casos de uso. De igual manera se puede hacer uso de tests manuales. |
| Success Criteria: | Los criterios de éxito están ligados a la ejecución del test según lo establecido en el digarama secuencial para los casos de uso. |
| Special Considerations: | Se debe tener en especial consideración el hecho de que el usuario puede tomar cualquier tipo de ruta, y que la prueba como tal no sera veraz si no se toman las rutas mas intuitivas que puede tomar el usuario. |

### Perfiles de desempeño

|  |  |
| --- | --- |
| Technique Objective: | Anticipar como sera el posible rendimiento y desempeño del software ante ciertos escenarios. |
| Technique: | Para la ejecución de este tipo de pruebas por lo general se hace uso de test llamados “benchmark”, pero para este caso lo mas acertado sera el hecho de analizar la duración que tiene cada proceso, y que ocurre durante este tipo, en este caso el software presentara una falencia ya que requerirá de cierto tiempo para un muestreo de datos, en donde el usuario tendrá que esperar sin ejecutar algún otro recurso. |
| Oracles: | En este aspecto la observación sera algo crucial para el análisis de desempeño, pero dicho proceso deberá ser interpretado de una una lógica unitaria con el fin de detectar en que apartado esta el déficit de desempeño |
| Required Tools: | En este apartado se debería a recurrir a herramientas relacionadas con el análisis de consumo de recursos de la computadora, generado por el software , pero en este caso el uso de herramientas relacionada con software no estará a disponibilad debido a factores de conocimiento, por ello se deberar recurrir al análisis de por observacion |
| Success Criteria: | Debera tomarse en consideración que aquel que haga uso de este software podrá enfocarse en la navegación aleatoria por este, asi que deberá hallarse e informarse puntos de quiebre en donde el software manifestara defectos de desempeño, asi el criterio de éxito sera mas grande en proporción de procentaje. |
| Special Considerations: | Se deberá tener en cuenta que para un buen diseño se debe tener una buena línea base, asi que el correcto diseño de la línea base permitirá una buena generación de actualización y correcciones de errores por ejemplo errores de desempeño. |

### Pruebas de carga

|  |  |
| --- | --- |
| Technique Objective: | Crear diversersos escenarios en donde se vea cual es el limite de carga para el software |
| Technique: | La técnica debería centrarse en la ejecucion de escenarios en donde el software se le cargen multiples procesos de diversa proporción, pero en el caso de este software debido a la lógica y cantidad de requerimientos, la inclusión de cargas sera algo de menor afectación, de igual manera se ha indicado también casos en donde si se carga un proceso de inicio de comunicación de datos y de solicitud de información, el proceso de comunicación no permitirá la ejecucion de otro proceso a este que pase el tiempo de ejecucion de este mismo. |
| Oracles: | En est caso se debe ejecutar una tarea de observación, verificando cambios según la ejecucion de ciertos planes de carga y como estos varian según la cantidad u tamaño de las cargas aplicadas en el software. |
| Required Tools: | En este una herramienta a usar sera el registro de comportamientos hallados en la observación para cada plan de carga, de tal manera que se ubiquen comportamientos específicos para la aplicación de cada plan de carga. |
| Success Criteria: | Para que el programa funcione u se ejecute con éxito se deberá hacer un análisis de la lógica de funcionamiento de este mismo, analizando que recurso tendrá mas afectación al momento de suministrarle una carga. |
| Special Considerations: | Tener en consideración que el software evidentemente tendrá problemas de carga si se fuerza específicamente procesos que impliquen transmisión de datos, ya que desde cierta perspectiva al comunicarse datos se están haciendo multiples procesos paralelamente. |

### Pruebas de stress

|  |  |
| --- | --- |
| Technique Objective: | Determinar cual es el tope de uso de memoria usada para la ejecucion del software, determinando los requerimientos minimos para la ejecucion de este mismo. |
| Technique: | Someter al software a condiciones de uso extrema, evaluando la cantidad de uso de memoria en (“Ram”) y de recursos graficos, en nuestro caso el uso de recursos no sera tan elevdado debido a los servicios que prestara el software de momento. Para ello se deberá someter a condiciones especificas (de minimo y máximos requerimiento) en donde se determine fornteras de uso de memoria. |
| Oracles: | Inicialmente se ejecutara el software y se creara una secuencia para los casos de uso a ejecutar, con esto tendremos diversos caminos para llevar a cabo una función. Paralelamente mediante la ayuda de un software u con la misma maquina se procede a determinar los rangos de uso de memoria física al momento de ejecutar el software ante las condiciones anteriormente mencionadas. |
| Required Tools: | Dentro de las herramientas a usar se tendrá los registros generados de rendimientos producidos en el control interno de la maquina. La maquina en donde se ejecutara el software por lo general trae de forma visual, un control de rendimiento en donde se indica específicamente de cuanto es el uso de la memoria (Ram) al tener un proceso abierto en este caso el software.  Adicional a ello existe gran variedad de programas destinados a esta actividad por lo cual también se tendrá en consideración el uso de estos. |
| Success Criteria: | Los criterios de éxito en este caso iran ligados a los recursos disponibles vs los recursos requeridos, por lo cual se deberá tener una buena delimitación de las fronteras de funcionamiento teniendo en cuenta rangos en el cual el software trabaja adecuadamente, por rangos se refiere a rangos de recursos minimos que deberá tener la maquina que ejecuta el software. |
| Special Considerations: | Se deberá definir el concepto de correcto funcionamiento, ya que el software puede funcionar en ciertos recursos minimos pero puede que no funcione adecuadamente. En nuestro caso los recursos usados no serán tan elevados. |

### Prueba de volumen

|  |  |
| --- | --- |
| Technique Objective: | Determinar el rango máximo de datos que se podrán manejar en el software y el tiempo máximo de ejecucion de estos |
| Technique: | Se deberá someter el software a un tiempo constante de uso, en donde se transmita cierto rango de datos ya sea en gran cantidad o en minima cantidad esto con el fin de hacer una relacion cantidad de datos vs máximo de tiempo de trabajo. Para ello ser indispensable de analizar y observar constantemente el comportamiento del software en la line de tiempo de ejecucion de este mismo. |
| Oracles: | Con respecto a la prueba, se deberá iniciar un análisis del software ante un funcionamiento normal y verificando el rango máximo de funcionamiento de este, para ello sera necesario someterlo a un uso constante y realizar un promedio de errores que se generen durante este proceso. Posterior a ellos se someterá al software a un carga de procesos mas pesada y se verificara cual es el tiempo máximo de funcionamiento de este sin que se presente anomalías. |
| Required Tools: | Para este proceso sera necesario llevar un registro en donde se informe acerca de la prueba realizada informadose los errores manifestados durante la prueba. Para ello se puede hacer uso de software, en nuestro caso deberá ser mediante observación continua hecha por un tester. |
| Success Criteria: | Se asumirá que el cirterio de éxito se asumirá cuando el software funcione continuamente durante un tiempo establecido sin que se manifieste error alguno. |
| Special Considerations: | Tener en consideración que se deben respectar los recursos minimos para el funcionamiento del software en caso contrario el factor de error sera mayor durante las pruebas. |

### Pruebas de Seguridad y control de acceso

|  |  |
| --- | --- |
| Technique Objective: | Verificar algunos aspectos de seguridad, en nuestro caso sera referente a la correcta protección hacia el usuario y la veracidad de la información obtenida a través de a comunicación. |
| Technique: | Se deberá hacer un análisis del factor de seguridad con respecto al aislamiento eléctrico, ya que el hardware a usar sera un sistema de adquisición en donde se envía y trasmiten señales eléctricas variantes en el tiempo, paralelamente se deberá asegurar que los datos obtenidos a través del hardware llegue de la forma correcta al software analizando factores de desviación y tiempos de muestreo. |
| Oracles: | En este caso se deberá aquirir una señal ejemplo y transmitir a tarves del puerto serial, como el sistema de adquisición se compontra de un circuito muy bien especificado y diseñado, tanto que si se manifiestase un error el tester no sabria si este se presentase en el software o hardware, se deberá hacer aplicando un ejemplo a menos a escala tal como la variación de un voltaje a través de un potenciómetro, esto con el fin del que hace el test logre determinar el origen de un error si se llegase a presentar |
| Required Tools: | Sistemas de adquisición, junto con una señal análoga variante en el tiempo, con esto se podrá evidenciar si la variación de los datos corresponde a las variables generadas en el software. |
| Success Criteria: | Se definirá que el programa funciona correctamente cuando se evidencie que los datos transmtidos desde el microcontrolador son los obtenidos en las variables generadas en el software. |
| Special Considerations: | Tener en consideración que al momento de realizar la prueba de esta manera se deberna definir puertos de comunicación, en este caso se deberá establecer en el Com3 en caso contrario el software generara error y no transmitirá la información generada al software. |

### Pruebas de fallos y recuperación

|  |  |
| --- | --- |
| Technique Objective: | La validación de los errores para posteriormente ser corregidos por métodos de recuperación al usar una línea base u corrigiendo el error directamanete |
| Technique: | Se generaran secuencias en las cuales se sabe que existe o existía un error para evidenciar si este surge y de que manera surge. Una vez se especifica el origen y causa de este se procede a ejecutar un proceso de corrección o en su efecto de recuperación, en donde se mira como el software en su caso se puede recuperar por si mismo u llamando versiones anteriores del software, por tal razón es crucial el uso de base de datos que contenga las versiones generadas. |
| Oracles: | Se deberá someter el software ante condiciones de error, estas se habran delimitado anteriormente al evidenciarse errores u se podrán delimitar al evidencia nuevos errores. En caso de ser un estado critico se procederá a un proceso de recuperación, tomando como base la línea base generada durante el diseño del software. |
| Required Tools: | -Base de datos.  -Repositorios.  -Registro de casos de uso que generan error. |
| Success Criteria: | Tendremos qu el cirterio de éxito se relacionara con la solvencion de errores para ejecutar la prueba. Con esto tendremos que al ejecutar las condiciones en donde antes se generaban error, se deberá ejecutar el proceso correctamente. |
| Special Considerations: | Generar un registro de ambientes generadores de error para realizar el respectivo análisis. |

### Pruebas de configuración

|  |  |
| --- | --- |
| Technique Objective: | Verificar el correcto funcionamiento de hardware y la delimitación de este. |
| Technique: | Se procederá a verificar que componentes de hardware fueron usadas junto con el software y posteriormente se procederá a verificar que estos componentes estén funcionando correctamente. |
| Oracles: | Para la verificación el correcto funcionamiento del hardware, se deberá generar una prueba practica, ya que el tester no se involucrara mucho con hardware especializado, en nuestro caso sera generar una señal referencia para emular la señal de ecg y verificar como es la información transmitida. |
| Required Tools: | -Tarjeta de adquisición.  -Generador de señal.  -Analisis de la señal evidenciando la comunicación en el puerto serial. |
| Success Criteria: | Como se aplicara una prueba generando una señal, el criterio de éxito se relacionara con la correcta transmisión de esa señal a través del puerto serial, verificando los datos enviados y la velocidad de transmisión. |
| Special Considerations: | Tener en cuenta que la comunicación de datos sera en 8 bits y tendrá que ser recibida en una variable de tipo int. |

### Pruebas de instalación

|  |  |
| --- | --- |
| Technique Objective: | Verificar la plataforma en el cual el software funcionara y el lenguaje usado en este, también se definirá como sera el procedo de udapte para el software. |
| Technique: | Verificar la funcionalidad del software en ciertas plataformas u S.O, definiendo esta como compatible, no funcional o con errores de funcionamiento, paralelamente se tendrá que definir la funcionalidad del software en cada versión. |
| Oracles: | Se deberá definir las plataformas en las cuales funciona el software, delimitando el lenguaje de programación esto con el fin de detectar y corregir errores en caso en que se manifiesten estos. |
| Required Tools: | Disponer de diversas plataformas para probarar la funcionalidad del software en cada una de esta, con esto se definirán exceptuaciones. |
| Success Criteria: | Se generaran registros de compotabilidad en donde se indique las especificaciones de diseño para el software |
| Special Considerations: | Tener en cuenta el enfoque de programación que tendrá el software ya que este definirá en gran medida el factor de compatibilidad. |

# Criterios de entrada y de éxito

## Plan de pruebas

### Criterio de entrada al plan de pruebas

Para inicar la prueba se debe disponer de un microcontrolador que tome datos análogos y los envie a través del puerto serial, una computadora con Windows 7 con java 1.8.0.

### Criterio de salida del plan de pruebas

Definiremos que el software saldré del plan de pruebas satisfactoriamente cuando los casos de usos planteados superen márgenes de error ante un uso u secuencia especificada. Tambien se puede definir que el software podrá salir del plan de pruebas cuando surgan errores de gran relevancia que requieran corrección, evidentemente una vez solventado esto deberá volverse a someter a otro plan de pruebas. En nuestro caso saldrá cuando se verifique que la comunicación serial es adecuada y la suministración de información sea acorde a lo especificado en los requerimientos.

### Criterio de suspensión y reinicio

Se deberá suspender la prueba cuando se encuentre errores de mayor relevancia que comprometan la integridad del hardware, en nuestro caso no existirá aspectos que involucren aquel caso debido a la implementación de aislamiento eléctrico. La prueba puede suspenderse también al notar fallos en la comunicación serial ya que este proceso sera una “base” para la ejecucion del software como tal.

## Ciclos de pruebas

### Criterio de entrada al ciclo de pruebas

Para iniciar el proceso de pruebas se debe tener en consideración inicialmente de disponer de los recursos necesarios para dicha tarea, en este caso del hardware (Tarjeta de adquisicion) y de los documentos en donde se especifica los requerimientos y los casos de uso, teniendo asi las herramientas y lógica de uso necesarias para la ejecucion del software con objetivo de prueba.

### Criterio de salida del ciclo de pruebas

Se definirá que la prueba estará en su fase de temincacion o de salida, cuando se halla verificado que cada uno de los requerimientos funcione según lo establecido. En nuestro caso se definirá este criterio cuando se verifique que la comunicación serial envie los datos correctamente y paralelamente la información de suministración de información este acorde a lo planteado en los documentos.

### Terminación anormal del ciclo de pruebas

Sera condición para una terminación anormal, cuando el software manifieste algún riesgo al hardware u algún proceso principal este involucrado en un caso de error. En nuestro caso si se manifiesta error en la comunicación serial se pocedera a una terminación anormal o inmediata debido a que otras funcionalidades se verán implicadas con este proceso.

# Entregables

[In this section, list the various artifacts that will be created by the test effort that are useful deliverables to the various stakeholders of the test effort. Don’t list all work products; only list those that give direct, tangible benefit to a stakeholder and those by which you want the success of the test effort to be measured.]

## Resumen de evaluación de las pruebas

[Provide a brief outline of both the form and content of the test evaluation summaries, and indicate how frequently they will be produced.]

## Reporte de cubrimiento de las pruebas

[Provide a brief outline of both the form and content of the reports used to measure the extent of testing, and indicate how frequently they will be produced. Give an indication as to the method and tools used to record, measure, and report on the extent of testing.]

## Reportes de la calidad percibida

[Provide a brief outline of both the form and content of the reports used to measure the perceived quality of the product, and indicate how frequently they will be produced. Give an indication about to the method and tools used to record, measure, and report on the perceived product quality. You might include some analysis of Incidents and Change Request over Test Coverage.]

## Log de incidentes y peticiones de cambios

[Provide a brief outline of both the method and tools used to record, track, and manage test incidents, associated change requests, and their status.]

## Conjunto de pruebas de humo y scripts de Soporte a las pruebas Scripts

[Provide a brief outline of the test assets that will be delivered to allow ongoing regression testing of subsequent product builds to help detect regressions in the product quality.]

## Productos de trabajo adicionales

[In this section, identify the work products that are optional deliverables or those that should not be used to measure or assess the successful execution of the **Test Plan**.]

### Resultados detallados de las pruebas

[This denotes either a collection of Microsoft Excel spreadsheets listing the results determined for each test case, or the repository of both test logs and determined results maintained by a specialized test product.]

### Scripts de pruebas automáticos adicionales

[These will be either a collection of the source code files for automated test scripts, or the repository of both source code and compiled executables for test scripts maintained by the test automation product.]

### Guías de pruebas

[Test Guidelines cover a broad set of categories, including Test-Idea catalogs, Good Practice Guidance, Test patterns, Fault and Failure Models, Automation Design Standards, and so forth.]

### Matrices de trazabilidad

[Using a tool such as Rational RequisistePro or MS Excel, provide one or more matrices of traceability relationships between traced items.]

# Flujo de pruebas

[Provide an outline of the workflow to be followed by the Test team in the development and execution of this **Test Plan**.]

The specific testing workflow that you will use should be documented separately in the project's Development Case. It should explain how the project has customized the base RUP test workflow (typically on a phase-by-phase basis). In most cases, we recommend you place a reference in this section of the **Test Plan** to the relevant section of the Development Case. It might be both useful and sufficient to simply include a diagram or image depicting your test workflow.

More specific details of the individual testing tasks are defined in a number of different ways, depending on project culture; for example:

* defined as a list of tasks in this section of the **Test Plan**, or in an accompanying appendix
* defined in a central project schedule (often in a scheduling tool such as Microsoft Project)
* documented in individual, "dynamic" to-do lists for each team member, which are usually too detailed to be placed in the **Test Plan**
* documented on a centrally located whiteboard and updated dynamically
* not formally documented at all

Based on your project culture, you should either list your specific testing tasks here or provide some descriptive text explaining the process your team uses to handle detailed task planning and provide a reference to where the details are stored, if appropriate.

For Master Test Plans, we recommend avoiding detailed task planning, which is often an unproductive effort if done as a front-loaded activity at the beginning of the project. A Master Test Plan might usefully describe the phases and the number of iterations, and give an indication of what types of testing are generally planned for each Phase or Iteration.

**Note**: Where process and detailed planning information is recorded centrally and separately from this Test Plan, you will have to manage the issues that will arise from having duplicate copies of the same information. To avoid team members referencing out-of-date information, we suggest that in this situation you place the minimum amount of process and planning information within the Test Plan to make ongoing maintenance easier and simply reference the "Master" source material.]

# Necesidades de ambiente

[This section presents the non-human resources required for the **Test Plan**.]

## Hardware base del sistema

The following table sets forth the system resources for the test effort presented in this *Test Plan*.

[The specific elements of the test system may not be fully understood in early iterations, so expect this section to be completed over time. We recommend that the system simulates the production environment, scaling down the concurrent access and database size, and so forth, if and where appropriate.]

[**Note**: Add or delete items as appropriate.]

| **System Resources** | | |
| --- | --- | --- |
| **Resource** | **Quantity** | **Name and Type** |
| Database Server |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| —Database Name |  | TBD |
| Client Test PCs |  |  |
| —Include special configuration requirements |  | TBD |
| Test Repository |  |  |
| —Network or Subnet |  | TBD |
| —Server Name |  | TBD |
| Test Development PCs |  | TBD |

## Elementos de software base del ambiente de pruebas

The following base software elements are required in the test environment for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Software Element Name** | **Version** | **Type and Other Notes** |
| --- | --- | --- |
| NT Workstation |  | Operating System |
| Windows 2000 |  | Operating System |
| Internet Explorer |  | Internet Browser |
| Netscape Navigator |  | Internet Browser |
| MS Outlook |  | eMail Client software |
| Network Associates McAfee Virus Checker |  | Virus Detection and Recovery Software |

## Herramientas de productividad y de soporte

The following tools will be employed to support the test process for this *Test Plan*.

[Note: Add or delete items as appropriate.]

| **Tool Category or Type** | **Tool Brand Name** | **Vendor or In-house** | **Version** |
| --- | --- | --- | --- |
| Test Management |  |  |  |
| Defect Tracking |  |  |  |
| ASQ Tool for functional testing |  |  |  |
| ASQ Tool for performance testing |  |  |  |
| Test Coverage Monitor or Profiler |  |  |  |
| Project Management |  |  |  |
| DBMS tools |  |  |  |

## Configuraciones del ambiente de pruebas

The following Test Environment Configurations needs to be provided and supported for this project.

| **Configuration Name** | **Description** | **Implemented in Physical Configuration** |
| --- | --- | --- |
| Average user configuration |  |  |
| Minimal configuration supported |  |  |
| Visually and mobility challenged |  |  |
| International Double Byte OS |  |  |
| Network installation (not client) |  |  |

# Responsabilidades, Recursos y necesidades de entrenamiento

[This section presents the required resources to address the test effort outlined in the **Test Plan**—the main responsibilities, and the knowledge or skill sets required of those resources.]

## Personas y roles

This table shows the staffing assumptions for the test effort.

[**Note**: Add or delete items as appropriate.]

| **Human Resources** | | |
| --- | --- | --- |
| **Role** | **Minimum Resources Recommended**  **(number of full-time roles allocated)** | **Specific Responsibilities or Comments** |
| Test Manager |  | Provides management oversight.  Responsibilities include:   * planning and logistics * agree mission * identify motivators * acquire appropriate resources * present management reporting * advocate the interests of test * evaluate effectiveness of test effort |
| Test Analyst |  | Identifies and defines the specific tests to be conducted.  Responsibilities include:   * identify test ideas * define test details * determine test results * document change requests * evaluate product quality |
| Test Designer |  | Defines the technical approach to the implementation of the test effort.  Responsibilities include:   * define test approach * define test automation architecture * verify test techniques * define testability elements * structure test implementation |
| Tester |  | Implements and executes the tests.  Responsibilities include:   * implement tests and test suites * execute test suites * log results * analyze and recover from test failures * document incidents |
| Test System Administrator |  | Ensures test environment and assets are managed and maintained.  Responsibilities include:   * administer test management system * install and support access to, and recovery of, test environment configurations and test labs |
| Database Administrator, Database Manager |  | Ensures test data (database) environment and assets are managed and maintained.  Responsibilities include:   * support the administration of test data and test beds (database). |
| Designer |  | Identifies and defines the operations, attributes, and associations of the test classes.  Responsibilities include:   * defines the test classes required to support testability requirements as defined by the test team |
| Implementer |  | Implements and unit tests the test classes and test packages.  Responsibilities include:   * creates the test components required to support testability requirements as defined by the designer |

## Necesidades de personal y entrenamiento

This section outlines how to approach staffing and training the test roles for the project.

[The way to approach staffing and training will vary from project to project. If this section is part of a Master Test Plan, you should indicate at what points in the project lifecycle different skills and numbers of staff are needed. If this is an Iteration Test Plan, you should focus mainly on where and what training might occur during the Iteration.

Give thought to your training needs, and plan to schedule this based on a Just-In-Time (JIT) approach—there is often a temptation to attend training too far in advance of its usage when the test team has apparent slack. Doing this introduces the risk of the training being forgotten by the time it's needed.

Look for opportunities to combine the purchase of productivity tools with training on those tools, and arrange with the vendor to delay delivery of the training until just before you need it. If you have enough headcount, consider having training delivered in a customized manner for you, possibly at your own site.

The test team often requires the support and skills of other team members not directly part of the test team. Make sure you arrange in your plan for appropriate availability of System Administrators, Database Administrators, and Developers who are required to enable the test effort.]

# Hitos de iteración

[Identify the key schedule milestones that set the context for the Testing effort. Avoid repeating too much detail that is documented elsewhere in plans that address the entire project.]

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Iteration Plan agreed |  |  |  |  |
| Iteration starts |  |  |  |  |
| Requirements baselined |  |  |  |  |
| Architecture baselined |  |  |  |  |
| User Interface baselined |  |  |  |  |
| First Build delivered to test |  |  |  |  |
| First Build accepted into test |  |  |  |  |
| First Build test cycle finishes |  |  |  |  |
| [Build Two will not be tested] |  |  |  |  |
| Third Build delivered to test |  |  |  |  |
| Third Build accepted into test |  |  |  |  |
| Third Build test cycle finishes |  |  |  |  |
| Fourth Build delivered to test |  |  |  |  |
| Fourth Build accepted into test |  |  |  |  |
| Iteration Assessment review |  |  |  |  |
| Iteration ends |  |  |  |  |

# Riesgos, dependencias, suposiciones y restricciones

[List any risks that may affect the successful execution of this **Test Plan**, and identify mitigation and contingency strategies for each risk. Also indicate a relative ranking for both the likelihood of occurrence and the impact if the risk is realized.]

| **Risk** | **Mitigation Strategy** | **Contingency (Risk is realized)** |
| --- | --- | --- |
| Prerequisite entry criteria is not met. | <Tester> will define the prerequisites that must be met before Load Testing can start.  <Customer> will endeavor to meet prerequisites indicated by <Tester>. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | <Customer> will ensure a full set of suitable and protected test data is available.  <Tester> will indicate what is required and will verify the suitability of test data. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | <System Admin> will endeavor to ensure the Database is regularly refreshed as required by <Tester>. | * Restore data and restart * Clear Database |

[List any dependencies identified during the development of this **Test Plan** that may affect its successful execution if those dependencies are not honored. Typically these dependencies relate to activities on the critical path that are prerequisites or post-requisites to one or more preceding (or subsequent) activities You should consider responsibilities you are relying on other teams or staff members external to the test effort completing, timing and dependencies of other planned tasks, the reliance on certain work products being produced.]

| **Dependency between** | **Potential Impact of Dependency** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any assumptions made during the development of this **Test Plan** that may affect its successful execution if those assumptions are proven incorrect. Assumptions might relate to work you assume other teams are doing, expectations that certain aspects of the product or environment are stable, and so forth].

| **Assumption to be proven** | **Impact of Assumption being incorrect** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[List any constraints placed on the test effort that have had a negative effect on the way in which this **Test Plan** has been approached.]

| **Constraint on** | **Impact Constraint has on test effort** | **Owners** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# Procedimientos y procesos de gestión

[Outline what processes and procedures are to be used when issues arise with the **Test Plan** and its enactment.]

## Medición y evaluación de la extension de las pruebas

[Outline the measurement and assessment process to be used to track the extent of testing.]

## Evaluación de los entregables de este pan de pruebas

[Outline the assessment process for reviewing and accepting the deliverables of this **Test Plan**]

## Reporte de problemas, Escalamiento, y resolución de incidentes

[Define how process problems will be reported and escalated, and the process to be followed to achieve resolution.]

## Gestión de los ciclos de prueba

[Outline the management control process for a test cycle.]

## Estrategias de trazabilidad

[Consider appropriate traceability strategies for:

* Coverage of Testing against Specifications — enables measurement the extent of testing
* Motivations for Testing — enables assessment of relevance of tests to help determine whether to maintain or retire tests
* Software Design Elements — enables tracking of subsequent design changes that would necessitate rerunning tests or retiring them
* Resulting Change Requests — enables the tests that discovered the need for the change to be identified and re-run to verify the change request has been completed successfully]

## Aprobación y terminación

[Outline the approval process and list the job titles (and names of current incumbents) that initially must approve the plan, and sign off on the plans satisfactory execution.]