

# Plan de Pruebas

Para

Nitrate

Versión 1.0 aprobada

Preparado por:

Josué Arrieta Salas  
Adrián López Quesada  
Seth Michael Stalley

31/12/2016

## Historial de Revisiones e identificadores de plan de pruebas

Nombre	Fecha	Descripción	Versión
Plan de Pruebas 001	31/12/2016	Creación del primer documento de pruebas. Acorde a la primera iteración de casos de uso para la primera versión del Sistema Nitrate.	1.0

# Índice

Referencias.....	4
Introducción .....	4
Ítems de prueba .....	5
Cuestiones de riesgo del software .....	6
UC-003 - Generar la fórmula de concentración .....	6
UC-005 Calcular el valor de “Sample” .....	7
Características a probar.....	8
Características que no se probarán .....	8
Enfoque .....	9
Pruebas automáticas.....	9
Pruebas manuales.....	10
Criterio de éxito o fallo de las pruebas .....	10
Subir archivo de texto.....	11
Leer archivo de texto .....	12
Generar la fórmula de concentración.....	12
Observar carpeta.....	13
Calcular el valor de “Sample” .....	13
Ingresar el valor de concentración estándar .....	14
Criterio de suspensión y reanudación de requerimientos .....	14
Documentos entregables de prueba .....	15
Tareas de pruebas pendientes.....	15
Necesidades de entorno .....	15
Necesidades de personal y capacitación .....	16
Responsabilidades.....	16
Calendario .....	17
Planeación de riesgos y contingencias .....	18
Aprobaciones.....	19
Glosario .....	20

## Referencias

Los siguientes documentos respaldan la elaboración de este plan. Para mejorar el mantenimiento de este documento, se tratará siempre de referenciar los otros escritos que se han realizado:

- Project Charter versión 1.0:  
<https://drive.google.com/open?id=0Bwn9E8E9d8OwWFIxTkVCbEtVUFk> .
- Documento de Visión y Alcance versión 1.0:  
<https://drive.google.com/open?id=0Bwn9E8E9d8OwS0Q0VXISYUV6VTQ> .
- Listado de casos de uso:  
<https://drive.google.com/open?id=0Bwn9E8E9d8OwZUEtNURlaFkxZU0> .
- Estatuto de requerimientos del sistema versión 1.0 (ERS):  
<https://drive.google.com/open?id=0Bwn9E8E9d8OwQXZIWmlUMVM5dDA> .

Para la elaboración de este plan se ha utilizado la plantilla propuesta en el documento IEEE-829. Tal plantilla se puede encontrar en:

<https://drive.google.com/open?id=0Bwn9E8E9d8OwbERqamYxbEtZd2c> .

## Introducción

Las anomalías y errores en un sistema son las principales causas por la cual un producto se puede llegar a considerar de mala calidad o con poca satisfacción al cliente. Por esta razón se considera crítico la elaboración de planes de prueba; con el propósito de detectar la mayor cantidad de errores en el sistema y poder entregar un producto de alta calidad. Es importante mencionar que es imposible garantizar que un software esté libre de errores.

El propósito de este documento es establecer cómo se realizarán estas pruebas y crear una guía para el *testing* del sistema Nitrate. Se pretende entregar a Laura Hernández un producto de calidad para garantizar la mejor experiencia del usuario. También cabe recalcar que el sistema está utilizando una metodología ágil y dirigida por pruebas; por lo tanto todo el desarrollo del producto estará marcado directamente por este plan de pruebas.

Es importante mencionar que este será el único documento relacionado con las pruebas que se hará, de tal manera que se considerara como un plan de tipo *Master*. No existen otros tipos de planes ni lo existirán.

También es importante mencionar que para la realización y ejecución de este plan no hay restricciones de dinero, ni de personal. Pero sí hay restricción de tiempo por el calendario tan ajustado que se tiene. Para más información sobre este problema se puede consultar el Project Charter en la sección: Manejo de riesgos. También en este mismo documento se puede revisar el calendario del proyecto que está más adelante.

El software está siempre en constante cambio. La herramienta de control de configuración del proyecto será Github. Esta nos mantiene de manera automática varias versiones del software. El log de control de cambios será la bitácora provista por Github. Esto con el propósito de utilizar herramientas automáticas o semi-automática y la eficiencia mejora. Se implementa el software de manera más rápida. Se planea hacer un *Tag* al final de cada iteración.

Cada vez que un integrante requiera realizar un cambio a un artefacto de software, es de suma importancia que se establezca un proceso de control de cambios. Como se mencionó anteriormente se utilizará Github una herramienta de control de cambios. Cada cambio requerido el integrante debe crear un *branch* (copia del repositorio donde se puede trabajar localmente) y cuando deba hacerle *commit* al repositorio principal, lo debe de hacer por medio de un *pull request*. Este *pull request* debe ser aceptado por cualquier miembro del equipo, excepto del que lo solicitó. Es de suma importancia que en los *commits* se menciona:

- Artefacto cambiado: nombre del artefacto que se cambió.
- Fecha: fecha en que el cambio se realizó:
- Responsable del cambio: integrante del equipo que realizó el cambio.
- Descripción del cambio: acción que el responsable del cambio realizó.
- Propósito y justificación del cambio: razones del porqué del cambio.

Si se quiere conocer con más detalle cómo los cambios serán procesados, se puede revisar el Project Charter en la sección: Plan de Control de Cambios.

## **Ítems de prueba**

Debido a que todo el proyecto está organizado por casos de uso (Consultar el documento de listado de casos de uso en caso de ser necesario) los ítems de prueba de este plan de prueba corresponderá a estos casos o funcionalidades. Para esta primera versión solo se tomarán en cuenta los casos de uso pertenecientes a la primera iteración. Si se quiere conocer con mayor

detalle la descripción y texto de cada caso de uso se puede consultar el ERS, y también los requerimientos funcionales que estos mismos implican. También en este mismo documento se mostrará el flujo de ejecución de los casos de uso más adelante. Se tienen los siguientes casos de uso:

Identificador	Caso de uso	Prioridad
UC-001	Subir archivo de texto	Alta
UC-002	Leer archivo de texto	Alta
UC-003	Generar la fórmula de concentración	Alta
UC-004	Observar carpeta	Media
UC-005	Calcular el valor de “ <i>Sample</i> ”	Media
UC-006	Ingresar valor de concentración estándar	Baja

### Cuestiones de riesgo del software

Las áreas más riesgosas y necesarias de probar en el software para esta primera versión del plan de pruebas son las siguientes:

#### *UC-003 - Generar la fórmula de concentración*

Si se calcula incorrectamente la cantidad de concentración de nitratos en una muestra de agua, se asegura que tal muestra es apta para el consumo humano. El software estaría creando resultados incorrectos. Esta situación podría influenciar indirectamente en la salud humana. También este software es pensado para realizar una investigación, y el nivel de correctitud de la investigación depende del sistema Nitrate. Por estas razones este es un caso de uso crítico y riesgoso. Debe ser probado extensivamente. Se tiene:

- Impacto: tanto la satisfacción del cliente como posible perjudicación de la salud humana se podría ver perjudicada.
- Implicaciones: se debe invertir más tiempo y recurso en asegurar que esta funcionalidad esté correcta.

### *UC-005 Calcular el valor de “Sample”*

Este caso de uso supone la predicción de la concentración de una muestra de agua a partir de una calibración existente. Si se logra una predicción incorrecta la salud humana se podría ver afectada indirectamente por el software. Así como la credibilidad de la investigación de Laura Hernández. .Por estas razones este es un caso de uso crítico y riesgoso. Debe ser probado extensivamente. Se tiene:

- Impacto: tanto la satisfacción del cliente como posible perjudicación de la salud humana se podría ver perjudicada.
- Implicaciones: se debe invertir más tiempo y recurso en asegurar que esta funcionalidad esté correcta.

Por otro lado también se tienen los siguientes riesgos propios del desarrollo y aplicación de este plan de pruebas:

- a. Errores en la comprensión de los requerimientos o ambigüedad en la documentación escrita anteriormente. O documentación escrita anteriormente de mala calidad. Esto se cree un riesgo producto del poco tiempo que se ha tenido para la elicitación de los requerimientos así como para la construcción de documentos técnicos.
  - i. Impacto: Puede causar pruebas no válidas o no necesarias; o también puede causar que no se realicen pruebas suficientes
  - ii. Implica: Analizar de manera profunda lo acordado anteriormente en otros documentos para mitigar el impacto de este riesgo.
- b. Errores en pruebas de elementos sencillos del software por la expectativa de funcionamiento al ser sencillo. Se tiene poco tiempo para desarrollar el software de manera que se cree que los elementos sencillos no deben ser probados o no lo suficiente.
  - i. Impacto: Si no se realizan pruebas sobre elementos que se asume su funcionalidad como correcta, puede causar errores en el producto final no deseados.
  - ii. Implica: Analizar exhaustivamente que todos los requerimientos sean probados sin importar la escala de facilidad de implementación que se maneje sobre ellos.

### **Características a probar**

Para la primera versión del y primera iteración del proyecto Nitrate se pueden listar las siguientes características esenciales:

- La aplicación debe de poder procesar y cargar archivos de texto con los valores de absorbancia por *wavelength* y metadatos de manera automática.
- Se tendrá una carpeta que el programa estará observando, de manera que cada vez que un archivo de texto es ingresado, este archivo es cargado de forma automática. Esta carpeta es seleccionada por el usuario.
- Por cada archivo de texto que fue cargado, se puede seleccionar un valor de desviación estándar (STD) para este.
- Calcular el valor de la concentración de una muestra a partir de una calibración seleccionada.
- Generar la fórmula de concentración con base en la correlación de las desviaciones estándares seleccionadas.

Estas características se pueden encontrar con mayor detalle en el Project charter en la sección de Alcance. También se puede consultar en el documento de visión y alcance en la sección de Alcance de la versión inicial. Es importante mencionar que tales características son las completas para la primera versión del software, incluyendo las tres iteraciones a realizar.

### **Características que no se probarán**

Es importante mencionar que la aplicación será probada en su totalidad, ya que se optó por una metodología guiada por pruebas. Sin embargo hay una serie de características que no están contempladas en este plan de prueba. No se probará:

- Escalabilidad del servidor y base de datos: no se considera crítico en este momento, ni tampoco se tienen los recursos necesarios para generar una prueba de este tipo.
- No se realizarán pruebas a los procedimientos almacenados a la base de datos. Se toma esta decisión por cuestiones de tiempo; además se considera que es una base de datos pequeña y fácil de probar. No es necesario un plan de pruebas.
- No se probarán errores que se pueden presentar debido a una mala conexión de Internet. No es el enfoque de la aplicación ya que solo se necesita conexión a Internet para la



validación de usuarios y al visualizar gráficos de otros usuarios utilizando la aplicación móvil.

- No se probará ningún aspecto relacionado con la seguridad: encriptación de datos y validación de cuentas de usuario de manera extensiva. Esta parte no formará parte de este plan.
- No se realizarán pruebas de concurrencia en la base de datos. No se considera un factor crítico en este momento. No se espera que haya mucho uso simultáneo de la aplicación y se está corto de tiempo.
- No se probarán ninguno de los casos de usos considerados como extra en el documento de casos de uso. Al menos no en esta revisión de este plan de pruebas.

## **Enfoque**

En primer lugar se ha de decir que se utilizará la siguiente configuración para todo tipo de pruebas para esta primera iteración. Al ser una aplicación de escritorio se necesitará:

- Computador con java 8 en adelante instalado.
- Windows como sistema operativo, ya que es el sistema de Laura Hernández. Una versión Windows 8 en adelante.
- 512mb de RAM.
- 20mb de disco duro.
- Procesador Intel i3 o superior.

Si se encuentra algún error en cualquier tipo de pruebas, se debe notificar inmediatamente al grupo por algún medio virtual. El equipo debe concentrarse a arreglar las anomalías encontradas antes de empezar a implementar alguna otra funcionalidad. Será un proceso guiado por pruebas. Habrán dos tipos de pruebas que se realizarán al sistema: automáticas y manuales.

### *Pruebas automáticas*

Estas se realizarán a diferentes funciones que se utilizarán en el sistema. Se implementarán con Junit 4, ya que se utilizará Java como lenguaje de programación. Estas pruebas automáticas serán de manejo interno, y no estarán contempladas en este plan de pruebas de manera extensiva. Se podría considerar un plan más especializado, más técnico, para este tipo de pruebas; pero por cuestiones de tiempo no se tomó esta decisión.

Cada vez que se realice una prueba automatizada (ya sea unitario o de integración), se tendrá que realizar una prueba de regresión en donde se efectúen todas las pruebas que se habían creado anteriormente. Se necesita la mayor cobertura de errores posible, y debido a que son automáticos no repercutirá en el tiempo de desarrollo del sistema Nitrate. Todos los componentes están comprendidos en estas pruebas y no se especificará en algún componente en específico. Este tipo de pruebas será de caja blanca, en donde en caso de que no se obtengan los resultados obtenidos, se podrá inspeccionar el código por dentro.

### *Pruebas manuales*

Serán de tipo de pruebas de usuario. No se necesita ningún tipo de herramienta adicional ni servicio, por lo tanto no es necesario realizar algún tipo de asesoría ni entrenamiento especial antes de efectuar el plan de pruebas.

Para las pruebas de este tipo se utilizará un formato de caja negra, en donde sólo interesa los valores de entrada y los resultados de la prueba. Se utilizará un enfoque más del lado del usuario. Se realizará la siguientes cantidades de pruebas para cada caso de uso:

- 4 pruebas para los casos de uso con prioridad baja.
- 8 pruebas para los casos de uso con prioridad media.
- 16 pruebas para los casos de uso de prioridad alta.

Es importante mencionar que cada caso de uso será probado de manera aislada y no en grupos. Tampoco se separaran los casos de uso en pequeños grupos, todo será probado en conjunto.

### **Criterio de éxito o fallo de las pruebas**

En este plan de pruebas se considera como exitosa una prueba, si los resultados al realizar la misma, son los esperados para cada caso de prueba. Se definirá un caso de prueba para cada caso de uso. Como fallo del sistema se pueden definir dos eventos, el primero sería que durante la ejecución de la prueba, el sistema experimente un fallo interno y el sistema colapse, o también se considera fallo si ante una prueba realizada, si a la hora de realizar la misma, no se reciben los datos esperados, o el sistema no informa lo esperado. También esto se verá en los casos de prueba.

El software se enfrentará ante este plan de pruebas durante todo su desarrollo ya que se optó una metodología ágil. Se considerará como aprobado si logra tener exitosas como mínimo un 90 por ciento del total de las pruebas realizadas sobre el sistema Nitrato. Sería bueno afirmar que sólo se aprobará el plan si se logra un 100 por ciento, pero esta meta es menos alcanzable para cualquier software. Aún más con el calendario ajustado que se tiene.

Se tienen los siguientes casos de prueba para cada caso de uso, donde se muestra específicamente su condición de éxito o fallo. Se utilizarán tres niveles de severidad: alta - media - baja. Esto indicará la importancia o el nivel de detalle que se debe ejecutar el caso de prueba. Para ver el encargado del diseño de estos casos de uso y de la ejecución de los mismos revisar la sección de Responsabilidades de este documento:

*Subir archivo de texto*

<b>Id</b>	<b>UC-001</b>
<b>Nombre</b>	Casos de prueba para subir un archivo de texto
<b>Severidad</b>	Alta
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrato. El archivo a subir de texto tuvo que haber sido creado externamente.
<b>Poscondiciones</b>	El archivo de texto es cargado al sistema Nitrato y tal situación es mostrada en pantalla.
<b>Pasos y datos</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción de <i>Open File</i> de la pantalla principal.</li><li>2. El usuario selecciona el archivo que desea subir.</li></ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"><li>1. Se espera que se muestre en pantalla una nueva pantalla con los archivos y directorios del computador.</li><li>2. Se espera que se muestre en pantalla el archivo cargado y los datos de él en la tabla principal del programa.</li></ol>

*Leer archivo de texto*

<b>Id</b>	<b>UC-002</b>
<b>Nombre</b>	Casos de prueba para leer la absorbancia de una archivo de texto
<b>Severidad</b>	Alta
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrate. El archivo tuvo que haber sido cargado anteriormente.
<b>Poscondiciones</b>	Se muestra en pantalla la absorbancia de dicho archivo en la tabla principal
<b>Pasos y datos</b>	<ol style="list-style-type: none"><li>1. El usuario ingresa en <i>Wavelength</i> la longitud de onda deseada.</li><li>2. El usuario selecciona la opción de <i>Absorbance</i>.</li></ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"><li>1. Se muestra en pantalla la longitud de onda ingresada en la fila del archivo.</li><li>2. Se muestra en pantalla la absorbancia de cada archivo en su respectiva fila.</li></ol>

*Generar la fórmula de concentración*

<b>Id</b>	<b>UC-003</b>
<b>Nombre</b>	Casos de prueba para genera la fórmula de concentración
<b>Severidad</b>	Alta
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrate. El archivo de texto deberá haber sido cargado anteriormente. Cada archivo deberá poseer una concentración ingresada manual o automáticamente.
<b>Poscondiciones</b>	Se muestra en pantalla la calibración y la generación de la fórmula: intersección con eje y, pendiente y coeficiente <i>Pearson</i> .
<b>Pasos y datos</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una serie de archivos..</li><li>2. El usuario selecciona una columna de absorbancia dada.</li><li>3. El usuario oprime el botón de <i>Calibrate</i>.</li></ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"><li>1. Se muestra en pantalla de manera distintiva los archivos seleccionados.</li><li>2. Se muestra en pantalla de manera distintiva la columna seleccionada.</li><li>3. Se muestra en pantalla la nueva calibración en la tabla de calibraciones con su respectiva fórmula.</li></ol>

*Observar carpeta*

<b>Id</b>	<b>UC-004</b>
<b>Nombre</b>	Casos de prueba para poner a observar una carpeta
<b>Severidad</b>	Media
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrate. La carpeta con sus archivos de texto deberá ser creada anteriormente.
<b>Poscondiciones</b>	La carpeta es observada y todos los archivos en ella cargados
<b>Pasos y datos</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona el submenú de <i>Tools</i> de la pantalla principal.</li><li>2. El usuario selecciona la opción de <i>Open Observer</i>.</li><li>3. El usuario selecciona la opción <i>Browse</i> para seleccionar una carpeta.</li><li>4. El usuario selecciona la carpeta o directorio a observar.</li><li>5. El usuario selecciona el botón <i>Start</i> para iniciar el observador</li></ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"><li>1. Se espera que se muestra en pantalla las opciones de dicho submenú.</li><li>2. Se muestra en pantalla la carpeta actualmente seleccionada, o ninguna carpeta en caso de que no se haya seleccionado ninguna anteriormente.</li><li>3. Se muestra en pantalla una nueva pantalla con los archivos y directorios del computador.</li><li>4. Se muestra la carpeta seleccionada.</li><li>5. Se muestra en pantalla los archivos cargados que estaban en la carpeta seleccionada</li></ol>

*Calcular el valor de “Sample”*

<b>Id</b>	<b>UC-005</b>
<b>Nombre</b>	Casos de prueba para calcular el valor de Sample
<b>Severidad</b>	Alta
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrate. La calibración seleccionada deberá haber sido creada anteriormente con el sistema Nitrate.
<b>Poscondiciones</b>	Se muestra en pantalla el valor de concentración calculado para todas las filas.

<b>Pasos y datos</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona una calibración anteriormente realizada en la tabla de calibraciones.</li> <li>2. El usuario oprime en el botón de <i>Concentration</i>.</li> </ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"> <li>1. Se muestra en pantalla tal fila de manera distintiva.</li> <li>2. Se muestra en pantalla el valor de la concentración calculado para todas las filas.</li> </ol>

*Ingresar el valor de concentración estándar*

<b>Id</b>	<b>UC-006</b>
<b>Nombre</b>	Casos de prueba para ingresar el valor de concentración estándar
<b>Severidad</b>	Baja
<b>Precondiciones</b>	El usuario deberá haber sido autenticado por el sistema Nitrate. El archivo de texto deberá haber sido cargado anteriormente
<b>Poscondiciones</b>	Para los archivos seleccionados se debe mostrar en pantalla la concentración manualmente escogida
<b>Pasos y datos</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona en <i>Type</i> el tipo STD para el archivo que desea ingresar la concentración estándar en la tabla principal.</li> <li>2. El usuario ingresa el valor estándar en la columna <i>Concentration</i>.</li> </ol>
<b>Resultados esperados de cada paso</b>	<ol style="list-style-type: none"> <li>1. Se espera que se muestre en pantalla tal archivo de tipo STD.</li> <li>2. Se muestra en pantalla el archivo seleccionado como STD con su respectiva concentración.</li> </ol>

### **Criterio de suspensión y reanudación de requerimientos**

Habrán tres criterios de suspensión de pruebas para esta primera iteración de casos de uso. Esto con el propósito de determinar que el testing que es estaría ejecutando no tendría valor y no tiene sentido continuar con las pruebas. Se estarían gastando recursos:

1. En primer lugar se detendrían las pruebas si los errores al momento comprenden al menos un 35% del total de pruebas a realizar. Esto si se ve desde un punto de vista general.
2. También es posible detener el proceso de pruebas si se detectan al menos una anomalía en los casos de uso considerados riesgosos. Estos serían UC-003 - Generar la fórmula de concentración y UC-005 Calcular el valor de “Sample”

3. También es importante mencionar que se considera aceptable continuar con el siguiente caso de prueba sólo si se obtiene al menos un 80% de resultados correctos para el caso de prueba anterior. En caso contrario se suspende

### **Documentos entregables de prueba**

Como parte de este plan se planea entregar también:

- Documento de plan de pruebas.
- Documento de casos de prueba (contenido en este mismo documento).
- Informe de ejecución de pruebas: tanto para pruebas automáticas (serán capturas de pantalla de los resultados de la herramienta Junit) como para las manuales (serán tablas resultados basadas en lo especificado por cada caso de uso en su caso de prueba). En caso de ser requerido se podría también entregar el código fuente de las clases en java de las pruebas unitarias y de integración automáticas. Se hará un informe general que se entregará al final de cada iteración. En este informe se contendrán los últimos registros de errores y registros de ejecución que se obtuvieron al final de la iteración.
- Toda documentación mencionada en la sección de referencias: ERS, documento de visión y alcance, listado de casos de uso y Project Charter.

### **Tareas de pruebas pendientes**

Como se mencionó anteriormente se escogió un proceso iterativo, y este plan comprende específicamente para los casos de uso comprendidos en la primera iteración. Queda pendiente la elaboración y ejecución de las pruebas tanto de la segunda iteración como de la tercera iteración. Esto con el propósito de evitar el gasto de recursos innecesarios. Esto sería para la versión 1 del sistema Nitrate.

### **Necesidades de entorno**

No hay necesidades específicas de hardware para la ejecución de este plan. De hecho se necesita un hardware bastante general: una computadora de escritorio con una versión reciente de Windows (tal especificación del hardware se puede encontrar en la subsección de Enfoque en este mismo documento). Para la validación de usuarios también es necesario que se tenga una

conexión a Internet estable para poder conectarse al servidor y consultar la base de datos. Este servidor se alquiló a Amazon y se encuentra en la nube de este.

Es importante mencionar que los datos de prueba (archivos de absorbancia por longitud de onda) son considerados especiales. Estos serán provistos por un espectrómetro y un software totalmente externos al sistema Nitrate. Estos archivos de texto también serán los mismos datos de prueba para futuras iteraciones; con el propósito de probar el sistema que se asemeje de la mejor manera posible al real.

Como se mencionó anteriormente, las funcionalidades que se van a probar del sistema serán probadas como un todo, no por separado, debido a que estas no se subdividen en partes.

### **Necesidades de personal y capacitación**

Adrián López será el encargado de que el recurso humano que se utilizará para las pruebas obtenga toda la capacitación necesaria para que se lleven a cabo las diferentes pruebas. Como se mencionó anteriormente se utilizará Junit para las pruebas automáticas y un proceso totalmente manual para las pruebas manuales. Junit es una herramienta que se ha utilizado con anterioridad por el equipo en otros proyectos, por lo tanto no se cree necesario que se necesite algún entrenamiento personal o de capacitación en el uso de la herramienta. El personal ya conoce la implementación de pruebas en un entorno de escritorio en un ambiente de desarrollo en Java.

Sin embargo, debido a que se realizará un software en un área que los desarrolladores no están muy relacionados (ámbito físico-químico); el personal debe familiarizarse con todos los términos de esta área para la realización de pruebas efectivas. Una pequeña capacitación es necesaria.

### **Responsabilidades**

La elaboración de este plan de pruebas ha estado a cargo de los desarrolladores del sistema Nitrate. Se tienen las siguientes necesidades específicas:

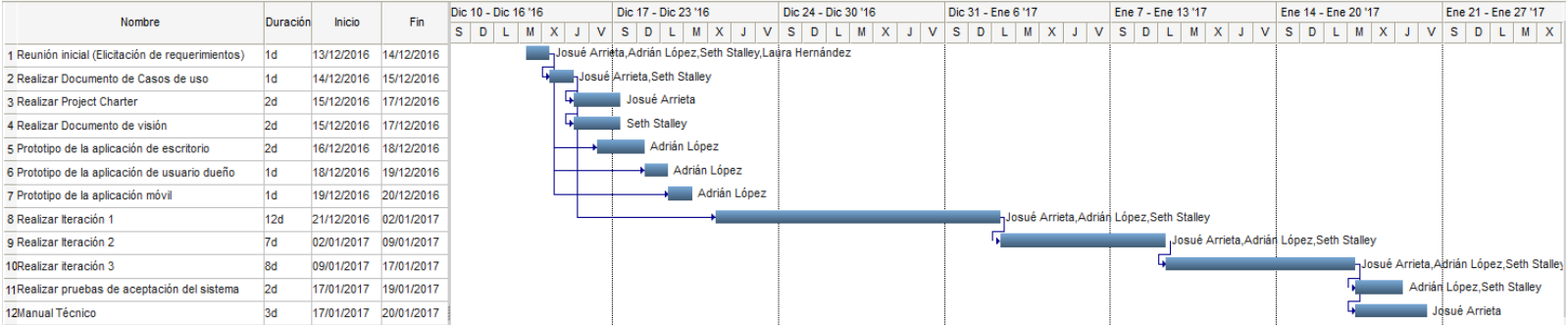
<b>Desarrollador</b>	<b>Responsabilidad</b>
Josué Arrieta	Será el administrador de las pruebas. Será el encargado de tomar las decisiones críticas de seguir o no para los ítems de prueba cubiertos en



	este plan de pruebas. Creará el calendario de ejecución de las pruebas.
Adrián López	Será el coordinador del proyecto. Se encargará de que las pruebas se realicen, y tendrá un control general del desarrollo del proyecto. Será el responsable de que el personal reciba la capacitación necesaria para las pruebas del sistema. Se encargará del diseño de las pruebas y creación de casos de prueba. Establecerá qué se probará y qué no se probará.
Seth Stalley	Será el encargado de la realización de las pruebas. Deberá seguir el plan de pruebas al pie de la letra y la programación de las pruebas automáticas.

Calendario

Como se mencionó anteriormente se seleccionó para la implementación de este proyecto una metodología ágil; de manera que no se puede especificar exactamente cuándo ocurrirá la ejecución de las pruebas. Estas se estarán realizando constantemente en todo momento, sobre todo las pruebas de regresión automáticas. Se tiene el siguiente calendario para la elaboración del proyecto (<https://drive.google.com/open?id=0Bwn9E8E9d8OwYzNySWxIbVJTXX28>):



La realización del plan de pruebas junto con su diseño de casos de prueba se realizará los primeros días de la iteración.

La ejecución de las pruebas manuales se efectuará en los últimos dos días antes de la entrega de cada iteración y obtener la aprobación del cliente. Si hay un atraso en la agenda de las pruebas, se invertirán más recursos (tiempo y personal) para la ejecución de estas y también se disminuirá el grado de aceptación de las pruebas. El factor tiempo es crítico en este proyecto. Si se quiere conocer con más detalle el manejo del tiempo, riesgos, entregables e hitos; favor revisar el Project Charter.

## **Planeación de riesgos y contingencias**

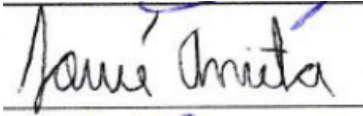

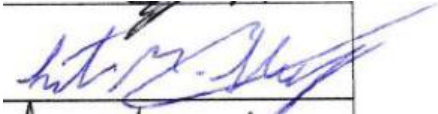
Se recomienda leer también la sección de manejos de riesgos del Project Charter. Los posibles riesgos con énfasis al proceso de pruebas son:

- Cambios en los requerimientos originales. En caso de esto, el número de pruebas se reducirá por un 50% y el número de pruebas aceptables por un 30% y se reforzará la cantidad y la calidad de pruebas para la siguiente iteración. Este riesgo se considera de probabilidad media de ocurrencia.
- Posibles atrasos en el calendario. Se tiene un calendario predefinido y corto para la entrega de cada iteración y de la primera versión del sistema Nitrato. En caso de que el riesgo se materialice se reducirá el número de pruebas por un 50%. También se descartó la posibilidad de mejorar, optimizar o la implementación de casos de usos adicionales . Este riesgo se considera de probabilidad alta de ocurrencia.
- Posibles retrasos en el entrenamiento del personal que realizará las pruebas. Esto provocaría que las pruebas se vean retrasadas. En caso de que el riesgo se materialice, se concentrará todos los recursos en capacitar rápidamente el personal. Este riesgo se considera de probabilidad baja de ocurrencia.
- Falta de personal al momento de realizar las pruebas. En caso de que el riesgo se materialice, se concentrará todos los recursos en conseguir el personal necesario. Se tendrá que saturar aún más la utilización de recursos ya que se tiene poco personal. Se forzará al equipo de trabajo operar horas extra. Este riesgo se considera de probabilidad media de ocurrencia.

Es importante mencionar que en ninguno de los casos se omitirá por completo el testing de software. Las soluciones serán siempre: trabajar horas extras y bajar la calidad de las pruebas. Pero nunca se quitará por completo.

## Aprobaciones

Cualquiera de los siguientes podrá marcar el proceso de prueba como aprobado y que se pueda continuar con la siguiente iteración. Es importante mencionar que para finalizar cada iteración se tendrá que obtener un aprobación de Laura Hernández:

Nombre	Firma
Josué Arrieta	
Adrián López	
Seth Stalley	

## Glosario

Se recomienda también revisar el glosario que se encuentra en el ERS para el mayor entendimiento del sistema. También se adjunta el glosario de este documento en particular:

<b>Término</b>	<b>Definición</b>
Caja blanca	Prueba que además de tomar en cuenta las entradas y salidas; también se centra en la inspección del código fuente del interior. Se prueban distintos flujos de ejecución.
Caja Negra	Prueba en donde solo se toman en cuenta las entradas que recibe y las salidas que produce, sin inspeccionar el funcionamiento interior.
Dirigido por pruebas	Es un proceso de desarrollo de software en que los requerimientos son convertidos en casos de prueba y las pruebas de unidad son repetidamente ejecutadas en el código fuente. Las pruebas son esenciales.
ERS	Documento técnico de software en donde se especifican los requerimientos o características que un sistema debe tener
Funciones	Corresponde a una funcionalidad que un componente de software puede realizar.
Junit	Es una herramienta que se utiliza para la creación de pruebas de unidad y de integración en Java.
Metodología ágil	Proceso de software contrario a casada: documentación, implementación, testing, aseguramiento de la calidad se realiza en todo momento en varias iteraciones de la creación del software. Estos pueden cambiar constantemente y evolucionar con el tiempo.
Pruebas de integración	Son un tipo de pruebas de software que verifica el funcionamiento entre 2 o más componentes de software.
Pruebas de regresión	Son aquel tipo de pruebas que cuando se realiza una prueba, también se tienen que realizar todas las pruebas anteriores a esta.
Pruebas de unidad	Es un tipo de prueba que solamente verifica la funcionalidad de un componente de software o una función.
RAM	Es la memoria principal de un computador donde se guardan los programas actualmente en ejecución. Es altamente accesada por el procesador.