



AgroFinderGround

***Ingeniería en Tecnologías de la información***

***Sistema de Consulta de Suelos de Cultivo.***



**AgroFinderGround**

**Plan de prueba**

**Historial de revision**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 26/03/2015 | 1.0 | Creación del documento | José Carlos Vázquez García |

Contenido

[1 INTRODUCCIÓN 4](#_Toc415613963)

[1.1 PROPÓSITO 4](#_Toc415613964)

[1.2 ALCANCE 4](#_Toc415613965)

[1.3 AUDIENCIA 5](#_Toc415613966)

[1.4 DOCUMENTO DE TERMINOLOGÍAS Y SIGLAS 5](#_Toc415613967)

[1.5 REFERENCIAS 5](#_Toc415613968)

[2 MISIÓN DE EVALUACIÓN Y PRUEBA DE MOTIVACIÓN 5](#_Toc415613969)

[2.1 ANTECEDENTES 5](#_Toc415613970)

[2.2 MISIÓN DE EVALUACIÓN 6](#_Toc415613971)

[2.3 PRUEBA MOTIVADORES 6](#_Toc415613972)

[3 OBJETIVO DE LOS ELEMENTOS DE PRUEBA 6](#_Toc415613973)

[4 ENFOQUE DE LAS PRUEBAS 7](#_Toc415613974)

[4.1 TIPOS DE TÉCNICA DE PRUEBAS 7](#_Toc415613975)

[4.1.1 PRUEBA DE INTEGRIDAD DE LA BASE DE DATOS 7](#_Toc415613976)

[4.1.2 PRUEBA DE FUNCIÓN 8](#_Toc415613977)

[5 CRITERIOS DE ENTRADA Y SALIDA 8](#_Toc415613978)

[5.1 PLAN DE PRUEBA 8](#_Toc415613979)

[5.1.1 CRITERIOS DE ENTRADA DEL PLAN MAESTRO DE PRUEBAS 9](#_Toc415613980)

[5.1.2 CRITERIO DE SALIDA DEL PLAN MAESTRO DE PRUEBAS 9](#_Toc415613981)

[5.1.3 CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN. 9](#_Toc415613982)

[6 NECESIDADES DE AMBIENTE 9](#_Toc415613983)

[6.1 HARDWARE BASE PARA AMBIENTE DE PRUEBAS 9](#_Toc415613984)

[6.2 SOFTWARE BASE PARA AMBIENTE DE PRUEBAS. 9](#_Toc415613985)

[6.3 HERRAMIENTAS DE SOPORTE A LAS PRUEBAS 9](#_Toc415613986)

[6.4 CONFIGURACIONES DE AMBIENTES DE PRUEBAS. 10](#_Toc415613987)

[7 RESPONSABILIDADES Y EQUIPO DE TRABAJO 10](#_Toc415613988)

[7.1 PERSONAS Y ROLES 10](#_Toc415613989)

[8 RIESGOS DE LAS PRUEBAS 10](#_Toc415613990)

# Introducción

## Propósito

El objetivo de este documento es recoger los casos de pruebas que verifican que el sistema satisface los requisitos especificados. Contiene la definición de los casos de prueba, la matriz de trazabilidad entre casos de pruebas y requisitos, y la estrategia a seguir en la ejecución de las pruebas. También permitirá definir los lineamientos a seguir para realizar la planeación de la etapa de pruebas sobre el proyecto “AgroFinderGround”, planteando una estrategia que conduzca al objetivo enfocado en el aseguramiento de calidad del software web.

El propósito del plan de pruebas es:

* Proveer un artefacto central que gobierne la planeación y control del esfuerzo de pruebas. Éste define el enfoque general que será empleado para probar el software y para evaluar los resultados de esas pruebas, y es el plan de más alto nivel que será usado por el o los administradores para guiar y dirigir el trabajo de pruebas detallado.
* Proveer visibilidad a los interesados en el esfuerzo de pruebas que han tenido las consideraciones adecuadas para varios aspectos que orientan el esfuerzo de pruebas, y dónde es apropiado que los interesados aprueben el plan.

Este Plan de Pruebas también soporta los siguientes objetivos específicos:

* Identificar los ítems que serán objeto de las pruebas.
* Enmarcar la metodología de pruebas que será utilizada
* Identificar los recursos requeridos y proveer un estimado del esfuerzo de las pruebas.
* Elaborar un listado de los elementos entregables del plan de pruebas.

## Alcance

El Plan de Pruebas describe el detalle de las diferentes pruebas a ser aplicadas, así como también las herramientas y metodologías a utilizar en cada una de estas. Las pruebas que serán realizadas son:

* **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 validaran las piezas individuales del software como una unidad independiente, bucles, condicionales, etc.
* **Pruebas de integración:** Se validará la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta.
* **Pruebas Funcionales o de Procedimientos:** Se validaran los procesos, reglas de negocio establecidas y los requerimientos funcionales.
* **Pruebas de sistema:** Las pruebas de sistema se determinarán en el momento que se entregue el documento de Requerimientos no funcionales, y así determinar qué tipos de prueba se realizarán y a qué casos de uso se aplicarán.
* **Pruebas de regresión:** Se validará que el sistema mantenga su correcta funcionalidad debido a la incorporación de un ajuste, corrección o nuevo requerimiento.

Adicionalmente y con el fin de centrar el plan de pruebas en ciertos factores que son críticos y de mayor relevancia para el proyecto, se determinan los tipos de pruebas que se realizarán para el proyecto, diseñando los factores de calidad y las pruebas especializadas para alcanzar estos atributos del software entregado. Con esta misión se identifican de acuerdo a las especificaciones del cliente los factores

Para este proyecto de acuerdo a los requerimientos, se definen los siguientes factores en los que se enfocarán las pruebas:

* Corrección.
* Conformidad.
* Facilidad de Uso.
* Portabilidad.
* Facilidad de Operación

## Audiencia

Este plan maestro de pruebas está dirigido a todas aquellas personas involucradas en la planeación, aprobación y ejecución del mismo.

## Documento de terminologías y Siglas

## Referencias

* Cronograma del Proyecto
* Especificación Requerimientos de Software

# Misión de evaluación y prueba de motivación

## Antecedentes

Se pretende realizar un levantamiento y análisis de información de los procesos de gestión de solicitudes de la parte responsable agrícola de la zona de Cuitláhuac con el fin de plantear una arquitectura de solución tecnológica con el fin de optimizar la eficiencia al realizar alguna consulta de la información del suelo y sus propiedades, tanto a nivel técnico como funcional de estos procesos que constituyen y representan un valor en los objetivos a alcanzar en este caso del representante agrícola.

## Misión de evaluación

La misión de la evaluación para el presente proyecto se define enfocada al aseguramiento de la calidad de los componentes y artefactos tecnológicos desarrollados, de manera que estos cumplan con la especificación de los requerimientos del cliente. Para esto se definen los siguientes lineamientos que constituyen la misión y objetivos dentro de este esfuerzo de pruebas:

* Descubrir tantos errores como sea posible.
* Notificar de los riesgos percibidos del proyecto.
* Examinar la aplicación para comprobar si hace o no lo que se supone, debe hacer. De igual forma verificar si ésta hace o no lo que se supone, no debe hacer.
* Validar y Verificar a través de la comparación del resultado de las pruebas del aplicativo con el resultado que el mismo tendría que producir de acuerdo a su especificación.
* Evaluar la calidad del producto y satisfacción de los interesados.

El proceso de evaluación y pruebas debe permitir detectar problemas desde el inicio de la especificación de requerimientos, antes de que sean de gran impacto en fases más adelantadas del proyecto, esto con el fin de disminuir los riesgos y de obtener un producto con calidad logrando mayor satisfacción del cliente.

## Prueba Motivadores

Dentro de los principales motivadores de pruebas del proyecto, están la necesidad del cliente de optimizar y gestionar la ejecución de sus procesos de negocio, verificar la confiabilidad de la información que posee sobre sus clientes y dar trámites ágiles y efectivos a las solicitudes que ellos generan a la organización.

Adicionalmente existen unos motivadores puntuales que van a contribuir a que se construya un software que satisfaga los requerimientos del cliente de la manera más óptima posible y siguiendo un proceso adecuado para conseguirlo. Estos son:

* Aseguramiento de la calidad.
* Solicitudes de cambios.
* Riesgos de calidad
* Verificación de los casos de uso.
* Comprobación de los requerimientos funcionales y no funcionales.

# Objetivo de los elementos de prueba

A continuación se listan los elementos (artefactos, entregables, documentos etc.) que serán objeto de prueba dentro del esfuerzo de pruebas:

**Fase Inicial**

* Documentación
* Especificación de Requerimientos
* Estimaciones
* Modelos - Diagramas

# Enfoque de las pruebas

El plan de pruebas se basará en su totalidad en pruebas funcionales, instalación, regresión y otras teniendo en cuenta los requerimientos no funcionales.

**Revisión de la documentación**: La estrategia para realizar estas pruebas, consiste en la revisión de la documentación y casos de uso verificando su completitud y concordancia en la información que se encuentra en ellos.

* **Pruebas unitarias:**Las estrategias para realizar estas pruebas consiste en generar casos de prueba necesarios:
  + Para que cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.
  + Para que cada condición tenga por lo menos una vez un resultado verdadero y al menos una vez uno falso.
  + Para probar varias veces el mismo bucle (en donde aplique) considerando los siguientes casos: Ignorar el bucle, pasar una vez, pasar dos veces, pasar n veces, pasar n-1 veces y n+1 veces.
* **Pruebas funcionales o de procedimientos:** La estrategia para realizar estas pruebas consiste en la elaboración y ejecución de Set de Pruebas, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos que permitan verificar lo siguiente:
* Los resultados esperados ocurren cuando se usan datos válidos.
* Se despliegan mensajes de error cuando se usan datos inválidos.
* **Pruebas de Regresión:**La estrategia para realizar estas pruebas consiste en repetir las pruebas (funcionales y de carga) ejecutadas antes de corregir defectos o de añadir nuevas funcionalidades, para comprobar que las modificaciones no provocan errores donde antes no los había.

## Tipos de Técnica de pruebas

### Prueba de integridad de la base de datos

|  |  |
| --- | --- |
| Objetivo de la Técnica | Determinar como la base de datos será cargada |
| Técnica | Bottom-up – Empezar con los modulos de nivel inferior llamando e insertando una serie de datos a la base de datos. |
| Herramientas Requeridas | * SQLMutation |
| Criterio de Complejidad | Todas las pruebas planeadas deben ser ejecutadas  Los defectos encontrados seran tomados en cuenta. |
| Criterio de exito | * Concordancia de los procedimientos del sistema con los requerimientos de usuario * Optimo manejo de excepciones y errores |

### Prueba de función

|  |  |
| --- | --- |
| Objetivo de la Técnica | Verificar el funcionamiento interno de los componentes desarrollados por medio de la comprobación del los procedimientos llevados a cabo por el software en cada invocación/llamado/respuesta, asi como el procesamiento de datos que tiene lugar en cada uno de esta acciones |
| Técnica | Prueba de caja negra |
| Herramientas Requeridas | * Debuggers como phpunit. * Sistema de seguimiento de errores |
| Criterio de exito | * Concordancia de los procedimientos del sistema con los requerimientos de usuario * Optimo manejo de excepciones y errores |

# Criterios de Entrada y Salida

## Plan de Prueba

### Criterios de Entrada del Plan Maestro de Pruebas

* Claridad en el procedimiento para el desarrollo de las pruebas.
* Tener un entorno de pruebas adecuado.
* Set de pruebas completo y claro.
* Toda la documentación requerida para la realización de las pruebas debe estar disponible.

### Criterio de Salida del Plan Maestro de Pruebas

* Que todos los set de pruebas diseñados para cada caso de uso se ejecuten de manera exitosa, cumpliendo los criterios de aceptación definidos para cada uno.

### Criterios de Suspensión y Reanudación.

* Una característica principal tiene un error que impide probar un área importante.
* El entorno de prueba no es lo suficientemente estable como para confiar en los resultados.
* El entorno de pruebas es muy diferente del entorno de producción.

# Necesidades de ambiente

## Hardware Base para ambiente de pruebas

La siguiente tabla relaciona los recursos de hardware que son necesarios para crear un ambiente inicial de pruebas en este proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Equipo** | **Procesador** | **DD** | **RAM** | **Aplicación a Instalar** |
| PC Windows (XP o superior) | Intel | 10 GB | 1 GB | Navegador Mozilla o Google Chrome |

## Software Base para ambiente de pruebas.

El software a probar en el equipo de pruebas es:

* Windows XP o superior
* Navegador Mozilla Firefox o Google Chrome.

## Herramientas de soporte a las pruebas

Las siguientes son herramientas a utilizar para la ejecución de las pruebas y la administración de sus resultados.

| **Categoría o Tipo** | **Nombre** | **Fabricante** | **Versión** |
| --- | --- | --- | --- |
| Pruebas Unitarias | ***PHPUnit*** | Sebastian Bergmann | 3.6.10 |

## Configuraciones de Ambientes de Pruebas.

En esta fase aún no se posee la información suficiente para determinar que Hardware y software será requerido para la ejecución de pruebas sobre los componentes tecnológicos desarrollados.

# Responsabilidades y equipo de trabajo

## Personas y Roles

Esta tabla representa el personal supuesto para el esfuerzo de pruebas.

| **Recursos Humanos** | |
| --- | --- |
| **Rol** | **Responsabilidades Específicas o Comentarios** |
| Administrador de Pruebas | * Administra el esfuerzo de las pruebas, aprueba los criterios de entrada y salida a las pruebas, monitorea avance del esfuerzo de pruebas, aprueba los casos de prueba, gestiona el alcance y misión de las pruebas, Certifica el nivel de calidad del producto construido. |
| Diseñador de Pruebas | * Es el responsable de diseñar los set de pruebas (estructura y enfoque) que se realizarán al sistema para una certificar que se construyó un producto que satisface los requerimientos definidos. |
| Analista de Pruebas | * Es el responsable de ejecutar los casos de prueba y realizar los reportes correspondientes sobre esta ejecución. * Realizar documentación técnica de las pruebas. |
|  |

# Riesgos de las pruebas

A continuación se expone una matriz en la cual se relacionan los 5 factores de prueba más críticos para el proyecto con los riesgos identificados para cada uno de ellos vs las fases en las que se ejecutan las pruebas.

| **Factor de Prueba** | **Requerimientos** | **Diseño** | **Software** |
| --- | --- | --- | --- |
| Conformidad | * RSK\_REQ\_001: Pasar por alto la prueba de requerimientos no funcionales clave que impliquen un gran impacto en la arquitectura propuesta. | * RSK\_DSN\_001: Alta complejidad en el diseño de las pruebas que evidencien la conformidad con los requerimientos de gobernabilidad y reusabilidad | * RSK\_SFW\_001: Omitir la ejecución de pruebas en las características menos comunes de utilización. |
| Portabilidad | * RSK\_REQ\_002: Identificar tardíamente problemas de compatibilidad con navegadores poco usables o sistemas operativos. | * RSK\_DSN\_002: No contar con la tecnología necesaria para probar aspectos del diseño enfocados a comprobar el bajo acoplamiento de la solución. | N/A |
| Facilidad de Uso | * RSK\_REQ\_003: No lograr captar la opinión de los usuarios finales para determinar los aspectos de facilidad de uso que ellos esperan. | * RSK\_DSN\_003: Realizar las pruebas con un enfoque muy técnico sin detectar aspectos que por diseño supongan complejidades altas en el uso de la aplicación | * RSK\_SFW\_002: Probar solo funcionalidades sin identificar problemas o mejoras en la facilidad de utilización de la aplicación web |
| Facilidad de Operación | * RSK\_REQ\_004: No incluir en las listas de chequeo de comprobación de los requerimientos, los aspectos relacionados con la facilidad de operación, por desconocimiento en los mismos | * RSK\_DSN\_004: No detectar a tiempo aspectos del diseño que se conviertan en impedimentos para permitir las fácil instalación y administración de las solución | * RSK\_SFW\_003: Detectar tardíamente problemas relacionados con la implementación de la aplicación web |
| Corrección | * RSK\_REQ\_005: No Encontrar requerimientos en una fase temprana con algún nivel de ambigüedad. | * RSK\_DSN\_005: No Identificar problemas para corregir defectos detectados en una fase avanzada del desarrollo. | * RSK\_SFW\_004: Presencia de errores en el producto que sean muy costosos de corregir cuando este ya se encuentre finalizado. |