****

**Plan de desarrollo de software**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del proyecto: | *Diagnóstico de plagas en los cultivos de la región.* | | |
| Alias del proyecto: | *Healthy-Plants* | | |
| Versión: | *1.0* | Fecha última  modificación: | *26/10/2016* |

Índice

[I Introducción 2](#_Toc465591281)

[1.1 Propósito 2](#_Toc465591282)

[1.2 Alcance 2](#_Toc465591283)

[1.3 Resumen 3](#_Toc465591284)

[II Vista general del proyecto. 4](#_Toc465591285)

[2.1 Propósito, alcance y objetivos 4](#_Toc465591286)

[2.2 Suposiciones y restricciones 5](#_Toc465591287)

[2.3 Entregables del proyecto 5](#_Toc465591288)

[2.4 Evolución del plan de desarrollo de software 7](#_Toc465591289)

[III Organización del proyecto 8](#_Toc465591290)

[3.1 Participantes en el proyecto 8](#_Toc465591291)

[3.2 Roles y responsabilidades 10](#_Toc465591292)

[3.3 Organigrama 11](#_Toc465591293)

[IV Gestión del proceso 12](#_Toc465591294)

[4.1 Fases 12](#_Toc465591295)

[4.2 Calendario de las actividades 14](#_Toc465591296)

# I Introducción

El Plan de Desarrollo de Software es una versión previa para ser incluida en la propuesta elaborada como respuesta al proyecto de Ingeniería de la Universidad Tecnológica del centro de Veracruz denominado Sistema de Apoyo al Diagnóstico de Enfermedades y Plagas en cultivos referenciado por su nombre Healthy Plants, este documento proporciona una visión general del proyecto.

El desarrollo del proyecto está basado en la metodología RUP (Rational Unified Process), este documento es as su vez uno de los artefactos generados.

## 1.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información para controlar y dirigir el proyecto describiendo el enfoque de desarrollo de software.

* El líder de lo utilizará para agendar las actividades, asignar los recursos disponibles y realizar su debido seguimiento.
* Los miembros del equipo lo usan para entender lo que deben hacer, en que momento deben hacerlo y cuáles son las debidas dependencias y relaciones existentes.

## 1.2 Alcance

Para esta versión del Plan de Desarrollo de Software se ha basado en la captura de requisitos principalmente por medio de encuestas mediante las cuales se ha considerado el sector al cual va dirigida la funcionalidad del producto. Este documento corresponde a la fase de Incepción por lo que es posible que su contenido sea modificado durante las fases siguientes produciendo nuevas versiones más actualizadas.

## 1.3 Resumen

Posterior a la introducción el documento consta las siguientes partes

**Vista general del proyecto:** Describe el propósito, alcance y objetivos del proyecto así como los artefactos que serán producidos y/o utilizados durante el desarrollo del mismo.

**Organización del proyecto:** Describe la estructura organizacional del equipo de desarrollo, los datos relevantes de cada uno de sus miembros y el rol que desempeñan dentro del equipo.

**Gestión del proceso:** Especifica la planificación estipulada para el proyecto indicando y describiendo cada una de las fases correspondientes.

# II Vista general del proyecto.

## 2.1 Propósito, alcance y objetivos

Healthy Plants es un proyecto que busca ayudar en el diagnóstico y tratamiento de enfermedades y plagas de los cultivos dependiendo de su localización geográfica y los distintos climas presentes a lo largo del año, esto haciendo uso de una serie de algoritmos que consideren las distintas variables al momento de realizar una búsqueda.

El proyecto debe contener los módulos responsables de realizar la búsqueda inteligente de enfermedades o plagas sugiriendo el o los resultados que mejor se adapten a la situación.

Como objetivos principales se tienen:

* Recabar información relacionada con la funcionalidad del proyecto.
* Diseñar planes que ayuden a la gestión del proyecto.
* Diseñar una base de datos que sirva como base a las pruebas.
* Montar un servicio web que sirva a las pruebas necesarias del software.
* Desarrollar un algoritmo de búsqueda inteligente.
* Desarrollar una aplicación con base en el algoritmo mencionado.
* Asociar dicho servicio con terceros.

## 2.2 Suposiciones y restricciones

Algunas de las suposiciones y restricciones del sistema y que se derivan de las encuestas previas realizadas con los fines antes mencionados en este documento son:

* Aseguramiento de la protección de los datos.
* Acuerdos de privacidad del usuario.
* Mantenimiento de la infraestructura necesaria para operar.
* Adecuación a la legislación de cada región de ser necesaria.

Cabe mencionar estos puntos pueden modificarse conforme a la etapa en la que se encuentre el desarrollo del producto o sistema.

## 2.3 Entregables del proyecto

Se describen cada uno de los artefactos que serán generados y utilizados durante el proyecto y que constituyen los entregables adecuándose a la perspectiva de la metodología RUP.

Así mismo es necesario mencionar que, de acuerdo a la metodología antes mencionada, todos los entregables pueden sufrir modificaciones a lo largo del proceso de desarrollo por lo cual solo al término de este se podrá tener una versión definitiva de cada uno de ellos, en cada iteración estos se irán perfeccionando.

* **Plan de Desarrollo de Software:** El presente documento.
* **Plan de negocios:** Describe los objetivos del proyecto, las estrategias y el modelo de negocio, se determina la viabilidad del proyecto y la imagen del ante terceras personas (clientes).
* **Caso de desarrollo:** Describe los detalles y datos referentes a la realización de las pruebas de funcionalidad dentro de un entorno real.
* **Plan de especificación de requisitos:** Es el documento que contiene la especificación de requisitos del sistema a desarrollar especificando la funcionalidad del mismo.
* **Documento de arquitectura de software:** Abarca todo el sistema y contiene información acerca de la estructura del mismo, está en constante cambio durante la fase de desarrollo.
* **Prototipo:** Es la versión previa de lo que pretende ser el sistema mostrando una funcionalidad mínima.
* **Casos de uso:** Es el diagrama que presenta las distintas acciones e interacciones que se realizan dentro del sistema.
* **Modelo de secuencia:** Describe las distintas interacciones entre los componentes principales del sistema.
* **Modelo de datos:** Describe de manera abstracta la manera en que se gestiona la información almacenada en la base de datos.
* **Plan de pruebas:** Es el documento que contiene las especificaciones de pruebas a realizar al producto terminado durante el testeo del mismo.
* **Diagrama de clases:** Describe la estructura de las clases que conforman el sistema y la manera en que se relacionan unas con otras.
* **Plan de integración:** Contiene los detalles referentes a la implementación, el alcance obtenido y el nivel de integración.
* **Resumen de evaluación de pruebas:** Contiene los resultados de las pruebas realizadas al producto final, las observaciones y el nivel de aprobación.
* **Notas de versión:** Contiene las especificaciones de los cambios realizados al sistema durante las distintas iteraciones o fases.
* **Artefactos de instalación:** Son los ejecutables o binarios generados posteriores a la fase de construcción, son propiamente dicho el sistema en sí.
* **Manuales del sistema:** Contiene las especificaciones e indicaciones de uso del sistema, así como las distintas soluciones a los problemas que se pudieran presentar.
* **Evaluación de resultados:** Contiene las notas finales de la conclusión del proyecto, las distintas observaciones, puntos de mejora y resultados obtenidos satisfactoriamente.

## 2.4 Evolución del plan de desarrollo de software

El plan se revisará al final de cada iteración y se harán las modificaciones necesarias antes del comienzo de la siguiente fase.

# III Organización del proyecto

## 3.1 Participantes en el proyecto

|  |  |
| --- | --- |
| **Nombre** | Contreras Ortiz José Manuel |
| **Rol** | Programador |
| **Actividades** | Codificación, depuración, correcciones |
| **Contacto** | 6886@utcv.edu.mx |

|  |  |
| --- | --- |
| **Nombre** | Domínguez Reyes Ulrico |
| **Rol** | Tester |
| **Actividades** | Testeo. |
| **Contacto** | 7015@utcv.edu.mx |

|  |  |
| --- | --- |
| **Nombre** | Hernández Chávez Celia |
| **Rol** | Líder |
| **Actividades** | Administración de recursos, creación de planes, dirección del proyecto. |
| **Contacto** | 6770@utcv.edu.mx |

|  |  |
| --- | --- |
| **Nombre** | Morales Argüello Sarahí |
| **Rol** | Programador |
| **Actividades** | Codificación, depuración, correcciones |
| **Contacto** | 7791@utcv.edu.mx |

|  |  |
| --- | --- |
| **Nombre** | Valencia Caballero Ana Fabiola |
| **Rol** | Diseñadora |
| **Actividades** | Diseño de interfaces, corrección de estilos. |
| **Contacto** | 7192@utcv.edu.mx |

|  |  |
| --- | --- |
| **Nombre** | Vera Avilés José Enrique |
| **Rol** | Analista |
| **Actividades** | Encuestas y cuestionarios, levantamiento de requisitos. |
| **Contacto** | 9107@utcv.edu.mx |

## 3.2 Roles y responsabilidades

|  |  |
| --- | --- |
| Rol | Descripción |
| Líder de proyecto | Es el responsable de detectar las necesidades de los usuarios y gestionar los recursos económicos, materiales y humanos, para obtener los resultados esperados en los plazos previstos y con la calidad necesaria. |
| Analista | El analista es alguien que es responsable de entender las necesidades del cliente, y asegurarse de que la solución que está siendo desarrollada se ajusta a esas necesidades. |
| Diseñador | Es el encargado del aspecto estético del sistema, por consecuencia también es responsable directo del grado de usabilidad del mismo. |
| Tester | Se realiza a menudo por los desarrolladores para los aspectos técnicos y los usuarios para los aspectos funcionales, es decir, son los encontrar posibles fallos en el producto de software. |
| Programador | Es el responsable de desarrollar el código de la aplicación de acuerdo a las especificaciones, además es responsable de documentarlo debidamente. |

## 3.3 Organigrama

# IV Gestión del proceso

## 4.1 Fases

El proyecto se llevará a cabo de acuerdo a las fases correspondientes a la metodología indicada con una o más iteraciones entre cada una de ellas, la siguiente tabla muestra los periodos de tiempo estipulados para los mismos.

|  |  |
| --- | --- |
| Fase | Periodo |
| Inicio | Septiembre-Diciembre 2016 |
| Elaboración | Enero-Abril 2017 |
| Construcción | Mayo-Agosto 2017 |
| Transición | Septiembre-Diciembre 2017 |

Los hitos que marcan el final de cada fase se indican en esta tabla.

|  |  |
| --- | --- |
| **Descripción** | **Hito** |
| Fase de inicio | En esta fase se elaborará, aparte del presente documento, el plan de negocio y se definirá el caso de desarrollo. |
| Fase de elaboración | En esta fase se realizará el levantamiento de los requerimientos del sistema, se definirá la estructura del sistema y se generará los primeros prototipos de interfaces. |
| Fase de construcción | En la primera parte de esta fase de elaborará los diagramas de casos de uso, secuencia, clases y se definirá el modelo de datos, posteriormente se realizará la codificación de la aplicación de acuerdo a las especificaciones indicadas, terminando con el plan de integración y el plan de pruebas. |
| Fase de transición | En esta última fase se realizarán las pruebas, se redactarán las notas de versión y se elaborarán los manuales correspondientes, se concluye con la generación de los ejecutables y la evaluación de resultados. |

## 4.2 Calendario de las actividades

A continuación, se presenta un calendario de las principales tareas del proyecto.



