Plan de Pruebas - NoMásAccidentes

Histórico de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Descripción/cambio | autor |
| 1.0.0 | 01/09/2019 |  | Eduardo Obreque |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Propósito del plan de pruebas |
| El propósito del el plan de pruebas es comunicar a todos los involucrados en el proyecto las características a ser probadas del software NoMásAccidentes, daremos también a conocer el alcance que tendrán las pruebas de software, la calendarización de las pruebas y los riesgos que puedan surgir mientras desarrollamos las pruebas con todo esto lo que buscamos es cumplir con las expectativas del cliente para verificar que el software desarrollado cumpla con el criterio de calidad establecido por el cliente. |

|  |
| --- |
| Alcance de las pruebas\* |
| Las pruebas se realizarán a los módulos del software usuarios, capacitaciones, asesorías y revisiones verificando si los métodos, interfaces y cada elemento del formulario cumple con su función establecida.  Los requisitos del ambiente de prueba a tener en cuenta son: las pruebas no se pueden ejecutar en el mismo pc que el desarrollador, los pc donde se realizarán las pruebas deben ser los más similar posible a los pc de producción, los pc deben tener instalado Windows 10, netbeans y visual studio 2017, las pruebas se deben ejecutar en netbeans y visual studio. |

|  |  |  |
| --- | --- | --- |
| Definición de roles y responsabilidades | | |
| Rol | Responsabilidades | Relevancia |
| Eduardo Obreque | Jefe de Proyecto | Alta |
| Mauricio González | Analista Funcional / QA | Alta |
| Simón Chávez | Analista Desarrollador / QA | Alta |

|  |
| --- |
| Tipos de pruebas a realizar |
| Los tipos de pruebas que desarrollaremos para este software son de tipo funcional para verificar las funcionalidades del software, también pruebas no funcionales para verificar la calidad del software y de mantención para verificar que las nuevas funcionalidades al software no hayan afectado a otras partes del sistema. |

|  |
| --- |
| Estrategia y técnicas de pruebas a aplicar |
| En esta oportunidad la más factible que podemos realizar son pruebas de caja gris ya que tenemos que supervisar tanto el código como las funcionalidades para poder cumplir con las expectativas de los clientes, las que más ocuparemos en este proyecto son:  Pruebas orientadas a objetos: Iniciaremos probando el correcto funcionamiento de los módulos para no tener inconvenientes más adelante.  Pruebas de regresión: Vamos a ejecutar el mismo subconjunto de pruebas que ya se han aplicado para asegurarse de que los cambios no han generado efectos colaterales.  Pruebas de Validación:Estas las realizaremos para comprobar que el sistema cumple con los requisitos del cliente.  Pruebas de estrés: Nos ayudaran a probar los límites de nuestro SW y ver en qué casos podemos mejorarlo  Pruebas de Carga: Para lograr ver cuantos usuarios se pueden conectar a la vez  Pruebas de interfaz de usuario: Esta es muy importante porque nuestro SW tiene que ser amigable con el usuario ya que las personas que trabajaran con el no tendrán mucho conocimiento de la computación y tienen que trabajar con él de forma instintiva. |

|  |
| --- |
| Definición del proceso de testing |
| Pruebas Unitarias o de Componente: Este tipo de pruebas son ejecutadas nuestro equipo de desarrollo y logra que nuestro desarrollador verifique que los componentes estén bajo condiciones soporten el ingreso de datos erróneos para poder lograr tratarlos de manera controlada.  Pruebas de Integración: Las ejecutan el equipo de desarrollo para poder comprobar que los elementos del SW interactúen entre sí de manera correcta.  Pruebas de Aceptación: Esto lo ve el cliente ya que él es quien tiene que verificar si el SW diseñado cumple con los requisitos entregados por el para poder así lograr la aceptación y la aprobación de este  Pruebas de Sistema: Esto lo ejecuta el equipo de pruebas idealmente ajeno al equipo de desarrollo y consiste en la ejecución de las pruebas en donde se podemos verificar que la funcionalidad del SW ha sido lograda con éxito de acuerdo con los documentos de especificación definidos con anterioridad en el proyecto. |

|  |
| --- |
| Requerimientos del test |
| Para comenzar necesitamos un ambiente donde podamos trabajar correctamente el SW a diseñar en lo posible que este mismo no esté en el computador personal del desarrollador, lo mejor es que se encuentre en un servidor compartido por lo equipos de pruebas además de esto utilizar nombres de dominio distintos a los de producción. Intentar siempre que el ambiente sea lo más parecido que el de producción ya sea tanto en aplicaciones locales o en las de clientes y realizar si las versiones de estos se encuentran actualizadas para lograr esto se puede apoyar en las herramientas de control de versiones ya que es posible que se esté trabajando paralelamente en otro o en el mismo software y nos ayudara a probar las distintas funcionalidades.  El ambiente de entrega tiene que ser similar al anterior para evitar los problemas a la hora de la instalación o la ejecución del software y poder ver que no existe ningún tipo de error que interfiera con el trabajo diario de los empleados si no que solamente se los facilite. |

|  |
| --- |
| Definición de ciclos de prueba a ejecutar |
| Ciclo de prueba unitaria: asegurar de que cada uno de los módulos funcionen correctamente por separado.  Ciclo de prueba de Integración: Asegurar el correcto funcionamiento del sistema o subsistema que se plantea en cuestión.  Ciclo de prueba de Sistema: Señalar cuáles serán los objetivos a corto o mediano plazo dentro de la fase de desarrollo.  Encargados: Ciclo de prueba unitaria: Eduardo Obreque.  Encargados: Ciclo de prueba integración: Mauricio González.  Encargados: ciclo de prueba sistema: Simón Chávez. |

|  |
| --- |
| Calendarización de las actividades de pruebas |
| Usuarios: tres días, 26/10/18 29/10/18 Eduardo Obreque.  Capacitaciones: tres días, 27/10/18 30/10/18 Emanuel Villanueva.  Asesorías: dos días, 27/10/18 29/10/18 Jordán Cid.  Revisiones: dos días, 28/10/18 30/10/18 Cristian Effica. |
| [Carta Gantt.xlsx](Carta%20Gantt.xlsx) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Resumen de riesgos | | | | | | |
|  | **Fase del proceso de pruebas** | | | | | **Riesgo** |
| **Planificación** | **Análisis y diseño** | **Implementación y ejecución** | **Evaluación** | **Cierre** |
| **Magnitud** |  |  |  |  |  |
|  | Significativo |  |  |  | Requisitos |
|  |  | Critico |  |  | *Verificación de casos de prueba* |
|  |  |  | Moderado |  | *Implementación* |
| Moderado |  |  |  |  | Recursos |
|  |  | Critico |  |  | Tiempo reducido |
|  |  | Moderado |  |  | Falta de coordinación en el equipo de testing. |
|  |  | Alto |  |  | Fallos en el sistema |
|  | Significativo |  |  |  | Diseño, se necesita una mejor arquitectura |
|  |  |  | Leve |  | Tecnologías desconocidas. |

|  |  |
| --- | --- |
| Definir clasificación de los defectos | |
| Nivel de Severidad | Descripción |
| Crítico. | Falla en el sistema. No es posible continuar con el procesamiento. |
| Significativo. | No es posible continuar con el proceso de la función seleccionada |
| Moderado. | Funciones restringidas, pero el procesamiento puede continuar |
| Leve. | Cambio de forma menor |

|  |  |
| --- | --- |
| Definición de artefactos*.* | |
| Artefacto | Descripción |
| Manual de Usuario | Manual de usuario con todas las especificaciones, restricciones, funcionalidades y manera de uso del Software. |
| Software Implementado | El sistema correctamente implementado, e instalado operativamente en la empresa. |

|  |
| --- |
| Condiciones de aceptación para cierre del proceso de pruebas |
| Debe cumplir tajantemente todos los requerimientos que expresó el cliente, en el tiempo indicado. |