DOCUMENTO DE PLAN DE PRUEBAS

1. Plantilla inspirada en el estándar IEEE 829-2008 y adaptada a las necesidades del curso de Construcción de Software

(Plantilla compilada por Ph.D. Franklin Parrales B.)

Tabla de contenido

[1. Introducción 3](#_Toc75630696)

[1.1. Propósito 3](#_Toc75630697)

[1.2. Objetivos 3](#_Toc75630698)

[2. Alcance 3](#_Toc75630699)

[2.1. Alcance de pruebas 3](#_Toc75630700)

[2.2. Limitaciones 4](#_Toc75630701)

[3. Definiciones, siglas, y abreviaturas 4](#_Toc75630702)

[4. Documentos de referencia 4](#_Toc75630703)

[5. Estrategia de pruebas 5](#_Toc75630704)

[5.1. Pruebas funcionales 5](#_Toc75630705)

[5.2. Pruebas de integración 5](#_Toc75630706)

[6. Criterios de entrada y de salida 5](#_Toc75630707)

[6.1. Criterio de entrada del plan de pruebas 6](#_Toc75630708)

[6.2. Criterio de aceptación del plan de pruebas 6](#_Toc75630709)

[6.3. Criterio de suspensión y reanudación 6](#_Toc75630710)

[7. Entregables 6](#_Toc75630711)

[7.1. Protocolo de pruebas 6](#_Toc75630712)

[7.2. Informe de pruebas 6](#_Toc75630713)

[7.3. Reporte de observaciones 7](#_Toc75630714)

[8. Ambiente de pruebas 7](#_Toc75630715)

[8.1. Requerimientos base de hardware 7](#_Toc75630716)

[8.2. Requerimientos base de software en el ambiente de pruebas 7](#_Toc75630717)

[8.3. Herramientas de apoyo para la ejecución de pruebas 8](#_Toc75630718)

[9. Cronograma de trabajo 8](#_Toc75630719)

[10. Riesgos, dependencias, suposiciones y restricciones 9](#_Toc75630720)

[10.1. Riesgos 9](#_Toc75630721)

[10.2. Dependencias 10](#_Toc75630722)

[10.3. Suposiciones 10](#_Toc75630723)

[10.4. Restricciones 10](#_Toc75630724)

[11. Aprobación 11](#_Toc75630725)

# Introducción

## Propósito

El propósito de este documento es garantizar que los requerimientos y especificaciones funcionales definidas en el Proyecto “**Organización de Partidos”**, se encuentren desarrolladas y cumplan con la funcionalidad y estándares de calidad esperados, minimizando los incidentes al momento de realizar las pruebas de aceptación del usuario funcional otorgando así una mayor confiabilidad en el uso del software.

## Objetivos

* Establecer casos y condiciones de prueba basados en las especificaciones funcionales, que al ejecutarse simulen un ambiente de producción.
* Identificar posibles inconvenientes al ejecutar los distintos casos de prueba.
* Tomar en consideración la efectividad de los distintos casos de prueba.
* Determinar las diferencias existentes entre las especificaciones funcionales o requerimientos y el desarrollo del producto.
* Validar y comprobar la calidad del producto a ser entregado, con base a los estándares definidos.

# Alcance

## Alcance de pruebas

|  |  |
| --- | --- |
| **Tipo de pruebas** | **Aplica según el proyecto (si / no)** |
| Pruebas funcionales | Si |
| Pruebas de regresión | Si |
| Pruebas de interfaz de usuario | No |
| Prueba de rendimiento | No |
| Pruebas de carga | No |
| Pruebas de seguridad y control de acceso | Si |
| Pruebas de instalación | Si |
| Pruebas de integración | No |
| Pruebas de caja blanca | No |
| Pruebas de control de bitácoras | Si |

## Limitaciones

# 

* *No se realizarán pruebas de validaciones en módulo contrato.*

# Definiciones, siglas, y abreviaturas

|  |  |
| --- | --- |
| **Siglas** | **Definición** |
| C. C | Creación de contrato del software. |
| R. U. B. D | Registro de usuarios en la base de datos |
| C. A. N | Creación de acta de nacimiento |

# Documentos de referencia

|  |
| --- |
| **DOCUMENTO VERSIÓN** |
| *Documentos de caso de uso* |
| *Especificación de requisitos* |
|  |
|  |

# Estrategia de pruebas

*La Estrategia de Pruebas se encarga de definir y planificar las pruebas que serán realizadas por el Equipo de Testing a las distintas entregas de los proyectos, estableciendo el conjunto de servicios a aplicar en cada proyecto en función de la tecnología, estado y criticidad.*

## Pruebas funcionales

*Las pruebas funcionales se realizan para velar que el producto esperado cumpla con las funciones específicas para los cuales fue creado. Este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, generalmente se requiere apoyo de los usuarios finales ya que ellos pueden aportar mucho en el desarrollo de casos de prueba complejos, enfocados al negocio. En el caso de nuestro proyecto tenemos los siguientes casos de pruebas organizados por módulos.*

|  |  |
| --- | --- |
| ***Código*** | CP-001 |
| ***Caso de prueba*** | Registrar Equipo |
| ***Responsable*** | Desarrolladores |
| ***Descripción de la prueba*** | * **Se ejecuta el método:**   + inicioTest() * **Se inicializan los datos:**   + user = “john”   + password = “1234” * **Se establecen los datos de la Conexión con el método:**   + Conexión.iniciarDatos(user,password) * **Se crea una instancia de ControlEquipo con el constructor parametrizado:**   + instance = new ControlEquipo(“Guayaquil-FC”, ”Ecuador”) * **Se ejecuta el método:**   + instance.registrarEquipo() * **Se presenta en un recuadro el mensaje:**   ¡Se registró correctamente el Equipo! |
| ***Requisito previo*** | Se debe tener permisos de creación en la Base de Datos |
| ***Resultado esperado*** | Se registrará el Equipo en la Base de Datos |
| ***Resultado obtenido*** | Se registró el Equipo en la Base de Datos |
| ***Estado*** | Exitoso |
| ***Observaciones*** |  |

|  |  |
| --- | --- |
| ***Código*** | CP-002 |
| ***Caso de prueba*** | Establecer Conexión |
| ***Responsable*** | Desarrolladores |
| ***Descripción de la prueba*** | * **Se inicializan los datos:**   + user = “john”   + password = “1234” * **Se establecen los datos de la Conexión con el método:**   + Conexión.iniciarDatos(user, password) * **Se ejecuta el método y se obtiene un objeto tipo *Connection*:**   + result = Conexión.establecerConexion() * **Se comprueba el objeto obtenido, con el método:**   + Conexión.validarConexion(result) * **Se presenta en un recuadro el mensaje:**   + ¡Conectado Exitosamente! |
| ***Requisito previo*** | Se debe tener permisos de conexión en la Base de Datos |
| ***Resultado esperado*** | Se establecerá la conexión a la Base de Datos |
| ***Resultado obtenido*** | Se estableció la conexión a la Base de Datos |
| ***Estado*** | Exitoso |
| ***Observaciones*** | * La prueba unitaria puede presentar el estado de *Failed* en el caso de que se comparen dos objetos de tipo *Connection* distintos. |

|  |  |
| --- | --- |
| ***Código*** | CP-003 |
| ***Caso de prueba*** | Eliminar Equipo |
| ***Responsable*** | Desarrolladores |
| ***Descripción de la prueba*** | * **Se ejecuta el método:**   + inicioTest() * **Se inicializan los datos:**   + user = “john”   + password = “1234” * **Se establecen los datos de la Conexión con el método:**   + Conexión.iniciarDatos(user, password) * **Se inicializan el nombre del Equipo a eliminar:**   + nombreBuscado = “Guayaquil-FC” * **Se crea una instancia de ControlEquipo con el constructor sin parámetros:**   + instance = new ControlEquipo() * **Se ejecuta el método:**   + instance.eliminarEquipo(nombreBuscado) * **Se presenta en un recuadro el mensaje:** * ¡Se eliminó correctamente el Equipo! |
| ***Requisito previo*** | Se debe tener permisos de eliminación en la Base de Datos |
| ***Resultado esperado*** | Se eliminará el Equipo en la Base de Datos |
| ***Resultado obtenido*** | Se eliminó el Equipo en la Base de Datos |
| ***Estado*** | Exitoso |

|  |  |
| --- | --- |
| ***Código*** | CP-004 |
| ***Caso de prueba*** | Buscar Equipo |
| ***Responsable*** | Desarrolladores |
| ***Descripción de la prueba*** | * **Se ejecuta el método:**   + inicioTest() * **Se inicializan los datos:**   + user = “john”   + password = “1234” * **Se establecen los datos de la Conexión con el método:**   + Conexión.iniciarDatos(user,password) * **Se inicializan el nombre del Equipo a buscar:**   + nombreBuscado = “Guayaquil-FC” * **Se crea una instancia de ControlEquipo con el constructor sin parámetros:**   + instance = new ControlEquipo() * **Se ejecuta el método:**   + instance.buscarEquipo(tablaEquipo ,nombreBuscado) * **Se presenta en un recuadro el mensaje:**   + ¡Se encontró correctamente al Equipo! |
| ***Requisito previo*** | Se debe tener permisos de búsqueda en la Base de Datos |
| ***Resultado esperado*** | Se buscará al Equipo en la Base de Datos |
| ***Resultado obtenido*** | Se encontró al Equipo en la Base de Datos |
| ***Estado*** | Exitoso |

|  |  |
| --- | --- |
| ***Código*** | CP-005 |
| ***Caso de prueba*** | Modificar equipo |
| ***Responsable*** | Desarrolladores |
| ***Descripción de la prueba*** | * **Se ejecuta el método:**   + inicioTest() * **Se inicializan los datos:**   + user = “john”   + password = “1234” * **Se establecen los datos de la Conexión con el método:**   + Conexión.iniciarDatos(user,password) * **Se inicializan el nombre del Equipo a modificar:**   + nombreBuscado = “Guayaquil-FC” * **Se crea una instancia de ControlEquipo con el constructor sin parámetros:**   + instance = new ControlEquipo() * **Se ejecuta el método:**   + instance.modificarEquipo(nombreBuscado) * **Se presenta en un recuadro el mensaje:**   + ¡Se modificó correctamente al Equipo! |
| ***Requisito previo*** | Se debe tener permisos de actualización en la Base de Datos |
| ***Resultado esperado*** | Se modificará al Equipo en la Base de Datos |
| ***Resultado obtenido*** | Se modificó al Equipo en la Base de Datos |
| ***Estado*** | Exitoso |

## Pruebas de integración

*La prueba de integración busca verificar que los módulos o componentes que lo conforman funcionan correctamente en conjunto o integrados. Las pruebas de integración en los proyectos de desarrollo de software, no solo se presentan en la integración entre módulos de una misma aplicación de software sino entre varias aplicaciones de software.*

# Criterios de entrada y de salida

*Es de suma importancia tener definidos apropiadamente estos criterios y comunicarlos a toda la organización para que no haya ninguna duda sobre los mismos y que todos deben cumplir.*

## Criterio de entrada del plan de pruebas

*Para contar que vamos por un buen camino tenemos:*

*Todo el paquete de código fuente está completo y ha sido revisado informalmente por el equipo de trabajo.*

*Todo el desarrollo del programa está respaldado en la nube.*

*Nuestro entorno de trabajo es el adecuado*

*Todas las secciones de trabajo están designadas*

## Criterio de aceptación del plan de pruebas

*Para contemplar que los requisitos de aceptación son los adecuados tenemos que:*

*100% de la ejecución de las pruebas especificadas al inicio de la iteración son completadas exitosamente.*

## Criterio de suspensión y reanudación

*Para saber si se necesita arreglar o suspender un diseño tenemos:*

# Entregables

*Dentro del proceso de pruebas se contemplan los siguientes entregables.*

## Protocolo de pruebas

*El documento de protocolo de pruebas contiene los casos de pruebas que se realizarán y los resultados.*

## Informe de pruebas

*El informe de pruebas contendrá la evaluación de resultados con los siguientes puntos:*

* *Resumen de pruebas, el cual incluye la identificación del caso de prueba y el resultado de la ejecución.*
* *Acciones sugeridas.*
* *Análisis de Resultados.*

## Reporte de observaciones

*Los reportes de observaciones serán elaborados para todos los bugs que se encuentren durante las pruebas, se incluirá su severidad y su priorización para la solución a través del líder de proyecto o por parte de ingeniería de requerimientos si el documento de caso de uso debe ser actualizado. Cualquier modificación a los requerimientos debe ser hecha a través de solicitudes de cambios, las cuales aseguran que el cambio propuesto sea revisado antes de ser incorporado en el producto.*

# Ambiente de pruebas

*Describe los recursos de infraestructura donde se llevarán a cabo las pruebas de la solución tecnológica desarrollada, este ambiente debe ser igual a producción.*

## Requerimientos base de hardware

*Elementos de hardware necesarios para poder realizar las pruebas, así como la cantidad requirente y sus especificaciones.*

|  |  |  |
| --- | --- | --- |
| **Recurso** | **Cantidad** | **Nombre y tipo** |
| *Equipos móviles* | *2* | *Aleatoria* |
| *Servidor de base de datos* | *1* | *Oracle, SQL, otras.* |
| *Equipos de escritorio* | 2 | Windows, Apple, otras. |

## Requerimientos base de software en el ambiente de pruebas

*La siguiente tabla define los elementos de software requeridos en el ambiente de pruebas.*

|  |  |  |
| --- | --- | --- |
| **Elemento de software** | **Versión** | **Tipo** |
| *Oracle* | 5.0 | SMBD |
| *Apache NetBeans* | *12.4* | *Java* |

## Herramientas de apoyo para la ejecución de pruebas

*La siguiente tabla define las herramientas utilizadas para soportar el proceso de pruebas de este plan.*

|  |  |  |
| --- | --- | --- |
| **Elemento de software** | **Versión** | **Tipo** |
| *Excel* |  | *Administración de Pruebas.* |
| *Project* |  | *Administración del Proyecto.* |
| *Otras.* |  |  |

# Cronograma de trabajo

*Generar un cronograma que incluya las actividades a ser realizadas durante la ejecución de las pruebas. Detalle además los principales hitos que determinan el avance de las pruebas.*

*El cronograma deberá elaborarse en Microsoft Project u otro software de vuestra elección y ser entregado al líder de Proyecto para la inclusión de los tiempos dentro del Plan de Proyecto.*



# Riesgos, dependencias, suposiciones y restricciones

*Se detallarán los factores de riesgo, dependencias, suposiciones y restricciones que contempla el plan de pruebas. Para cada subsección se ha incorporado ejemplos de guía.*

## Riesgos

*Describa los factores de Riesgos para el presente Plan de Pruebas, su estrategia de mitigación y contingencia.*

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Estrategia de mitigación** | **Contingencia** |
| *Falta de tiempo* | *Aplicar al cronograma planteado inicialmente una adición de tiempo equivalente al 10% sobre el tiempo inicialmente pactado* | *Contratar más recursos* |
| *Atrasos en corrección de errores* | *Dar prioridad a errores funcionales y bloqueantes que impidan la continuación de las pruebas* | *Contratar más personal en desarrollo* |
| *Plan de Pruebas deficiente* | *Para el caso en que algún probador ejecute mal un plan de pruebas, se debe contar con un segundo probador que valide la correcta aplicación de las estrategias de pruebas, en puntos aleatorios.* | *Negociar previamente una adición de un 10% más de tiempo a la fase de pruebas inicialmente planeada* |
| *Otras.* |  |  |

## Dependencias

*Describa los factores de Dependencia con otras áreas y su impacto.*

|  |  |
| --- | --- |
| **Dependencia con otras áreas** | **Impacto de la dependencia** |
| *Interdependencia entre proyectos.* | *Medio* |
| *Accesos (Permisos) a otros sistemas.* | *Medio* |
| *Comunicación con sistemas externos.* | *Bajo* |
| *Comunicación con sistemas internos.* | *Alto* |
| *Otras.* |  |

## Suposiciones

*Describa las suposiciones y su impacto.*

|  |  |
| --- | --- |
| **Suposición a ser probada** | **Impacto de suposición incorrecta** |
| *El ambiente de pruebas debe contar con las especificaciones mínimas de hardware y software.* | * *Pruebas deficientes.* * *Caídas frecuentes.* * *Atrasos en el cronograma.* |
| *Requerimientos funcionales depurados y consistentes.* | * *Pruebas inconsistentes.* * *Atrasos en el cronograma.* * *Redefinición de requerimientos.* * *Cambios en otros módulos del sistema.* |
| *Datos de pruebas suficientes.* | * *Inversión de tiempo en la generación de datos de prueba.* * *Atraso en el cronograma.* * *Inconsistencia en las pruebas.* |
| *Otras.* |  |

## Restricciones

|  |  |
| --- | --- |
| **Restricciones** | **Impacto de la restricción** |
| *La fecha límite para la finalización del plan de pruebas es la primera semana del mes de…* | * *Iniciar a tiempo el paso a producción de la solución* * *Incumplimiento en la entrega del proyecto.* |
| *El máximo número de recursos disponibles para pruebas es de XX.* | * *Atrasos en el cronograma.* |
| *Utilización de herramientas libres y/o licenciadas.* | * *Incurrir en sanciones por efectos de utilización de software pirata* * *Herramientas con funcionalidades limitadas o nulas* * *Herramientas con fecha de caducidad* |
| *Otras.* |  |

# Aprobación

(Fecha)

|  |  |
| --- | --- |
| Elaborado por: | Revisado por: |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Nombre  Cargo | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Nombre  Cargo |
| Aprobado por:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Nombre  **Líder del Proyecto** | Aprobado por:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Nombre  **Director de la Unidad de Tecnologías de la Información y Comunicaciones de la Empresa X** |