Propuesta de Colaboración Profesional

**Plan de Pruebas – *IEEE 829***

Proyecto: MASTER DETAIL.

06/12/2016

Preparado por:Romero Ruiz Cristina

**ÍNDICE TEMÁTICO**

**CONTENIDOS Nº Pág.**

[**1** **VERSIÓN DEL PLAN 4**](#_gjdgxs)

[**2** **ALCANCE 4**](#_30j0zll)

[**3** **ESTRATEGIAS 4**](#_1fob9te)

[*3.1.1* *Pruebas Unitarias 4*](#_3znysh7)

[*3.1.2* *Pruebas de Integración 5*](#_2et92p0)

[*3.1.3* *Pruebas de Arquitectura Cliente-Servidor 5*](#_tyjcwt)

[3.1.3.1 Prueba de Base de Datos 5](#_3dy6vkm)

[3.1.3.2 Pruebas de Servidor 5](#_1t3h5sf)

[3.1.3.3 Prueba de Transacción y Comunicación de Red 6](#_4d34og8)

[*3.1.4* *Pruebas de Sistema 6*](#_2s8eyo1)

[3.1.4.1 Pruebas de Validación 6](#_17dp8vu)

[3.1.4.2 Pruebas de Verificación 7](#_lnxbz9)

[*3.1.5* *Pruebas de Aceptación 8*](#_35nkun2)

[**4** **CATEGORIZACIÓN DE LA CONFIGURACIÓN 8**](#_2jxsxqh)

[4.1 Criterios de Aceptación 9](#_z337ya)

[**5** **RECURSOS 10**](#_3j2qqm3)

[**6** **CALENDARIO 11**](#_2xcytpi)

[*6.1.1* *Grafo de Dependencias 11*](#_3whwml4)

[**7** **RIESGOS 11**](#_2bn6wsx)

[8 Responsables 12](#_3as4poj)

**ÍNDICE DE TABLAS**

[Tabla 1 - Alcance 4](#_1fob9te)

[Tabla 2 - Pruebas de Validación 7](#_3rdcrjn)

[Tabla 3 - Pruebas de Validación 7](#_26in1rg)

[Tabla 4 - Tabla Aceptación Prueba 8](#_44sinio)

[Tabla 5 - Categorización de la Configuración 9](#_z337ya)

[Tabla 6 - Recursos Pruebas Unitarias 10](#_1y810tw)

[Tabla 7 - Recursos Pruebas de Aceptación 10](#_4i7ojhp)

[Tabla 8 – Calendario 11](#_1ci93xb)

[Tabla 9 - Riesgos 12](#_qsh70q)

[Tabla 10 - Responsables 12](#_1pxezwc)

1. **Versión del plan**

Versión: ***PlanPruebas\_v1.doc.***

1. **Alcance**

A continuación se indican los tipos de pruebas que serán realizadas tanto en el código del software, como también la cohesión que existe entre cada uno de los mantenedores y la coherencia que debe existir según el diseño realizado.

|  |  |
| --- | --- |
| **Tipos de Pruebas** | **Elementos que se ven afectados** |
| Unitarias | Parte del código perteneciente al Módulo registro y venta. |
| Integración | Todos los módulos mencionados funcionando correctamente juntos. |
| Arquitectura Cliente -Servidor | Probar que la base de datos, transacciones y comunicación en la red funcionan. |
| Sistema | Informe Especificación de Requerimientos (IEEE 830) y el Software. Informe de Diseño (IEEE 1471) y Software. |
| De Aceptación | Software |

**Tabla 1 - Alcance**

1. **Estrategias**

* + 1. **Pruebas Unitarias**

Se realizarán pruebas unitarias a medida que se obtenga un pequeño avance respecto a los diferentes módulos, haciendo uso de la técnica de programar y probar (desarrollo común). Junto a esto, una vez finalizada una porción de código, se utilizará la técnica de valores límites para probar que la parte del programa funciona correctamente, es decir, al ingresar los valores extremos que puede soportar un cierto intervalo, funciona y muestra los mensajes de alerta correspondientes, junto también con pruebas de bucles y condiciones. Las pruebas unitarias se realizarán de acuerdo a la programación en el calendario posteriormente indicado.

Se identificarán los casos especiales para ser probados:

* Probar siempre los límites.
* Al tener listas, vectores y tablas:

1. Probar listas de un solo valor y listas vacías.
2. Probar siempre distintos tamaños.
3. Comprobar primer elemento, el elemento central y el último elemento de los elementos mencionados.

* Pasar *null* en vez del objeto.

Solo se documentan las pruebas realizadas a los módulos terminados y estas deben tener el resultado esperado y el obtenido, junto con la fecha realizada, la descripción de la prueba y su tipo (Unitaria, Integración) deben ir reflejadas en una tabla de pruebas, ubicada en el directorio de pruebas con el nombre *TablaCasosPruebas\_v1.xlsx.*

* + 1. **Pruebas de Integración**

Las pruebas de integración no se podrán realizar hasta que **el software esté finalizado** por lo menos dos módulos del software. Los resultados obtenidos se documentará en la misma tabla de prueba que las pruebas Unitarias. Para realizar esto, se hará uso de la técnica de Integración ascendente *(Bottom to Top),* es decir, se integran los módulos moviéndose en dirección jerárquica de menor a mayor, comenzando por el modulo principal (Modulo Ingreso). Para seguir posteriormente con los otros módulos del software y llegar al más importante que es el Módulo Orden de Trabajo.

Para probar se pide al programador utilizar además las pruebas de particiones las cuales son de caja negra. Estas consisten en dividir las posibles entradas y salidas en conjuntos de características similares.

* + 1. **Pruebas de Arquitectura Cliente-Servidor**

Permite probar la arquitectura aplicada en el Software MASTER, la cual se realizará una vez que se realice la implementación del Software y Base de Datos en la organización Uthermosillo Esta prueba se divide en tres:

* + - 1. **Prueba de Base de Datos**

La finalidad es probar la integridad y exactitud de los datos que van siendo ingresados por medio del software por medio de los usuarios Administrador y Usuarios. Esto servirá para comprobar si los datos son almacenados, borrados, actualizados o bien recuperados de forma apropiada.

* + - 1. **Pruebas de Servidor**

Esta prueba permitirá probar el desempeño del servidor respecto al tiempo de respuesta que tiene en cada transacción, como también en el procesamiento de los datos.

* + - 1. **Prueba de Transacción y Comunicación de Red**

Por un lado se verificarán las transacciones que se realizarán a través del software como el inicio de sesión de algún usuario, como también obtener datos de la Base de Datos y manipularlos como por ejemplo creando informes.

Por otro lado, la prueba de comunicación de red será básicamente estableciendo un inicio de sesión y obteniendo los mensajes satisfactorios programados en el software.

* + 1. **Pruebas de Sistema**

Estas pruebas se subdividen en dos tipos de prueba, por un lado de validación y otro de verificación.

* + - 1. **Pruebas de Validación**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombres de las Pruebas** | **Descripción de la Prueba** | | **¿Por qué se realiza o no la prueba?** |
| REQ01 | Realizar el ingreso de órdenes de registro. Realizar el procedimiento 5 veces. | | Fundamentalmente para determinar que el problema principal que se le quiere solucionar al cliente funcione, en este caso, que los registros se realicen y que no importa el número de intentos. |
| REQ02 | Los usuarios con el perfil de “cajero” solamente pueden ingresar a los módulos de registro. | | Usuarios sólo deben acceder a estos módulos. NO son administradores. |
| REQ03  REQ04  REQ05  REQ06  REQ07 | Estas pruebas se enfocan en los módulos de proveedores, usuarios, productos, probar que la inserción, búsqueda, actualización y eliminación de los registros funciona correctamente. | | Esta prueba se realiza para revisar que el Módulo de Usuarios funciona correctamente. |
| REQ09 | Creación de un nuevo usuario y comprobar que este funcione y pueda cumplir su labor. |  | |
| REQ010 | Probar que la venta se realice correctamente. | De suma importancia debido a que uno de los objetivos del software es controlar esto. | |

|  |  |  |
| --- | --- | --- |
| NOMBRES DE PRUEBAS | DESCRIPCION | PORQUE SE REALIZA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R-NF1 | | Verificar que la instalación del software sea a la planteada en el Informe de Requerimientos de Software. | | Verificar el requerimiento de Mínimo Conocimiento de Computación |
| R-NF2 | | Crear 2 usuarios con los diferentes perfiles para probar que las acciones que puedan realizar sean las correctas de acuerdo a los casos de Uso. | | Verificar Fiabilidad del Módulo Ingreso |
| R-NF3 | | Verificar que el software pueda ser instalado a través de un CD. | |  |
| R-NF4 | | Realizar una matriz que verifique si el software está instalado correctamente. | |  |

**Tabla 3 - Pruebas de Validación**

* + - 1. **Pruebas de Verificación**

Se comprueba y corrobora que la construcción del software corresponde a lo planteado en el documento de diseño *(IEEE 1471)*, verificando que los componentes de software cumplan los distintos diagramas diseñados. Por otra parte, se comprobará que las actividades presentes en el software correspondan a actividades diseñadas anteriormente, verificando de esta forma que las actividades sean las que se encuentran en el diseño.

* + 1. **Pruebas de Aceptación**

Son realizadas por el cliente, el cual deberá indicar los casos de pruebas realizados junto con el indicador de pruebas superadas para la aceptación del software e irán en un anexo del documento de pruebas *TablaCasosPruebas\_v1.xlsx.*

Una prueba de Aceptación es aprobada cuando se haya firmado el formulario de pruebas por el supervisor de la prueba por parte del cliente y también por el jefe del proyecto. El formulario contiene la siguiente estructura:

|  |
| --- |
| Responsable: |
| Fecha: |
| Descripción: |
| Caso de Pruebas: |
| Resultado Esperado: |
| Resultado Obtenido: |
| Observaciones: |
| Estado:  Aprobado \_\_  Rechazado \_\_  Firma Cliente \_\_\_\_\_\_\_\_\_\_\_ |

1. **Categorización de la configuración**

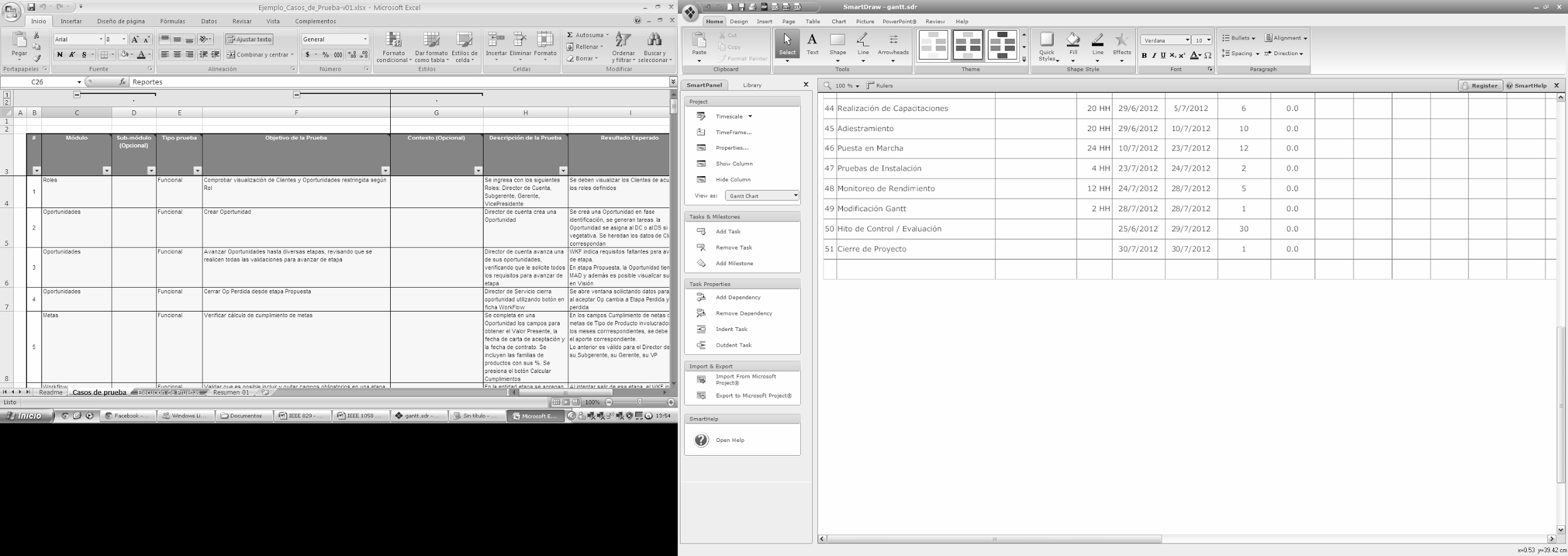
Esta categorización existe para las Pruebas de Integración, Sistema y Aceptación:

|  |  |
| --- | --- |
| **Estado** | **Condiciones** |
| Suspendido | Se suspenderá el seguir aplicando el plan de pruebas en el caso que:   * El proyecto no pueda continuar * Atraso necesario para comenzar una etapa. Por ejemplo, que para las pruebas de integración no se hayan realizado las pruebas unitarias (ver grafo de dependencia presentado posteriormente). |
| Repetido | Se repetirá el todo el plan de pruebas en el caso de falla de alguna de las pruebas.  En el caso de hacer modificaciones al código, por el resultado de alguna prueba, se realizarán las pruebas nuevamente. |
| Culminado | Se dará por finalizado o culminado la aplicación del plan cuando se obtenga en *TablaCasosPruebas\_v1.xls* un resultado del 100% de la aplicación de estas y de forma correcta. |

**Tabla 5 - Categorización de la Configuración**

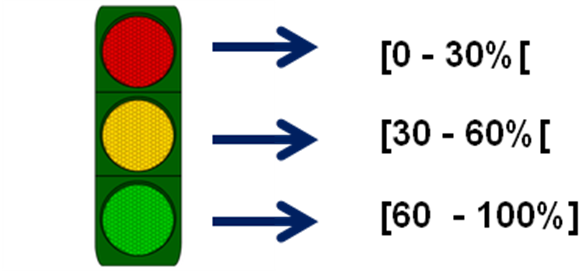
* 1. ***Criterios de Aceptación***

Cada prueba será llevada a la tabla *TablaCasosPruebas\_v1.xlsx,* dentro de la cual se debe definir el tipo de prueba que se realizará y el objetivo que esta tiene.



Una vez realizadas las pruebas, se definirá el ciclo de cada prueba realizada junto a la fecha planificada para su ejecución, el encargado a cargo de la prueba y el porcentaje de éxito que esta tenga junto a sus observaciones.

Se considerará que una prueba debe repetirse 3 veces si su porcentaje de éxito se encuentra en el color rojo, 2 veces en el color amarillo y 1 vez en el color verde.



1. **Recursos**

Los recursos de hardware para realizar las pruebas son:

Para las pruebas unitarias, integración y de casos de uso. Respecto al Lenguaje de Programación y Base de Datos utilizada, se encuentra especificado en el documento *IEEE 1471.*

|  |  |
| --- | --- |
| Nombre | Samsung |
| Procesador |  |
| Memoria RAM | 2 GB |
| Disco Duro | 500 GB |
| SO | Windows 7 |

Para las Pruebas de Aceptación que el cliente realizará, se cuenta con el siguiente equipo:

|  |  |
| --- | --- |
| Nombre | DELL |
| Procesador | Intel® CORE i7 |
| Memoria RAM | 6 GB |
| Disco Duro | 1 TB |
| SO | Windows 10 PRO |

1. **Calendario**

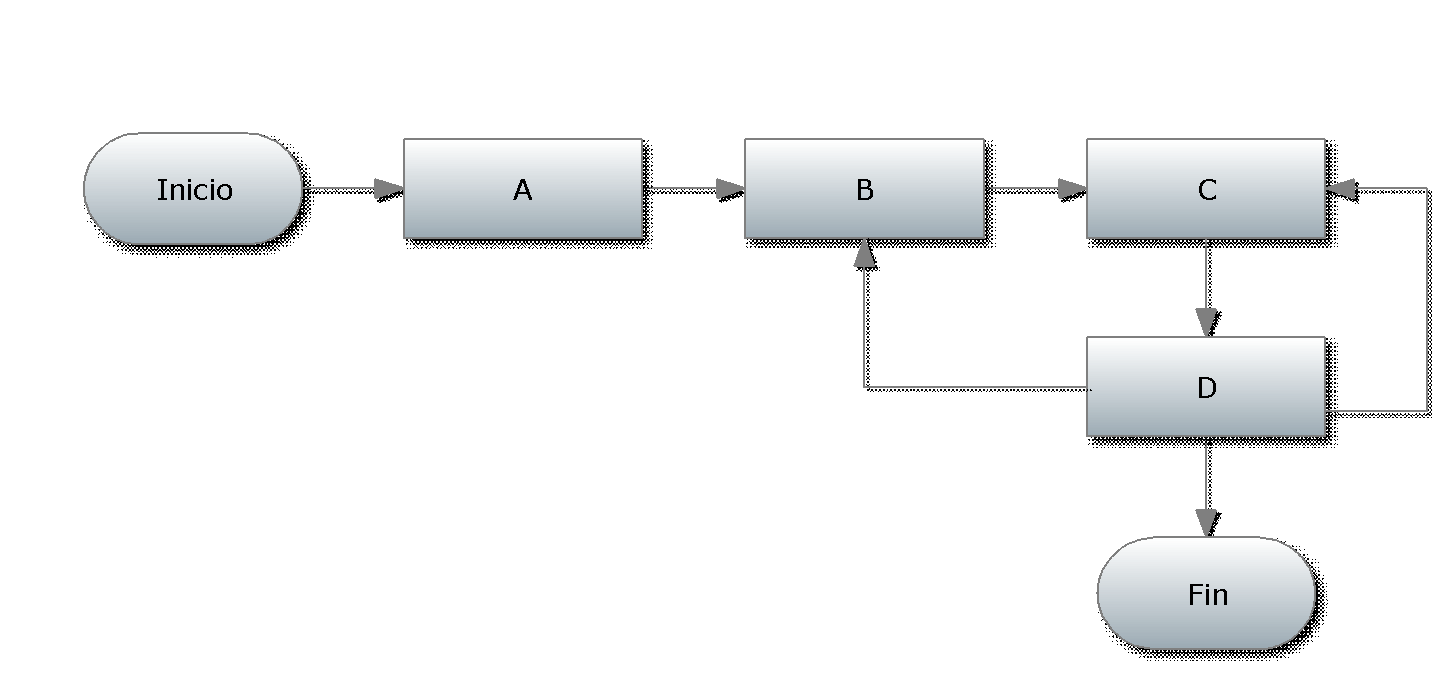
Se presenta a continuación la calendarización de las pruebas junto a su duración estimada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificador** | **Prueba** | **Fecha inicio** | **Fecha termino** | **Duracion (Horas)** |
| A | Unitarias | 10/NOV/16 | 11/NOV/2016 | 10 |
| B | Integración | 12/NOV/2016 | 23/NOV/16 | 14 |
| C | Sistema | 13/NOV/16 | 14/NOV/16 | 6 |
| D | Aceptación | 14/NOV/16 | 15/NOV/16 | 3 |

**Tabla 8 – Calendario**

* + 1. **Grafo de Dependencias**

Lo anterior demostrado en un gráfico para un mayor entendimiento.



1. **Riesgos**

|  |  |
| --- | --- |
| **Identificación del Riesgo** | IRP1 |
| **Descripción del Riesgo:** | Atraso en la codificación, por lo tanto, en el inicio de alguna de las pruebas |
| **Probabilidad de Ocurrencia:** | Baja |
| **Impacto en caso de ocurrir:** | Alto |
| **Responsable de Mitigarlo:** | Jefe de Proyecto(Romero) |
| **Mitigación del Riesgo:** | * Control con la carta Gantt en la actividad de codificación. |
| **Plan de Contingencia:** | * Pedir asesoría a los profesores guías * Pedir más tiempo en la entrega de proyecto final |
| **Identificación del Riesgo** | IRP2 |
| **Descripción del Riesgo:** | Sobrepasar el tiempo destinado a las pruebas |
| **Probabilidad de Ocurrencia:** | Baja |
| **Impacto en caso de ocurrir:** | Alto |
| **Responsable de Mitigarlo:** | Jefes de Proyecto(Romero) |
| **Mitigación del Riesgo:** | * Control con la carta Gantt en la actividad de codificación. * Realizar las pruebas en el menor tiempo necesario y corregir los problemas que se presenten menor al tiempo estimado |
| **Plan de Contingencia:** | * Acortar los tiempos de implantación para designar ese tiempo a la etapa de pruebas |
| **Identificación del Riesgo** | IRP3 |
| **Descripción del Riesgo:** | No disponer de los recursos físicos especificados |
| **Probabilidad de Ocurrencia:** | Baja |
| **Impacto en caso de ocurrir:** | Alto |
| **Responsable de Mitigarlo:** | Jefes de Proyecto(Romero) |
| **Mitigación del Riesgo:** | * Una semana antes de comenzar la etapa de pruebas realizar un chequeo a los computadores. En el caso que requieran un formateo realizarlo para no tener problemas. |
| **Plan de Contingencia:** | * En caso de presentar problemas con contar con ellos, conseguirse un equipo de la Universidad para realizar las pruebas del software. |

1. **Responsables**

A continuación se presentan los responsables de llevar a cabo cada tarea:

|  |  |
| --- | --- |
| **Tarea** | **Responsable dentro del equipo de trabajo** |
| A | Romero |
| B | Romero |
| C | Romero |
| D | .Romero |
| E | Romero |