**PLAN Y ESTRATEGIA DE PRUEBAS DEL SISTEMA**

Contenido

[1. Propósito del documento de pruebas del sistema 2](#__RefHeading__427_1425452620)

[2. Alcance 2](#__RefHeading__429_1425452620)

[3. Estrategia 2](#__RefHeading__431_1425452620)

[3.1 Definición Estrategia 2](#__RefHeading__433_1425452620)

[4. Plan de Pruebas 3](#__RefHeading__435_1425452620)

[4.1 Diseño conceptual: 3](#__RefHeading__437_1425452620)

[4.2 Materiales para el desarrollo de pruebas: 4](#__RefHeading__439_1425452620)

[4.3 Tiempo de pruebas 5](#__RefHeading__441_1425452620)

[4.4 Pruebas a realizar 5](#__RefHeading__443_1425452620)

[4.5 Resultados esperados: 6](#__RefHeading__445_1425452620)

**1. Propósito del documento de pruebas del sistema**

El propósito principal del plan y la estrategia de pruebas del sistema es garantizar que los requerimientos cumplan con la funcionalidad requerida y la con la calidad mínima del componente.

**2. Alcance**

El actual plan de pruebas y estrategia satisface los requerimientos para el ciclo 2 del proyecto y se ajusta al tamaño de la solución propuesta.

**3. Estrategia**

Revisar las estrategias propuestas por TSPi para el realizar la planeación y desarrollo de pruebas, que permitan al grupo definir la funcionalidad del producto según las necesidades del cliente, a su vez deberá permitir saber la calidad del sistema con respecto a los niveles exigidos por el equipo de trabajo y su funcionalidad en condiciones normales y adversas.

3.1 Definición Estrategia

* **Objetivos del sistema - Ciclo 2**

* Calcular los data bindings con el ambiente externo y los data binding de cada una de sus funciones.
* Calcular los data bindings con el entorno externo de cada procedimiento.
* Calcular el acoplamiento del programa y sus funciones.
* Calcular la complejidad ciclomatica de McCabe del programa.
* Calcular la complejidad ciclomatica de McCabe de los procedimientos.
* Calcular Fuerza de los data bindings para el programa y sus funciones.
* Calcular el índice de empaquetamiento/robustes
* **Especificación de la estrategia**

Tomando en cuenta que el aplicativo que se desarrollará en este segundo ciclo no es de gran tamaño y buscando una estrategia efectiva que disminuya en su gran mayoría los problemas al realizar la integración se opta por tomar la estrategia “The One-at-a-time Strategy” enfatizando en los siguientes puntos:

* Validar los métodos principales con pruebas unitarias.
* Realizar inspecciones de código según CheckList propuesto.
* Obtener 2 partes que componen el software e ir añadiendo una tercera, cuarta y quinta parte hasta su totalidad y por cada adición realizar una prueba que permita determinar algún problema.
* Por cada prueba realizada hacer análisis en ambientes normales y adversos, revisar usabilidad, rendimiento y calidad.

**4. Plan de Pruebas**

Las pruebas de software son parte fundamental del desarrollo ya que logra minimizar en gran cantidad los errores expuestos al usuario, proporcionan una mayor calidad y disminuyen los tiempos y costos de la realización software. Debido a esto se ha definido el siguiente plan de pruebas;

4.1 Diseño conceptual:

1. Todas las pruebas que se realizarán serán de caja negra
2. Pruebas unitarias, cada parte del producto debe ser probada, validando que cumpla con los requerimientos y realice su fin de forma adecuada.
3. Pruebas de integración, la estrategia a utilizar para las pruebas de integración es “the one-at-a-time”

4.2 Materiales para el desarrollo de pruebas:

Cada prueba realizada debe estar soportada por un script en el cual se documentaran los resultados de “Pruebas Unitarias - Pruebas de Integración”, a través del formato que se muestra a continuación:

Sample Test Log (Form Logtest)

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Instructor:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part/Level: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cycle: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Test/Phase | Product | Start | Stop | Interruption Time | Delta Time | Problems | Comments |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

4.3 Tiempo de pruebas

* *Pruebas Unitarias*

Durante la fase de implementación se tomará un tiempo de una hora/integrante del grupo para realizar las pruebas unitarias de cada uno de los componentes entregados.

* *Pruebas de integración*

Durante la fase de integración se realizarán las pruebas de integración basados en el estándar “the one-at-a-time” y se dedicara el 40% (12 horas grupo) del tiempo de la tarea de implementación a la realización de estas.

4.4 Pruebas a realizar

Las pruebas durante el ciclo 1, se realizarán a los componentes listados a continuación:

1. RespuestaConteo
2. Metodo
3. Clase

Según la estrategia “The one-at-a-time”, los pasos para la integración de los componentes está dado por:

* Metodo - Clase
* Metodo - Clase - RespuestaConteo

4.5 Resultados esperados:

Mitigar los riesgos debidos a:

* Partes de baja calidad.
* Componentes fallidos.
* Componentes que no cumplan su objetivo.
* Controlar la integración de las partes del sistema.
* Generar un adecuado control de cambios.