[Nombre del proyecto]

Modelo de Construcción y Pruebas de Software

Versión [1.0]

[Este documento es la plantilla base para elaborar el documento Modelo de Construcción y pruebas de Software. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda. Para actualizar la tabla de Contenido, haga clic con el botón derecho del ratón sobre cualquier línea del contenido de la misma y seleccione *Actualizar campos*, en el cuadro que aparece seleccione *Actualizar toda la tabla* y haga clic en el botón Aceptar]

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| [dd/mm/aaaa] | [x.x] | [detalles] | [nombre] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[Nombre del proyecto] 1

Modelo de Diseño Construcción de Software 1

Versión [1.0] 1

Historia de revisiones 1

Contenido 2

1. Introducción 3

1.1. Propósito 3

1.2. Alcance 3

1.3. Definiciones, siglas y abreviaturas. 3

1.4. Referencias 3

1.5. Visión general 3

2. Diseño de Casos de Uso 3

2.1. Diseño del Caso de Uso [nombre del caso de uso 1] 3

2.1.1. Diagrama de paquetes 3

2.1.2. Diagrama de Interacción 3

2.1.3. Diseño de Flujo de eventos 3

2.1.4. Requerimientos especiales o de implementación 4

2.2. Diseño del Caso de Uso [nombre del caso de uso 2] 4

3. Diseño de Objetos 4

3.1. [Objeto 1] 4

3.2. [Objeto 2] 4

4. Diseño de Subsistemas 4

4.1. Subsistemas Específicos 4

4.1.1. [Nombre del Subsistema Específico 1] 4

4.1.2. [Nombre del Subsistema Específico 2] 5

4.2. Subsistemas de Soporte 5

4.2.1. [Nombre del Subsistema de soporte 1] 5

4.2.2. [Subsistema de soporte 2] 5

5. Revisión de la Interfaz de Usuario 5

5.1. Formatos individuales de interfaz de pantalla 6

5.2. Controles y elementos de diseño de interfaz de pantalla 6

5.3. Forma de navegación de interfaz de pantalla 6

5.4. Formatos de impresión 6

6. Diseño de Datos 6

6.1. Modelo de Datos del Núcleo 6

6.2. Modelo de Datos General 6

6.3. Especificación de la Distribución de Datos 7

**7. PLAN DE PRUEBAS DE SOFTWARE**………………………………………………………… 8

Alcance de las Pruebas 8

Elementos de Pruebas 8

Nuevas Funcionalidades a Probar 8

Pruebas de Regresión 9

Funcionalidades a No Probar 9

Enfoque de Pruebas (Estrategia) 9

Criterios de Aceptación o Rechazo 9

Criterios de Aceptación o Rechazo 9

Criterios de Suspensión 9

Criterios de Reanudación 9

Entregables 9

Recursos 10

Requerimientos de Entornos – Hardware 10

Requerimientos de Entornos – Software 10

Herramientas de Pruebas Requeridas 11

Personal 11

Entrenamiento 12

Planificación y Organización 12

Procedimientos para las Pruebas 12

Matriz de Responsabilidades 13

Cronograma 13

Premisas 14

Dependencias y Riesgos 14

1. Introducción

La introducción debe proporcionar una visión general del documento Modelo de Diseño Construcción y pruebas de Software.

* 1. Propósito

Esta sección debe indicar el propósito del documento **Modelo de Construcción y pruebas** y la audiencia esperada para este documento.

* 1. Alcance

Esta sección es una breve descripción de a que se aplica el documento **Modelo de Construcción y Pruebas**, que es afectado o influenciado por este documento.

* 1. Definiciones, siglas y abreviaturas.

Esta sección debe proporcionar las definiciones de todos los términos, las siglas, y abreviaturas requeridas para interpretar apropiadamente el documento **Modelo de Construcción y Pruebas**. Esta información puede proporcionarse por la referencia al Glosario del proyecto.

* 1. Referencias

Esta sección debe proporcionar una lista completa de todos los documentos a los que se hace referencia en el documento **Modelo de Construcción y Pruebas**. Cada documento debe identificarse por el título, número del informe (si se aplica), fecha, y organización que lo publica. Especifique las fuentes de las que pueden obtenerse las referencias. Esta información puede proporcionarse por la referencia a un apéndice o a otro documento.

* 1. Visión general

Esta sección describe que contiene el resto del documento **Modelo de Construcción y Pruebas** y explica cómo se organiza este documento.

1. Diseño de Casos de Uso

En esta sección se especifica el comportamiento de los casos de uso, mediante subsistemas u objetos de diseño que interactúan, y se determinan las operaciones e interfaces de los distintos subsistemas u objetos de diseño.

Un objeto de diseño es una abstracción de un objeto o varios objetos en la implementación del sistema. Se define en base a parámetros, reglas, operaciones que realiza, requerimientos de implementación, interfase de usuario (si corresponde) y relaciones con otros objetos.

Esta especificación se realiza mediante Diagrama de Paquetes, Diseño de flujo de eventos, Diagramas de interacción y Requerimientos especiales o de implementación.

* 1. Diseño del Caso de Uso [nombre del caso de uso 1]
     1. Diagrama de paquetes

En esta sección se identifican los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones.

* + 1. Diagrama de Interacción

Se sugiere realizar un diagrama de interacción para representar la realización de **cada Caso de Uso**.

* + 1. Diseño de Flujo de eventos

Una descripción escrita en términos de objetos y subsistemas de diseño, que explica y complementa el diagrama de interacción y sus niveles.

* + 1. Requerimientos especiales o de implementación

Es una descripción que recoge los requerimientos (no funcionales) en la realización de un caso de uso.

* 1. Diseño del Caso de Uso [nombre del caso de uso 2]

1. Diseño de Objetos
   1. [Objeto 1]

Se especifican los parámetros, reglas, condiciones usando la misma sintaxis del lenguaje, el código y métodos son especificados en pseudo código.

* 1. [Objeto 2]

...

1. Diseño de Subsistemas
   1. Subsistemas Específicos
      1. [Nombre del Subsistema Específico 1]

**Propósito**

Descripción de por qué el subsistema existe. Este atributo debe dar la razón de la creación del subsistema. Como ser la funcionalidad específica y los requerimientos de performance por los cuales fue creado. También describe requerimientos especiales que se deben lograr con él que no están incluidos en la especificación de requerimientos del software.

**Función**

Declara qué hace el subsistema. Establece la transformación aplicada a las entradas del subsistema para producir la salida deseada.

**Subordinados**

Se identifican los objetos de diseño y subsistemas de diseño que componen el subsistema que se describe.

Se propone representar esta información con un diagrama de paquetes.

**Dependencias**

Descripción de la relación de este subsistema con otros subsistemas de diseño. Identifica los usos o la necesidad de presencia de la relación para un subsistema. Describe la naturaleza de cada interacción incluyendo características como tiempo y condiciones de la interacción. Estas interacciones pueden involucrar la iniciación, orden de ejecución, datos compartidos, creación, duplicación, uso o almacenamiento.

Se propone representar esta información con una tabla de dependencias.

|  |  |  |
| --- | --- | --- |
| **Subsistema del que depende** | **Naturaleza de interacción** | **Características** |
| Identificación del subsistema del que depende | Condiciones para que se realice la interacción | Características de la interacción, como ser, pasaje de parámetros, mensajes, datos compartidos, etc. |

**Recursos**

Identifica y describe a todos los recursos externos al diseño que necesita el subsistema para realizar su función. Se deben especificar las reglas de interacción y métodos para usar el recurso.

Este atributo brinda información sobre elementos como dispositivos físicos (impresoras, particiones de disco, bancos de memoria), servicios de software (librerías, servicios del sistema operativo), y recursos de procesamiento (ciclos de CPU, ubicación de memoria, buffers).

Se deben describir características de uso como el tiempo de proceso al cual se debe adquirir el recurso e incluir la cantidad de tiempo de uso. Debe incluir también la identificación de capacidad potencial y facilidades de manejo del recurso.

**Interfases**

Descripción de cómo los otros subsistemas interactúan con éste. Describe los métodos de interacción y las reglas que gobiernan éstas interacciones. Los métodos de interacción incluyen los mecanismos para invocar o interrumpir al subsistema, para comunicarse a través de parámetros, áreas comunes de información o mensajes, y para acceso directo a información interna. Las reglas que gobiernan la interacción incluyen el protocolo de comunicaciones, formato de la información, valores aceptables, y el significado de cada valor.

Este atributo también debe dar una descripción de los rangos de entrada, el significado de entradas y salidas, el tipo y formato de cada entrada y salida, y los códigos de error de salida. Se propone representar esta información con una lista del tipo:

|  |  |  |
| --- | --- | --- |
| **Interacción** | **Método de interacción** | **Reglas de la interacción** |
| Objeto que implementa la interacción |  |  |

* + 1. [Nombre del Subsistema Específico 2]

...

* 1. Subsistemas de Soporte
     1. [Nombre del Subsistema de soporte 1]

**Propósito**

**Función**

**Subordinados**

**Dependencias**

**Recursos**

**Interfases**

* + 1. [Subsistema de soporte 2]

...

1. Revisión de la Interfaz de Usuario

En esta sección se realiza el diseño detallado del comportamiento de la interfaz de usuario a partir de la especificación de requerimientos y, en caso que existan, de las pautas para interfase de usuario, además debe estar de acuerdo con el entorno tecnológico definido. Si se hubiera realizado un prototipo de la interfaz de usuario, éste se tomaría como punto de partida para el diseño.

Además, se incluyen las ventanas alternativas o elementos de diseño que han aparecido nuevos como consecuencia del diseño de realización de Casos de Uso.

Se revisa: la interfaz de usuario, la navegación entre ventanas, los elementos que forman cada interfaz, sus características, su disposición, y cómo se gestionan los eventos relacionados con los objetos.

En aquellos casos en los que se realicen modificaciones significativas sobre la interfaz de usuario, será conveniente la validación de la misma por parte del usuario, siendo opcional la realización de un nuevo prototipo.

* 1. Formatos individuales de interfaz de pantalla

Definir los formatos de los distintos tipos de pantallas.

En el caso de desarrollo con software especializado en diseño de interfaz; Definir el formato general ejemplo como para cada tipo de objeto Transaction, WorkPanel, WebPanel y/o para las pantallas más importantes del sistema.

* 1. Controles y elementos de diseño de interfaz de pantalla

Definir para los controles y elementos de diseño de interfaz de pantalla sus características, su disposición, y cómo se gestionan los eventos relacionados con ellos.

* 1. Forma de navegación de interfaz de pantalla

Indicar la forma de navegación dentro de las pantallas y entre las mismas.

* 1. Formatos de impresión

Definir los distintos tipos de formatos de impresión que se utilizarán en el sistema.

Definir el formato general para cada tipo de objeto que tenga como resultado una impresión: Report, Procedure y/o para los listados más importantes del sistema.

1. Diseño de Datos

En esta sección se define la estructura de datos que utilizará el sistema, a partir de los requisitos funcionales y no funcionales establecidos para el sistema y las particularidades del entorno tecnológico, que consiga una mayor eficiencia en el tratamiento de los datos.

* 1. Modelo de Datos del Núcleo

En el caso de desarrollo con “X”: En esta sección se realiza el diseño de datos que pertenecen al Núcleo del sistema.

Se identifican las principales entidades de datos y los datos por los que están compuestas a partir de la especificación de requerimientos.

Además, se describen las dependencias entre dichas entidades mediante diagramas.

* 1. Modelo de Datos General

En el caso de desarrollo: En esta sección se realiza el diseño de datos que no pertenecen al Núcleo pero que son relevantes para la arquitectura del sistema.

Se identifican las principales entidades de datos y los datos por los que están compuestas a partir de la especificación de requerimientos.

Además, se describen las dependencias entre dichas entidades mediante diagramas.

* 1. Especificación de la Distribución de Datos

En esta sección se especifica el modelo de distribución de datos indicando la ubicación de los manejadores de bases de datos o sistemas de archivos, así como los distintos elementos de la estructura física de datos (base de datos, tablas, índices), en los nodos correspondientes.

Para elaborar esta sección el Arquitecto trabajará en conjunto con el Especialista Técnico.

**7. PLAN DE PRUEBAS DE SOFTWARE**

# Resumen ejecutivo

Resumen de todo el contenido del plan de pruebas de software, describe cuál es su propósito, establece si es un plan maestro o un plan detallado, identifica el alcance del plan de pruebas en relación con el plan de proyecto de software, restricciones (por ejemplo, de recursos o presupuesto), alcance del esfuerzo de pruebas entre otros aspectos.

# Alcance de las pruebas

## Elementos de pruebas

Listado de todos los módulos, componentes o elementos que se van a probar. Si es de alto nivel, se listan las áreas funcionales (módulos o procesos que cubre el Testing), por otro lado, si es de un nivel detallado se listan los programas, unidades o módulos.

## Nuevas funcionalidades a probar

Es un listado de lo que se va a probar “desde el punto de vista del usuario”. No es una descripción técnica del software sino sus características y funcionalidades. Se incluyen tanto las que son nuevas como las que se están modificando.

## Pruebas de regresión

Listado de las funcionalidades no directamente involucradas en el desarrollo, pero cuyos componentes están siendo afectados y por ende deben probarse para asegurar que continúan funcionando adecuadamente. Al igual que en el punto anterior, se describen desde el punto de vista del usuario.

## Funcionalidades a no probar

Listado de las funcionalidades que no se van a probar. Debe incluir información de las razones por las cuales no se van a probar y los riesgos que se están asumiendo.

## Enfoque de pruebas (estrategia)

La estrategia de pruebas puede definirse como un documento por separado, o puede ser incluido dentro del plan de pruebas según su extensión. Aquí pueden definirse los tipos de pruebas a realizar (funcionales, de desempeño, de interfaces, no funcionales, etc.), requerimientos especiales de las pruebas, configuraciones a probar, subconjuntos de datos a considerar, nivel de pruebas de regresión, entre otros aspectos.

# Criterios de aceptación o rechazo

## Criterios de aceptación o rechazo

Son los criterios que serán considerados para dar por completado el plan de pruebas de software, por ejemplo: Completar 100% de pruebas unitarias, cierto porcentaje de casos exitosos, cobertura de todos los componentes y líneas de código, porcentaje de defectos corregidos, entre otros.

## Criterios de suspensión

Establece claramente bajo qué condiciones se detienen un conjunto de casos de pruebas, por ejemplo, en caso de existir defectos que impidan la ejecución de más casos de pruebas, cierto porcentaje de casos fallidos, o cualquier otro que se especifique.

## Criterios de reanudación

Luego de haber suspendido las pruebas, aquí se establece bajo qué criterios se reanudarán.

# Entregables

Establece que se entregará como parte de la ejecución del plan, por ejemplo: Documento de plan de pruebas, casos de pruebas, especificación de diseño de casos, log´s de errores, reportes de incidencias, evidencias de pruebas, reportes emitidos por herramientas de pruebas y cualquier otro que se establezca.

# 

# Recursos

## Requerimientos de entornos – Hardware

Lista de los requerimientos de equipos, hardware y red necesarios para completar las actividades del plan de pruebas de software. Incluye servidores de aplicación, bases de datos, equipos de PC que necesitan los Testers, conectividad a la red (incluyendo accesos), entre otros.

## Requerimientos de entornos – Software

Lista de los requerimientos de software necesarios para completar las actividades de prueba, puede incluir accesos a Sistemas (en entorno de pruebas) y bases de datos, así como instalación de software en los Computadores asignados a los Testers.

## Herramientas de pruebas requeridas

Especifica las herramientas de software, metodologías o técnicas especiales empleadas en las pruebas, por ejemplo, herramientas de automatización de pruebas, software de gestión de pruebas, entre otros.

## Personal

Lista del personal necesario para completar las actividades de pruebas, especificando sus roles, por ejemplo: Un (1) líder de pruebas, cinco (5) analistas de pruebas (Testers), dos (2) especialistas en automatización de pruebas, entre otros.

## Entrenamiento

Necesidades de entrenamiento en el sistema o aplicación, así como en las herramientas de prueba a utilizar.

# Planificación y organización

## Procedimientos para las pruebas

Especifica los procedimientos o metodología de pruebas a emplear durante la ejecución del plan de pruebas de software.

## Matriz de responsabilidades

Lista cada una de las personas integrantes del equipo de QA y sus responsabilidades. Se puede hacer uso de una matriz RACI (responsable, aprobador, consultado, informado).

## Cronograma (Gantt)

Debe estar basado en estimaciones de actividades realizadas por el equipo de prueba. En él se identifican los hitos relevantes en las pruebas de software, se establecen las dependencias (actividades predecesoras) y demás aspectos componentes de un cronograma.

## Premisas

Las premisas relacionadas con las tareas de pruebas de software, incluyendo limitaciones de tiempo, disponibilidad de recursos que se asumen, uso de una metodología de pruebas, uso de una herramienta, entre otros.

## Dependencias y Riesgos

Aquí se listan los riesgos asociados con el proceso de pruebas de software, por ejemplo, algunas fuentes de riesgos suelen ser:

* Dependencias con desarrollos.
* Dependencias con otros proyectos.
* Disponibilidad de recursos.
* Restricciones de tiempo.
* Premisas que resulten no ser ciertas.

Los riesgos se pueden clasificar en función de su probabilidad e impacto, cada uno debe contemplar un plan de mitigación para evitar que ocurra o plan de contingencia cuando el riesgo no puede mitigarse y tiene que aceptarse.

# Referencias

Lista de todos los documentos que pueden citarse como apoyo o para ampliar el contenido del plan de pruebas. Algunos ejemplos de lo que se puede hacer referencia aquí son:

* Plan de proyecto.
* Especificaciones de requerimientos.
* Diseño general.
* Diseño detallado.
* Procedimientos y estándares de desarrollo.
* Procedimientos y estándares de pruebas.
* Metodologías, procedimientos y estándares corporativos.