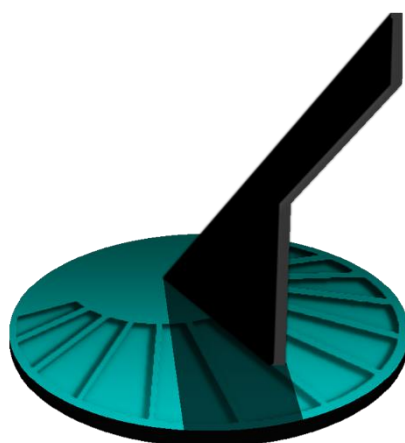


Plan de SQA

TEMPUS



TEMPUS

Gestión de Horarios de Cursada y Mesas de Examen

La Calidad del Software tiene como objetivo brindar la confianza de que el producto final logrará satisfacer los requisitos del cliente.

En el Plan de SQA se reflejan las evaluaciones a realizar, los estándares a aplicar, los productos a realizar, los procedimientos a seguir en la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Tabla de contenido

Propósito	5
<i>Referencias</i>	<i>5</i>
Gestión	6
<i>Organización</i>	<i>6</i>
<i>Actividades</i>	<i>7</i>
Ciclo de vida del software cubierto por el Plan	7
Actividades de calidad a realizarse	8
Relaciones entre las actividades de SQA y la planificación.....	9
<i>Responsables.....</i>	<i>9</i>
Documentación	10
<i>Propósito</i>	<i>10</i>
<i>Documentación mínima requerida.....</i>	<i>10</i>
Especificación de requerimientos del software.....	10
Modelo de casos de uso	11
Modelo de diseño	12
Plan de pruebas	13
Documentación de usuario.....	13
<i>Plan de Gestión de configuración.....</i>	<i>13</i>
<i>Otros documentos.....</i>	<i>14</i>
Estándares, prácticas, convenciones y métricas.....	14
<i>Objetivos</i>	<i>14</i>
<i>Métricas de proceso</i>	<i>15</i>
Cantidad de módulos reutilizados	15
<i>Métricas de proyecto</i>	<i>15</i>
Cantidad de reuniones por iteración	16
Tareas creadas sobre tareas resueltas.....	16
<i>Métricas de producto</i>	<i>18</i>
Puntos de caso de uso	18
<i>Estándar de documentación.....</i>	<i>18</i>
<i>Estándar de base de datos</i>	<i>18</i>
<i>Estándar de programación.....</i>	<i>19</i>
Revisiones y auditorías	19
<i>Objetivo.....</i>	<i>19</i>

<i>Requerimientos mínimos</i>	19
Revisión de especificación de requerimientos del software	19
Revisión del modelo de casos de uso	19
Revisión de diseño crítico	20
Auditoría funcional	20
Auditoría física	20
<i>Otras revisiones</i>	20
Revisión de documentación de usuario.....	20
Verificación	20
Reporte de problemas y acciones correctivas	20
<i>Revisiones</i>	20
<i>Auditorias</i>	21
Herramientas, técnicas y metodologías	22
<i>Herramientas</i>	22
Herramientas de apoyo	22
GitHub.....	22
PHPUnit.....	23
Netbeans IDE	23
Taiga.....	24
Tortoise SVN	25
Visual Studio Code	25
XAMPP	26
Xdebug.....	26
<i>Técnicas</i>	27
<i>Metodologías</i>	27
Gestión de riesgos	28
Anexos.....	28
<i>Formulario de Pedidos y Detección de Cambios</i>	28

Plan de SQA

Propósito

El propósito es el establecimiento de pautas y actividades que deben llevarse a cabo para garantizar la calidad del producto de software a desarrollar. Además, se busca verificar que el software y la documentación a entregar cumplan con todos los requerimientos.

Los objetivos específicos que se buscan con el presente Plan de SQA son:

- Cumplir con los estándares, normas y convenciones aceptadas por los integrantes del grupo de desarrollo.
- Generar el compromiso de cada uno de los integrantes del grupo de desarrollo para el aseguramiento de la calidad y la detección de errores.
- Asegurar el cumplimiento de cada uno de los requerimientos especificados.
- Controlar la configuración del software y la documentación asociada.
- Especificar las pruebas, revisiones y controles a realizar para el aseguramiento de la calidad del software.

Los procedimientos definidos en el presente documento se utilizarán para examinar las prestaciones que brindará el sistema, así como examinar la documentación para determinar que ambos cumplieron con los requerimientos.

Referencias

- ANSI/IEEE Std 730.1-1989, IEEE Standard for Software Quality Assurance.
- Documento Plantilla Gestión de Riesgos.
- Documento Plan de Gestión de Riesgos.
- Documento Plantilla de Revisión de SQA.
- Documento Informe Final de SQA.
- Documento Plantilla de Revisión Técnica Formal.
- Documento Plan de Pruebas del Software.
- Documento Plan de Gestión de Configuración.
- Documento Especificación de Casos de Uso.

- SQuaRE, ISO 25000:2005, Quality management systems – Requirements ISO 9001:2008

Gestión

La gestión del proyecto está a cargo del líder del proyecto, sin embargo será monitoreada en conjunto con los gerentes de calidad. Se intenta llevar un control que permita establecer que las actividades se ajustan al plan propuesto y minimizar las posibles desviaciones.

Organización

La mayoría de las actividades que se realizan durante el desarrollo del proyecto impactan en la calidad del producto final. Las líneas de trabajo que tienen un impacto directo son las siguientes:

- Requerimientos.
- Análisis.
- Diseño.
- Implementación.
- Pruebas.

El equipo de trabajo está compuesto por tres personas que se organizan de la siguiente forma:

Rol	Descripción	Responsable
Administrador de configuración	Administra los elementos de configuración del proyecto.	Oyarzo
Analista	Obtiene información para comprender el problema, capturar y ajustar las prioridades para los requerimientos.	Oyarzo, Quiroga y Márquez
Arquitecto	Responsable de diseñar la arquitectura del software.	Quiroga
Diseñador	Responsable de diseñar los casos de uso para el proyecto.	Oyarzo

Documentador	Responsable de mantener la información generada durante el proceso de desarrollo.	Quiroga
Gerente de calidad	Responsable de realizar y modificar las tareas relacionadas a la calidad.	Márquez
Ingeniero de pruebas	Realiza las pruebas identificadas y definidas previamente.	Oyarzo, Quiroga y Márquez
Líder del proyecto	Encargado de establecer las condiciones de trabajo. Dirige, asigna y coordina.	Márquez
Programador	Responsable de desarrollar parte del sistema.	Oyarzo, Quiroga y Márquez
Rol general	Responsable de realizar tareas generales.	Oyarzo, Quiroga y Márquez

Actividades

Ciclo de vida del software cubierto por el Plan

Las actividades contempladas son:

- Captura de requerimientos.
- Análisis y diseño.
- Implementación.
- Pruebas.

Los productos contemplados son:

- Especificación de Requerimientos de Software.
- Modelo de Datos.
- Modelo de Casos de Uso.
- Modelo de Diseño.
- Arquitectura del Sistema.

- Plan de pruebas.
- Manual de usuario.

Actividades de calidad a realizarse

Las tareas a ser llevadas a cabo deberán reflejar las evaluaciones a realizar, los estándares a seguir, los productos a revisar, los procedimientos para la elaboración de los distintos productos y los procedimientos para informar de los defectos detectados a sus responsables y realizar el seguimiento de los mismos hasta su corrección.

Las actividades que se realizarán son:

- Revisar cada producto.
- Realizar Revisión Técnica Formal (RTF).

Revisar cada producto

En el marco de esta actividad se revisan los productos definidos como clave.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en esta revisión. Se revisan los productos contra los estándares, utilizando la lista de comprobación definida para el producto.

Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión de SQA, este documento debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas.

Se definen los siguientes documentos para cada uno de los elementos clave:

- Revisión de SQA – ERS: Especificación de Requerimientos de Software.
- Revisión de SQA – MDA: Modelo de Datos.
- Revisión de SQA – MCU: Modelo de Casos de Uso.
- Revisión de SQA – MDI: Modelo de Diseño.
- Revisión de SQA – AS: Arquitectura del Sistema.
- Revisión de SQA – PP: Plan de pruebas.
- Revisión de SQA – MU: Manual de usuario.

Para cada Revisión de SQA se debe generar un documento que debe estar almacenado dentro del repositorio en el directorio de gestión de calidad.

Realizar Revisión Técnica Formal (RTF)

El objetivo de la RTF es descubrir errores en la función, la lógica o la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas. Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el responsable de SQA e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

La duración de la reunión no debe ser mayor a dos horas.

Como salida se obtiene el Informe de RTF. Los documentos que serán sometidos a Revisión Técnica Formal son los siguientes:

- Especificación de Requerimientos de Software.
- Arquitectura del Sistema.
- Modelo de Datos.
- Modelo de Casos de Uso.
- Manual de Usuario.

Relaciones entre las actividades de SQA y la planificación

Las actividades de SQA deberán desarrollarse a lo largo del proyecto. Se establecen puntos de control en el fin de cada iteración para cada fase.

En cada Iteración se deben controlar y documentar las métricas que se establecen.

Responsables

Los responsables de llevar a cabo los controles de calidad son el Líder de Proyecto junto con los Gerentes de Calidad.

Las revisiones deben realizarse con el Líder del Proyecto junto con los demás integrantes del grupo, salvo que se establezca lo contrario. Para cada actividad que se planifique se debe llevar a cabo el registro de errores encontrados.

Para la puesta en marcha de las actividades se deberá seguir el siguiente ciclo de prevención:

1. Ejecutar una tarea
2. Realizar un control de revisiones, para decidir la aceptación o necesidad de corrección de dicha tarea.
3. En caso de que en la revisión se presenten errores se realizara un análisis causal para determinar el motivo de estos. Se analiza un determinado error, se establece una hipótesis de su posible causa, se trata de deducir en qué momento se produjo y por qué. Luego se deberá realizar la corrección del mismo y tomar una acción correctiva con el fin de eliminar la causa del problema.
4. El resultado del análisis causal es ingresado a una base de datos para mantener un registro y poder obtener métricas.
5. Se comienza nuevamente el ciclo ejecutando la tarea.

Documentación

Propósito

Identificación de la documentación relativa a desarrollo, verificación y validación, uso y mantenimiento del software.

Se establece cómo los documentos van a ser revisados para chequear consistencia: se confirman criterio e identificación de las revisiones.

Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la implementación lograr satisfacer los requerimientos definidos.

Para cada documento se indica cual es el objetivo, qué estándar o norma debe seguir y que información mínima debe contener para cumplir con las definiciones del mismo.

Especificación de requerimientos del software

El documento de especificación de requerimientos deberá describir, de forma clara y precisa, cada uno de los requerimientos esenciales del software además de las interfaces externas. Dicho documento describe completamente el comportamiento externo del sistema, los requerimientos no funcionales y restricciones de diseño.

El cliente deberá obtener como resultado del proyecto una especificación adecuada a sus necesidades en el área de alcance del proyecto, de acuerdo al compromiso inicial del trabajo y a los cambios que este haya sufrido a lo largo del proyecto, que cubra aquellos aspectos que se haya acordado detallar con el cliente.

La especificación debe:

- Ser completa:
 - Externa, respecto al alcance acordado.
 - Internamente, no deben existir elementos sin especificar.
- Ser consistente, no puede haber elementos contradictorios.
- Ser no ambigua, todo término referido al área de aplicación debe estar definido en un glosario.
- Ser verificable, debe ser posible verificar siguiendo un método definido, si el producto final cumple o no con cada requerimiento.
- Estar acompañada de un detalle de los procedimientos adecuados para verificar si el producto cumple o no con los requerimientos.
- Incluir requerimientos de calidad del producto a construir.

Los requerimientos de calidad del producto a construir son considerados dentro de atributos específicos del software que tienen incidencia sobre la 'calidad en el uso'.

- Funcionalidad: Adecuación a las necesidades, precisión de los resultados, interoperabilidad, seguridad de los datos.
- Confiabilidad: Madurez, tolerancia a fallas, recuperabilidad, etc.
- Usabilidad: Comprensible, aprendible, operable, atractivo, etc.
- Eficiencia: Utilización de los recursos.
- Mantenibilidad: Analizable, modificable, estable, verificable, etc.
- Portabilidad

Cada uno de estos atributos debe cumplir con las normas y regulaciones aplicables a cada uno.

Modelo de casos de uso

El Modelo de Casos de Uso describe la funcionalidad propuesta para el sistema. El modelo se basa en la descripción de elementos o usuarios externos al sistema (actores) y de la funcionalidad del mismo (casos de uso).

El Modelo de Casos de Uso describe los requerimientos funcionales de un actor en términos de las interacciones. El documento presenta al sistema desde la perspectiva de sus usos y esquematiza como proporcionara valor a los usuarios. También sirve como un acuerdo entre los clientes y desarrolladores para limitar las funcionalidades con las que contara el sistema.

Dado que este documento es la entrada fundamental para el análisis, diseño y pruebas, debe contar con las siguientes características:

- Identificación y descripción de actores. Se deben describir los actores identificados para conocer el entorno externo del sistema.
- Identificación y descripción de casos de uso. Se deben describir los casos de uso identificados. Cada caso de uso representa la funcionalidad que el sistema ofrece y que aporta un resultado de valor al usuario.
 - Establecer precondiciones.
 - Establecer post condiciones.
 - Flujo de sucesos principal.
 - Flujos alternativos.
- Diagramas asociados: Diagramas de casos de uso, actividad, interacción.

El Modelo de Casos de Uso debe estar redactado en un lenguaje natural para la comprensión de los clientes. Debe ser elaborado primero en iteraciones tempranas y luego ser gradualmente extendido hasta obtener un documento detallado.

Modelo de diseño

El documento de diseño especifica como el software será construido para satisfacer los requerimientos. El propósito del Modelo de Diseño es empezar a realizar los casos de uso identificados durante etapas anteriores. Este modelo es utilizado por los desarrolladores para la implementación.

Deberá describir los componentes y subcomponentes del diseño del software, incluyendo interfaces internas. Este documento deberá ser elaborado primero como Preliminar y luego será gradualmente extendido hasta llegar a obtener el Detallado.

El cliente deberá obtener como resultado del proyecto el diseño de un producto de software que cubra aquellos aspectos que se haya acordado con el cliente incorporar al diseño, en función de la importancia que estos presenten y de sus conexiones lógicas.

El diseño debe:

- Corresponder a los requerimientos a incorporar: Todo elemento del diseño debe contribuir a algún requerimiento

- La implementación de todo requerimiento a incorporar debe estar contemplada en por lo menos un elemento del diseño.
- Ser consistente con la calidad del producto

Plan de pruebas

El Plan deberá identificar y describir los métodos a ser utilizados en:

- La verificación de que:
 - Los requerimientos descritos en el documento de requerimientos han sido aprobados por una autoridad apropiada.
 - Los requerimientos descritos en el documento de requerimientos son implementados en el diseño expresado en el documento de diseño.
 - El diseño expresado en el documento de diseño esta implementado en código.
- Validar que el código, cuando es ejecutado, se adecua a los requerimientos expresados en el documento de requerimientos.

Documentación de usuario

La documentación de usuario debe especificar y describir los datos y entradas de control requeridos, así como la secuencia de entradas, opciones, limitaciones de programa y otros ítems necesarios para la ejecución exitosa del software.

Todos los errores deben ser identificados y las acciones correctivas descritas.

Como resultado del proyecto el cliente obtendrá una documentación para el usuario de acuerdo a los requerimientos específicos del proyecto.

Plan de Gestión de configuración

El Plan de gestión de configuración debe contener métodos para identificar componentes de software, control e implementación de cambios, y registro y reporte del estado de los cambios implementados.

La Gestión de Configuraciones permite controlar el sistema como producto global a lo largo de su creación, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores durante el mismo, lo que se traduce en un aumento de calidad del proceso de desarrollo y de mejora de la productividad en la organización.

La Gestión de Configuración facilita además el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.

Para más detalles se debe acceder al Plan de Gestión de Configuración.

Otros documentos

Otros documentos que tienen incidencia en la calidad del proyecto son:

- Plan de proyecto: Este documento está compuesto por toda la información necesaria para llevar a cabo la dirección del proyecto. Es utilizado por la dirección del proyecto para dirigir las actividades a realizar durante el proceso de desarrollo del software, este comprende un conjunto de artefactos que son desarrollados durante la fase de inicio para luego ser utilizados durante todo el ciclo de vida del proyecto (gestión de riesgos, aseguramiento de calidad, resolución de problemas, entre otros).
- Plan de estimación: Una parte importante de la toma de decisiones al comenzar un nuevo proyecto de desarrollo de software está dada por el costo que éste tendrá.
- Plan de iteración: El objetivo de este plan es definir detalladamente para cada una de las iteraciones a realizarse un conjunto de tareas, actividades y recursos. Para cada iteración existen una serie de objetivos los cuales son usados como referencia de evaluación para determinar diferentes aspectos, como el grado de terminación de una determinada función, rendimiento, niveles de calidad, etc.

Estándares, prácticas, convenciones y métricas

En esta sección del documento se realiza una identificación de estándares, practicas, convenciones y métricas que serán utilizadas durante el desarrollo del proyecto. Se establece como será monitoreado y asegurado el cumplimiento de cada uno de los ítems que se mencionan.

Objetivos

Las métricas son una herramienta poderosa y fundamental para el trabajo en SQA. Su aporte fundamental son las medidas preventivas que pueden surgir a raíz de su estudio. Sin duda aportan conclusiones que muchas veces no se aprecian a simple vista y que ayudan a mejorar la eficiencia del grupo de trabajo y la calidad de los productos. Aportan un caudal de información para hacer controles estadísticos de la calidad. Además cabe resaltar que nunca debe dejarse de buscar nuevas métricas de acuerdo a las nuevas variaciones y tendencias de las estadísticas.

Existen dos objetivos importantes que se persiguen dentro del programa de métricas:

- Documentar las metas a la hora de establecer un programa de métricas. Esto tiene sentido a la hora de decidir exactamente qué debe lograrse antes de gastar recursos estableciendo un programa de este tipo.
- Identificar la información (la métrica) necesaria para lograr estas metas y establecer el marco de referencia de donde puede ser obtenida.

Métricas de proceso

Las métricas de proceso se recopilan de todos los proyectos y durante un largo periodo de tiempo. Su intención es proporcionar un conjunto de indicadores de proceso que conduzca a mejorar el proceso de software a largo plazo. Este tipo de métricas se caracteriza por:

- Control y ejecución del proyecto.
- Medición de tiempos de las fases.

Para este proyecto se trabajará con las siguientes métricas del proceso:

- Cantidad de módulos reutilizados.

Cantidad de módulos reutilizados

Se establecen la cantidad de módulos que se han reutilizado junto con el grado aproximado de reutilización. El líder del proyecto, junto con los desarrolladores, deberá establecer los módulos a reutilizar y registrarlos. El grado de reutilización sirve para establecer qué porcentaje aproximado del módulo ha podido utilizarse.

Modulo	Descripción	Grado de reutilización
Usuarios	Crear, buscar, borrar y modificar.	
Roles	Crear, buscar, borrar y modificar.	
Permisos	Crear, buscar y borrar.	
Total módulos reutilizados = 3.		

Métricas de proyecto

Las métricas del proyecto permiten evaluar el estado del proyecto y seguir la pista de los riesgos. Para el presente proyecto de desarrollo se trabajara con las siguientes métricas del proyecto:

- Cantidad de reuniones por iteración.
- Tareas creadas sobre tareas resueltas.

Cantidad de reuniones por iteración

Se definen la cantidad de reuniones que se llevan a cabo durante el periodo de una iteración junto con el tiempo utilizado.

En cada cierre de Iteración se deberán evaluar la cantidad de reuniones que se han desarrollado. Para ello se hará uso de los documentos “Resumen de Reunión” donde se establecen los objetivos, hora de inicio, hora de fin y temario. Se utilizara la duración de cada reunión para calcular el tiempo utilizado en cada iteración.

La información debe cargarse en la siguiente tabla donde se indica la Iteración, la fase a la que corresponde, la cantidad de reuniones que se han desarrollado y la cantidad de horas utilizadas.

Fase	Iteración	Cantidad de reuniones	Cantidad de horas
Inicio	1	4	8
	2	0	0
Elaboración	1	0	0
	2		
	3		
Construcción	1		
	2		
	3		
Transición	1		
Total		4	8

Con la recopilación de esta información se podrán planificar las reuniones necesarias para cada iteración subsiguiente junto con la cantidad de horas promedio a utilizar.

Tareas creadas sobre tareas resueltas

Se definen la cantidad de tareas o actividades que se han creado durante un determinado periodo junto con la cantidad de tareas que se han resuelto.

El periodo definido será la duración de una Iteración. Se consideraran las actividades que se han planificado al inicio de una Iteración y la cantidad de tareas que se han resuelto al finalizar la misma. Para ello se puede hacer uso del documento “Plan de Iteración” junto con la herramienta “Taiga”.

El objetivo de esta métrica es conocer la cantidad de actividades que el grupo de desarrollo puede realizar durante un periodo de tiempo. Es decir, el grado de cumplimiento con lo planificado.

Al finalizar cada Iteración, se debe completar la siguiente tabla:

Fase	Iteración	Actividades	
		Planificadas	Resueltas
Inicio	1	37	21
	2	16	13
Elaboración	1	20	18
	2		
	3		
Construcción	1		
	2		
	3		
Transición	1		
Total		73	52

Observaciones:

- La Iteración 1 – Fase de Inicio, no ha cumplido los objetivos. Se tomó la decisión de pasar las actividades sin resolver a una nueva iteración (Iteración 2).
- La iteración 2 – Fase de Inicio, contaba con tareas de investigación que no han sido resueltas. No afecta en los objetivos de la misma.

Métricas de producto

Las métricas del producto se centran en las características del software y no en cómo fue producido. También son productos los artefactos, documentos, modelos, y componentes que conforman el software. Se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad, y el esfuerzo

Para este proyecto se trabajará con las siguientes métricas del producto:

- Puntos de Caso de Uso

Puntos de caso de uso

Se establecen como métrica la cantidad de puntos de caso de uso.

Estándar de documentación

Como estándar de documentación se define un documento denominado “Estándar de Documentación”. Este estándar establece los lineamientos para la preparación y administración de la documentación que se genera a lo largo del desarrollo del proyecto.

Para obtener mayor detalle de las características se debe acceder al documento mencionado. La documentación que se genere a lo largo del proyecto debe cumplir con los lineamientos del estándar de documentación.

Estándar de base de datos

Como estándar de base de datos se define un documento denominado “Estándar de Base de Datos” que se encuentra dentro de la sección de estándares en el repositorio del proyecto.

Este documento establece convenciones para la nomenclatura de tablas y campos de una base de datos realizando una adaptación del estándar CamelCase. Se constituyen convenciones, estilos y reglas para:

- Nomenclatura de base de datos.
- Nomenclatura de tablas.
- Nomenclatura de campos: Identificadores de tablas, claves foráneas y otros campos.

La base de datos del sistema debe estar de acuerdo a las reglas que se establecen en el documento de estándar de base de datos. Para obtener más detalle de las características del estándar se debe acceder al documento disponible.

Estándar de programación

Se define el documento “Estándar de Programación PHP” donde se definen las características principales que debe contener el código generado durante la implementación.

En dicho documento se establecen estilos y reglas para la nomenclatura de clases, variables, constantes y métodos. Además, se indica la forma correcta de realizar los comentarios al código generado.

Para obtener más detalle se debe acceder al documento disponible.

Revisiones y auditorías

Objetivo

Una auditoria consiste en realizar una investigación para determinar el grado de cumplimiento y la adecuación de los procedimientos, instrucciones, estándares u otros requisitos de tipo contractual establecidos aplicables.

Las revisiones técnicas se utilizan para detectar defectos que puedan ser eliminados antes que avancen por las distintas fases de desarrollo. Se puede definir a la revisión como una reunión formal en la que se presenta el estado actual de los resultados del proyecto.

Requerimientos mínimos

En esta sección se especifican las revisiones y auditorías que deben realizarse como mínimo, así como la agenda para la realización de las mismas.

Revisión de especificación de requerimientos del software

La revisión de requerimientos se realiza para asegurar que se ha cumplido con los requerimientos especificados por el cliente. Se debe generar el documento de revisión correspondiente con los resultados obtenidos.

Revisión del modelo de casos de uso

La revisión de casos de uso se realiza para asegurar que se ha cumplido con los estándares y que la especificación contiene los elementos mínimos para continuar con el proceso. El revisor debe generar un documento de revisión e indicar la aprobación o reprobación de la especificación.

Revisión de diseño crítico

La revisión de diseño crítico se realiza para asegurar la consistencia del diseño detallado con la especificación de requerimientos. Esta revisión debe ser documentada en informada por el revisor.

Auditoría funcional

Esta auditoría se realiza previa a la liberación del software, para verificar que todos los requerimientos especificados en la especificación de requerimientos del software fueron cumplidos. Todos los integrantes del grupo de desarrollo deben estar presentes en la auditoría y se debe generar el documento de auditoría correspondiente.

Auditoría física

Esta revisión se realiza para verificar que el software y la documentación son consistentes y están aptos para la liberación. Todos los integrantes del grupo de desarrollo deben estar presentes y se debe generar el documento de auditoría correspondiente.

Otras revisiones

Revisión de documentación de usuario

Se revisa la completitud, claridad y aplicación de uso.

Verificación

La finalidad del plan de pruebas es realizar las pruebas al sistema con el fin de encontrar aquellos defectos que el mismo pueda contener y tratando de realizarlo con el mínimo costo posible.

La información detallada de las pruebas junto con su planificación se encuentra disponible en el documento “Plan de Pruebas de Software” dentro del repositorio del proyecto.

Reporte de problemas y acciones correctivas

En esta sección se incluye la descripción de las prácticas y procedimientos que se seguirán para el reporte, seguimiento, y resolución de los problemas surgidos en el desarrollo del sistema.

Revisiones

Se realizan las revisiones planificadas para el proyecto. Luego, los integrantes del grupo de calidad reportaran el resultado de las revisiones, la aprobación o reprobación y una serie de recomendaciones. Las revisiones se utilizan para asegurar que:

- El proceso se está realizando de forma correcta y se trabaja de forma efectiva.
- El proceso se está siguiendo correctamente pero no se trabaja bien.
- El proceso no se está siguiendo.

El proceso de reporte está orientado al Líder del Proyecto, el cual los utilizara para tomar las decisiones correspondientes. Finalmente, el proceso de reporte de revisiones y acciones correctivas queda establecido de la siguiente forma:

1. Se realizan las revisiones planificadas.
2. Se genera el reporte por parte del revisor con las recomendaciones pertinentes. Se debe completar el documento "Revisión de SQA" según el documento a revisar.
3. Se presenta el reporte al líder del proyecto.
4. El líder del proyecto toma decisiones según el reporte de revisión.
5. Si se decide, se aplican las correcciones necesarias. En caso de ser necesario, se realiza una nueva revisión y el proceso inicia nuevamente.

Auditorias

En principio se realizan las auditorias planificadas para el proyecto. Luego, los integrantes del grupo de calidad reportaran el resultado de las auditorias y una serie de recomendaciones. El reporte mencionado se utiliza para asegurar que:

- El proceso se está realizando de forma correcta y se trabaja de forma efectiva.
- El proceso se está siguiendo correctamente pero no se trabaja bien.
- El proceso no se está siguiendo.

El proceso de reporte está orientado hacia el Líder del Proyecto el cual los utilizara para:

- Saber si los procesos de desarrollo son acatados por el grupo de desarrollo.
- Para indicar el acuerdo, desacuerdo o aplazamiento de las recomendaciones proporcionadas.

Finalmente, el proceso de reporte y acciones correctivas queda establecido de la siguiente forma:

1. Se realizan las auditorias planificadas.
2. Se genera el reporte por parte del grupo de calidad con las recomendaciones pertinentes.
3. Se presenta el reporte al Líder del Proyecto.
4. El líder del proyecto toma decisiones según el reporte de auditoria.

5. Si se decide, se aplican las correcciones necesarias. En caso de ser necesario, se realiza una nueva auditoría y el proceso inicia nuevamente.

Herramientas, técnicas y metodologías

Para el desarrollo del sistema se hace uso de toda herramienta disponible por el sistema operativo, documentos de ayuda, analizadores de estructuras, analizadores de código, estándares y software de desarrollo.

Herramientas

Herramientas de apoyo

- Microsoft Excel, Microsoft Power Point, Microsoft Word.
- Navegador web: Chrome, Internet Explorer, Mozilla.
- Herramientas para la gestion del Proyecto: GanttProject.
- Aplicaciones de video conferencia: Zoom y Google Meet.
- Visualizador de documentos en format PDF: Foxit Reader.
- Para trabajar con archivos comprimidos: 7-ZIP.
- Herramientas de diagramas UML: Visual Paradigm.
- Herramientas de base de datos: MySQLWorkbench, DB-Main y Día.

GitHub

GitHub es una plataforma de **desarrollo colaborativo de software** para alojar proyectos utilizando el sistema de control de versiones GIT. El código y documentación del proyecto se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.



GitHub aloja el repositorio de código y brinda **herramientas** muy útiles para el **trabajo en equipo**, dentro de un proyecto. Actualmente, Github ofrece múltiples herramientas:

- Una wiki para el mantenimiento de las distintas versiones de las páginas.

- Un sistema de seguimiento de problemas que permiten a los miembros del equipo detallar un problema con el software o una sugerencia que deseen hacer.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas del repositorio.

PHPUnit

PHPUnit fue creado con el objetivo de realizar pruebas unitarias en el lenguaje de programación PHP. Es un framework desarrollado en PHP que pertenece a la familia de xUnit y que ha sido creado por Sebastián Bergmann. Como el objetivo es detectar errores, PHPUnit nos permite diseñar y ejecutar un conjunto de pruebas en una forma sencilla.



Las pruebas unitarias nos permiten aislar ciertas partes de nuestro programa y demostrar que esas partes de forma individual funcionan correctamente. Se debe recordar que las pruebas deben estar diseñadas para encontrar problemas en las fases iniciales del desarrollo.

Para más información se debe observar el Manual de configuración de PHPUnit que queda disponible en el repositorio del proyecto.

Netbeans IDE

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.



Netbeans permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web con Java, JavaScript, HTML5, PHP, C / C ++ y más. A su vez, permite crear aplicaciones Web con PHP 7, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en PHP. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE.

El IDE se usará para desarrollar la aplicación web del proyecto utilizando la extensión de PHP. Además se hará uso de las extensiones necesarias para facilitar el trabajo.

Taiga

Taiga es una herramienta de gestión de proyectos para equipos ágiles multifuncionales. Tiene un amplio conjunto de funciones y, al mismo tiempo, es muy sencillo comenzar a través de su interfaz de usuario intuitiva.



Scrum es una metodología de desarrollo de software ágil iterativa e incremental para gestionar el desarrollo de productos.

- El trabajo acumulado del producto es lo que finalmente se entregará, ordenado en la secuencia en la que se debe entregar. Los pedidos acumulados de productos se dividen en fragmentos manejables y ejecutables llamados sprints.
- Cada cierto tiempo, el equipo inicia un nuevo sprint y se compromete a entregar una cierta cantidad de historias de usuarios del trabajo acumulado, de acuerdo con sus habilidades, capacidades y recursos. El proyecto avanza a medida que se agota el trabajo atrasado.

La metodología **Kanban** se utiliza para dividir el desarrollo del proyecto (cualquier tipo de proyecto) en etapas.

- Una tarjeta kanban es como una tarjeta de índice o una nota post-it que detalla cada tarea (o historia de usuario) en un proyecto que debe completarse. El tablero Kanban se usa para mover cada carta de un estado de finalización al siguiente y, al hacerlo, ayuda a rastrear el progreso.
- El proyecto avanza tan rápido como el tiempo promedio de ciclo de vida de una tarjeta.

Las actividades planificadas en cada iteración son cargadas en Taiga para que cada uno de los integrantes del grupo pueda monitorear los avances y el equipo docente verificar el avance del proyecto.

Tortoise SVN

TortoiseSVN es un cliente Apache Subversion, implementado como una extensión de shell de Windows. Es intuitivo y fácil de usar, ya que no requiere que se ejecute el cliente de línea de comandos de Subversion. Y es de uso gratuito, incluso en un entorno comercial.



Se hará uso de Tortoise SVN para administrar los cambios y actualizaciones del repositorio del proyecto.

Visual Studio Code

Visual Studio Code (VSC) es un editor de código desarrollado por Microsoft para Windows, Linux y MacOS. Incluye soporte para la depuración, control integrado de GIT, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.



Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado. Muchas de las características de

Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos.

En este proyecto se hace uso de esta herramienta para desarrollar la aplicación móvil.

XAMPP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.



El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

Se hace uso de XAMPP para desplegar el sitio web y administrar las características de la base de datos.

Xdebug

XDebug es una extensión de PHP que proporciona al desarrollador ciertas características para la depuración. Dichas características son:

- Seguimientos de pila. Salida detallada de la ruta que se siguió, incluyendo parámetros enviados a las funciones llamadas.
- Una salida al estilo VAR_DUMP que produce información estructurada.
- Generador de perfiles para detectar cuellos de botella en la aplicación. Esto se puede visualizar con una herramienta externa como KCacheGrind o WinCacheGrind.
- Un depurador remoto que se puede utilizar para establecer una conexión entre XDebug de forma remota con un cliente.

XDebug le permite al desarrollador de software romper la ejecución del código e inspeccionar todas las variables en el alcance durante una solicitud. Se puede decir que durante una sola iteración es posible encontrar problemas sin llenar el código de alertas o logs. Básicamente, ahorra una gran cantidad de tiempo al momento de rastrear problemas y ayuda a que el flujo de trabajo sea más eficiente.



Esta herramienta encuentra su mayor utilidad al momento de encontrar y solucionar un problema cuya causa es totalmente desconocida para el programador. Muchas veces, es posible saber la causa de un problema y rastrearlo en forma sencilla, pero un comportamiento extraño causa un análisis mayor. Los puntos de interrupción que brinda XDebug permiten pausar el código en cualquier momento haciendo que el seguimiento sea mucho más sencillo.

Para más información se debe observar el Manual de configuración de Xdebug que queda disponible en el repositorio del proyecto.

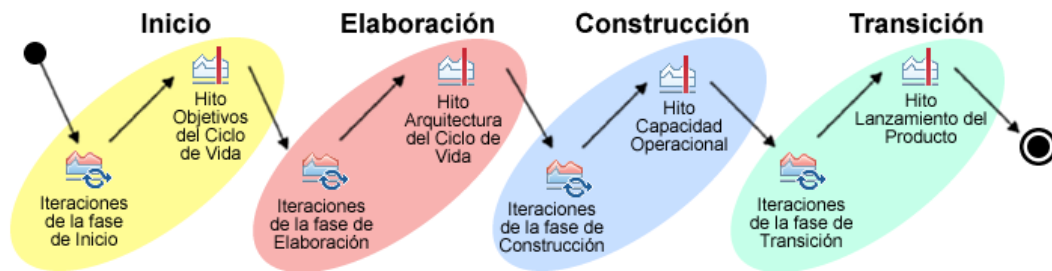
Técnicas

En las técnicas se incluirá la revisión en el uso de estándares, inspecciones de software, rastreabilidad de requerimientos y casos de uso, verificación y validación de requerimientos y diseño, mediciones y análisis de la lógica de negocio.

- Estándares:
 - Estándar de Documentación.
 - Estándar de Programación PHP.
 - Estándar de Base de Datos.

Metodologías

- Metodología PSI (<https://www.uarg.unpa.edu.ar/psi/>).
 - *“La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, con el fin de lograr una mayor confiabilidad, mantenimiento y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.”*



Flujo de trabajo de la metodología PSI

- Paradigma de programación: Orientado a objetos.

Gestión de riesgos

La Gestión de Riesgos es un elemento clave dentro del desarrollo del proyecto y de gran utilidad para la identificación de potenciales peligros. Es deber del Líder del Proyecto, con participación de los integrantes del grupo de desarrollo, hacer uso de las técnicas de gestión de riesgos.

Como parte de la Gestión de Riesgos se debe hacer una especificación de métodos y procedimiento para definir, monitorear y controlar las áreas de riesgo durante el proyecto.

Los riesgos identificados, la estrategia de mitigación, monitoreo y plan de contingencia a llevar a cabo, son descriptos en el documento de "Plan de Gestión de Riesgos". Durante el proceso de Gestión de Riesgos se debe generar también la siguiente documentación:

- Gestión de Riesgos, Anexo I. Se deberá completar un nuevo documento basado en la plantilla "Gestión de Riesgos Anexo I" cada dos iteraciones del proyecto. En caso de tener iteraciones de corto plazo, pueden realizarse un análisis para una fase completa.
- Seguimiento de Riesgos. Se debe completar un documento basado en la plantilla "Seguimiento de Riesgos" el cual será actualizado con el correr del proyecto.

Para obtener más detalle sobre la Gestión de Riesgos se debe acceder a los documentos mencionados que se encuentran disponibles en la sección de gestión de riesgos dentro del repositorio del proyecto.

Anexos

Formulario de Pedidos y Detección de Cambios

Formulario de Pedidos y Detección de Cambios	
Fecha de Petición:	

Nombre y Versión del Elemento	
Nombre del Solicitante:	
Necesidad de Cambio:	
Descripción del cambio pedido:	
Prioridad:	
Estado:	
Fecha del cambio:	
Identificador de la nueva versión:	
Que fue afectado por este cambio	