

**INSTITUTO TECNOLÓGICO DE MORELIA**

**Plan de SQA**  
**Fases del Ciclo de Vida del Software**  
11 de mayo del 2022  
Versión 3.0

Información del documento		
<b>Título</b> <b>Plan de SQA – FASES CICLO DE VIDA DEL SOFTWARE</b>	<b>Identificador</b> 1011	
	<b>Versión</b> 3.0	
<b>Archivo</b> Plan de SQA POR FASES.doc		
<b>José Ángel Villalón Villagómez</b>	11/05/2022	<b>Michoacán</b>

Aprobación del documento		
<b>Tester / Analista dde pruebas</b> <i>Francisco Javier Villa Díaz</i>	<i>Francisco Javier Villa Díaz</i>	11/05/2022
<b>Ingeniero de SQA</b> <i>José Ángel Villalón Villagómez</i>	<i>José Ángel Villalón Villagómez</i>	1/05/2022
<b>Project Manager / Líder Provento</b> <i>Ricardo Hernández Salgado</i>	<i>Ricardo Hernández Salgado</i>	11/05/2022
<b>Analista del Provento</b> <i>Jocabed Ríos Saucedo</i>	<i>Jocabed Ríos Saucedo</i>	11/05/2022
<b>Ingeniero de SQA</b> <i>Dagoberto Erik Arreola Frayle</i>	<i>Dagoberto Erik Arreola Frayle</i>	11/05/2022
<b>Tester / Analista dde pruebas</b> <i>Andrés López Sierra</i>	<i>Andrés López Sierra</i>	11/05/2022

## Contents

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>3</b>
1.1	PROPÓSITO .....	3
1.2	ALCANCE .....	3
1.3	IDENTIFICACIÓN .....	3
1.4	DESCRIPCIÓN DEL SISTEMA .....	4
1.5	DESCRIPCIÓN DEL DOCUMENTO .....	5
1.6	DOCUMENTOS RELACIONADOS .....	6
1.7	DOCUMENTOS REFERENCIADOS .....	6
1.8	GLOSARIO DE TÉRMINOS .....	6
1.9	ACRÓNIMOS .....	6
<b>2</b>	<b>GESTIÓN .....</b>	<b>7</b>
2.1	ORGANIZACIÓN .....	7
2.2	RESPONSABILIDADES .....	8
2.3	CALENDARIZACIÓN .....	9
<b>3</b>	<b>FASES DEL CICLO DE VIDA DEL SOFTWARE .....</b>	<b>10</b>
<b>4</b>	<b>ESTÁNDARES .....</b>	<b>11</b>
<b>5</b>	<b>REVISIONES Y AUDITORÍAS .....</b>	<b>11</b>
<b>6</b>	<b>PRUEBA .....</b>	<b>11</b>
<b>7</b>	<b>INFORMES DE PROBLEMAS Y ACCIONES CORRECTIVAS .....</b>	<b>12</b>
<b>8</b>	<b>HERRAMIENTAS, TÉCNICAS Y METODOLOGÍAS .....</b>	<b>12</b>
<b>9</b>	<b>AGRUPACIÓN, MANTENIMIENTO Y RETENCIÓN DE REGISTROS .....</b>	<b>12</b>

---

## 1 Introducción

### 1.1 Propósito

El propósito de presente plan es definir la organización, las actividades y responsabilidades asociadas al proceso de SQA durante el proyecto de software para aprendizaje autónomo. Además, entregar guías para la ejecución de las actividades de SQA, definir los estándares, los procedimientos y las convenciones que serán utilizados durante estas actividades y establecer las herramientas, técnicas y metodologías que soportarán las prácticas de SQA. Por lo tanto, el plan de SQA está dirigido al jefe de proyectos, los desarrolladores y al grupo de SQA, responsable de la elaboración, actualización y monitoreo del plan.

### 1.2 Alcance

El presente documento establece, de acuerdo a la política organizacional, las actividades de SQA que deberán ser ejecutadas durante el ciclo de vida del software definido para el proyecto de software para aprendizaje autónomo. El ciclo de vida comprende las etapas de planificación, especificación de requerimientos, diseño, implementación, integración y pruebas, aceptación y entrega, y mantenimiento.

El objetivo de SQA es entregar a la administración una visibilidad adecuada del proceso utilizado y los productos construidos durante el proyecto de software para aprendizaje autónomo mediante acciones planificadas y sistemáticas que aseguren la calidad de los procesos y productos.

### 1.3 Identificación

Producto de trabajo	Identificador	Acrónimo
JustLearn	I004	JL

Tabla 1 Productos de trabajo

1. JustLearn: El producto JustLearn consiste en un software donde cualquier usuario mediante un dashboard dar de alta cualquier curso de cualquier materia y complementarlo con cualquier objeto de aprendizaje que crea oportuno. Y a su vez, permitir a cualquier usuario desde una interfaz inicial, poder tomar el curso desde cualquier lugar y a su propio ritmo.

---

## **1.4 Descripción del sistema**

Se pretende desarrollar un sistema que sea capaz de gestionar objetos de aprendizaje, mediante la creación de un curso académico a un usuario, dichos objetos de aprendizaje son videos, lecciones, documentos, prácticas de retroalimentación, etc.

También se debe de permitir a cualquier usuario consultar el contenido en todo momento del curso que ya fue creado, para poder tomarlo desde cualquier lugar, las veces que quiera y a su propio ritmo.

---

## **1.5 Descripción del documento**

A través de la implementación del SQA se pretende cumplir con los requerimientos explícitamente establecidos para cada fase del ciclo de vida de software, así como las actividades correspondientes en cada una.

### **FASE 1**    Requerimientos:

El análisis de sistemas es vital para determinar cuáles son las necesidades de una empresa, así como cómo pueden satisfacerse, quién será responsable de las partes individuales del proyecto y qué tipo de cronograma se debe esperar. Aquí es donde los equipos consideran los requisitos funcionales del proyecto o la solución. También es donde tiene lugar el análisis del sistema, o el análisis de las necesidades de los usuarios finales para garantizar que el nuevo sistema pueda satisfacer sus expectativas.

### **FASE 2**    Diseño:

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, así como decidir la estructura general del mismo. El diseño es una etapa compleja y su proceso debe realizarse de manera iterativa.

Es posible que la solución inicial no sea la más adecuada, por lo que en tal caso hay que refinarla. No obstante, hay catálogos de patrones de diseño muy útiles que recogen errores que otros han cometido para no caer en la misma trampa.

### **FASE 3**    Codificación:

En esta parte se realizan las tareas que comúnmente se conocen como programación, que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior. Esta tarea la realiza el programador siguiendo por completo los lineamientos impuestos en el diseño y en consideración siempre a los requisitos funcionales y no funcionales (ERS) especificados en la primera etapa.

### **FASE 4**    Prueba:

Como errar es humano, la fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Por supuesto, lo ideal es hacerlo antes de que el usuario final se los encuentre. Se dice que una prueba es un éxito si se detecta algún error.

### **FASE 5**    Implementación:

En esta parte se trata de elegir las herramientas adecuadas, un entorno de desarrollo que haga más sencillo el trabajo y el lenguaje de programación óptimo.

Esta decisión va a depender del diseño y el entorno elegido.

Es importante tener en cuenta la adquisición de productos necesarios para que el software funcione.

### **FASE 6**    Mantenimiento:

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

- Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
- Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
- Añadirle nuevas funcionalidades (mantenimiento perfectivo).

Aunque suene contradictorio, cuanto mejor es el software más tiempo hay que invertir en su mantenimiento. La principal razón es que se usará más (incluso de formas que no se habían previsto) y, por ende, habrá más propuestas de mejoras.

## 1.6 Documentos relacionados

El plan de SQA forma parte del plan de proyecto, por lo tanto, los recursos y la calendarización de las actividades de SQA se encuentran detalladas en este último plan. Cabe resaltar que el plan de SQA debe ser ejecutado conjuntamente al plan de SCM.

## 1.7 Documentos referenciados

- ANSI / IEEE – STD 830 *Guide for Software Requirements Specifications*
- ANSI / IEEE – STD 1016 *Recommended Practice for Software Design Descriptions*
- ANSI / IEEE – STD 1008 *Standard for Software Unit Testing*
- ANSI / IEEE – STD 1063 *Standard for Software User Documentation*
- ANSI / IEEE – STD 1028 *Standard for Software Reviews and Audits*
- Documento de Actividades de Gestión de Calidad – A. Delgado & B. Pérez 2000

## 1.8 Glosario de términos

- **Aseguramiento de la calidad del software (SQA)** – El propósito de SQA es entregar a la administración una visibilidad adecuada del proceso utilizado y los productos construidos mediante acciones planificadas y sistemáticas que aseguren la calidad de dichos procesos y productos.
- **Auditoría** – Evaluación independiente de los productos de trabajo y de un conjunto de procesos de software para asegurar la adherencia con las especificaciones, los estándares, procedimientos y otros acuerdos.
- **Gestión de la configuración del software (SCM)** – El propósito de SCM es establecer y mantener la integridad de los productos a través de todo el ciclo de vida del software, proveyendo un adecuado control de los cambios en los diversos ítems de configuración.
- **Revisión** – Metodología definida, estructurada y disciplinada para la detección e identificación de defectos en los productos de trabajo durante el ciclo de vida del software.
- **Prueba (Testing)** – Actividad que valúa los atributos y la capacidad de un programa o sistema para determinar si se cumple con los resultados definidos.

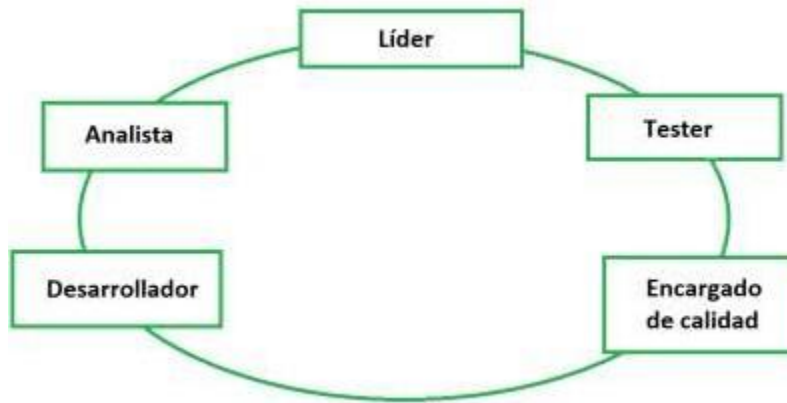
## 1.9 Acrónimos

Acrónimo	Significado
SQA	<i>Software Quality Assurance</i> , Aseguramiento de la Calidad del Software
SCM	<i>Software Configuration Management</i> , Gestión de Configuración del Software
WBS	<i>Work Breakdown Structure</i>

---

## 2 Gestión

### 2.1 Organización



1. **Ingeniero SQA** – Responsable de monitorear el cumplimiento de las actividades planificadas en el plan de SQA y de garantizar la calidad de los entregables, la documentación y de los procesos utilizados para producir software.
  - Establecer un programa de calidad para cada proyecto de desarrollo de software de acuerdo a las políticas organizacionales.
  - Revisar y aprobar el plan de SQA para el proyecto.
  - Resolver cualquier conflicto relacionado con las actividades de SQA.
2. **Tester:**
  - Probar y Validar software
  - Maximizar la ausencia de defectos
  - Realizar pruebas durante y posterior al desarrollo del software



---

### 3. Project Manager:

- Establecer un programa de calidad para el proyecto de desarrollo de software de acuerdo a las políticas organizacionales.
- Identificar las actividades de SQA requeridas para el proyecto.
- Revisar y aprobar el plan de SQA para el proyecto.
- Identificar los participantes de las actividades de SQA.
- Implementar las actividades de SQA de acuerdo al plan.
- Monitorear las actividades de SQA planificadas en el plan.
- Identificar los factores de calidad para la implementación del software.
- Identificar, desarrollar y mantener la documentación del proyecto.

### 4. Desarrolladores :

- Revisar y entregar sus observaciones sobre el plan de SQA para el proyecto.
- Implementar las actividades de SQA de acuerdo al plan.
- Participar de la solución de los problemas detectados por las actividades de SQA que sean de su competencia.
- Implementar las prácticas, procesos y procedimientos definidos en el plan de proyecto y en otros planes o documentos complementarios.

### 5. Analista:

- Especificar requerimientos en la fase 1
- Realizar diagramas UML, BPMN, Casos de Uso
- Diseñar pruebas junto al gerente técnico
- Analizar la UX

## 2.2 Responsabilidades

En la siguiente tabla se adjunta una matriz de responsabilidades sobre las actividades de SQA.

Actividad	Tester	Jefe proyecto	SQA	Analista	Desarrollador	Repr. Prueba
Evaluación de la selección los productos de trabajo	X	X	X	X		
Evaluación de las herramientas	X	X	X			
Evaluación de la planificación y el monitoreo del proyecto	X	X	X	X		
Evaluación de la especificación de requerimientos	X	X	X	X	X	X
Evaluación del diseño	X	X	X	X	X	X
Evaluación de la implementación y de la prueba de unidad			X		X	X

Evaluación de la integración y prueba			X		X	X
Evaluación del producto antes de su liberación		X	X	X	X	X
Evaluación del proceso de revisión		X	X			
Evaluación de las acciones correctivas	X	X	X	X		
Evaluación del proceso de SCM	X	X	X	X		
Verificar la implementación de los procesos		X	X			
Establecer las auditorías		X	X	X		

## 2.3 Calendarización

Adjuntada en archivo .xls en la misma carpeta.

Calendarización de Actividades / Validación y Verificación de Software							
PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTÓNOMO							
JUSTICIA							
SIMANA	1	2	3	4	5	6	7
PERIODO	4/abril/22 5/abril/22	25/abril/22 26/abril/22	2/maio/22 3/maio/22	9/maio/22 10/maio/22	16/maio/22 17/maio/22	23/maio/22 24/maio/22	30/maio/22 31/maio/22
RESPONSABLE	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina	Agencia de SQA, Jefe de Oficina
FASES	1	2, 3	3	4, 5	4, 5	5, 6	
ACTIVIDADES	RFI, determinar requisitos funcionales y no funcionales	Requisitos de sistema	RFI-1 a RFI-2	Pruebas Conceptuales	Implementación	Evaluación	Entrega Proyecto
	Plan de SQA	Modelado UML (Diagrama de Casos de Uso)	RFI-1 a RFI-2	Pruebas Unitarias	Extracción	Mapas	
	Diagrama BPMN y Diagrama de Casos de Uso	Diagrama de base de datos	RFI-3	Pruebas de Sistema	Pruebas de aceptación	Corrección de errores	
			RFI-4	Pruebas de integración			

EQUIPO DE TRABAJO	ROLES	FASES	ROLES
Agencia de SQA, Jefe de Oficina	Desarrollador	1	Requisitos
Agencia de SQA, Jefe de Oficina	Project Manager	2	Diseño
Agencia de SQA, Jefe de Oficina	Analista	3	Análisis/Definición
Agencia de SQA, Jefe de Oficina	Desarrollador	4	Pruebas
Agencia de SQA, Jefe de Oficina	Desarrollador de SQA, Jefe de Oficina	5	Implementación
Agencia de SQA, Jefe de Oficina	Desarrollador	6	Mantenimiento



### 3 **FASES DEL CICLO DE VIDA DEL SOFTWARE**

	Actividad	Producto Resultado	Rol Responsable	Roles Participantes
<b>FASE 1: REQUERIMIENTOS</b>				
	ERS PLAN de SQA Diagrama BPMN Diagrama de Casos de Uso	ERS.pdf Plan SQA.pdf Diagrama BPMN.png Casos de Uso.png	Líder de Proyecto	Ingeniero de SQA  Analista
<b>FASE 2: DISEÑO</b>				
	Modelado del Sistema Arquitectura del Sistema	Modelados.pdf Arquitectura.png	Líder de Proyecto	Analista Desarrollador
<b>FASE 3: CODIFICACIÓN</b>				
	RF31-RF2 RF2.1-RF2.2-RF2.3 RF3 RF5	Vistas.php Index.php Curso.php Registro.php Admin.php	Líder de Proyecto	Desarrollador Tester
<b>FASE 4: PRUEBAS</b>				
	Pruebas Componentes Pruebas Unitarias Pruebas de Sistema Pruebas de Integración	Casos de prueba.xls  Resultados de casos de prueba.xls	Líder de Proyecto	Desarrollador Tester Analista
<b>FASE 5: IMPLEMENTACIÓN</b>				
	Implementación Evaluación Pruebas de Aceptación	Sistema Integro	Líder de Proyecto	Tester Ingeniero de SQA
<b>FASE 6: MANTENIMIENTO</b>				
	Evaluación Mejoras Corrección de errores	Manual de mantenimiento.pdf	Líder de Proyecto	Analista Ingeniero de SQA

## 4 Estándares

Para alcanzar la calidad total de los productos y la mejora continua, se utilizan los siguientes estándares:

- IEEE STD-730: “El aseguramiento de la calidad de software es un modelo planificado y sistemático de todas las acciones necesarias a fin de asegurar que el ítem o producto cumpla con los requerimientos técnicos establecidos”.
- IEEE STD-729: Análisis de los requerimientos de Software.
- IEEE STD-828: Estándar para planes del manejo de las configuraciones de Software.
- IEEE STD-829: Estándar para la documentación de pruebas de software.
- IEEE STD-830: Estándar para las especificaciones de requerimientos de software.
- IEEE STD-1012: Estándar para la planificación de verificación y validación de Software.

## 5 Revisiones y auditorías.

A continuación, se identifican las revisiones y auditorías requeridas para el presente proyecto:

Fase del ciclo de vida del sw	Revisiones y auditorías requeridas
Especificación de requerimientos	<ul style="list-style-type: none"><li>• Revisión de la Especificación de Requerimientos</li><li>• Auditoría del proceso (opcional)</li></ul>
Diseño	<ul style="list-style-type: none"><li>• Revisión del diseño preliminar</li><li>• Revisión del diseño detallado</li><li>• Revisión del plan de pruebas</li><li>• Revisión de la especificación y procedimiento de prueba</li><li>• Auditoría del proceso (opcional)</li></ul>
Implementación	<ul style="list-style-type: none"><li>• Revisión del código</li><li>• Revisión de la prueba de unidad</li><li>• Auditoría del proceso (opcional)</li></ul>
Integración y prueba	<ul style="list-style-type: none"><li>• Revisión de las pruebas</li><li>• Auditoría funcional</li><li>• Auditoría del proceso (opcional)</li></ul>
Aceptación y entrega	<ul style="list-style-type: none"><li>• Revisión del producto final</li><li>• Revisión de la documentación usuaria</li><li>• Auditoría física</li><li>• Auditoría del proceso (opcional)</li></ul>

## 6 Prueba

En JustLearn las actividades de prueba incluyen los niveles de unidad, integración, aceptación y del sistema, las cuales se desarrollan según el procedimiento descrito en el estándar de “Proceso de prueba”.

---

## **7 Informes de problemas y acciones correctivas**

En el presente proyecto se utilizarán los siguientes informes con el objeto de informar sobre los problemas detectados y sobre el estado de las acciones correctivas derivadas de ellos:

1. **Informe de auditoría** – Informe del estado y los resultados de l proceso de auditoría a la institución/unidad auditada.
2. **Informe de discrepancias** – Informe para el registro de las disconformidades detectadas durante el ciclo de vida del software. Por disconformidad se entiende cualquier desviación del producto o del proceso de los requerimientos, estándares y procedimientos definidos. Este informe permite además registrar y monitorear las acciones correctivas derivadas de la disconformidad.
3. **Informe de actividades de SQA** – Informe para el registro del estado de desarrollo de las diferentes actividades de SQA. Permite monitorear la adherencia al plan de SQA.

También serán utilizados los informes asociados a los procesos de revisión y prueba.

## **8 Herramientas, técnicas y metodologías**

Utilizaremos la herramienta de Trello teniendo así un controlador de tareas y un administrador general del proyecto.

## **9 Agrupación, mantenimiento y retención de registros**

Los informes generados por las actividades de SQA durante el ciclo de vida del proyecto, serán almacenados y mantenidos en la librería del software por un período mínimo de **1 mes y medio**.