



# PLAN DE PRUEBA IEEE 829

## (PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTÓNOMO)

1.IDENTIFICADOR DEL PLAN DE PRUEBA.....	2
2.REFERENCIAS.....	2
3.INTRODUCCIÓN .....	2
4.ELEMENTOS DE PRUEBA .....	2
4.1 SOFTWARE QUE SE VA A PROBAR .....	2
4.2 DOCUMENTOS A PROBAR.....	3
5. PROBLEMAS DE RIESGO DE SOFTWARE.....	3
6. CARACTERÍSTICAS A PROBAR.....	3
7.CARACTERÍSTICAS QUE NO DEBEN PROBARSE.....	3
8.ENFOQUE.....	4
9.ELEMENTO CRITERIOS DE APROBACIÓN/RECHAZO.....	6
10.CRITERIOS DE SUSPENSIÓN Y REQUISITOS DE REANUDACIÓN .....	6
11.ENTREGABLES DE PRUEBA .....	6
12.TAREAS DE PRUEBA RESTANTES.....	7
13.NECESIDADES DE PERSONAL Y CAPACITACIÓN .....	7
14.RESPONSABILIDADES .....	7
15.CALENDARIO.....	8
16.PLANIFICACIÓN DE RIESGOS Y CONTINGENCIAS.....	8
17.APROBACIONES.....	9

## **1.IDENTIFICADOR DEL PLAN DE PRUEBA**

PLNPRBAS-001

## **2.REFERENCIAS**

- ERS
- Plan SQA
- Diseño de casos pruebas
- Maquetado del sistema

## **3.INTRODUCCIÓN**

El siguiente documento tiene como finalidad mostrar una planificación de pruebas y gestión de pruebas para el “PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTÓNOMO”, donde se describirá el alcance, enfoque, los recursos y el cronograma de actividades con respecto a las pruebas generando así resultados y sus respectivos criterios de aprobación.

## **4.ELEMENTOS DE PRUEBA**

Los elementos a probar dentro del proyecto de Software de Aprendizaje Autónomo se dividen en dos:

- 1) Sistema de software funcional.
- 2) Documentación del sistema.

### **4.1 SOFTWARE QUE SE VA A PROBAR**

- a. Registrarse (RF-1) Version 1.0
- b. Login (RF-2) Versión 1.0
- c. Menú Instructor (RF3) Versión 1.0
- d. CRUD Curso (RF-4) Versión 1.0
  - I. Agregar un Curso (RF-4.1) Versión 1.0
  - II. Editar un Curso (RF-4.2) Versión 1.0
  - III. Eliminar un Curso (RF-4.3) Versión 1.0

- e. CRUD Contenido (RF-5) Versión 1.0
  - I. Agregar Contenido (RF-5.1) Versión 1.0
  - II. Editar Contenido (RF-5.2) Versión 1.0
  - III. Eliminar Contenido (RF-5.3) Versión 1.0
- f. Inicio Instructor (RF-6) Versión 1.0
- g. Inicio Usuario (RF-7) Versión 1.0
- h. Inicio Curso (RF-8) Versión 1.0

## **4.2 DOCUMENTOS A PROBAR**

- a. Especificación de Requerimientos (ERS) Versión 3.0
- b. Documento de maquetado del sistema

## **5. PROBLEMAS DE RIESGO DE SOFTWARE**

Existen algunos riesgos de software uno de ellos es reiniciar la aplicación en medio de un proceso, es un factor crítico para la confiabilidad de la aplicación. Esto en el caso de subir archivos, ya que una vez que los datos se extraen de la maquina del usuario se almacenan en la base de datos y al momento de haber una pérdida de conexión puede haber perdida o daño de estos .

## **6. CARACTERÍSTICAS A PROBAR**

- Funcionalidad
- Eficiencia
- Portabilidad

## **7. CARACTERÍSTICAS QUE NO DEBEN PROBARSE**

- Aspectos visuales (Paleta de color, Logo, estilo).

## **8.ENFOQUE**

Las pruebas para el PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTÓNOMO se describen a continuación, se considera tener al menos una prueba por cada etapa dentro del desarrollo de software.

### **PRUEBAS BASADAS EN REQUISITOS**

Las pruebas basadas en los requisitos son ejecutadas en la etapa de Análisis de requerimientos, estas principalmente tienen dos objetivos:

- 1) Validar que los requisitos sean correctos, completos, no ambiguos y coherentes desde un punto de vista lógico.
- 2) Definir el conjunto de casos de prueba más pequeño que sea necesario para asegurarse de que el diseño y el código cumplen por completo los requisitos.

### **PRUEBAS BASADAS EN EL DISEÑO**

Las pruebas basadas en el diseño como su nombre lo indica son ejecutadas en la etapa de diseño dentro del ciclo de desarrollo de software, su principal objetivo es:

- Revisar que los casos de prueba son diseñados en base a la arquitectura y/o al diseño detallado de un componente o sistema.

### **PRUEBAS DE COMPONENTES**

Estas pruebas también son conocidas por pruebas unitarias, estas son ejecutadas en la etapa de desarrollo una vez que ya existan componentes del sistema lo que se desea obtener con esta prueba es:

- Validar que los componentes funcionen de acuerdo a lo esperado, para estas pruebas por lo regular se usan herramientas de depuración donde se va revisando línea a línea el comportamiento del código, al momento de detectar algún defecto se procede a analizarlo y resolverlo.

## **PRUEBAS DE INTEGRACIÓN**

Las pruebas de integración son ejecutadas en la etapa de pruebas, estas requieren una mejor estructura y organización ya que al menos se requieren un plan de pruebas para ejecutarlas, las funciones de estas pruebas son:

- 1) Probar el software para verificar que este funcione de manera conjunta.
- 2) Se valida el software a través de varias interfaces y casos de uso tomando en cuenta que la salida de un componente es la entrada de otro.

## **PRUEBAS DE SISTEMA**

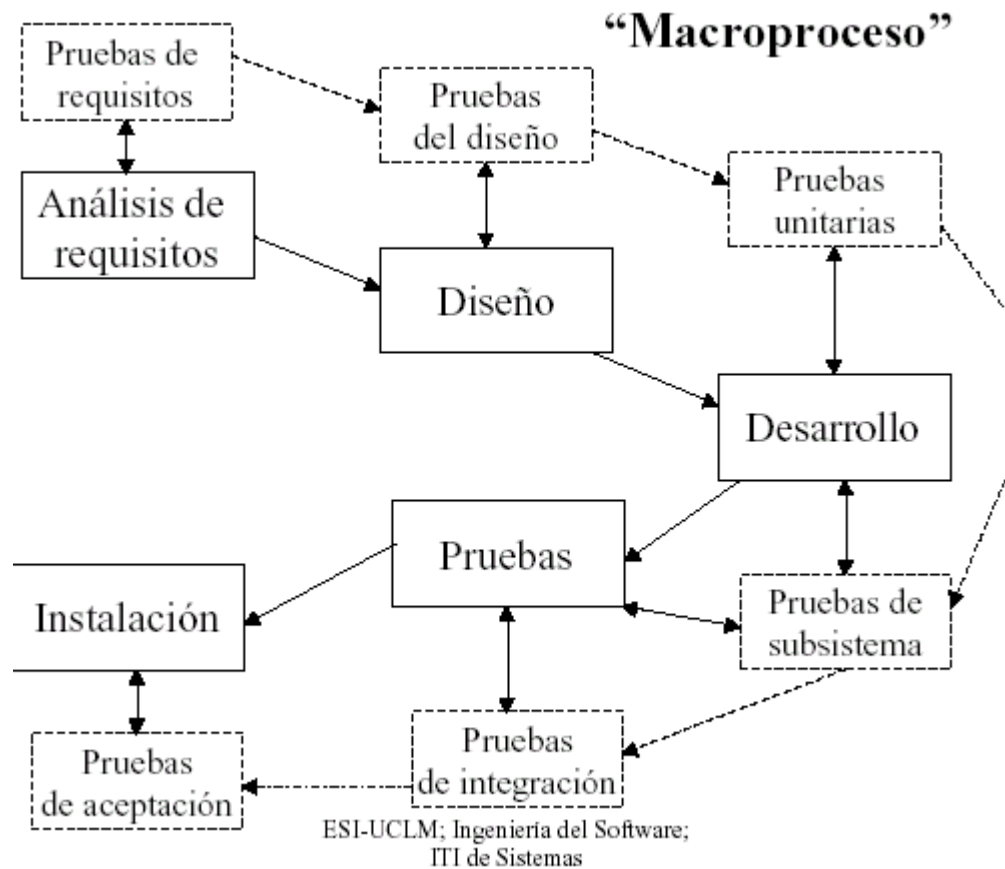
Estas pruebas son las que validan que todas las funciones y componentes del sistema colaboran correctamente. Estas pruebas son ejecutadas una vez se concluyeron las de componentes y las de integración a diferencia de estas dos las pruebas de sistema son realizadas desde el punto de vista del usuario su objetivo es:

- Validar en su totalidad los requerimientos del usuario especificados en el documento de ERS con el sistema.

## **PRUEBAS DE ACEPTACIÓN**

Estas pruebas son realizadas para determinar si el software cumple las expectativas y es viable para sus usuarios finales, es decir, si es aceptable y apto para salir al mercado. Objetivo:

- verificar que el software cumple las expectativas desde el punto de vista del cliente y de los usuarios finales.



## 9.ELEMENTO CRITERIOS DE APROBACIÓN/RECHAZO

El proceso para ejecutar una prueba y saber si esta se considerara aprobada o rechazada depende de el resultado que esta arroje, el resultado dependerá de los requerimientos del documento de ERS, si la prueba cumple con lo que se describe en este documento se considerara APROBADA y si no cumple con lo estipulado en el documento se considera como un RECHAZO de esta misma.

## 10.CRITERIOS DE SUSPENSIÓN Y REQUISITOS DE REANUDACIÓN

No existe.

## 11.ENTREGABLES DE PRUEBA

- Plan de prueba.
- Procedimientos de prueba y resultados IEEE 829

## 12.TAREAS DE PRUEBA RESTANTES

Tarea	Asignado a:
Pruebas Basadas En Requisitos	Tester
Pruebas Basadas En El Diseño	Tester
Pruebas De Componentes	Tester
Pruebas De Integración	Tester
Pruebas De Sistema	Tester
Pruebas De Aceptación	Tester


## 13.NECESIDADES DE PERSONAL Y CAPACITACIÓN

Se prefiere que haya al menos un Tester de tiempo completo asignado al proyecto para las fases de prueba de aceptación/integración y sistema del proyecto. Esto requerirá la asignación de esta persona a tiempo parcial al comienzo del proyecto para participar en las revisiones, etc.

## 14.RESPONSABILIDADES

Actividad	Desarrollador	Tester	Analista	Calidad	Líder
Evaluación de la selección los productos de trabajo.		X	X	X	X
Evaluación de las herramientas.		X		X	X
Evaluación de la planificación y monitoreo del proyecto.		X	X	X	X
Evaluación de la especificación de requerimientos.	X	X	X	X	X
Evaluación de diseño.	X	X	X	X	X
Evaluación de la prueba de unidad.	X			X	

## 15.CALENDARIO

Calendario de Actividades / Validación y Verificación de Software								EQUIPO DE TRABAJO	ROLES	FASES	ROLES
PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTÓNOMO								Arreola Frayle Dagoberto Erik	Desarrollador	1	Requerimientos
JUSTLEARN								Hernández Salgado Ricardo	Project Manager	2	Diseño
SEMANA	1	2	3	4	5	6	7	Rios Saucedo Jacobed	Analista	3	Desarrollo/Codificación
PERIODO	4/abril/22 8/abril/22	25/abril/22 29/abril/22	2/mayo/22 6/mayo/22	9/mayo/22 13/mayo/22	16/mayo/22 20/mayo/22	23/mayo/22 27/mayo/22	30/mayo/22 03/junio/22	Villa Díaz Francisco Javier	Tester	4	Pruebas
RESPONSABLE	Ingeniero de SQA, Analista, Líder de Proyecto	Analista, Desarrollador, Líder de Proyecto	Desarrollador, Tester y Líder de Proyecto	Desarrollador, Tester, Analista y Líder de Proyecto	Tester, Ingeniero de SQA y Líder de Proyecto	Analista, Ingeniero de SQA, Líder de Proyecto	Todo el Equipo de trabajo	Villalón Villagómez José Ángel	Ingeniero de SQA	5	Implementación
FASES	1	2, 3	3	3, 4	4, 5	5, 6		Andrés López Sierra	Desarrollador	6	Mantenimiento
ACTIVIDADES:	ERS, determinar requisitos funcionales, y no funcionales	Arquitectura de sistema	RF31 RF2	Pruebas Componentes	Implementación	Evaluación	Entrega Proyecto	 Ricardo Hernández Salgado Project Manager			
	PLAN de SQA	Modelado UX UI (Adobe XD)	RF2.1 RF2.2 RF2.3	Pruebas Unitarias	Evaluación	Mejoras					
	Diagrama BPMN y Diagrama de Casos de Uso	Creación de base de datos	RF3	Pruebas de Sistema	Pruebas de aceptación	Corrección de errores					
			RF5	Pruebas de Integración							
			RF5	Si hubo algún retraso con un RF							

## 16.PLANIFICACIÓN DE RIESGOS Y CONTINGENCIAS

El equipo de desarrollo asignado al PROYECTO DE SOFTWARE PARA APRENDIZAJE AUTONOMO puede tener demoras con respecto a los tiempos, ya que este puede encontrarse con dificultades como un mal levantamiento de ERS o el retraso de revisiones del software por parte del cliente.



## **17.APROBACIONES**

Líder- Hernández salgado Ricardo	
Analista – Ríos Saucedo Jocabed	
Tester – Villa Diaz Francisco Javier	
Ingeniero de SQA- Villalon Villagómez Ángel	
Desarrollador – Arreola Frayle Dagoberto Erik	
Desarrollador – López cierra Andrés	