

## CAPÍTULO 3

### MARCO METODOLÓGICO

---

#### 3.1. Metodología a ser utilizada

Es importante señalar que al desarrollar software educativo es necesaria la incorporación de aspectos de índole pedagógico y didáctico para satisfacer las necesidades del usuario al que va dirigido el software, así como la incorporación del mismo en el proceso de desarrollo para identificar, necesidades sentidas y expresadas y/o problemas específicos, y poder establecer mecanismos que den respuesta a esos problemas.

Para desarrollar el software educativo se utiliza una adaptación del modelo propuesto por Díaz de Fijoo (2002) en “Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica” ya que sugiere un enfoque sistemático y secuencial basado en *Rational Unified Process* (RUP) con incorporaciones de los aspectos didácticos y pedagógicos antes mencionados, utilizando una adaptación de sus características, así como la utilización de algunos entregables (documentos o versiones del software) primordiales.

Se escoge dicha metodología ya que, además de obtener las mejores prácticas de la metodología RUP, incorpora un diseño instruccional e involucra a los usuarios en el proceso de creación del software, añade una

adaptación del Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software (Mendoza y otros.. 2002), al desarrollo de software educativo, permitiendo garantizar el cumplimiento de las condiciones mínimas deseadas de manera de facilitar y garantizar la satisfacción de las necesidades educativas del público al que va dirigido el software.

### **3.2. Rational Unified Process (RUP)**

*Rational Unified Process* (RUP) es un proceso de Ingeniería de Software planteado por Krutchen (1996) cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Cubre el ciclo de vida de desarrollo de software.

#### **3.2.1. Fases de RUP para el desarrollo de software educativo**

- **Fase de Comienzo o Inicio**

Está principalmente dirigida al entendimiento de los requerimientos y determinar el alcance del esfuerzo de desarrollo. Se define la idea, la visión y el alcance del proyecto. Se incluye un análisis de las necesidades educativas y del entorno educativo, así como el diseño instruccional del proyecto. Esta fase se culmina con los objetivos del ciclo de vida. (Díaz y otros. 2001)

- **Fase de Elaboración**

Planificar las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura del software. Esta fase culmina con la arquitectura del ciclo de vida.

- **Fase de Construcción**

Desarrollar el producto y evolucionar la visión; la arquitectura y los planes hasta que el producto en una primera versión esté listo para ser enviado a la comunidad de usuarios. Esta fase culmina con la capacidad inicial de operación.

- **Fase de Transición**

Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto hasta que el cliente esté satisfecho. Esta fase culmina con la versión de producto, la cual a su vez concluye el ciclo.

### **3.3. Calidad de software educativo**

Cuando se habla de calidad de software educativo, se requiere un producto que satisfaga tanto las expectativas de los docentes como de los usuarios, a un menor costo, libre de defectos y cumpliendo con ciertas especificaciones.

La calidad del software educativo está determinada no sólo por los aspectos técnicos del producto sino por el diseño pedagógico y los

materiales de soporte. Este último aspecto es uno de los más problemáticos ya que existen pocos programas que ofrezcan un soporte didáctico.

### **Adecuación de MOSCA para software educativo**

En el trabajo “*Instrumento de Evaluación de software educativo bajo un enfoque sistémico*” (Díaz-Antón y otros.) se hace una adaptación del modelo original de MOSCA (Modelo Sistémico de Calidad del Software. Mendoza y otros (2002)) para la evaluación del software educativo específicamente. La estructura del modelo consta de cuatro niveles.

**Nivel 0:** Dimensiones. Efectividad Producto.

**Nivel 1:** Categorías. Se contemplan tres categorías:

- **Funcionalidad (FUN):** Es la capacidad del producto del software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es utilizado bajo ciertas condiciones.
- **Usabilidad (USA):** Esta categoría se refiere a la capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.
- **Fiabilidad (FIA):** La fiabilidad es la capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas.

**Nivel 2:** Características. Cada categoría tiene asociado un conjunto de características.

**Nivel 3:** Subcaracterísticas. Para algunas de las características se asocian un conjunto de subcaracterísticas.

**Nivel 4:** Métricas. Para cada característica se propone una serie de métricas utilizadas para medir la calidad sistémica. Se evalúan métricas adicionales relacionadas con Funcionalidad, Usabilidad y Fiabilidad, que permitan adaptar MOSCA en el área de software educativo.

El modelo de evaluación de calidad consta de una serie de categorías, características y subcaracterísticas y métricas.

CATEGORÍA	CARACTERÍSTICAS	SUBCARACTERÍSTICAS
FUNCIONALIDAD (FUN)	FUN.1 Ajuste a los propósitos	FUN.1.1 General FUN.1.2 Objetivos de aprendizaje FUN.1.3 Contenidos de aprendizaje FUN.1.4 Actividades de aprendizaje FUN.1.5 Ejemplos FUN.1.6 Motivación FUN.1.7 Retroalimentación FUN 1.8 Ayudas FUN.1.9 Evaluación y registro de datos FUN.1.10 Metodología de enseñanza
	FUN.2 Precisión	
	FUN.3 Seguridad	

USABILIDAD (USA)	USA.1 Facilidad de comprensión	USA.1.1 General USA.1.2 Interactividad USA.1.3 Diseño de la interfaz USA.1.4 Guías didácticas
	USA.2 Capacidad de uso USA.3 Interfaz Gráfica USA.4 Operabilidad	
FIABILIDAD (FIA)	FIA.1 Madurez FIA.2 Recuperación FIA.3 Tolerancia a fallas	

**Tabla 2.** Categorías, características, sub-características y número de métricas, para el modelo basado en MOSCA. Fuente: Díaz-Antón y otros.

### 3.3.1. Algoritmo para la evaluación de la calidad de software educativo

El algoritmo consta de dos fases, la primera fase describe los pasos a seguir para la preselección del software, en caso de que sea adquirido por vía comercial. Por ser este Trabajo Especial de Grado un desarrollo independiente, no se tomará en cuenta dicha fase, por lo que se describe a continuación la segunda del algoritmo.

**Fase 2. Estimar la calidad del producto de software educativo a través de un enfoque sistémico.** En esta fase se evalúa la calidad del producto siguiendo tres actividades:

### **1. Estimar la calidad de la FUNCIONALIDAD del producto.**

Para la categoría de FUNCIONALIDAD (FUN) se debe cumplir la característica de “Ajuste a los propósitos” y al menos una de las dos restantes, “Precisión” o “Seguridad”. (Díaz y otros. 2001)

### **2. Estimación de la calidad para cada categoría.**

Para las dos (2) categorías restantes, USABILIDAD y FIABILIDAD, se debe:

- Aplicar las métricas propuestas en el sub-modelo del producto para las categorías seleccionadas y normalizar los resultados de las métricas a una escala de 1 al 5. (Díaz-Antón y otros.)
- Verificar que el 75% de las métricas se encuentran dentro de los valores óptimos (mayor o igual a 4) para cada una de las subcaracterísticas y características. Si no se cumple el 75% de las métricas asociadas, entonces esa subcaracterística o característica tendrá calidad nula.
- Evaluar la categoría. Una categoría es satisfecha si el número de características es altamente satisfecho; es decir, si satisface el 75% de las características asociadas a la categoría. Tratándose de software educativo, donde existen características que son imprescindibles que estén presentes, se proponen las características

mínimas satisfechas que debe tener cada categoría del producto para que ésta pueda ser satisfecha.

**3. Estimar la calidad del producto partiendo de las categorías evaluadas.**

Para que un software educativo obtenga calidad intermedia, deben ser satisfechas las categorías de FIABILIDAD y USABILIDAD. Para alcanzar la calidad avanzada todas las categorías deben ser satisfechas. (Díaz y otros. 2001).