



Universidad  
**Inca Garcilaso de la Vega**  
Nuevos Tiempos. Nuevas Ideas

**ESCUELA DE POSGRADO**  
**Dr. Luis Claudio Cervantes Liñán**

**UN MODELO PARA EL DESARROLLO DE SOFTWARE  
EDUCATIVO EN EL PROCESO DE ENSEÑANZA – APRENDIZAJE  
EN LA ETAPA ESCOLAR**

Tesis para optar el grado académico de:  
Maestro en Ciencias en Ingeniería de Sistemas y Computación

**Presentado por:**

Jorge Christian Almóguer Martínez

LIMA-PERÚ  
2015

**Dedicatoria:**

**A quienes me dieron la vida:  
Papá y Mamá.**

**A quienes comparten mi vida:  
Esposa e hijos.**

## INDICE

<b>Resumen</b> .....	5
<b>Abstract</b> .....	6
<b>Introducción</b> .....	7
<b>Capítulo I: Fundamentos Teóricos</b> .....	9
1.1 Antecedentes de la Investigación .....	9
1.2 Marco Teórico .....	10
1.2.1 Ciclos de Vida del Software .....	10
1.2.2 Metodología de Desarrollo de Software .....	21
1.2.3 Software Educativo .....	33
1.2.4 Metodologías de Desarrollo de Software Educativo .....	36
1.2.5 Educación en Nuestro País. ....	48
1.3 Marco Conceptual .....	56
<b>Capítulo II: El Problema de Investigación</b> .....	58
2.1 Descripción de la Realidad Problemática .....	58
2.2 Delimitación de la Investigación .....	61
2.3 Planteamiento del Problema .....	61
2.4 Objetivos de la Investigación .....	62
2.4.1 Objetivo General .....	62
2.4.2 Objetivo Específico .....	62
2.5 Hipótesis de la Investigación .....	62
2.6 Variables e Indicadores .....	62
2.6.1 Identificación de Variables .....	62
2.6.2 Definición Operacional de las Variables .....	63
2.7 Justificación e Importancia del Estudio .....	63
<b>Capítulo III: Metodología</b> .....	64
3.1 Población y Muestra .....	64
3.2 Método y Diseño de Investigación .....	65
3.3 Técnicas e Instrumentos de Recolección de Datos .....	66
3.4 Técnicas de procedimiento y análisis de datos. ....	67

<b>Capítulo IV: Presentación y Análisis de los Resultados</b> .....	68
4.1 Propuesta del Modelo de Investigación.....	68
4.2 Presentación de Resultados .....	82
4.3 Análisis e interpretación de los resultados .....	83
4.4 Contrastación de Hipótesis .....	90
4.5 Discusión de resultados .....	93
<b>Capítulo V: Conclusiones y Recomendaciones</b> .....	95
5.1 Conclusiones .....	95
5.2 Recomendaciones .....	96
<b>Referencias bibliográficas</b> .....	97
<b>Anexos</b> .....	99

## RESUMEN

La educación es sin lugar a dudas la base de crecimiento de cualquier país y representa el verdadero motor de cualquier política económica y social; responsable por la generación de conocimientos y destrezas intelectuales en las personas, especialmente en nuestros estudiantes de educación básica regular, capaces de aportar un mayor crecimiento económico y protagonizar el cambio hacia la modernidad y eficiencia en el bienestar y calidad de vida de nuestra sociedad.

Por otro lado, nuestro país ha presentado durante los últimos años un crecimiento económico sostenible y reducción de la pobreza extrema; es así, que los acuerdos nacionales han permitido establecer planificaciones a largo plazo. Así mismo, existen diversas políticas para la mejora educativa, entre las que destaca el Plan Nacional de Educación, el cual ha impulsado la carrera pública magisterial. Si bien se han producido avances en estos aspectos, la situación de nuestra educación no ha cambiado significativamente; es por ello, que se debe seguir buscando mejoras que permita un aporte a nuestra educación.

Partiendo desde nuestra perspectiva profesional podemos señalar que un aporte significativo estará orientado en aplicar herramientas tecnológicas, es decir, programas educativos donde los estudiantes mejoren el proceso de enseñanza-aprendizaje; las cuales deberán tener modelos de desarrollo de software adecuados a nuestra realidad.

Esta tesis se orienta en construir un modelo de desarrollo de software educativo en el proceso de enseñanza - aprendizaje, para la etapa escolar, que permita integrar de manera coherente la ingeniería de software y el enfoque pedagógico educativo, a fin de producir software que contribuya en mejorar la educación como parte del proceso de enseñanza – aprendizaje. Esto será de gran aporte porque estaremos formando estudiantes con capacidades y destrezas que permita un aprendizaje significativo.

**Palabra clave:** software educativo, pedagogía socio cognitiva humanista, diseño instruccional.

## ABSTRACT

Education is undoubtedly the basis of growth of any country and represents the true engine of any economic and social policy; responsible for the generation of knowledge and intellectual skills in people, especially our regular basic education students capable of providing greater economic growth and to bring about change towards modernity and efficiency in the welfare and quality of life of our society.

On the other hand, our country has made in recent years, sustainable economic growth and reducing extreme poverty; so that national agreements have established long-term planning. Likewise, there are different policies for educational improvement, most notably the National Education Plan, which has fueled public teaching career. While progress has been made in these respects, the situation of our education has not changed significantly; It is for this reason that you should keep looking for improvements to allow a contribution to our education.

Starting from our professional perspective we note that a significant contribution will be geared to implement technological tools, namely, educational programs where students improve the teaching-learning; which should be development models suited to our reality software.

This thesis aims to build a model of development of educational software in the teaching - learning for school stage, enabling consistently integrate software engineering and educational pedagogical approach, to produce software that helps in improve education as part of the teaching - learning. This will be a great contribution we will be training students with skills and abilities that allow meaningful learning.

**Keyword:** Educational software, social cognitive humanist pedagogy, instructional design.

## INTRODUCCION

La evolución constante de la educación ha acelerado el ritmo de nuestras vidas, creando una sociedad basada en el conocimiento en donde las tecnologías de la información son utilizadas no solo para facilitar la vida cotidiana de las personas; sino ha contribuido en mejor el proceso de enseñanza - aprendizaje de los estudiantes.

El empleo de las tecnologías de la información ha contribuido enormemente en el proceso o desarrollo educativo. En ese sentido, los programas educativos (PE) existentes orientan al estudiante en fortalecer sus conocimientos y se han convertido en una herramienta poderosa que los docentes emplean en el desarrollo de sus sesiones de aprendizaje.

En ese sentido, los programas educativos existentes son importados a nuestro país, en su gran mayoría, y son muy pocas, las organizaciones que desarrollan programas educativos contextualizados a nuestra realidad. Por tanto, se demanda de las Instituciones educativas; es decir universidades e instituciones superiores con carreras o especialidades en computación o informática; preparar a sus estudiantes con técnicas que ayuden a la elaboración de software educativo que permita fortalecer el proceso de enseñanza aprendizaje en nuestras Instituciones de educación básica regular.

La presente tesis propone un modelo para el desarrollo de software educativo en el proceso de enseñanza - aprendizaje en la etapa de educación escolar. Esto permitirá que los especialistas en la construcción de programas de ordenadores puedan emplear este modelo como una herramienta para la elaboración de programas educativos y en especial adaptados a las propuestas pedagógicas educativas contemporáneas.

La presente tesis se ha desarrollado en cinco capítulos que permite el desarrollo formal del trabajo de investigación.

En el capítulo I, denominado Fundamentos teóricos, que describe el conocimiento previo que toda investigación debe tener; en ese sentido nos presenta los antecedentes de investigación, el marco teórico y el marco conceptual.

En el capítulo II, denominado Planteamiento del problema que describe el pensamiento que tiene el investigador, nos presenta la descripción de la realidad problemática,

delimitación de la investigación, planteamiento del problema, objetivo e hipótesis de la investigación, variables e indicadores, justificación e importancia del estudio y propuesta del modelo de tesis.

En el capítulo III, denominado Metodología que describe la población y muestra del estudio, método y diseño de investigación, técnicas e instrumentos de recolección de datos y técnicas de procedimientos y análisis de datos.

En el capítulo IV, denominado Presentación y Análisis de los resultados que presentan los resultados de la encuesta aplicada para efectos de obtener información, análisis e interpretación de los resultados, contrastación de hipótesis planteada y discusión de resultados.

Finalmente el capítulo V, denominado conclusiones y recomendaciones se presentan como resultado de un análisis integral del estudio, las mismas que están en concordancia con el objetivo de la tesis.



# CAPÍTULO I

## 1. FUNDAMENTOS TEORICOS

### 1.1 ANTECEDENTES DE LA INVESTIGACIÓN

La educación es sin lugar a dudas la base del desarrollo en cualquier país y representa el verdadero motor de cualquier política económica y social; responsable por la generación de conocimientos y destrezas intelectuales en las personas capaces de generar un mayor crecimiento económico, y protagonizar el cambio hacia la modernidad y la eficiencia en el bienestar y calidad de vida que requieren nuestros pueblos.

Bajo este contexto el software educativo durante los últimos años, ha tenido un creciente desarrollo y gran parte del mismo ha sido realizado en forma desordenada y poco documentada; considerando este aumento de forma exponencial, surgió la necesidad de lograr metodologías de desarrollo la cual conjugue los métodos que proporciona la ingeniería de software y los enfoques de las teorías educativas, la cual permitan la construcción de programas educativos de calidad. Es así, como el software educativo se ha convertido en una herramienta fundamental para la enseñanza aprendizaje en las aulas.

Se debe entender que un software educativo (SE) representa programas de ordenadores realizados con la finalidad de ser utilizado como facilitadores del proceso de enseñanza - aprendizaje. En tal sentido, un componente de aporte para su elaboración lo representan los diseños instruccionales, los mismos que representan procesos tales como: el desarrollo de materiales educativos y las actividades instruccionales.

Uno de los problemas que se presentan para el desarrollo Software Educativo es la falta de capacitación de las personas involucradas en su elaboración, en donde se presenta la carencia de fundamentos pedagógicos y que en la mayoría caen en la construcción de programas con fundamentos conductistas, por otra lado la existencia de docentes de áreas que orientan a la construcción de SE sin fundamentos informáticos; estas dos formas no favorece el resultado de elaborar software educativos de calidad.

## **1.2 MARCO TEÓRICO**

Para un mejor entendimiento de la tesis se mencionará algunos fundamentos teóricos que ayuden a comprender la investigación; entre ellos tenemos al ciclo de vida del software, metodologías de desarrollo de software, ingeniería de software educativo y el sistema educativo peruano:

### **1.2.1 CICLOS DE VIDA DEL SOFTWARE**

El ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. Las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados. [\[16\]](#)

Estos programas se originan en el hecho de que es muy costoso rectificar los errores que se detectan tarde dentro de la fase de implementación. El ciclo de vida permite que los errores se detecten lo antes posible y por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

El ciclo de vida básico de un software consta de los siguientes procedimientos: definición de objetivos, análisis de los requisitos y su viabilidad, diseño general y en detalle, programación, prueba, integración, validación, documentación, implementación y mantenimiento.

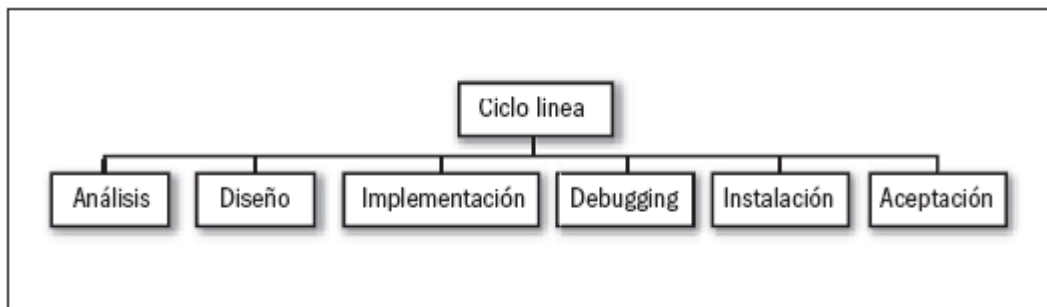
A continuación se citan algunos modelos para el desarrollo del ciclo de vida del software:

#### **1.2.1.1 Ciclo de vida Lineal**

Consiste en descomponer la actividad global del proyecto en etapas separadas que son realizadas de manera lineal, es decir cada etapa se realiza una sola vez, a continuación de la etapa anterior y antes de la etapa siguiente. Con un ciclo de vida lineal es muy fácil dividir las tareas, y prever los tiempos (sumando linealmente los de cada etapa).[\[6\]](#)

Las actividades de cada una de las etapas mencionadas deben ser independientes entre sí, es decir, que es condición primordial que no haya retroalimentación entre ellas, aunque si pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión, requiere también que se conozca desde el primer momento, con excesiva rigidez, lo que va a ocurrir en cada una de las distintas etapas antes de comenzarlas. Esto

último minimiza, también las posibilidades de errores durante la codificación y reduce al mínimo la necesidad de requerir información del cliente o del usuario. En la figura 1.1 se muestra el ciclo de vida lineal.

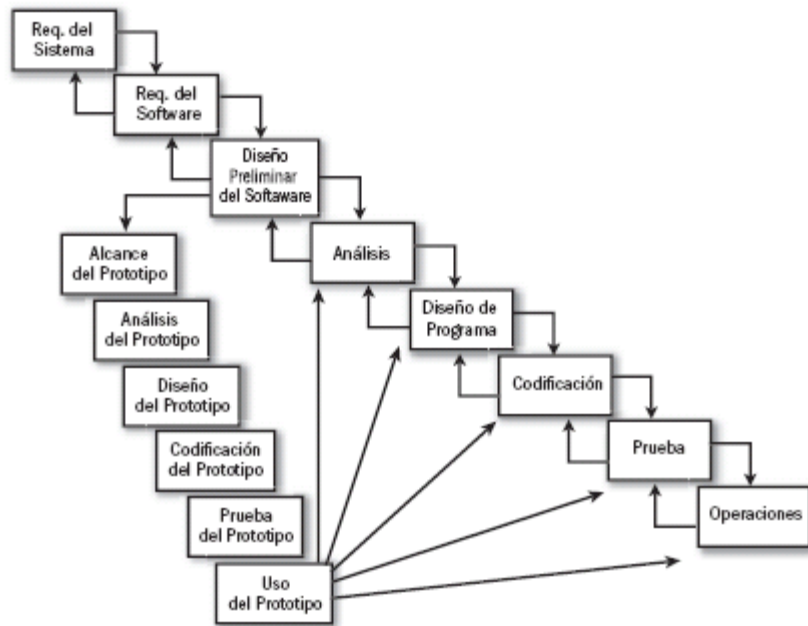


**Figura 1.1 Ciclo de vida Lineal**

Se destaca como **ventaja** la sencillez de su gestión y administración tanto económica como temporal, ya que se acomoda perfectamente a proyectos internos de una empresa para programas muy pequeños de altas, bajas y modificaciones sobre un conjunto de datos. Tiene como **desventaja** que no es apto para desarrollos que superen mínimamente requerimientos de retroalimentación entre etapas, es decir, es muy costoso retomar una etapa anterior al detectar una falla.

#### **1.2.1.2 El Modelo Cascada Puro.**

La versión original del modelo en cascada, fue presentada por Winston Royce en 1970. Es un ciclo de vida que admite iteraciones, contrariamente a la creencia de que es un ciclo de vida secuencial como el lineal. Después de cada etapa se realiza una o varias revisiones para comprobar si se puede pasar a la siguiente. Es un modelo rígido, poco flexible, y con muchas restricciones. Aunque fue uno de los primeros, y sirvió de base para el resto de los modelos de ciclo de vida. En la figura 1.2 se muestra el ciclo de vida en cascada. [\[6\]](#)

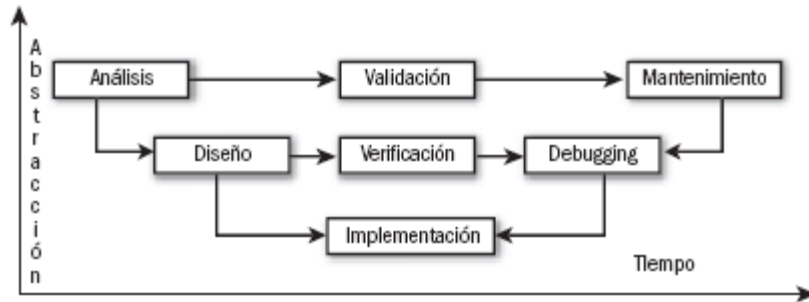


**Figura 1.2 Ciclo de vida en Cascada**

Una de sus **ventajas**, además de su planificación sencilla, es la de proveer un producto con un elevado grado de calidad sin necesidad de un personal altamente calificado. Se pueden considerar como **desventaja** la necesidad de contar con todos los requerimientos (o la mayoría) al comienzo del proyecto, y si se han cometido errores y no se detectan en la etapa inmediata siguiente, es costoso y difícil volver atrás para realizar la corrección posterior. Además, los resultados no los veremos hasta que estemos en las etapas finales, por lo que, cualquier error detectado nos trae retraso y aumenta el costo del desarrollo en función del tiempo que insume la corrección de estos.

### 1.2.1.3 Ciclo de Vida en V

Este ciclo fue diseñado por Alan Davis, contiene las mismas etapas que el ciclo de vida en cascada puro. A diferencia de aquel, a este se le agregaron dos subetapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y debugging, como se muestra en la figura 1.3. [6]



**Figura 1.3 Ciclo de vida en V**

Podemos utilizar este modelo en aplicaciones que si bien son simples (pequeñas transacciones sobre bases de datos por ejemplo), necesitan una confiabilidad muy alta.

Las ventajas y desventajas de este modelo son las mismas que el ciclo de vida en cascada, con el agregado de los controles cruzados entre etapas para lograr una mayor corrección.

#### 1.2.1.4 Ciclo de vida tipo Sashimi

Es un ciclo parecido al ciclo de vida cascada puro, con la diferencia de que el ciclo en cascada no se pueden solapar las etapas, y en este sí. Esto suele, en muchos casos, aumentar su eficiencia ya que la retroalimentación entre etapas se encuentra implícitamente en el modelo. En la figura 1.4 se muestra el ciclo de vida tipo Sashimi. [6]



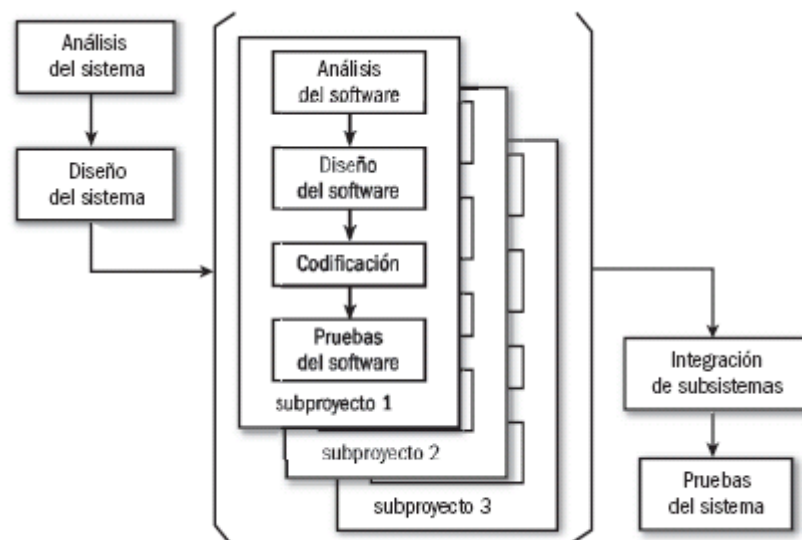
**Figura 1.4 Ciclo de vida tipo Sashimi**

Se hace notar como **ventajas** la ganancia de calidad en lo que respecta al producto final y la falta de necesidad de una documentación detallada (el ahorro proviene por el solapado

de las etapas). Sus **desventajas** también se refieren al solapamiento de las etapas: es muy difícil gestionar el comienzo y fin de cada etapa y los problemas de comunicación, si aparecen, generan inconsistencias en el proyecto.

### 1.2.1.5 Ciclo de vida en cascada con subproyectos

Sigue el modelo de ciclo de vida en cascada. Cada una de las cascadas se divide en subetapas independientes que se pueden desarrollar en paralelo. En la figura 1.5 se muestra el ciclo de vida en cascada con subproyectos. [6]



**Figura 1.5 Ciclo de vida en cascada con subproyectos**

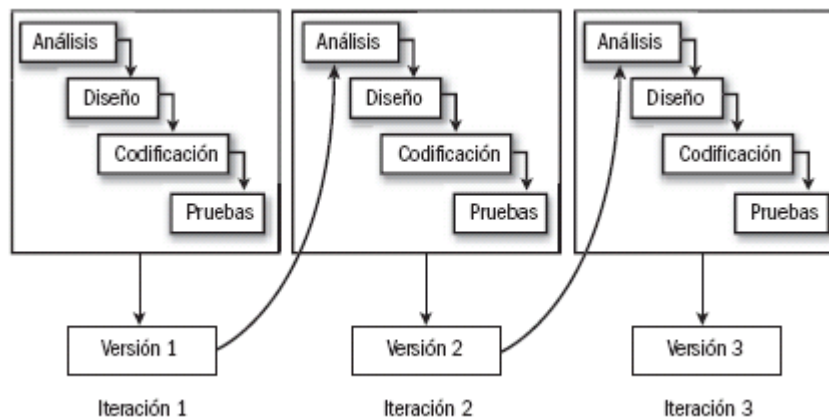
La ventaja es que se puede tener más gente trabajando al mismo tiempo, pero la desventaja es que pueden surgir dependencias entre las distintas subetapas que detengan el proyecto temporalmente si no es gestionada de manera correcta. Podemos utilizar este modelo para administrar cualquier proyecto mencionados en los modelos anteriores, pero cuidando en administrar muy bien el tiempo.

### 1.2.1.6 Ciclo de vida Iterativo

Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.

Es la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es

quien luego de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repiten hasta obtener un producto que satisfaga al cliente. En la figura 1.6 se muestra el ciclo de vida iterativo. [6]



**Figura 1.6 Ciclo de vida Iterativo**

Se suele utilizar en proyectos en los que los requerimientos no están claros de parte del usuario, por lo que se hace necesaria la creación de distintos prototipos para presentarlos y conseguir la conformidad del cliente.

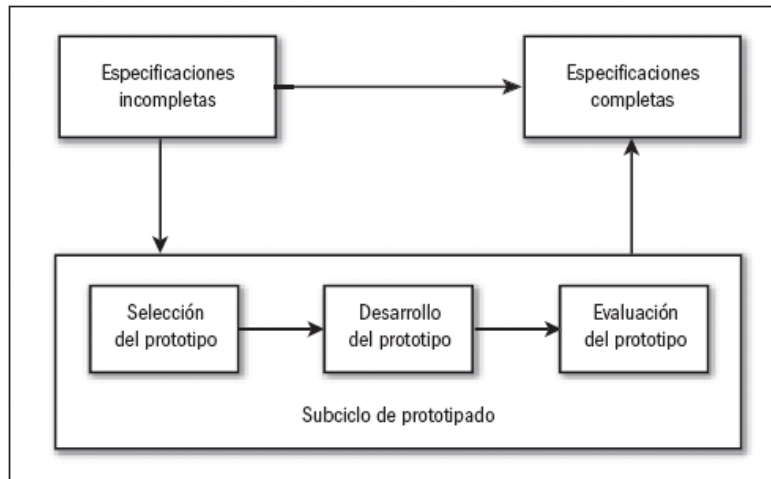
Podemos adoptar este modelo en aplicaciones medianas a grandes, en las que el usuario o cliente final no necesita todas las funcionalidades desde el principio del proyecto. Quizás una empresa que debe migrar sus aplicaciones hacia otra arquitectura, y desea hacerlo paulatinamente, es una opción ideal para este tipo de ciclo de vida.

### 1.2.1.7 Ciclo de Vida por Prototipos

El uso de programas prototipo no es exclusivo del ciclo de vida iterativo. En la práctica los prototipos se utilizan para validar los requerimientos de los usuarios en cualquier ciclo de vida. [6]

Si no se conoce exactamente como desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un prototipo, o sea, un producto parcial y provisional. En este modelo, el objetivo es lograr un producto intermedio, antes de realizar el producto final, para conocer mediante el prototipo como responderán las funcionalidades previstas para el producto final.

Antes de adoptar este modelo de ciclo debemos evaluar si el esfuerzo por crear un prototipo vale realmente la pena adoptarlo. En la figura 1.7 se observa el ciclo de vida por prototipos.



**Figura 1.7 Ciclo de vida por Prototipos**

Se utiliza mayoritariamente en desarrollos de productos con innovaciones importantes, o en el uso de tecnologías nuevas o poco probadas, en las que la incertidumbre sobre los resultados a obtener, o la ignorancia sobre el comportamiento, impiden iniciar un proyecto secuencial.

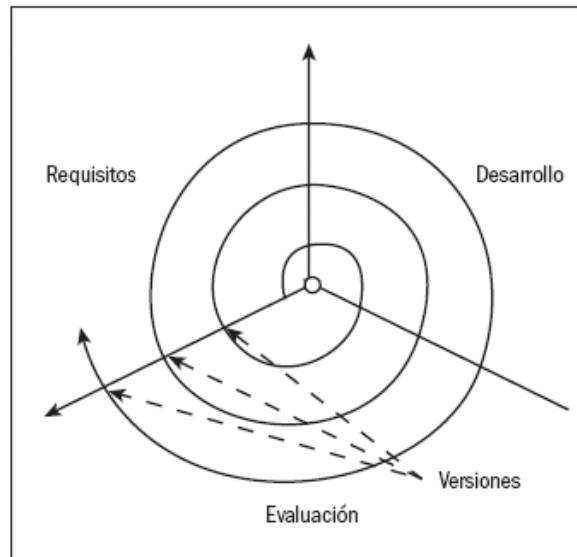
La ventaja de este ciclo se basa en que el único apto para desarrollos en los que no se conoce a priori sus especificaciones o la tecnología a utilizar. Como contrapartida, por este desconocimiento, tiene la desventaja de ser altamente costoso y difícil para la administración temporal.

#### **1.2.1.8 Ciclo de vida Evolutivo**

Este modelo acepta que los requerimientos del usuario pueden cambiar en cualquier momento. [6]

La práctica nos demuestra que obtener todos los requerimientos al comienzo del proyecto es extremadamente difícil, no sólo por la dificultad del usuario de transmitir su idea, sino porque estos requerimientos evolucionan durante el desarrollo y de esta manera, surgen nuevos requerimientos a cumplir. El modelo de ciclo de vida evolutivo afronta este problema mediante una iteración de ciclos requerimientos-desarrollo- evaluación. En la figura 1.8 se observa este ciclo de vida.





**Figura 1.8 Ciclo de vida Evolutivo**

Resulta ser un modelo muy útil cuando desconocemos la mayoría de los requerimientos iniciales, o estos requerimientos no están completos.

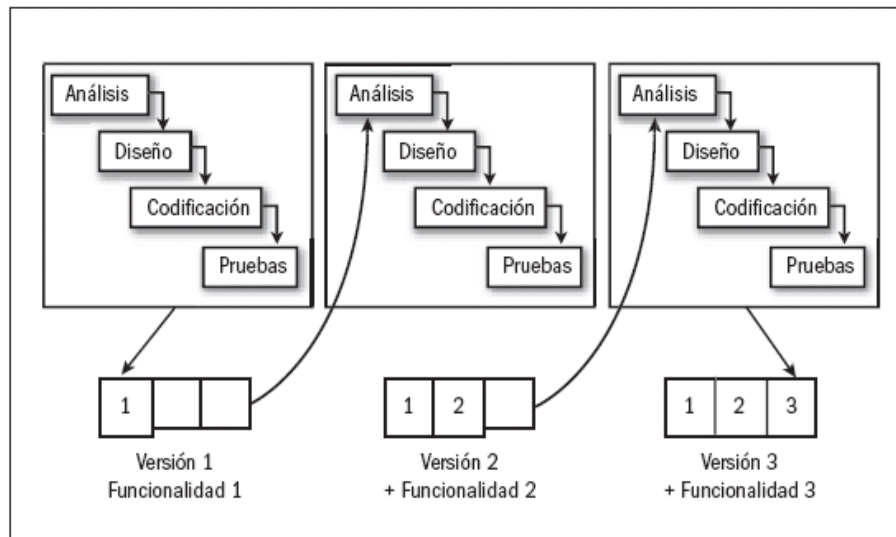
Tomemos como ejemplo un sistema centralizado de stock-ventas-facturación, en el cual hay muchas áreas que utilizarán la aplicación. Tenemos dos complicaciones: la primera, los usuarios no conocen de informática, la segunda, no es uno, sino varios los sectores que nos pueden pedir modificaciones o hacer nuevas solicitudes. Además, el pedido de un sector puede influir en los requerimientos del otro. Se hace necesario, entonces, lograr que la aplicación evolucione hasta lograr las satisfacciones de los todos los sectores involucrados.

#### **1.2.1.9 Ciclo de vida Incremental**

Este modelo se basa en la filosofía de construir incrementando funcionalidades del programa. [6]

Se realiza construyendo por módulos que cumplen las diferentes funciones del sistema. Esto permite ir aumentando gradualmente las capacidades del software. Este ciclo facilita la tarea del desarrollo permitiendo a cada miembro del equipo desarrollar un modelo particular en el caso de que el proyecto sea realizado por un equipo de programadores.

Es una repetición del ciclo cascada, aplicándose este ciclo en cada funcionalidad del programa a construir. Al final de cada ciclo le entregamos una versión al cliente que contiene una nueva funcionalidad. Este ciclo nos permite realizar una entrega al cliente antes de terminar el proyecto. En la figura 1.9 se observa el ciclo de vida incremental.

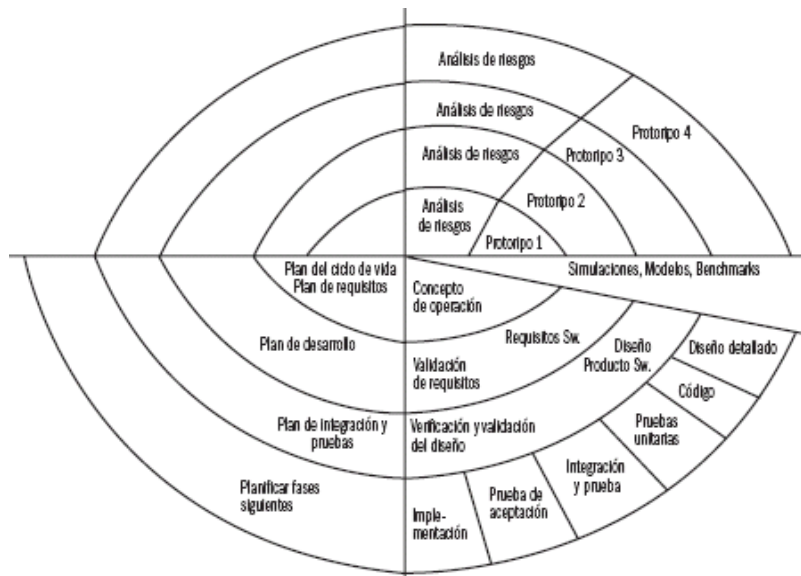


**Figura 1.9 Ciclo de vida Incremental**

Este modelo no está pensado para cierto tipo de aplicaciones, sino está orientado a cierto tipo de usuario o cliente. Podremos utilizar este modelo para casi cualquier proyecto, pero será verdaderamente útil cuando el usuario necesite entregas rápidas, aunque sean parciales.

#### 1.2.1.10 Ciclo de vida en Espiral

Este ciclo puede considerarse una variación del modelo con prototipado, fue diseñado por Boehm. El modelo se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final. Toma los beneficios de los ciclos de vida incremental y por prototipos, pero se tiene más en cuenta el concepto de riesgo que aparece debido a las incertidumbres e ignorancias de los requerimientos proporcionados al principio del proyecto o que surgirán durante el desarrollo. A medida que el ciclo se cumple, se van obteniendo prototipos sucesivos que van ganando la satisfacción del cliente o usuario, tal como se muestra en la figura 1.10. [6]



**Figura 1.10 Ciclo de vida en Espiral**

En este modelo hay cuatro actividades que envuelven a las etapas.

- **Planificación:** Relevamiento de requerimientos iniciales o luego de una iteración.
- **Análisis de Riesgo:** De acuerdo con el relevamiento de requerimientos decidimos si continuamos con el desarrollo.
- **Implementación:** Desarrollamos un prototipo basado en los requerimientos.
- **Evaluación:** El cliente evalúa el prototipo, si da su conformidad, termina el proyecto. Caso contrario, incluimos los nuevos requerimientos solicitados por el cliente en la siguiente iteración.

La ventaja más notoria de este modelo de desarrollo de software es que puede comenzarse el proyecto con un alto grado de incertidumbre, se entiende también como ventaja el bajo riesgo de retraso en caso de detección de errores, ya que se puede solucionar en la próxima rama del espiral.

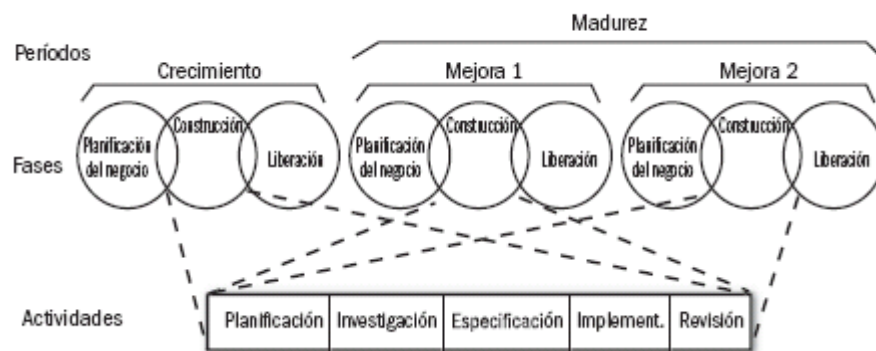
Alguna de las desventajas son: el costo temporal que suma cada vuelta del espiral, la dificultad para evaluar los riesgos y la necesidad de la presencia o la comunicación continúa con el cliente.

### 1.2.1.11 Ciclo de vida Orientado a Objetos

Puede considerarse como un modelo pleno a seguir, como así también una alternativa dentro de los modelos anteriores. [6]

En esta propuesta cada funcionalidad o requerimiento solicitado por el usuario, es considerado un objeto. Los objetos están representados por un conjunto de propiedades, a los cuales denominamos **atributos**, por otra parte, al comportamiento que tendrán estos objetos los denominados **métodos**.

Vemos que tanto la filosofía de esta metodología, los términos utilizados en ella y sus fines, coinciden con la idea de obtener un concepto de objeto sobre casos de la vida real. En la figura 1.11 se observa el ciclo de vida orientado a objetos.



**Figura 1.11 Ciclo de vida Orientado a Objetos**

La característica principal de este modelo es la abstracción de los requerimientos de usuario, por lo que es mucho más flexible que los restantes, que son rígidos en requerimientos y definición, soportando mejor la incertidumbre que los anteriores, aunque sin garantizar la ausencia de riesgos. La abstracción es lo que nos permite analizar y desarrollar las características esenciales de un objeto, despreocupándonos de las menos relevantes.

En este modelo se utiliza las llamadas fichas CRC (**clase, responsabilidades, colaboración**) como herramienta para obtener las abstracciones y mecanismos clave de un sistema analizando los requerimientos del usuario. En la ficha CRC se escribe el nombre de la clase u objeto, sus responsabilidades (los métodos) y sus colaboradores. Estas fichas, además, nos ayudan a confeccionar los denominados **Casos de Uso**.

## 1.2.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida.

Por tanto, no hay un consenso entre los autores sobre el concepto de **metodología de desarrollo de software**, por lo tanto no existe una definición universalmente aceptada. Sí hay un acuerdo en considerar a la metodología como “un conjunto de pasos y procedimientos que deben seguirse para el desarrollo del software”.<sup>[3]</sup>

Generalizando, Piattini llega a la definición de metodología de desarrollo como “un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”. Normalmente consistirá en fases o etapas descompuestas en subfases, módulos, etapas, pasos, etc. Esta descomposición ayuda a los desarrolladores en la elección de las técnicas a utilizar en cada estado del proyecto, facilitando la planificación, gestión, control y evaluación de los proyectos.

Por otro lado, todo desarrollo de software es riesgoso y difícil de controlar, pero si no consideramos una metodología de por medio, lo que obtendremos son clientes descontentos con el resultado y desarrolladores aún más insatisfechos.

Muchas veces realizamos el diseño de nuestro software de manera rígida, con los requerimientos que el cliente nos solicitó, de tal manera que cuando el cliente en la etapa final (etapa de prueba), solicita un cambio se nos hace muy difícil realizarlo, pues si lo hacemos, altera muchas cosas que no habíamos previsto, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido.

En la actualidad existen diversas metodologías de desarrollo de software, entre ellas las tradicionales y las ágiles; en tal sentido citaremos alguna de ellas:

### 1.2.2.1 Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software: [\[16\]](#)

- **Inicio**, el objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, en esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**, en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión**, el objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes, ver **figura 1.12**.

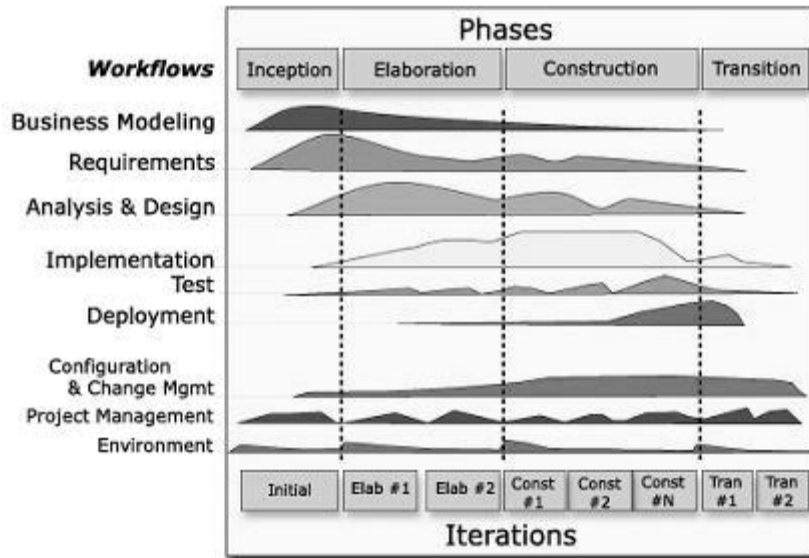
Es relevante mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

#### **Disciplina de Desarrollo**

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

#### **Disciplina de Soporte**

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto



**Figura 1.12: Fases e Iteraciones de la Metodología RUP**

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se **convierta luego en un entregable al cliente**. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

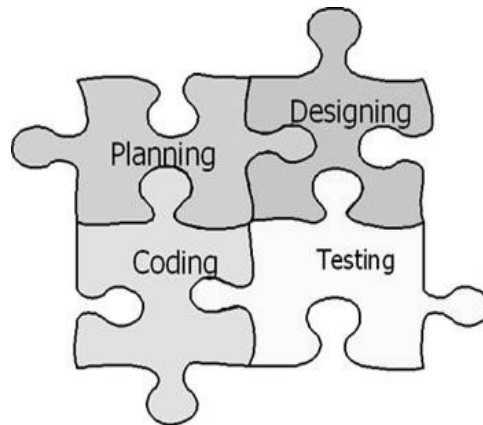
Los elementos del RUP son:

- **Actividades**, con los procesos que se llegan a determinar en cada iteración.
- **Trabajadores**, vienen hacer las personas o entes involucrados en cada proceso.
- **Artefactos**, un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

### 1.2.2.2. Extreme Programing (XP)

Es una de las metodologías de desarrollo de software ágil que consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. En la **figura 1.13** se aprecia los componentes importantes de esta propuesta.[\[16\]](#)



**Figura 1.13: Metodología Extreme Programing.**

**Características de Extreme Programing**, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

#### **Propuesta de Extreme Programing**

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.



- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

### **Derechos del Cliente**

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses.

### **Derechos del Desarrollador**

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- La **comunicación** entre los usuarios y los desarrolladores
- La **simplicidad** al desarrollar y codificar los módulos del sistema
- La **retroalimentación** concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales

### **1.2.2.3 SCRUM**

Es una metodología ágil de desarrollo, aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. [\[16\]](#)

Es una metodología de desarrollo simple, que responde al ciclo iterativo e incremental; que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve (normalmente de 30 días).

Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente.

#### Características de SCRUM

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como ítems de la lista Product Backlog.
- El producto se construye en una serie de Sprints de un mes de duración.

A continuación se describirán algunos conceptos de Scrum:

- ***Herramientas y Prácticas***

Scrum no requiere ni provee prácticas de Ingeniería. En lugar de eso, especifica prácticas y herramientas de gerencia que se aplican en sus distintas fases para evitar el caos originado por la complejidad e imposibilidad de realizar predicciones.

- ***Lista Product Backlog***

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para un Sprint.

El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Existe un rol asociado con esta lista y es el de Product Owner. Si alguien quiere realizar cualquier modificación sobre la lista por ejemplo: agregar o incrementar la prioridad de sus elementos tiene que convencer al Product Owner.

- **Sprints**

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

El objetivo de un Sprint debe ser expresado en pocas palabras para que sea fácil de recordar y esté siempre presente en el equipo.

- **Burn down Chart**

La *burn down chart* es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

- **Sprint Backlog**

Es el punto de entrada de cada Sprint. Es una lista que tiene los ítems de la Product Backlog List que van a ser implementados en el siguiente Sprint.

Los ítems son seleccionados por el Scrum Team, el Scrum Master y el Product Owner en la Sprint Planning Meeting a partir de la priorización de los ítems y los objetivos que se marcaron para ese Sprint. A partir de los objetivos a cumplir durante el Sprint el Scrum

Team determina que tareas debe desempeñar para cumplir el objetivo. De esto surge el Sprint Backlog.

- ***Stabilization Sprints***

En estos Sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad. Suelen aplicarse cuando se prepara un producto para el release. Son útiles cuando se están realizando pruebas beta, se está introduciendo a un equipo en la metodología de Scrum o cuando la calidad de un producto no alcanza los límites esperados.

No fueron definidos por Scrum pero han sido recomendados por su utilidad al aplicar esta metodología.

- ***Scrum of Scrums o MetaScrum***

Los equipos de Scrum suelen tener entre 5 y 10 personas, sin embargo esta metodología ha sido aplicada en proyectos que involucran más de 600 personas. Esto ha sido llevado a cabo dividiendo a los accionistas en equipos de pequeños de hasta 10 personas aproximadamente. Y definiendo jerárquicamente personas que pertenecen a dos equipos, es decir, además de su rol específico dentro de un equipo tienen el rol de enlace en un equipo superior.

- ***Roles y responsabilidades en Scrum***

Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: propietario del producto, equipo, gestor de Scrum (también Scrum Manager o Scrum Master) y “otros interesados”.

- ***Product Owner***

El *Product Owner* representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

- ***ScrumMaster (o Facilitador)***

El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

- **Equipo**

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

- **Usuarios**

Es el destinatario final del producto. Como bien lo dice la paradoja, El árbol cae en el bosque cuando no hay nadie ¿Hace ruido? Aquí la definición sería Si el software no es usado ¿fue alguna vez escrito?.

- **Stakeholders (Clientes, Proveedores).**

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

- **Managers**

Es la gente que establece el ambiente para el desarrollo del producto.

- A continuación se señalan los tipos de Reuniones en Scrum

- **Daily Scrum**

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama "daily standup". El scrum tiene unas guías específicas, la cual se detalla a continuación:

- La reunión comienza puntualmente a su hora. A menudo hay castigos -acordados por el equipo- para quién llega tarde (por ejemplo: dinero, flexiones, llevar colgando una gallina de plástico del cuello, etc)
    - Todos son bienvenidos, pero solo los responsables pueden hablar.
    - La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
    - Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta)
    - La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.
    - Durante la reunión, cada miembro del equipo contesta a tres preguntas:[]
    - ¿Qué has hecho desde ayer?
    - ¿Qué es lo que estás planeando hacer hoy?

- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).

- **Scrum de Scrum**

Cada día normalmente después del “Daily Scrum”. Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración. Asiste una persona asignada por cada equipo.

La agenda será la misma como del Daily Scrum, además de las siguientes cuatro preguntas: ¿Qué ha hecho tu equipo desde nuestra última reunión?. ¿Qué hará tu equipo antes que nos volvamos a reunir? ¿Hay algo que demora o estorba a tu equipo? ¿Estás a punto de poner algo en el camino del otro equipo?

- **Reunión de Planificación del Sprint (Sprint Planning Meeting)**

Al inicio del ciclo Sprint (cada 15 o 30 días), una “Reunión de Planificación del Sprint” se lleva a cabo. Selecciona que trabajo se hará y se prepara con el equipo completo, el Sprint Backlog que detalla el tiempo que tomara hacer el trabajo. Se Identifica y comunica cuánto del trabajo es probable que se realice durante el actual Sprint. Ocho horas como limite; y al final del ciclo Sprint, dos reuniones se llevaran a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”

- **Reunión de Revisión del Sprint (Sprint Review Meeting):** Revisar el trabajo que fue completado y no completado Presentar el trabajo completado a los interesados (alias “demo”) en un tiempo de cuatro horas como límite.
- **Retrospectiva del Sprint (Sprint Retrospective):** Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

#### 1.2.2.4 MICROSOFT SOLUTION FRAMEWORK (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. En la **figura 1.14** se aprecia los componentes de esta propuesta. [6]



**Figura 1.14: Metodología Microsoft Solution Framework (MSF)**

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

- **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.

- **Modelo de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
- **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores.



### **1.2.3 SOFTWARE EDUCATIVO**

El Software Educativo se pueden considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza – aprendizaje.

Se caracterizan por ser altamente interactivos, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones de experimentados profesores, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico.

El software educativo pueden tratar las diferentes materias (Matemática, Idiomas, Geografía, Dibujo), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los estudiantes, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los estudiantes y más o menos rico en posibilidades de interacción.[\[2\]](#)

#### **1.2.3.1 Características del Software Educativo**

Entre ellos podemos mencionar las siguientes características:

- Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Facilita las representaciones animadas.
- Incide en el desarrollo de las habilidades a través de la ejercitación.
- Permite simular procesos complejos.
- Reduce el tiempo de que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- Facilita el trabajo independiente y a la vez un tratamiento individual de sus conocimientos.
- Entorno de comunicación o interfaz, el cual permite un dialogo posibilitando la interactividad característica de estos materiales.

### **1.2.3.2 Categorización del Software Educativo**

Según su naturaleza informática, los podemos categorizar como:

- De consulta: Como por ejemplo los atlas geográficos.
- Tutoriales: Son aquellos que transmiten conocimiento al estudiante a través de pantallas que le permiten aprender a su propio ritmo, pudiendo volver sobre cada concepto cuantas veces lo desee.
- Ejercitación: Permiten al estudiante reforzar conocimientos adquiridos con anterioridad, llevando el control de los errores y llevando una retroalimentación positiva. Proponen diversos tipos de ejercicios tales como “completar”, “unir con flechas”, “selección múltiple” entre otros.
- Simulación: Simulan hechos y/o procesos en un entorno interactivo, permitiendo al usuario modificar parámetros y ver cómo reacciona el sistema ante el cambio producido.
- Lúdicos: Proponen a través de un ambiente lúdico interactivo, el aprendizaje, obteniendo el usuario puntaje por cada logro o desacierto. Crean una base de datos con los puntajes para conformar un “cuadro de honor”.
- Micromundos: ambiente donde el usuario, explora alternativas, puede probar hipótesis y descubrir hechos verdaderos.

### **1.2.3.3 Funciones del Software Educativo**

El software educativo cuando se aplican a la realidad educativa, realizan las funciones básicas propias de los medios didácticos en general y además, en algunos casos, según la forma de uso que determina el profesor, pueden proporcionar funcionalidades específicas, entre ellas podemos mencionar:

- Función informativa: la mayoría de los programas a través de sus actividades presentan unos contenidos que proporcionan una información estructuradora de la realidad a los estudiantes. Los programas tutoriales y, especialmente, las bases de datos, son los programas que realizan más marcadamente una función informativa.
- Función instructiva: todos los programas educativos orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a facilitar el logro de unos objetivos educativos específicos. Con todo, si bien el computador actúa en general como mediador en la construcción del conocimiento y el metaconocimiento de los estudiantes, son los programas tutoriales los que realizan de manera más explícita esta

función instructiva, ya que dirigen las actividades de los estudiantes en función de sus respuestas y progresos.

- Función motivadora: generalmente los estudiantes se sienten atraídos e interesados por todo el software educativo, ya que los programas suelen incluir elementos para captar la atención de los estudiantes, mantener su interés y, cuando sea necesario, focalizarlo hacia los aspectos más importantes de las actividades.
- Función evaluadora: la interactividad propia de estos materiales, que les permite responder inmediatamente a las respuestas y acciones de los estudiantes, les hace especialmente adecuados para evaluar el trabajo que se va realizando con ellos.
- Función investigadora: los programas no directivos, especialmente las bases de datos, simuladores y Micromundos, ofrecen a los estudiantes, interesantes entornos donde investigar: buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc. Además, tanto estos programas como los programas herramienta, pueden proporcionar a los profesores y estudiantes instrumentos de gran utilidad para el desarrollo de trabajos de investigación que se realicen básicamente al margen de los computadores.
- Función expresiva: dado que los computadores son unas máquinas capaces de procesar los símbolos mediante los cuales las personas representamos nuestros conocimientos y nos comunicamos, sus posibilidades como instrumento expresivo son muy amplias.
- Función metalingüística: mediante el uso de los sistemas operativos (MS/DOS, WINDOWS) y los lenguajes de programación (BASIC, LOGO...) los estudiantes pueden aprender los lenguajes propios de la informática.
- Función lúdica: trabajar con los computadores realizando actividades educativas es una labor que a menudo tiene unas connotaciones lúdicas y festivas para los estudiantes.
- Función innovadora: aunque no siempre sus planteamientos pedagógicos resulten innovadores, los programas educativos se pueden considerar materiales didácticos con esta función ya que utilizan una tecnología recientemente incorporada a los centros educativos y, en general, suelen permitir muy diversas formas de uso. Esta versatilidad abre amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

## 1.2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE EDUCATIVOS

En el campo de desarrollo de software Educativo las investigaciones han ido evolucionando y en ese sentido se evalúa muy poco énfasis en el aspecto pedagógico para su desarrollo. Sin embargo, se citarán algunas propuestas de autores que de una y otra forma escribieron al respecto:

### 1.2.4.1 Metodología basada en el Prototipo Evolutivo.

Está basado en el **prototipo evolutivo** con refinamientos sucesivos, la cual presenta las siguientes características: [4]

- Cuando el software a desarrollar es requerido por el cliente y es de interés tener una idea de cómo será el programa lo antes posible, para tal fin se entregaran prototipos con funcionalidades en forma incremental.
- Estos prototipos entregados permitirán una revisión de los requerimientos y un refinamiento a fin de acercarse al producto final.
- El cliente identifica si el producto está de acuerdo a sus necesidades y consideraciones.

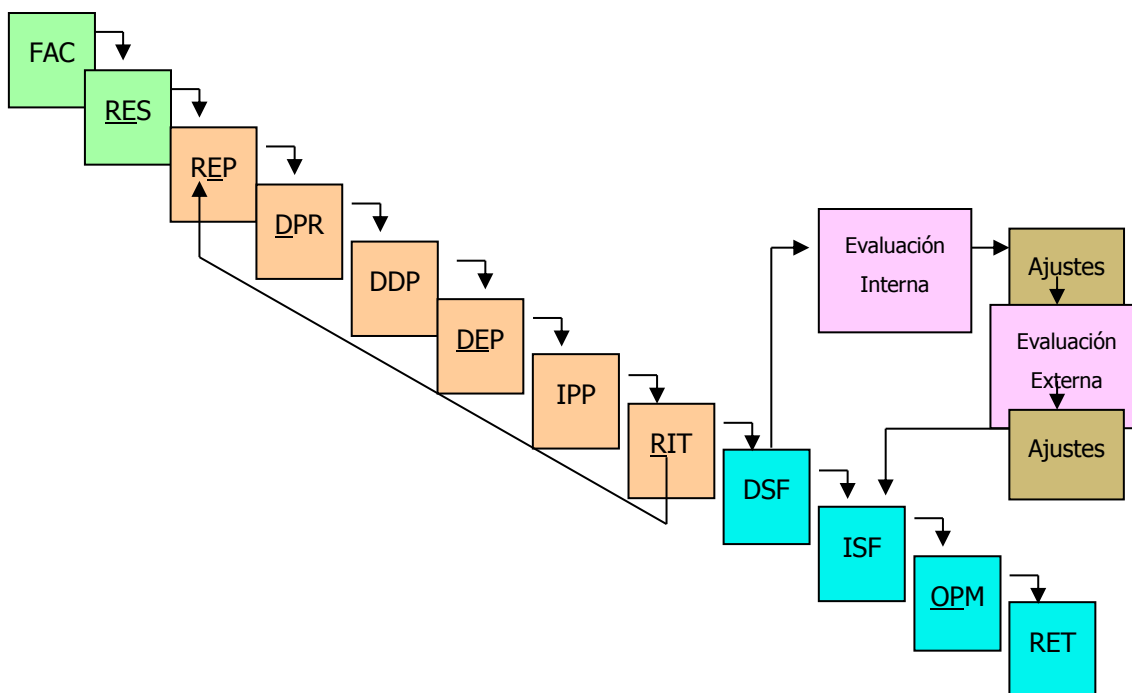
El ciclo de vida de esta metodología se define en las siguientes etapas, como se aprecia en la **Tabla 1.1**:

<b>Etapas</b>	<b>Descripción</b>
<b>Factibilidad (FAC)</b>	<ul style="list-style-type: none"><li>• Se define el producto software y se determina su factibilidad en el ciclo de vida,</li><li>• También se determina las ventajas y desventajas respecto de otros productos.</li></ul>
<b>Requisitos del sistema (RES)</b>	<ul style="list-style-type: none"><li>• Se definen las funcionalidades requeridas para el desarrollo del sistema (o programa), las interfaces y el tipo de diseño.</li></ul>
<b>Especificación de requisitos del prototipo (REP)</b>	<ul style="list-style-type: none"><li>• Se especifican las funciones requeridas, las interfaces y el rendimiento para el prototipo.</li></ul>
<b>Diseño del prototipo (DPR)</b>	<ul style="list-style-type: none"><li>• Poner en ejecución el plan del prototipo, ya que una vez fijadas las restricciones con el usuario, hay que mostrar el mismo funcionando.</li><li>• Se hace un análisis de cómo se va a trabajar, qué módulos se van a hacer, con qué lógica y qué funciones se van a usar.</li></ul>

<b>Diseño detallado del prototipo (DDP)</b>	<ul style="list-style-type: none"> <li>• Especificación de la estructura de control, la estructura de los datos, las relaciones de interfaces, el tamaño, los algoritmos básicos y las estimaciones de cada componente del programa.</li> <li>• Se definen y se documentan los algoritmos que llevarán a cabo la función a realizar por cada uno de los módulos.</li> <li>• El diseño de software, es un proceso que se centra en cuatro atributos distintos del programa: la estructura de datos, la arquitectura del software, el detalle procedimental y la caracterización de la interface.</li> <li>• También se deben traducir los requisitos a una representación del software que pueda ser establecida de forma que se obtenga la calidad requerida antes de que comience la codificación.</li> </ul>
<b>Desarrollo del prototipo (codificación) (DEP)</b>	<ul style="list-style-type: none"> <li>• Consiste en realizar la codificación o diseño detallado, en forma legible para la máquina.</li> </ul>
<b>Implementación y prueba del prototipo (IPP)</b>	<ul style="list-style-type: none"> <li>• Consiste en lograr un funcionamiento adecuado del producto software en el sistema informático, funcionando operacionalmente, incluyendo objetivos tales como la conversión del programa y datos (si la hubiere), la instalación y el entrenamiento.</li> <li>• La prueba debe asegurar que las entradas producidas produzcan los resultados que se esperan realmente.</li> </ul>
<b>Refinamiento iterativo de las especificaciones del prototipo (RIT)</b>	<ul style="list-style-type: none"> <li>• Es un aumento de la funcionalidad del sistema, para luego volver REP a fin de aumentar la funcionalidad del prototipo o continuar, si se logró el objetivo y alcance deseados.</li> </ul>
<b>Diseño del sistema final (DSF)</b>	<ul style="list-style-type: none"> <li>• Consiste en ajustar las restricciones o condiciones finales e integrar los módulos.</li> </ul>
<b>Implementación del sistema final (ISF)</b>	<ul style="list-style-type: none"> <li>• Sistema informático funcionando operativamente, incluyendo tales objetivos como conversión del programa y datos, (si la hubiere), la instalación y la capacitación del personal.</li> </ul>
<b>Operación y Mantenimiento (OPM)</b>	<ul style="list-style-type: none"> <li>• Puesta en funcionamiento del sistema informático, objetivo que se repite para cada actualización.</li> </ul>
<b>Retiro (RET)</b>	<ul style="list-style-type: none"> <li>• (si corresponde) Es una transición adecuada de las funciones realizadas para el producto y sus sucesores.</li> </ul>

Tabla 1.1: ETAPAS DEL CICLO DE VIDA DE LA METODOLOGÍA POR ZULMA CATALDI [4]

En la siguiente **figura 1.15**, se muestra la secuencia de cada una de las etapas, así como la recursividad la cual evidencia un refinamiento en los prototipos del software educativo.



**Figura 1.15: Ciclo del tipo de vida del Prototipo Evolutivo [4]**

Luego de la definición de las etapas, se procede a la **construcción del documento** denominado matriz de actividades, donde se definen las actividades a desarrollar en cada fase, los procesos asociados a cada una de ellas.

- **Aspecto Pedagógico**

Las actividades agregadas a la metodología incluyen aquellas concernientes al desarrollo de software teniendo en cuenta los aspectos educativos. Sobre este último esta metodología no define una teoría educativa en particular, sino que las actividades permiten trabajar con un enfoque cognitivista – constructivo.

Según esta propuesta se identificó que deben ser extendidas a las fases. Por lo tanto hubo necesidad de incorporar **nuevos procesos** que cautelaran las necesidades pedagógicas-didácticas, como se muestra en la **tabla 1.2**.

<p style="text-align: center;"><b>PROCESO DE IDENTIFICACIÓN DE LA NECESIDAD EDUCATIVA (FAC)</b></p>	<ul style="list-style-type: none"> <li>• Identificar la necesidad del programa educativo</li> <li>• <b>Seleccionar la teoría educativa a utilizar.</b></li> </ul>
<p style="text-align: center;"><b>PROCESO DE ANÁLISIS DE LOS REQUISITOS EDUCATIVOS. (FAC -RES)</b></p>	<ul style="list-style-type: none"> <li>• Definir los objetivos educativos</li> <li>• Definir las características del grupo destinatario</li> <li>• Definir contenidos y el recorte de contenidos</li> <li>• Definir estrategias didácticas</li> <li>• Definir las actividades mentales a desarrollar</li> <li>• Definir nivel de integración curricular</li> <li>• <b>Definir tipo de uso del programa y nivel de interactividad.</b></li> <li>• Definir efectos motivantes.</li> <li>• Definir posibles caminos pedagógicos</li> <li>• Definir tiempo y modo de uso del programa</li> <li>• <b>Definir hardware asociado.</b></li> </ul>
<p style="text-align: center;"><b>PROCESO DE EVALUACIÓN DE LOS PROTOTIPOS DE SOFTWARE (REP)</b></p>	<ul style="list-style-type: none"> <li>• Confeccionar el instrumento de evaluación</li> <li>• Evaluar prototipos del programa</li> <li>• Elaborar los resultados</li> <li>• Identificar cambios y ajustes a realizar</li> <li>• Llevar a cabo modificaciones pertinentes</li> <li>• Archivar resultados</li> </ul>
<p style="text-align: center;"><b>PROCESO DE EVALUACIÓN INTERNA Y EXTERNA DEL SOFTWARE (IPP)</b></p>	<ul style="list-style-type: none"> <li>• Confeccionar el instrumento de evaluación</li> <li>• Evaluar interna y externamente el programa</li> <li>• Elaborar los resultados</li> <li>• Identificar cambios y ajustes a realizar</li> <li>• Llevar a cabo modificaciones pertinentes</li> <li>• Archivar resultados</li> </ul>
<p style="text-align: center;"><b>PROCESO DE EVALUACIÓN CONTEXTUALIZADA (IPP)</b></p>	<ul style="list-style-type: none"> <li>• Diseñar la evaluación: definir grupos: de control y experimental, docente, tiempo, modo.</li> <li>• Aplicar la prueba</li> <li>• Identificar posibles problemas</li> <li>• Realizar las modificaciones y ajustes de la versión</li> </ul>

<b>PROCESO DE DOCUMENTACIÓN DIDÁCTICA (ISF)</b>	<ul style="list-style-type: none"> <li>• Planificar la documentación didáctica</li> <li>• Elaborar guía didáctica</li> <li>• Adjuntar la información didáctica pertinente</li> <li>• Producir la documentación y adjuntarla al programa.</li> </ul>
---	---

**Tablas 1.2: PROCESOS NUEVOS QUE SE INCORPORAN Y SUS ACTIVIDADES [4]**

En la **tabla 1.3**, se definen nuevas actividades que cautelasen cuestiones pedagógicas en los procesos existentes, tales como se señalan:

<b>PROCESO DE SELECCIÓN DEL MODELO DE CICLO DE VIDA (FAC)</b>	<ul style="list-style-type: none"> <li>• Seleccionar un modelo de ciclo de vida acorde con la teoría educativa elegida.</li> </ul>
<b>PROCESO DE EXPLORACIÓN DE CONCEPTOS (FAC)</b>	<ul style="list-style-type: none"> <li>• Identificar las necesidades educativas</li> <li>• Formular posibles soluciones potenciales</li> <li>• Formular soluciones potenciales compatibles.</li> </ul>
<b>PROCESO DE ANÁLISIS DE LOS REQUISITOS DE SOFTWARE (FAC)</b>	<ul style="list-style-type: none"> <li>• Definir el tipo de programa a desarrollar</li> <li>• Definir el tipo de interactividad</li> <li>• Integrar requisitos educativos y de software</li> </ul>
<b>PROCESO DE INICIACIÓN, PLANIFICACIÓN Y ESTIMACIÓN DEL PROYECTO (RES)</b>	<ul style="list-style-type: none"> <li>• Establecer la matriz de actividades considerando la teoría educativa elegida</li> </ul>
<b>PROCESO DE ASIGNACIÓN DEL SISTEMA (RES)</b>	<ul style="list-style-type: none"> <li>• Definir las funcionalidades del programa</li> <li>• Desarrollar la arquitectura del programa en base a la teoría educativa elegida.</li> </ul>
<b>PROCESO DE DISEÑO (REP – DPR)</b>	<ul style="list-style-type: none"> <li>• Definir la organización de los menús</li> <li>• Definir tipo de iconos a usar</li> <li>• Seleccionar efectos a usar (sonido, vídeo, etc.)</li> <li>• Seleccionar textos a usar</li> </ul>



	<ul style="list-style-type: none"> <li>• Asegurar facilidad de lectura</li> <li>• Realizar diseño de las pantallas</li> <li>• Realizar diseño de los menú</li> <li>• Realizar storyboards</li> <li>• Definir los criterios de navegación</li> <li>• Definir las actividades (búsqueda, ejercitación, etc. )</li> <li>• Definir tipo de módulos (problemas, evaluación, etc.)</li> <li>• Definir tipos de ayudas didácticas (errores, mensajes)</li> </ul>
<p style="text-align: center;"><b>PROCESO DE DOCUMENTACION TÉCNICA (ISF)</b></p>	<ul style="list-style-type: none"> <li>• Incluir los resultados de las evaluaciones</li> </ul>

**Tabla 1.3: LAS NUEVAS ACTIVIDADES INCORPORADAS A LOS PROCESOS EXISTENTES. [4]**

#### **1.2.4.2 Metodología basada en un enfoque de Calidad Sistémica.**

Se centra en el estudio y extensión de Rational Unified Process (RUP) al ámbito educativo, debido a que es una metodología abierta y adaptable al desarrollo de SE, lo cual garantiza que se lleven a cabo solo aquellas actividades y modelos que sean necesarios o útiles para el proyecto a desarrollar. [5]

El proceso iterativo de RUP se organiza en **fases**, según figura 1.16, cada una concluye con una piedra de milla, es decir que la inclusión de estas es sumamente importante para esta propuesta y constituyen una revisión de los requerimientos establecidos para cada fase, basadas en los controles de calidad. De esta manera si un producto o proceso no pasa el punto de revisión se rediseña o se cancela, evitando con ello costos adicionales de trabajo y de productos de mala calidad.

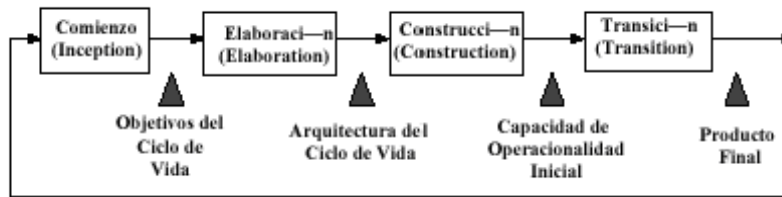


Figura 1.16: FASE DE RATIONAL UNIFIED PROCESS (RUP) [5]

A continuación se presenta los objetivos de cada fase:

- **Fase de comienzo o inicio:** dirigida al entendimiento de los requerimientos y determinar el alcance del esfuerzo de desarrollo. Aquí se define la idea, visión y el alcance del proyecto. Esta fase incluye el análisis y diseño; es decir un análisis de las necesidades educativas y del entorno educativo, se establece el desarrollo del plan creativo de la interfaz, en este se integra el trabajo de diseñadores gráficos, analistas de sistemas y docentes especialistas del área. Esta fase culmina con los objetivos del ciclo de vida. El aporte educativo se muestra en la **tabla 1.4**.

Modelo RUP	Actividades Educativas agregadas a RUP
<b>Fase de comienzo o inicio</b>	<ul style="list-style-type: none"> <li>• Un análisis de las necesidades educativas y del entorno educativo.</li> <li>• Un estudio sobre las teorías de aprendizaje y diseño instruccional que definen el formato del programa.</li> <li>• Una lista de requerimientos pedagógicos relacionados con el contenido y la población estudiantil a la que va dirigida el programa.</li> <li>• Revisión de los objetivos y contenidos del material educativo del programa.</li> <li>• Establecer los límites de las áreas educativas que se van a desarrollar.</li> <li>• Establecer un diseño instruccional para el proyecto multimedia, incluyendo los instrumentos de evaluación del usuario sobre lo aprendido.</li> <li>• Realizar un estudio sobre las pautas de diseño de interfaz adecuadas a la población estudiantil</li> </ul>

	<p>a la que va dirigida el programa.</p> <ul style="list-style-type: none"> <li>• Establecer los criterios de evaluación del software educativo basados en las características de funcionalidad, usabilidad y fiabilidad.</li> </ul>
--	--

**Tabla 1.4: FASE DE COMIENZO O INICIO: APORTE PEDAGÓGICO [5]**

- **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 y el aporte educativo se muestra en la tabla 1.5.

<b>Modelo RUP</b>	<b>Actividades Educativas agregadas a RUP</b>
<b>Fase de Elaboración</b>	<ul style="list-style-type: none"> <li>• Refinar los modelos instruccionales que se utilizan en el proyecto.</li> <li>• Refinar los requerimientos de diseño gráfico y aspectos comunicacionales en base a las pautas pedagógicas establecidas.</li> </ul>

**Tabla 1.5: FASE DE ELABORACIÓN: APORTE PEDAGÓGICO [5]**

- **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, el aporte educativo se muestra en la tabla 1.6.

<b>Modelo RUP</b>	<b>Actividades Educativas agregadas a RUP</b>
<b>Fase de Construcción</b>	<ul style="list-style-type: none"> <li>• Probar el diseño instruccional, comunicacional y gráfico, contra los criterios de <b>evaluación previamente establecidos</b></li> </ul>

**Tabla 1.6: FASE DE COMIENZO O INICIO: APORTE PEDAGÓGICO [5]**

- **Fase de Transición:** Es decir la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto hasta que el cliente este satisfecho. Esta fase culmina con la versión del producto, la cual a su vez concluye el ciclo. El aporte educativo se muestra en la **tabla 1.7**.

Modelo RUP	Actividades Educativas agregadas a RUP
<b>Fase de Transición</b>	<ul style="list-style-type: none"> <li>• Realizar la evaluación del producto por parte del docente y del estudiante objeto del programa educativo.</li> </ul>

Tabla 1.7: FASE DE TRANSICIÓN: APOORTE PEDAGÓGICO [5]

#### 1.2.4.3 Metodología Basada en el Modelo Cascada.

Metodología de diseño y desarrollo de software educativo basada en secuencia de fases ordenadas en forma lineal, permitiendo iteraciones en determinados momentos, en donde el resultado del producto se evalúa y frecuentemente se detecta la conveniencia de introducir cambios”. Partiendo de una idea inicial sobre el “**qué**” (materia y nivel) y el “**cómo**” (estrategia didáctica) ha de ir desarrollándose un proceso complejo en el que generalmente es necesaria la colaboración de *diferentes especialistas* para obtener un producto de calidad. [2]

Las etapas básicas de esta metodología son:

- **Génesis de la idea.** La elaboración de un programa educativo siempre parte de una **idea inicial** que parece potencialmente poderosa para favorecer los procesos de enseñanza / aprendizaje y que va tomando forma poco a poco; una idea que configura unas actividades atractivas para el alumno que potencialmente pueden facilitar la consecución de unos determinados objetivos educativos. Sus autores casi siempre son profesores y pedagogos, diseñadores de software educativo.

La idea inicial de un programa constituye una intuición global de lo que se quiere crear, contiene la semilla del **QUÉ** (materia y nivel) se quiere trabajar y del **CÓMO** (estrategia didáctica), y se irá completando y concretando poco a poco a medida que se elabore el primer diseño del programa: el diseño funcional. Su génesis puede realizarse: por libre iniciativa de los diseñadores o por encargo.

- **Prediseño o diseño funcional.** Elaborado a partir de una **idea inicial** (idea-semilla), el prediseño (diseño funcional) constituye un primer guión del programa que pondrá la énfasis en los aspectos pedagógicos del proyecto: contenidos, objetivos, estrategia didáctica, etc. En caso de que se elabore por encargo o por iniciativa empresarial, este primer guión servirá para presentarlo al jefe del proyecto y a los clientes para que lo sometan a un test de oportunidad y determinen su conformidad o disconformidad con el diseño. En todo caso, el diseño funcional también podrá distribuirse a otros profesores, buenos conocedores de los estudiantes a los que se dirige el material, para que aporten su opinión y sus sugerencias.

Frecuentemente el diseño funcional de los programas lo realiza una única persona, generalmente un profesor, pero resulta recomendable que intervenga un equipo de especialistas, el **equipo de diseñadores pedagógicos**, integrado por: profesor, pedagogo ó psicopedagogo y especialista en tecnología educativa.

- **Estudio de viabilidad y marco del proyecto.** El estudio de viabilidad, determinará si el proyecto es factible de realizarse en este contexto. considerará: aspectos pedagógicos, aspectos funcionales, aspectos técnicos, aspectos económicos y aspectos comerciales. Por lado el marco del proyecto, si el estudio de viabilidad resulta positivo, se concretará el marco de desarrollo del proyecto, el cual considera: presupuesto, personal que va intervenir, plan de trabajo y temporalización, especificaciones técnicas y plataforma de distribución, especificaciones pedagógicas y plataforma de desarrollo.
- **Diseño completo o diseño orgánico.**
- **Programación y elaboración del prototipo alfa-test.** Los analistas informáticos, programadores y especialistas en multimedia desarrollarán el primer prototipo interactivo del material. Se pueden considerar los siguientes trabajos: Fase de análisis, Fase de programación, producción de los elementos audiovisuales, digitalización de los elementos audiovisuales, acción de los elementos.
- **Redacción de la documentación del programa.**
- **Evaluación interna.** La evaluación interna la realizan los integrantes de los equipos de diseño y desarrollo del material. Se realizará siguiendo una determinada metodología que **considerará los criterios de calidad** propios de estos materiales, y terminará con

una reunión de todos los que han participado en el proceso. Considerará: aspectos técnicos, aspectos pedagógicos y aspectos funcionales.

- **Ajustes y elaboración del prototipo beta-test.**
- **Evaluación externa.** La evaluación externa es realizada por personas ajenas al proyecto, poniendo a prueba el producto terminado.
- **Ajustes y elaboración de la versión 1.0,** es el producto final en su primera versión, el cual será implementado a fin de realiza
- **Publicación y mantenimiento del producto.** el cual contempla la puesta en funcionamiento y la constante actualización del mismo

#### **1.2.4.4 METODOLOGÍA ORIENTADO POR OBJETOS**

Metodología que integra el modelo orientado a objetos con la metodología de ingeniería de software educativo (ISE) propuesta por Alvaro Galvis, para enriquecer el proceso de desarrollo de materiales educativos computarizados (MEC) altamente interactivos.[\[6\]](#)

Esta propuesta, refleja el ciclo de vida de un programa educativo, la cual puede tener dos maneras de ejecución, en función de los resultados de la etapa de análisis: sentido manecillas del reloj se procede a diseñar, desarrollar y probar lo que se requiere para atender una necesidad; y en sentido contrario se somete a prueba aquello que se encontró puede satisfacer la necesidad.

Siguiendo el ciclo de vida de un MEC, como se muestra en la figura 1.17, la siguiente descripción permite entender cada una de sus etapas, enriquecidas con el enfoque orientado a objetos.

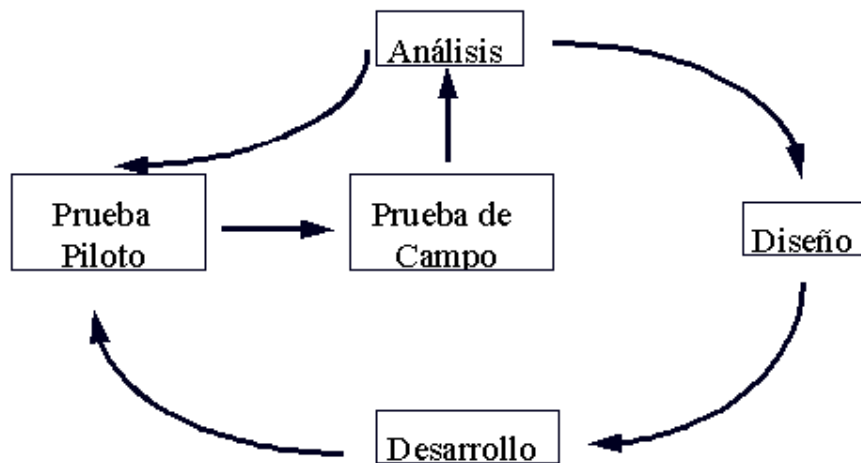


Figura 1.17: Metodología de ISE propuesta por Galvis. [6]

- **Análisis:** Se determina el contexto en el cual se va a crear la aplicación y derivar de allí los requerimientos que deberá atender la solución interactiva, como complemento a otras soluciones basadas en uso de otros medios. En esta etapa se establece como mínimo la siguiente información: Características de la población objetivo, conducta de entrada y campo vital, problema o necesidad a atender, identificado el problema se debe establecer las bases para resolverlo y justificación de uso de los medios interactivos.
- **Especificación de requerimientos:** como parte de la etapa de análisis se debe formular los requerimientos que deberá atender el material interactivo que se desea obtener. Esta deberá contener: descripción de la aplicación, dejar claro las restricciones que tendrá y una descripción de los escenarios, así como las restricciones relacionadas con población objetivo
- **Diseño:** Este se realiza en tres niveles diferentes: educativo, comunicacional y computacional. Se va a realizar el diseño usando el enfoque orientado a objetos, formalizando mucho de los aspectos relacionados con la aplicación, las restricciones existentes y los escenarios de interacción.
  - **Diseño educativo:** Tomando como punto de partida la necesidad o problema se debe establecer lo que hay que enseñar o reforzar para subsanar con apoyo del MEC las necesidades encontradas. Como resultado se debe tener el contenido y su estructura, micromundo (compuesto por argumento, mundo, escenarios, retos, personajes y herramientas), sistema de motivación y sistema de evaluación.
  - **Diseño Comunicacional:** Se define la interfaz de la aplicación, como también los objetos que debe contener cada pantalla y cuales elementos del mundo son

usados/afectados. Es importante conseguir que la interfaz sea amigable, flexible y agradable de usar, también debe ser consistente, es decir, cuidando de los mensajes, la distribución en pantalla, los juegos de colores, etc.

- **Diseño Computacional:** al final de esta etapa se tiene como resultado cada una de las diferentes clases de objetos, incluyendo sus atributos, el conjunto de métodos y las clases generales. Otro aspecto a realizarse en esta etapa es el refinamiento de los diferentes objetos como la interface, adecuándolas a las posibilidades de la herramienta de desarrollo que se va utilizar.
- **Desarrollo:** se implementa la aplicación usando toda la información obtenida anteriormente. Se toma la definición de clases y se implementa en el lenguaje escogido, tomando en cuenta las restricciones computacionales que se tengan. La herramienta de desarrollo se escogerá teniendo en cuenta: costo, disponibilidad en el mercado, portabilidad de la aplicación desarrollada, facilidades al desarrollador.

En el desarrollo se busca que el modelo del mundo sea independiente de la interfaz. Esto facilita el trabajo y permite trabajar en paralelo.

- **Prueba:** esta forma de desarrollo permite ir depurando los componentes del modelo generado, haciendo la validación de los prototipos durante la etapa de diseño y la prueba de uno a uno de los módulos desarrollados. En esta etapa se pone a prueba un módulo beta del micromundo interactivo, la prueba se realiza verificando que efectivamente la aplicación satisface las necesidades y cumplimiento de la funcionalidad requerida.

### 1.2.5 EDUCACION EN NUESTRO PAIS

Es la influencia ordenada y voluntaria ejercida sobre una persona para formarlo y desarrollarlo; de ahí que la acción ejercida por una generación adulta sobre una joven para transmitir y conservar su existencia colectiva. Es un ingrediente fundamental en la vida del hombre y la sociedad y apareció en la faz de la tierra desde que apareció la vida humana. Es la que da vida a la cultura, la que permite que el espíritu del hombre la asimile y la haga florecer, abriéndole múltiples caminos para su perfeccionamiento.



### **1.2.5.1 SISTEMA EDUCATIVO PERUANO**

La educación nacional representa un proceso ordenado e interrelacionado de elementos, sujeto a través de los cuales se desarrolla la acción educativa, de acuerdo con las características, necesidades e intereses de la realidad histórica, económica y cultural.

Por ello, el órgano más importante con relación al Sistema Educativo Nacional está representado por el Ministerio de Educación (MED), en nuestro caso el MED ([www.minedu.gob.pe](http://www.minedu.gob.pe)) la cual tiene como misión la promoción del desarrollo de las personas; por tanto la herramienta principal que permitirá llevar a cabo dicha misión será la existencia de un sistema educativo que verdaderamente brinde una información integral y permanente orientada a desarrollar equitativamente las capacidades individuales de todos.

La base de esta formación está fundada en la cultura de valores que los peruanos debemos poseer. Que respete la identidad individual y colectiva de nuestros ciudadanos y favorezca el desarrollo de las capacidades que nos permitan comprender y actuar sobre el entorno.

De esta manera, la consecuencia necesaria de estas iniciativas será una contribución a la mejora de la calidad de vida de los peruanos y la creación de condiciones propicias para el desarrollo social.

Los instrumentos identificados para ello son la promoción de la investigación científica y la tecnología productiva de la cultura en todas sus formas y ámbitos, así como de la educación física y el deporte. En síntesis, la búsqueda permanente de la excelencia.

### **1.2.5.2 MINISTERIO DE EDUCACIÓN (MED) EN EL PERU**

El Ministerio de Educación, órgano rector del sector, es la "empresa de servicios" más grande del país pues atiende a más de seis millones de estudiantes en el sistema público; controla cuarenta y cuatro mil centros educativos públicos y diecisiete mil programas no escolarizados que dependen directamente del Ministerio de Educación. Esto significa el 85% del servicio educativo que se ofrece en el Perú. [30]

Entre sus funciones principales están la formulación de políticas nacionales sobre educación, a partir de las cuales ejerce sus atribuciones normativas sobre todo el sistema sectorial y garantiza su cumplimiento mediante una adecuada supervisión. Sólo así, será posible contribuir a la mejora de la calidad de vida de los peruanos y a la creación de

condiciones favorables para el progreso, encontrándose la investigación científica, la tecnología, la cultura, la educación física y el deporte vistas bajo esta óptica como instrumentos de desarrollo en la búsqueda permanente de la excelencia.

Desde el punto de vista pedagógico el ministerio de educación cuenta con un **viceministerio de gestión pedagógica**, la cual es la encargada proponer los lineamientos de política pedagógica de todos los niveles y modalidades educativas que están bajo la administración del Ministerio de Educación. Es la que define, articula, monitorea y evalúa la aplicación de las estructuras curriculares básicas y otros elementos de tecnología educativa; diseña los planes de formación y capacitación de personal docente y presta la asesoría pedagógica para la correcta ejecución de la política educativa. Tiene a su cargo la dirección, el seguimiento y evaluación de los factores de calidad de la educación, identifica y promueve investigaciones de carácter pedagógico, propicia el uso de nuevas tecnologías de enseñanza y promueve la participación comunitaria. Presta un servicio de documentación e información pedagógica. También, establece las relaciones intersectoriales y con instituciones de la Sociedad Civil que actúan en favor de la educación. Entre los órganos dependientes del viceministerio de gestión pedagógica tenemos: Educación Básica Regular, Educación Básica Alternativa, Educación Superior y Técnico Profesional, Educación Básica Especial, Educación Comunitaria y Ambiental, Educación Intercultural Bilingüe y Rural, Investigación, Supervisión y Documentación Educativa, Tutoría y Orientación Educativa, Promoción Escolar, Cultura y Deporte; y Proyecto Huascarán. En la **figura 1.18** se muestra la estructura organiza de esta dependencia.

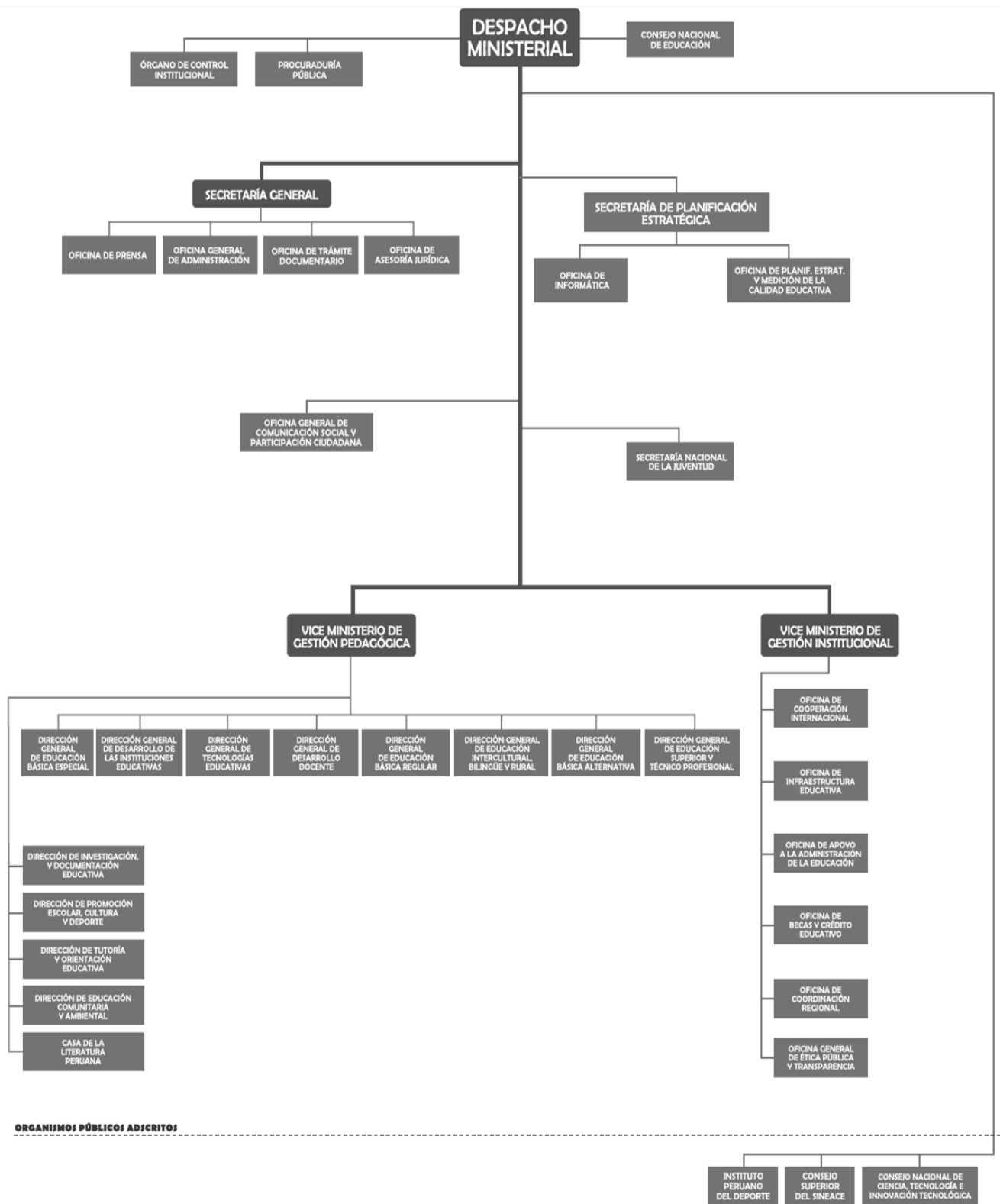


Figura 1.18. ORGANIGRAMA DEL VICEMINISTERIO DE GESTIÓN PEDAGÓGICA DEL PERÚ  
(FUENTE PÁGINA WEB DEL MINISTERIO DE EDUCACIÓN) [10]

Según el organigrama podemos señalar que la **Educación Básica Regular** es responsable de asegurar la formulación y propuesta articulada de la política, objetivos, estrategias pedagógicas, normas y orientaciones de alcance nacional para los niveles de Educación Inicial, Primaria y Secundaria. Entre sus funciones están: [24]

- Formular y proponer la política, objetivos y estrategias pedagógicas en los niveles de Educación Inicial, Primaria, y Secundaria, de manera coordinada con las respectivas Direcciones.
- Formular el diseño curricular nacional de la Educación Básica Regular y las orientaciones referidas al nivel de Educación Inicial, Primaria y Secundaria en coordinación con otras Direcciones Generales.
- Definir los criterios técnicos para la programación, diversificación, implementación y evaluación curricular, así como para el desarrollo de las acciones educativas y el diseño, producción, uso y distribución de materiales educativos.
- Normar la gestión pedagógica en las Instituciones Educativas y programas de Educación Inicial, Primaria y Secundaria en coordinación con las Instancias de Gestión Educativa Descentralizada.
- Promover estrategias para la difusión y consulta nacional de las propuestas pedagógicas y de innovación de la Educación Inicial, Primaria y Secundaria.
- Promover y gestionar proyectos de cooperación internacional dirigidos al mejoramiento de la calidad de la Educación Inicial, Primaria y Secundaria en el marco de las políticas de la Educación Básica Regular y el Proyecto Educativo Nacional.
- Impulsar la elaboración, adaptación y uso de nuevas tecnologías aplicadas a la educación, concordantes con la modernización del currículo, en coordinación con el Proyecto Huascarán.
- Promover la concertación y el consenso de acciones multisectoriales con otros organismos del Estado, gobiernos locales, empresas, organismos no gubernamentales, instituciones y asociaciones, en beneficio de la atención integral de niñas, niños y adolescentes.
- Promover mecanismos de sensibilización, participación y cogestión de padres de familia y de la comunidad en la Educación Básica Regular, a través de los medios de comunicación.
- Asegurar en los tres niveles educativos un enfoque intercultural, bilingüe, inclusivo, ambiental y comunitario en coordinación con las Direcciones Generales correspondientes.

- Proponer una política de textos y material educativo coherente con las necesidades de cada uno de los niveles y en un marco de descentralización.
- Implementar las acciones de mejoramiento educativo a partir de la información del Sistema de Evaluación, Acreditación y Certificación de la Calidad Educativa que correspondan a los niveles de la Educación Básica Regular.
- Regular los procesos de diseño, producción, distribución y uso de los materiales educativos asegurando su coherencia y concordancia con la articulación entre los niveles educativos.

Por otro lado, la **Educación Básica Alternativa** está dirigida a estudiantes que no tuvieron acceso a la Educación Básica Regular. Enfatiza la preparación para el trabajo y el desarrollo de capacidades empresariales. La **Educación Básica Especial** tiene un enfoque inclusivo y atiende a personas que tienen un tipo de discapacidad que dificulte un aprendizaje regular y a niños y adolescentes superdotados o con talentos específicos. La **Educación Superior y Técnico-Profesional** es responsable de formular, proponer y orientar la política pedagógica para la Educación Superior en la formación inicial y en servicio en los ámbitos pedagógico, tecnológico y artístico, así como en la Técnico-Productiva; y coordinar con el Sistema Universitario.

La **Educación Comunitaria y Ambiental** es responsable de promover, reconocer y valorar los aprendizajes que se logran en las organizaciones de la sociedad civil, así como normar y coordinar la educación ambiental para el desarrollo sostenible, la conservación, el aprovechamiento de los ecosistemas y la gestión de riesgos y prevención de desastres. La Dirección de **Educación Intercultural, Bilingüe y Rural**, es responsable de normar y orientar la política nacional de Educación Intercultural, Bilingüe y Rural en las etapas, niveles, modalidades, ciclos y programas del sistema educativo nacional.

La **Dirección de Investigación, Supervisión y Documentación Educativa** es responsable de realizar investigaciones y supervisión educacional para el desarrollo de la calidad del sistema educativo. Otras áreas propias de este viceministerio de gestión pedagógica tenemos: Tutoría y Orientación Educativa, Promoción Escolar, Cultura y Deporte; y Proyecto Huascarán

A continuación se describe las funciones de la dirección de educación secundaria de nuestro país; por ser materia de estudio.

- **Dirección de Educación Secundaria**

La Dirección de Educación Secundaria depende de la Dirección Nacional de Educación Básica Regular. Sus funciones son: [24]

- a. Formular y proponer la política, objetivos y estrategias pedagógicas del Nivel de Educación Secundaria.
- b. Elaborar y validar el diseño curricular nacional en lo que al nivel corresponde y las orientaciones pedagógicas de la Educación Secundaria en coordinación y articulación con las Direcciones de Inicial y Primaria.
- c. Proponer lineamientos de política de textos y materiales educativos para la Educación Secundaria.
- d. Normar, orientar, monitorear y evaluar los procesos de adecuación, diversificación, implementación y evaluación curricular, así como la producción, adquisición, uso y distribución de materiales educativos para el nivel.
- e. Establecer las necesidades de formación docente que deberá tener en cuenta el Sistema de Formación Inicial y Formación en Servicio para docentes, a cargo de los programas escolarizados y a distancia de Educación Secundaria.
- f. Promover estrategias para la difusión y consulta nacional de las propuestas de innovación y mejoramiento de la Educación Secundaria.
- g. Articular la Educación Secundaria con los niveles educativos de la Educación Básica Regular, en concordancia con los requerimientos y necesidades de los educandos y de la sociedad, en el marco de una educación intercultural, bilingüe e inclusiva.
- h. Formular lineamientos de política, acciones y estrategias para la elaboración y uso de las tecnologías de la información aplicadas a la educación, concordantes con la modernización del currículo, en coordinación con el Proyecto Huascarán.
- i. Fortalecer la educación inclusiva de calidad para los púberes y adolescentes con necesidades educativas especiales, asociadas a discapacidad o cualidades excepcionales en coordinación con la Dirección Nacional de Educación Básica Especial.
- j. Promover la participación de los padres de familia, instituciones y organismos de la comunidad en la formación integral de las y los estudiantes.
- k. Elaborar normas y orientaciones pedagógicas administrativas referidas al ingreso, promoción, certificación, diplomado, traslado y convalidación en la Educación Secundaria.

- l. Elaborar los módulos ocupacionales del Área de Educación para el Trabajo, en equivalencia con la Educación Técnico-Productiva y en correspondencia a los requerimientos del sector productivo.
- m. Orientar la aplicación de las políticas de Educación en áreas rurales, a distancia, especial, comunitaria y ambiental en el nivel, coordinando con las Direcciones Regionales de Educación y las Unidades de Gestión Educativa Local.
- n. Normar, implementar, monitorear y evaluar los lineamientos técnicos sobre los procesos pedagógicos de desarrollo curricular, evaluación del aprendizaje y apoyo a las acciones de tutoría y de prevención integral.
- o. Regular una adecuada y coherente articulación de la Educación Secundaria con la Educación Superior.
- p. Promover y gestionar proyectos de cooperación internacional dirigidos al incremento del acceso y al mejoramiento de la calidad de la Educación Secundaria, en coordinación con la Oficina de Cooperación Internacional.
- q. Promover la concertación y el consenso de acciones multisectoriales con otros organismos del Estado, gobiernos regionales y locales, municipios, empresas, organismos no gubernamentales, instituciones y asociaciones en beneficio integral de los adolescentes pres púberes y adolescentes plenos en Educación Secundaria.

### 1.3 MARCO CONCEPTUAL

- **Tecnología:** Aplicación de los conocimientos científicos para facilitar la realización de las actividades humanas. Supone la creación de productos, instrumentos, lenguajes y métodos al servicio de las personas. [9]
- **Información:** Datos que tienen significado para determinados colectivos. La información resulta fundamental para las personas, ya que a partir del proceso cognitivo de la información que obtenemos continuamente con nuestros sentidos vamos tomando las decisiones que dan lugar a todas nuestras acciones. [9]
- **Diseño Instruccional:** Se define Diseño Instruccional (Instructional Design) como el proceso que genera especificaciones instruccionales por medio del uso de teorías instruccionales y teorías de aprendizaje para asegurar que se alcanzarán los objetivos planteados. [15]  
En el diseño instruccional se hace un completo análisis de las necesidades y metas educativas a cumplir y posteriormente se diseña e implementa un mecanismo que permita alcanzar esos objetivos. Así, este proceso involucra el desarrollo de materiales y actividades instruccionales, y luego las pruebas y evaluaciones de las actividades del alumno. El Diseño Instruccional hace énfasis en crear planes para desarrollar materiales instruccionales que aumenten el aprendizaje de cada individuo.
- **Aprender a aprender:** implica enseñar a aprender (enseñar a pensar) desarrollando capacidad/destrezas y valores/actitudes. En la escuela clásica implica “trucos para aprender contenidos” (técnicas de estudio). En la escuela activa, aprender a aprender implica el uso adecuado de estrategias de aprendizaje para desarrollar procesos cognitivos y afectivos. [10]
- **Capacidad:** es una habilidad general, que utiliza o puede utilizar un aprendiz para aprender, cuyo componente fundamental es cognitivo. La inteligencia del aprendiz, como macrocapacidad, está constituida por una 30 o 40 capacidades, que pueden clasificarse en cognitivas (intelectivas), comunicación, psicomotoras y de intersección social. Estas capacidades constituye los objetivos fundamentales. [10]
- **Destrezas:** es una habilidad específica, que utiliza o puede utilizar un aprendiz para aprender cuyo componente fundamental es cognitivo. Un conjunto de destrezas constituye una capacidad. A nivel práctico cada capacidad, para ser trabajada



didácticamente, se ha de descomponer en destrezas. Los objetivos complementarios están constituidos por destrezas o capacidades pequeñas. [10]

- **Curriculum:** desde un aspecto pedagógico el curriculum es una selección cultural. Y como tal indica, la cultura social convertida en cultura escolar por medio de las instituciones escolares y los profesores. Y desde esta perspectiva la cultura escolar posee los mismos elementos que la cultura y por ello afirmamos que curriculum indica las capacidades, los valores, los contenidos y los métodos que los adultos queremos que aprendan en la Escuela. Estos son los elementos básicos que han de estar presentes en todos los proyectos y diseños curriculares. [10]
- **Contenidos curriculares:** los avances psicopedagógicos ha puesto de manifiesto que existen tres clases de conocimiento, que es necesario enseñar: conocimiento del mundo real (que), conocimiento estratégico (como) y conocimiento condicional (por qué). Esto origina tres tipos de contenidos que todo profesor debe enseñar en cualquier materia: Conceptos, procedimientos y valores o actitudes. Esto representa hoy, los contenidos curriculares. [11]

## CAPÍTULO II

### 2 EL PROBLEMA DE INVESTIGACIÓN

#### 2.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA

La educación es sin lugar la responsable de la generación de conocimientos y destrezas intelectuales en las personas capaces de producir un crecimiento económico y protagonizar el cambio hacia la modernidad y eficiencia en el bienestar y calidad de nuestro país.

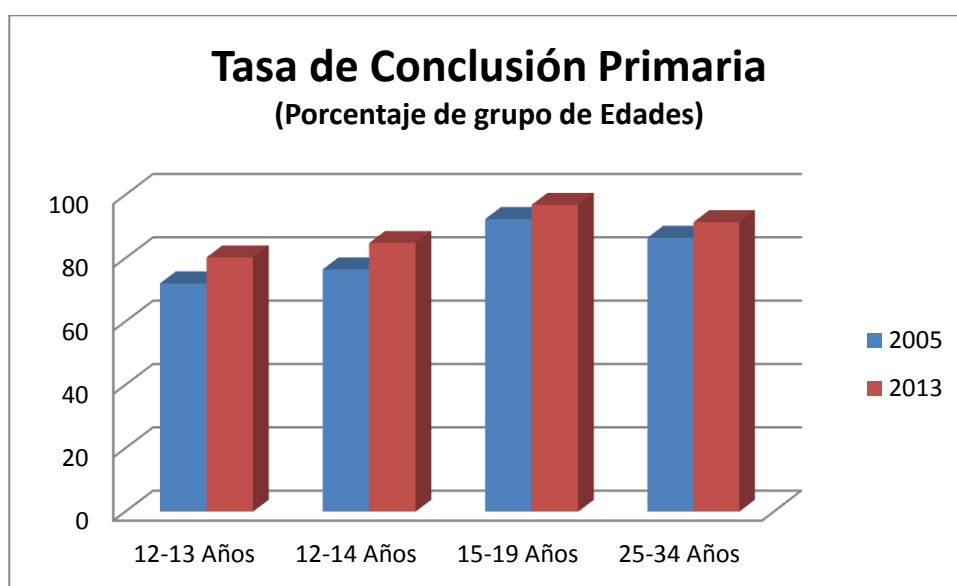
Nuestra educación requiere de reformas institucionales y esto no solo recae en la política educativa, sino en todas las entidades y profesionales que desean hacer cambios significativos en la educación de nuestro país. Entonces podemos preguntarnos sobre el estado actual de la educación en nuestro país; los avances en este campo habrían mostrado ligeras mejoras en algunos indicadores, tales como la disminución en las tasas de deserción y repetición en los niveles de la educación escolar; pero quizás el avance más importante este en las diversas iniciativas orientadas a promover **estándares educativos**, los cuales tienen como objetivo establecer metas de aprendizaje que guíen y articulen el proceso de enseñanza en estos niveles educativos.

Por otro lado, se ha presentado durante los últimos años un crecimiento económico sostenible y redujo levemente la pobreza extrema; es así, que se ha institucionalizado el Acuerdo Nacional como mecanismo para construir convenios importantes, tal es el caso, del Proyecto Educativo Nacional de largo plazo. Así mismo, existen diversas políticas para el cambio educativo, entre las que destacan el Plan Nacional de Educación para todos, el impulso de la carrera pública magisterial. Si bien se han producido avances en estos aspectos, la situación de la educación en nuestro país sigue siendo baja.

Finalmente, partiendo desde la perspectiva profesional podemos señalar que un aporte significativo estará orientado en aplicar herramientas tecnológicas, es decir, programas educativos donde los estudiantes mejoren el proceso de enseñanza - aprendizaje; las cuales deberán tener modelos de desarrollo de software adecuados a nuestra realidad. Las estadísticas demuestran que nuestro país está sufriendo cambios, pero estos deberán estar apoyados con nuevas estrategias que ayuden a mejorar la calidad.

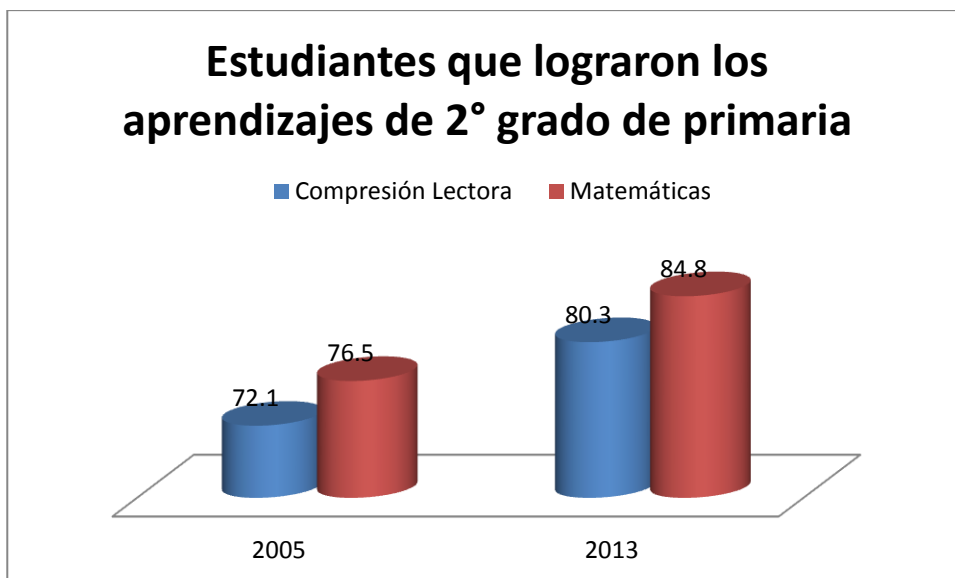
**Estadística del Ejercicio Educativo en el Perú.** En este punto daremos a conocer estadísticas que nos ayudaran a percibir que nuestra educación si bien ha vivenciado mejoras en los últimos años, todavía está faltando propuestas pedagógicas que ayuden a mejorar. [25]

La información que a continuación se presenta representa la tasa de conclusión del nivel primaria para grupos de edades entre 12 a 34 años; en ese sentido se evidencia estudiantes que ha terminado su etapa primaria en edades tardías; tal como se aprecia en la figura 2.1



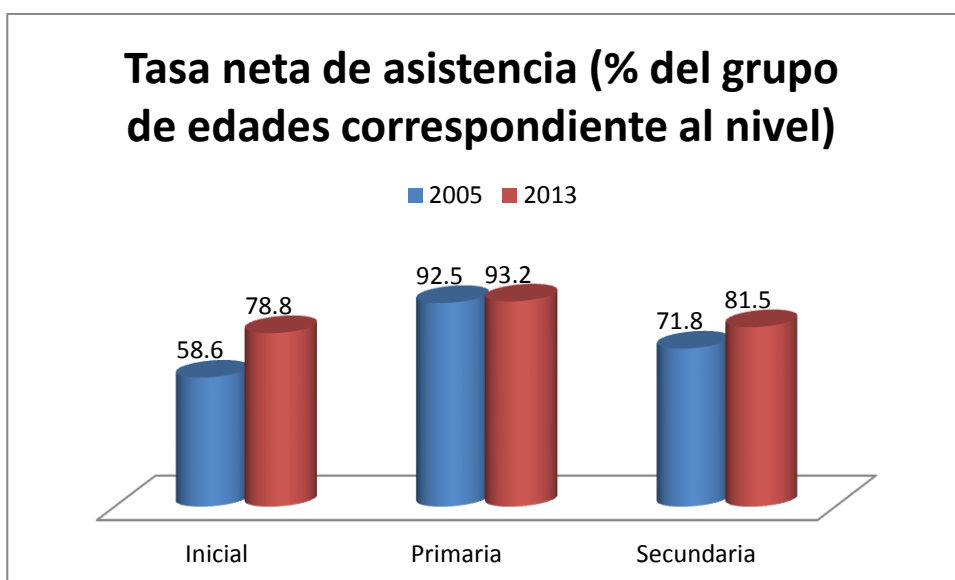
**Figura 2.1: TASA DE CONCLUSIÓN DE LA ETAPA PRIMARIA POR GRUPO DE EDADES.** [25]

Por otro lado, en la siguiente figura 2.2, se aprecia a los estudiantes que lograron los aprendizajes del 2° grado de primaria para los años 2008 y 2013, en ese sentido se evidencia un porcentaje de crecimiento para las áreas de comprensión lectora y matemáticas, pero no es la mejor toda vez que no se llega al 50%.



**Figura 2.2: ESTUDIANTES QUE LOGRARON LOS APRENDIZAJES DE 2° GRADO DE PRIMARIA.** [25]

Otro aspecto a evaluar es la tasa de asistencia de los estudiantes en los tres niveles de la educación básica regular, en ese sentido se presenta la figura 2.3 donde se muestra la tasa de asistencia neta según el porcentaje del grupo de edades correspondientes al nivel para los años 2005 y 2013. En ese sentido queremos señalar que para brindar una buena educación estos valores deben ser los más altos para con ello garantizar un buen proceso de aprendizaje.



**Figura 2.3: TASA NETA DE ASISTENCIA (% DEL GRUPO DE EDADES CORRESPONDIENTE AL NIVEL)** [25]

## **2.2 DELIMITACIÓN DE LA INVESTIGACIÓN**

El modelo a construir estará enmarcado en el contexto de la educación básica regular (EBR) del nivel secundario del Perú y en especial de la educación secundaria de la Institución Educativa “San Francisco de Borja”, en ese sentido los tipos de enseñanza de educación inicial y primaria no formarán parte en la elaboración de esta propuesta. Por otro lado, no se contemplará la gestión pedagógica de otras áreas propuestas por el Ministerio de Educación, es decir, la educación superior, técnica profesional, la básica especial, educación comunitaria y ambiental entre otras.

Para efecto de validar el modelo para el desarrollo de software educativo, se aplicará a estudiantes de 4° y 5° grado de educación secundaria.

El caso de estudio estará implementado en la Institución educativa “San Francisco de Borja” de educación inicial, primaria y secundaria con más de 45 años en la enseñanza de estudiantes del distrito de San Borja.

## **2.3 PLANTEAMIENTO DEL PROBLEMA**

¿De qué manera la construcción de un modelo para el desarrollo de software educativo influye en el proceso de enseñanza aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”?

## **2.4 LOS OBJETIVOS DE LA INVESTIGACIÓN**

### **2.4.1 Objetivo General**

Construir un modelo para el desarrollo de software educativo en el proceso de enseñanza aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”.

### **2.4.2 Objetivo Especifico**

- Estudiar y evaluar los modelos de desarrollo de software educativo.
- Proponer un modelo para el desarrollo de software educativo en la etapa escolar.
- Estudiar y evaluar métodos de enseñanza - aprendizaje actual.

## **2.5 LA HIPÓTESIS DE LA INVESTIGACIÓN**

La construcción de un modelo para el desarrollo de software educativo influye positivamente en el proceso de enseñanza - aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”.

## **2.6 LAS VARIABLES Y LOS INDICADORES**

### **2.6.1 Identificación de Variables**

- **Variable Independiente:** La construcción de un modelo para el desarrollo de un software educativo.
- **Variable Dependiente:** Proceso de enseñanza aprendizaje en la etapa escolar.

## 2.6.2 Definición Operacional de las Variables

Variables	Indicadores
<b>Variable Independiente:</b>  La construcción de un modelo para el desarrollo de un software educativo.	<ul style="list-style-type: none"><li>• Software educativo es fácil de utilizar.</li><li>• Interfaces gráficas amigable.</li><li>• Software contiene las destrezas del área.</li><li>• Software explicita las capacidades del área.</li></ul>
<b>Variable Dependiente:</b>  Proceso de enseñanza aprendizaje en la etapa escolar.	<ul style="list-style-type: none"><li>• Desempeño en las calificaciones de las áreas académicas</li><li>• Reforzar las competencias del área.</li></ul>

Tabla 2.1: DEFINICIÓN OPERACIONAL DE LA VARIABLES.

## 2.7 JUSTIFICACIÓN E IMPORTANCIA DEL ESTUDIO

En la actualidad no existe un estándar de software educativo, los actuales métodos presentan dificultades en la elaboración de programas educativos, estos no presentan un contexto pedagógico que ayude a los estudiantes de la etapa escolar a construir su conocimiento en base a sus capacidades. Por lo que se hace necesario construir un modelo para el desarrollo de software educativo en el proceso de enseñanza - aprendizaje en esta etapa.

Por otro lado, se podrá construir software educativo que represente un contexto de nuestra realidad en forma eficiente, es decir la industria de software peruana podrá contribuir a mejorar el proceso de enseñanza - aprendizaje que nuestro país requiere.

## CAPÍTULO III

### METODOLOGÍA

#### 3.1 POBLACIÓN Y MUESTRA

##### 3.1.1 POBLACIÓN

Estuvo conformado por 160 estudiantes de ambos sexos, los cuales están matriculados en los grados de 4to. y 5to, de educación secundaria, pertenecientes a la Institución Educativa “San Francisco de Borja” ubicado en el distrito de San Borja, correspondiente a la Unidad de gestión Educativa Nro. 7.

##### 3.1.2 MUESTRA

La muestra se seleccionó con la técnica de muestreo aleatorio simple. En ese sentido, se definió el tamaño de la muestra, aplicando para ello la siguiente formula correspondiente a poblaciones finitas:

$$n = \frac{N x Z^2 (p x q)}{(N - 1) x E^2 + Z^2 (p x q)}$$

**Dónde:**

n = Tamaño de la muestra

N = Población (160 estudiantes)

Z = Nivel de confianza (1.96)

p = Tasa de prevalencia de objeto de estudio (0.50)

q = (1-p) es decir 0.50

E = Error de precisión 0.05

Realizando el reemplazo:

$$n = \frac{160 x 1.96^2 (0.5 x 0.5)}{(160 - 1) x 0.05^2 + 1.96^2 (0.5 x 0.5)}$$

$$\mathbf{n = 113}$$



Según el resultado de la fórmula, el tamaño de la muestra se estimó en 113 estudiantes. No obstante al contar con una mayor accesibilidad se determinó 115 estudiantes la muestra final de investigación, lo cual equivale a 72% de la población total.

### 3.2 MÉTODO Y DISEÑO DE INVESTIGACIÓN

#### 3.2.1 Método de Investigación

La investigación aplico básicamente el método explicativo porque establece una relación causal entre las variables. Nivel aplicativo porque permitió recoger los datos y para la comprobación de la hipótesis se utilizaron instrumentos y técnicas.

#### 3.2.2 Diseño de Investigación

El diseño corresponde a la investigación no experimental, es decir no se manipulan las variables.

$$M - O_x r_y$$

**Dónde:**

M	=	Muestra
O	=	Observación
x	=	Construcción de un modelo para el desarrollo de Software Educativo.
y	=	Proceso de enseñanza aprendizaje en la etapa escolar.
r	=	Relación entre variables.

### **3.3 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS**

La técnica de recolección de datos aplicada fue la directa; es decir la información se obtuvo a través de encuestas en muestras representativas de la población citada, al mismo tiempo se aplicaron técnicas de observación directa.

Así mismo, se señala que la técnica de muestreo es el aleatorio simple, para la determinación del tamaño de la muestra.

Por otro lado, el instrumento que se utilizó es la encuesta que se realizó a la Institución Educativa San Francisco de Borja del distrito de San Borja, para los grados de 4to. y 5to. de Educación Secundaria; el cual está constituido por 7 ítem de tipo cerrado y con sus respectivas respuestas.

### **3.4 TÉCNICAS Y PROCEDIMIENTOS DE ANÁLISIS DE DATOS**

Para el procesamiento de datos se siguió el siguiente procedimiento:

- Cálculo de las frecuencias.
- Cálculo de los puntajes obtenidos.
- Gráficos respectivos.

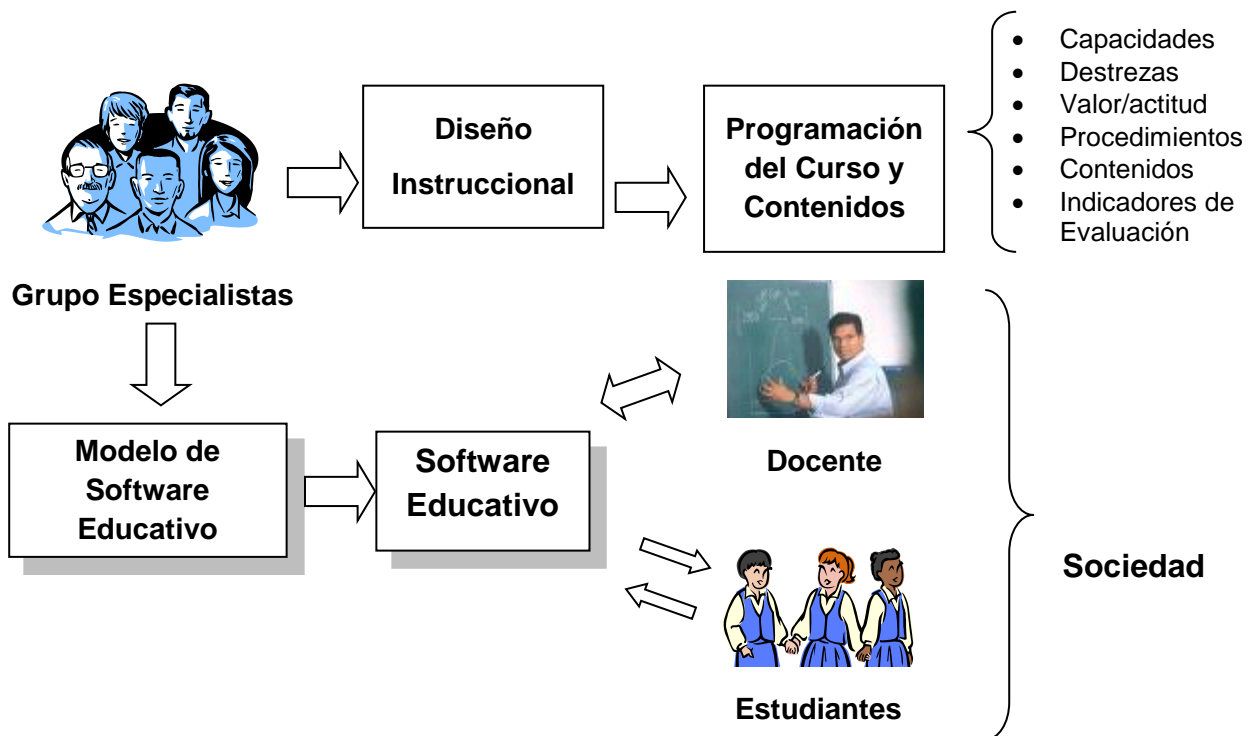
La prueba de hipótesis se realizó aplicando chi cuadrado que consiste en determinar la existencia de relación o no entre las variables de investigación.

## CAPÍTULO IV

### PRESENTACION Y ANALISIS DE LOS RESULTADOS

#### 4.1 PROPUESTA DEL MODELO DE INVESTIGACIÓN

El modelo presentado busca esquematizar en forma general los participantes así como los procesos a desarrollar para la construcción de un software educativo en la etapa escolar, en la figura 4.1, se presenta la propuesta del modelo Enseñanza - Aprendizaje.



**Figura 4.1: Modelo de Enseñanza - Aprendizaje usando Software Educativo**

Para poder entender este modelo podemos indicar que el **grupo de especialistas** estará conformado por programadores, diseñadores, docentes especializados entre otros; en ese sentido el trabajo estará coordinado durante toda la etapa de la construcción del software; el **diseño Instruccional** representa una serie de actividades propias del proceso pedagógico socio cognitivo, en donde se deberán determinar claramente las capacidades y destrezas en las diferentes áreas de la educación escolar. En ese sentido el resultado permitirá determinar la **programación del curso y los contenidos** a desarrollarse los

cuales estarán relacionados con el proceso pedagógico elegido; y en él se podrá identificar las capacidades elegidas según el área a desarrollar y destrezas que permitan fortalecer dichas capacidades. Otro aspecto importante de esta programación lo representan los valores y actitudes como objetivos afectivos propios de la institución, los procedimientos a seguir propios de las destrezas, los contenidos temáticos y los instrumentos de evaluación. Los **docentes y estudiantes** serán los usuarios de este software y ellos serán los únicos que podrán determinar que este modelo propuesto capture sus expectativas; por otro lado, ellos como participante de este proceso de enseñanza - aprendizaje no pueden estar aislados del contexto social, el cual influye directamente en el escenario de la globalización.

Finalmente, el **modelo de software educativo** obedece a una propuesta del ciclo de vida modificada que involucre en su desarrollo actividades pedagógicas y en donde el diseño instruccional ayude a su desarrollo para su implementación; en ese sentido la participación de los especialistas del área recojan los alcances anteriormente señalados y permita obtener un software educativo en la etapa escolar.

### **2.8.1 Propuesta del Diseño Instruccional**

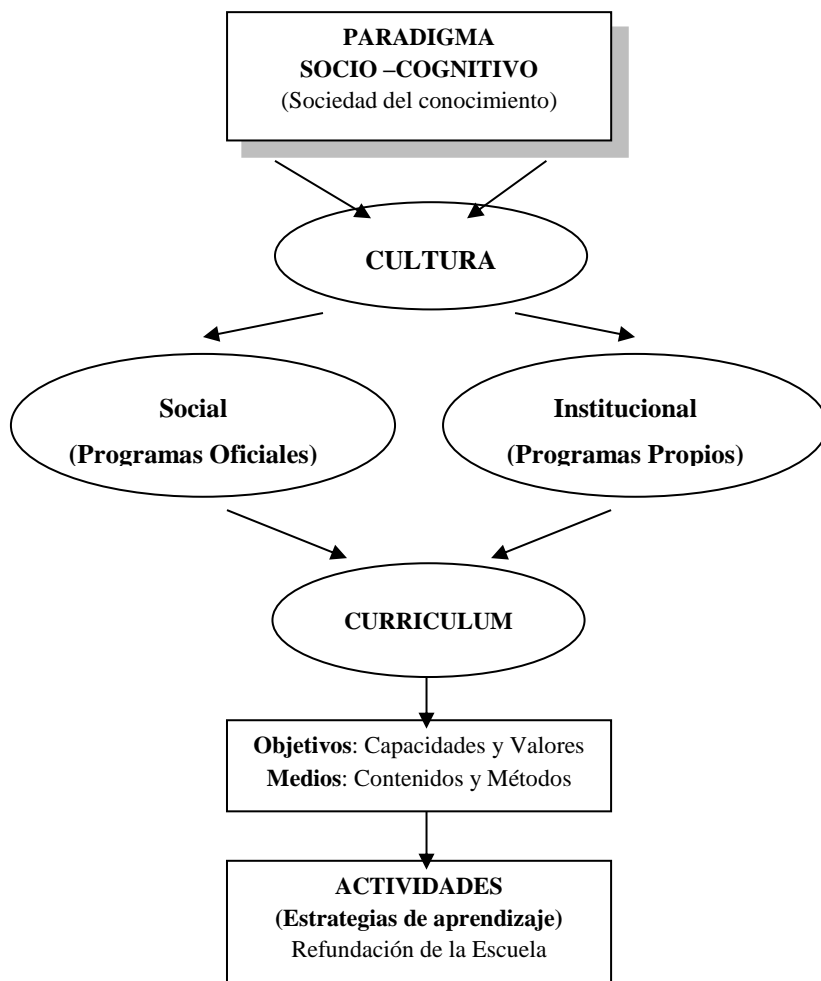
Esta actividad propia de la propuesta, consiste en determinar una serie de tareas que permita obtener una programación adecuada y el desarrollo de contenidos temáticos, tomando durante todo este proceso el aspecto pedagógico que ayude al alumno a fortalecer sus capacidades de aprendizaje. Para ello el diseño instruccional estará representado por el modelo ADDIE el cual ha sido adaptado al proceso pedagógico socio cognitivo, el cual se fundamenta en la refundación de la escuela.

Esta escuela nueva debe ser una organización que aprende y no que enseña (donde el aprendizaje ha de prevalecer sobre la enseñanza; en ese sentido se busca que los estudiantes generen sus propios conocimientos, convirtiéndose el docente en un facilitador del conocimiento. Esto es una transición del paradigma conductista al socio – cognitivo, que integre el escenario (globalización) y el conocimiento (herramientas para aprender y seguir aprendiendo).

Román Martiniano (2004), señala que el estudiante debe aprender a aprender, esto implica enseñar a aprender (enseñar a pensar) desarrollando en ellos capacidades o destrezas y valores o actitudes. En la escuela clásica, aprender a aprender implica “trucos para aprender contenidos” (técnicas de estudio). En la escuela de hoy, es decir la Activa,

aprender a aprender implica el uso adecuado de estrategias de aprendizaje para desarrollar procesos cognitivos y afectivos. [10]

En la figura 4.2, nos ayuda a comprender el cambio de un contexto socio cognitivo a la transición de unidades de aprendizaje propias de un currículo por capacidades y actitudes, buscando desarrollar con ello nuevas herramientas estratégicas (software educativo) para un aprendizaje significativo.



**Figura 4.2: TRANSICIÓN DESDE EL PARADIGMA SOCIO-COGNITIVO A LAS ACTIVIDADES DE APRENDIZAJE**

El diseño instruccional ADDIE, ha sido adaptado según las necesidades de la propuesta pedagógica socio cognitivo. La figura 4.3, muestra las etapas del ciclo de desarrollo instruccional, así como el resultado que ellas producen propio de cada actividad.

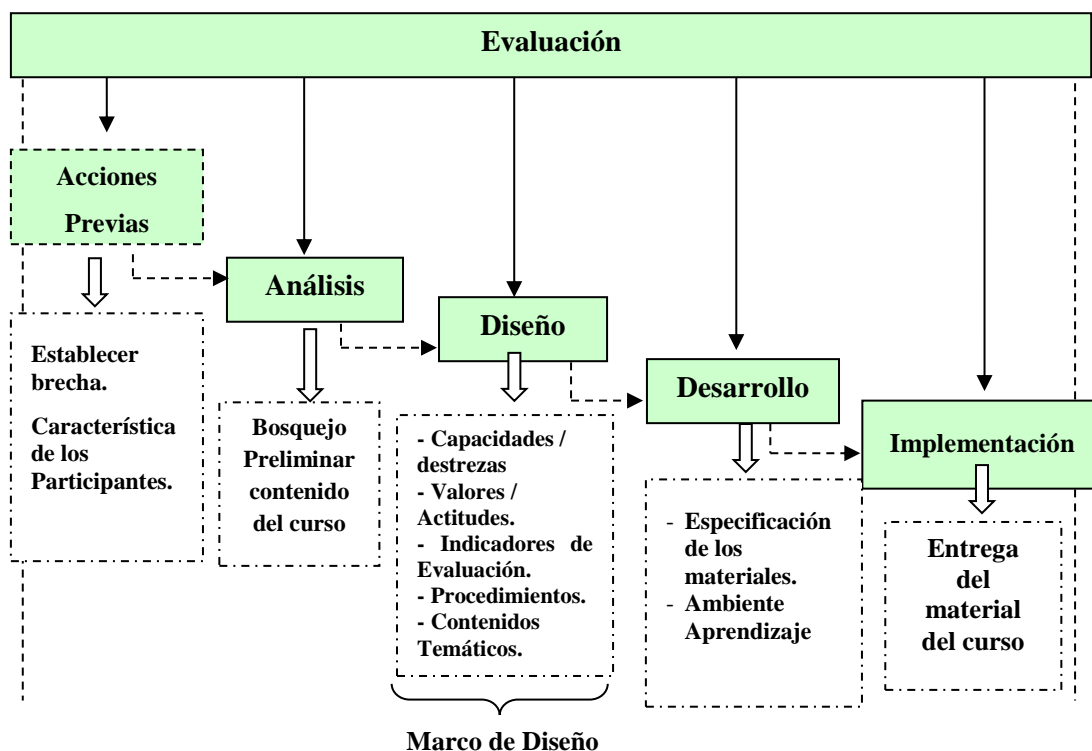


Figura 4.3: DISEÑO INSTRUCCIONAL ADDIE ADAPTADO A LA PEDAGOGÍA SOCIO COGNITIVA

- **Acciones Previas**, esta actividad toma en cuenta ciertas condiciones pre-existentes, tales como el *plan estratégico de la institución* (misión y objetivos institucionales) y *limitaciones e intereses de los participantes*. El resultado de todo ello es el **establecimiento de la brecha** existente entre los estudiantes y el nivel actual del conocimiento; así como las **características de los participantes**.
- **Análisis**, se debe tomar información detallada del curso, esta información corresponde a una descripción del entorno (contexto externo al sistema), *valoración organizativa* (eventos y tendencias dentro del sistema), *dominio de la materia* (bosquejo preliminar del contenido), *necesidades de aprendizaje* y *características de los participantes* (actitudes e intereses). El resultado de todo ello es el **bosquejo del contenido** del curso.
- **Diseño**, se inicia tomando como bosquejo el contenido del curso. En esta fase el resultado estará orientado en determinar los aspectos propios de la pedagogía propuesta, identificando claramente las capacidades y destrezas, valores y actitudes, preparación de los indicadores de evaluación, establecer los procedimientos y métodos para desarrollar un curso adecuado y finalmente determinar los contenidos temáticos

en relación al curso seleccionado. Es en esta fase donde el modelo instruccional ADDIE se fortalece con el cambio propuesto por esta tesis y agregamos en esta etapa las capacidades, destrezas, valores, actitudes, procedimientos y contenidos, como a continuación de señala:

- **Determinar las capacidades** propias del tema a desarrollar, en donde las capacidades definidas estarán relacionadas con el título del Software. En la **tabla 4.1**, se representa el panel de capacidades generales por aéreas propuesto por Martiniano Román Pérez, cabe señalar que otras aéreas no consideradas en esta figura, deberán tomar las capacidades afines como punto de partida. Cabe señalar que el docente especialista tendrá la labor de identificar dichas capacidades, que luego permitirá en la siguiente etapa identificar las destrezas adecuadas.

<p style="text-align: center;"><b><u>Cognitivas</u></b></p> <ul style="list-style-type: none"> <li>• Comprender</li> <li>• Imaginar</li> <li>• Sintetizar</li> <li>• Utilizar</li> <li>• Pensamiento creador</li> <li>• Relacionar</li> <li>• Planificar el conocimiento</li> <li>• Razonamiento lógico</li> <li>• Razonamiento inductivo</li> <li>• Interiorizar conceptos</li> <li>• Pensamiento y sentido crítico.</li> <li>• Observación Sistemática</li> <li>• Clasificar</li> <li>• Globalizar</li> <li>• Percibir</li> <li>• Memorizar</li> <li>• Simbolizar</li> </ul>	<p style="text-align: center;"><b><u>Psicomotoras</u></b></p> <ul style="list-style-type: none"> <li>• Orientación Espacial</li> <li>• Orientación temporal</li> <li>• Creatividad</li> <li>• Aplicar</li> <li>• Expresión corporal</li> <li>• Coordinación Psicomotriz</li> <li>• Utilizar</li> <li>• Construir</li> <li>• Automatizar</li> <li>• Manipular</li> <li>• Explorar</li> </ul>	<p style="text-align: center;"><b><u>Comunicación</u></b></p> <ul style="list-style-type: none"> <li>• Expresión Oral</li> <li>• Expresión escrita.</li> <li>• Expresión gráfica.</li> <li>• Expresión artística</li> <li>• Expresión corporal</li> <li>• Expresión musical</li> <li>• Expresión plástica</li> <li>• Expresión dinámica</li> <li>• Expresión icónica</li> <li>• Expresión informática</li> <li>• Dialogar</li> </ul>
<p style="text-align: center;"><b><u>Inserción Social</u></b></p> <ul style="list-style-type: none"> <li>• Integración en el medio</li> <li>• Relacionarse</li> <li>• Colaborar</li> <li>• Convivir</li> <li>• Participar</li> <li>• Aceptar críticamente</li> <li>• Compartir</li> <li>• Asumir responsabilidades</li> <li>• Sentido de equipo</li> <li>• Compresión de la realidad</li> </ul>		

**Tabla 4.1: PANEL DE CAPACIDADES GENERALES DE MARTINIANO ROMÁN PÉREZ. [10]**



- **Determinar las Destrezas** relacionadas a las capacidades identificadas, aquí las habilidades específicas representan las destrezas; y el docente representa un rol importante para el proceso aprendizaje. La propuesta se orienta a que el software represente una herramienta que permita fomentar las destrezas del alumno.

En la siguiente Tabla 4.2, se muestra un ejemplo del panel de destrezas según algunas capacidades identificadas.

<b>Comunicación</b>	<b>Inserción Social</b>	<b>Cognitivas</b>	<b>Psicomotoras</b>
<b><u>Expresión Oral:</u></b>	<b><u>Relacionarse:</u></b>	<b><u>Observación Sistemática</u></b>	<b><u>Expresión Corporal</u></b>
Observar	Escuchar	Observar	Conocimiento del Cuerpo
Ordenar	Comunicar	Experimentar	Lateralidad
Dicción	Respetar	Clasificar	Relación con otros
Fluidez	Dialogar	Deducir	
Vocabulario	Exponer	Describir	
Opinar	Aceptar	Interpretar	
Comprender	Informar	Esquematizar	<b><u>Orientación</u></b>
Nombrar	Colaborar	Recogida de Datos	<b><u>Espacio - Tiempo:</u></b>
Leer	Comunicarse	Comparar	Situarse
Dialogar	Opinar		Explorar
Articular	Disfrutar		Representar mentalmente
Pronunciar	Intervenir		Relacionar
Inventar		<b><u>Relacionar Conocimiento:</u></b>	Medir
Memorizar	<b><u>Participar:</u></b>	Analizar	Colocar
Comentar	Oír- Escuchar	Comparar	Localizar
	Aceptar	Identificar	Clasificar
<b><u>Dialogar:</u></b>	Colaborar	Conocer	Representar
Escuchar	Disfrutar	Distinguir	Desplazar
Atender			Cambiar
Ceder		<b><u>Razonamiento Lógico:</u></b>	Orientarse
Aceptar	<b><u>Compartir:</u></b>	Numerar	
Comprender	Prestar	Seriar	
Juicio Crítico	Ayudar	Calcular	
Respetar	Ceder	Operar	
Opiniones	Colaborar	Diferenciar	
Comunicación			

<p><b><u>Expresión</u></b></p> <p><b><u>Escrita:</u></b></p> <p>Escribir</p> <p>Repres.</p> <p>Gráficamente</p> <p>Relacionar imag.- palab</p> <p>Puntuación</p> <p>Ortografía</p> <p>Vocabulario</p> <p>Narrar</p> <p>Describir</p> <p>Elaborar frases</p> <p>Redactar</p> <p>Crear</p> <p>Comentar</p> <p>Analizar</p>	<p>Representar</p> <p>Interpretar</p> <p>Resolver</p> <p>Comparar</p> <p>Ordenar</p> <p>Razonar</p> <p>Elaborar</p> <p>Medir</p> <p>Inducir</p> <p>Analizar</p>	<p><b><u>Manipular:</u></b></p> <p>Tocar</p> <p>Manejar</p> <p>Modelar</p> <p>Puzzles</p> <p>Distinguir</p> <p>Texturas</p>
--	---	---

**Tabla 4.2. PANEL DE DESTREZAS SEGÚN CAPACIDADES IDENTIFICADAS.**  
**Por MARTINIANO ROMÁN PÉREZ. [10]**

- **Determinar los Valores y Actitudes**, aquí debemos representar los objetivos afectivos generales, esto debe obedecer a un plan estratégico institucional previstos en la etapa de acciones previas. En ese sentido aquellos valores y actitudes que identifiquen a la institución deberán ser comunes a todas las áreas del saber. El método propuesto identifica que valores y actitudes serán los más adecuados para que el alumno a través del software educativo sean fortalecidos como eje transversal.

En la **Tabla 4.3**, se muestra un panel de valores y actitudes donde se identifican las destrezas propias de las capacidades señaladas.

<b>Orden y Limpieza</b>	<b>Respeto</b>	<b>Amistad</b>	<b>Belleza</b>
Recoger Asearse Ordenar Cooperar Cuidar Conservar Organizar Guardar Presentar	Aceptar Colaborar Compartir Ceder Escuchar Valorar Cuidar Tolerar Dialogar	Querer Perdonar Jugar Escuchar Compartir Respetar Entregar Regalar	Sensibilizar Admirar Observar Ordenar
<b>Solidaridad</b>	<b>Tolerancia</b>	<b>Responsabilidad</b>	<b>Autoestima</b>
Comprometerse Ayudar Compartir Cumplir Cuidar Cooperar Perdonar Aceptar Respetar Colaborar Implicarse	Respetar Apertura Aceptar Valorar Escuchar Comunicar	Trabajo Constancia Coherencia Honradez Participar Respetar Cuidar Ayudar Compartir Organizar Interesarse Disciplina Limpieza Conservar Puntualidad Esfuerzo Colaborar	Valorarse Aceptarse Juicio Crítico Superarse

**Tabla 4.3: PANEL DE VALORES Y ACTITUDES.**

**Por MARTINIANO ROMÁN PÉREZ. [10]**

- **Determinar los Indicadores de Evaluación**, el diseño instruccional ADDIE propone un plan de evaluación para saber que tanto ha aprendido el estudiante, bajo este contexto existen *diferentes formas de evaluar*, se tomará el propuesto por Kirkpatrick (1995), la cual propone dos formas: la evaluación de reacción y evaluación de aprendizaje.

La **evaluación de reacción** está relacionada con una **evaluación formativa** orientada a la formación integral del alumno; permitiendo evaluar capacidades, destrezas y valores; en esta forma de evaluación se implementará un formulario de ingreso, en donde el docente podrá registrar a través del software educativo sus observaciones.

La **evaluación de aprendizaje** está relacionada con una evaluación **sumativa** es decir cuantificable y se orienta a la evaluación de contenidos en función de las capacidades según el área o asignatura seleccionada; para ello se desarrollaran formularios de evaluación en donde el alumno podrá interactuar con el software educativo para una medición cuantificable de su aprendizaje. Posteriormente será traducido a un valor cualitativo propio del nivel primario.

- **Determinar los contenidos**, para su desarrollo el equipo deberá presentarlo según el área, y estos podrán ser de dos tipos: los **saberes factuales**, es decir hechos, ejemplos y experiencias; y los **saberes conceptuales**, representados por principios, teorías, leyes, conceptos e hipótesis. Desde el punto de vista del software educativo se deberá construir una base de datos que permita el almacenamiento de los contenidos en base al área, capítulo y tema a desarrollar, con ello se tendrá una base de datos organizada para su explotación.
- **Determinar los procedimientos**, representa las actividades generales que el alumno desarrollará (forma de hacer) para alcanzar sus destrezas y por consiguiente las capacidades del área. En ese sentido el docente cumple el papel de facilitador en el proceso de aprendizaje. Esta tarea es importante ya que se consolida la actividad que realizará el software educativo en relación a la destreza y contenido conceptual según el tema propuesto.

En la **Tabla 4.4**, se muestra la propuesta entre las capacidades, destrezas, contenidos y procedimientos según el tema seleccionado para el software educativo a construir.

Capacidades / Destrezas	Contenidos Conceptuales	Procedimientos
Capacidad 1 1. Destreza 2. Destreza	Contenido 1 Contenido 2 ..... .....	1.a Redacción de la actividad a realizar según la destreza y el contenido. .....
Capacidad 2 3. Destreza	Contenido N	3.a Redacción de la actividad a realizar según la destreza y el contenido.

**Tabla 4.4: PROPUESTA ENTRE LAS CAPACIDADES, DESTREZAS, CONTENIDOS Y PROCEDIMIENTOS SEGÚN EL TEMA SELECCIONADO**

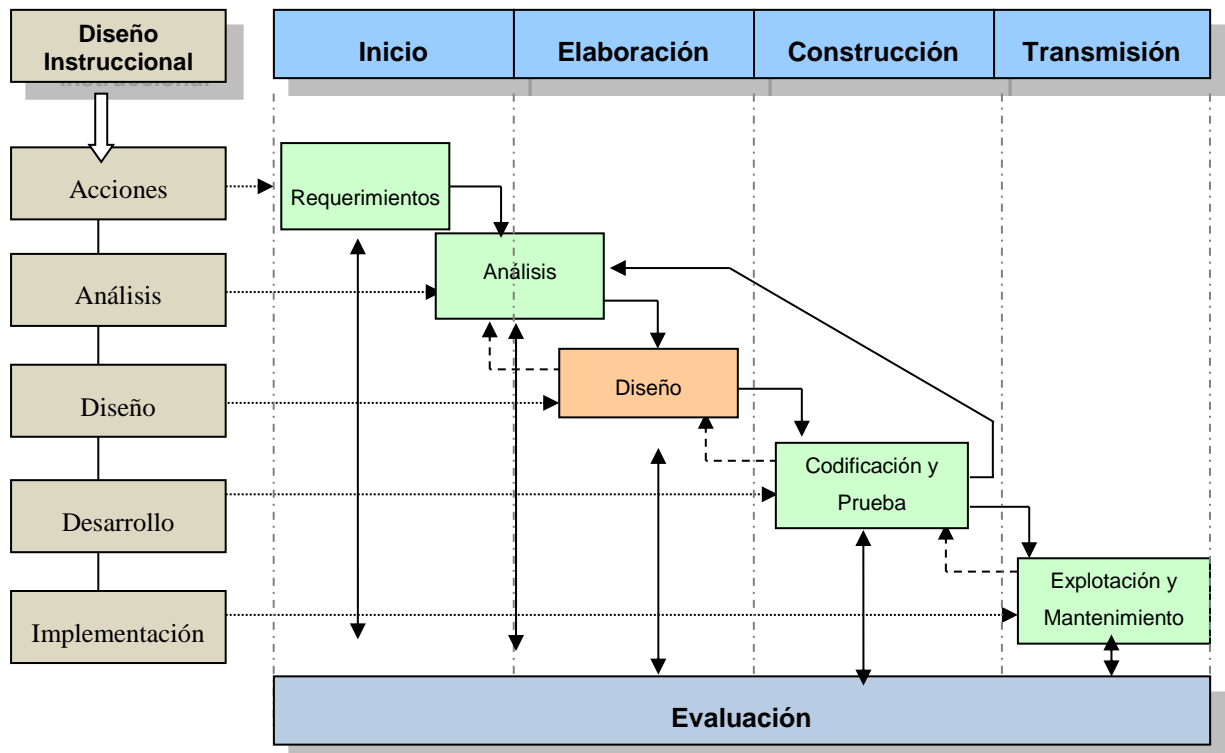
- **Desarrollo**, se toma como insumo el plano o marco para los elementos fundamentales del curso y su objetivo es agregar el contenido al marco del diseño. El resultado consiste en establecer las *especificaciones de los materiales* del curso (digitalizar la información) y un *ambiente de aprendizaje* propio del software educativo a desarrollar.
- **Implementación**, aquí el docente deberá estar listo y con una predisposición para establecer el proceso de aprendizaje con la ayuda del software educativo. En esa etapa se hará entrega del material del curso.

Por otro lado, la **fase de evaluación** al ser un eje transversal en el modelo será contemplado en cada una de las etapas con acciones que permitan garantizar que el material del curso preparado sea el apropiado.

### 2.8.2 Propuesta de Integración entre Diseño Instruccional y el Ciclo de Vida

La propuesta de desarrollo para la presente tesis se centra en la integración del ciclo de vida del software y el diseño instruccional, con ello se establece actividades donde el software educativo a construir contenga ingredientes que otras propuestas no la contemplan. Esta propuesta estará representando por la integración del diseño instruccional y el ciclo de vida como se muestra **figura 4.4**.

Por otro lado, se debe reconocer que al ser el ciclo de vida RUP una propuesta de desarrollo de software robusto, esta propuesta adaptará ciertos procesos que permita un desarrollo ágil e integrado.



**Figura 4.4: Propuesta de Integración del Diseño Instruccional y Ciclo de Vida.**

### 2.8.2.1 Fase de Inicio

Dirigida al entendimiento de los *requerimientos del programa* y *determinar el alcance del Software Educativo (SE)* a construir. Por otro lado, en esta fase debemos establecer actividades relacionadas con las **acciones previas** propias del diseño instruccional, la cual permita fortalecer la etapa de requerimiento propio del ciclo de vida. Así mismo se dará inicio a la etapa de análisis siempre y cuando el control de calidad sea superado propio de la etapa de evaluación.

En la **tabla 4.5**, se muestra las actividades propias de la fase de inicio y las actividades del diseño instruccional.

Adaptación del ciclo de vida	Diseño Instruccional
<ul style="list-style-type: none"> <li>• Formulación del problema</li> <li>• Necesidades y características del Software Educativo.</li> <li>• Reglas del negocio.</li> <li>• Glosario de términos.</li> <li>• Identificar los casos de usos por Actores.</li> <li>• Establecer las prioridades de casos de uso.</li> <li>• Identificación de los Actores.</li> </ul>	<p><b>Acciones Previas</b></p> <ul style="list-style-type: none"> <li>• Identificar el plan estratégico de la organización.</li> <li>• Establecer las limitaciones e intereses</li> </ul> <p><b>Análisis</b></p> <ul style="list-style-type: none"> <li>• Descripción del entorno externo</li> <li>• Establecer las características de los participantes.</li> <li>• Necesidades de aprendizaje.</li> </ul> <p><b>Además:</b></p> <ul style="list-style-type: none"> <li>• Definir el grupo de especialistas.</li> <li>• Proporcionar conceptos y un lenguaje común entre el <i>grupo especialista</i>.</li> </ul>

Tabla 4.5: Actividades de la fase de Inicio del Ciclo RUP.

### 2.8.2.2 Fase de Elaboración

Se debe **especificar las características y el diseño de la arquitectura del sistema** proveyendo las bases para el proceso de diseño e implementación. Esta definición de la arquitectura debe tener en cuenta los requerimientos de la fase inicio y proveer alternativas para el control de riesgos.

En la **tabla 4.6**, se detalla las actividades propias del ciclo de vida y el modelo instruccional, las cuales proporcionan actividades de análisis y diseño que permitan fortalecer ciclo de vida del software:

Adaptación del ciclo de vida	Diseño Instruccional
<ul style="list-style-type: none"> <li>• Asegurar que los requerimientos y definiciones hayan contemplado y aminorado todos los riesgos que puedan afectar el desarrollado SE.</li> <li>• Especificación de los casos de uso.</li> <li>• <b>Arquitectura del software:</b> <ul style="list-style-type: none"> <li>○ Vista de Casos de Uso Final</li> <li>○ Vista lógica.</li> <li>○ Vista de implementación.</li> <li>○ Vista de despliegue</li> </ul> </li> <li>• Verificar necesidades de hardware e infraestructura.</li> <li>• Afinar el diseño de la arquitectura a fin de verificar la solidez y cumpla con los requerimientos del Software Educativo.</li> <li>• Construir el prototipo inicial.</li> </ul>	<p><b>Análisis</b></p> <ul style="list-style-type: none"> <li>• Dominio de la materia.</li> </ul> <p><b>Diseño</b></p> <ul style="list-style-type: none"> <li>• Bosquejo del contenido del curso.</li> <li>• Identificar las capacidades y destrezas propias del curso a desarrollar.</li> <li>• Identificar los valores y actitudes institucionales.</li> <li>• Determinar los indicadores de evaluación.</li> <li>• Establecer los procedimientos para el desarrollo del curso.</li> <li>• Determinar los contenidos temáticos.</li> </ul>

**Tabla 4.6 ACTIVIDADES DE LA FASE DE ELABORACIÓN DEL CICLO VIDA.**

### 2.8.2.3 Fase de Construcción

Corresponde al desarrollo del software, tomando como punto de inicio el prototipo inicial definido en la etapa de elaboración, el cual ha sido observado por los patrocinados y el cual permita resolver los requerimientos definidos en las etapas anteriores. En esta etapa **se propone un proceso recursivo el cual permita refinar el software educativo a desarrollar** y cumpla con el control de calidad de lo contrario no se podrá pasar a la fase de transmisión del ciclo de vida.

En la **tabla 4.7**, se detalla las actividades de desarrollo propios del modelo instruccional, las cuales proporcionan actividades que permitan fortalecer ciclo de vida del software:



Adaptación del ciclo de vida	Diseño Instruccional
<ul style="list-style-type: none"> <li>• Desarrollar los componentes del Software Educativo.</li> <li>• Probar los componentes contra los criterios de evaluación definidos.</li> <li>• Gerenciar en paralelo la Base de datos con el desarrollo de los componentes.</li> <li>• Instalar y probar el Software Educativo como versión inicial.</li> </ul>	<p><b>Desarrollo</b></p> <ul style="list-style-type: none"> <li>• Plano o marco de diseño.</li> <li>• Establecer las especificaciones del <b>contenido del curso</b> y la <b>Digitalización</b>.</li> <li>• Ambiente de aprendizaje propio del Software Educativo a construir.</li> </ul> <p><b>Además</b></p> <ul style="list-style-type: none"> <li>• Desarrollo de actividades de evaluación relacionados con los indicadores definidos en la etapa de diseño.</li> </ul>

**Tabla 4.7: Actividades de la fase de construcción del Ciclo de Vida.**

#### 2.8.2.4 Fase de Transmisión

Corresponde a la **entrega del Software Educativo a los usuarios**. Esta etapa finaliza cuando el cliente queda satisfecho con el software educativo elaborado. En la **tabla 4.8**, se detalla las actividades de desarrollo propios del modelo instruccional, las cuales proporcionan actividades que permitan fortalecer ciclo de vida del software:

Adaptación del ciclo de vida	Diseño Instruccional
<ul style="list-style-type: none"> <li>• Instalación del software en el entorno final de trabajo.</li> <li>• Ajustar el software educativo a la organización.</li> </ul>	<p><b>Implementación</b></p> <ul style="list-style-type: none"> <li>• Capacitación a los docentes involucrados.</li> <li>• Evaluación del docente y del estudiante en relación a las capacidades y destrezas establecidas.</li> </ul>

**Tabla 4.8: ACTIVIDADES DE LA FASE DE TRANSMISIÓN DEL CICLO RUP.**

## 4.2 PRESENTACION DE RESULTADOS

En este capítulo se ha plasmado los resultados de la investigación, teniendo en cuenta lo siguiente:

Se aplicó una encuesta a los estudiantes del nivel secundaria de los grados de 4to. y 5to. de la Institución educativa San Francisco de Borja del distrito de San Borja, la misma que fue confeccionada de acuerdo a las variables e indicadores de la investigación y consto de 7 preguntas de tipo cerrada. En ese sentido, por la cantidad de estudiantes se implementó una plataforma virtual de encuestas **Limesurvey**; el cual es totalmente web permitiendo que los estudiantes contestaran en forma amigable y totalmente anónima.

Se procedió a vaciar y tabular los datos obtenidos a través de la encuesta, luego dichos datos se presentaron en los cuadros y gráficos respectivos.

Luego, se procedió a contrastar la hipótesis planteada, la misma que se realizó a través de la prueba no paramétrica chi cuadrado.

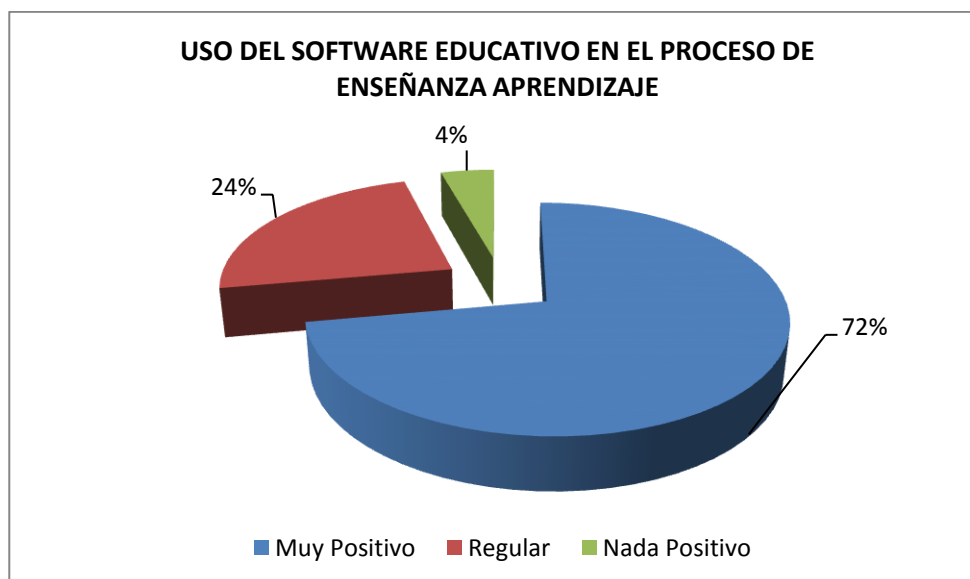
Finalmente, se efectuó la discusión de los resultados, debidamente argumentada a través de sustento teórico relacionado al tema de investigación.

### 4.3 ANALISIS E INTERPRETACIÓN DE LOS RESULTADOS

En el presente apartado se dará a conocer los resultados obtenidos como consecuencia de aplicar la encuesta a 115 estudiantes, en ese sentido se presentara cuadro por cuadro los resultados de cada una de las preguntas y luego se realizará la interpretación pertinente:

**CUADRO N°1**  
**USO DEL SOFTWARE EDUCATIVO EN EL PROCESO DE ENSEÑANZA**  
**APRENDIZAJE**

Respuestas	Nro.	Porcentaje
Muy positivo	83	72%
Regular	27	24%
Nada positivo	5	4%
<b>Total</b>	<b>115</b>	<b>100%</b>



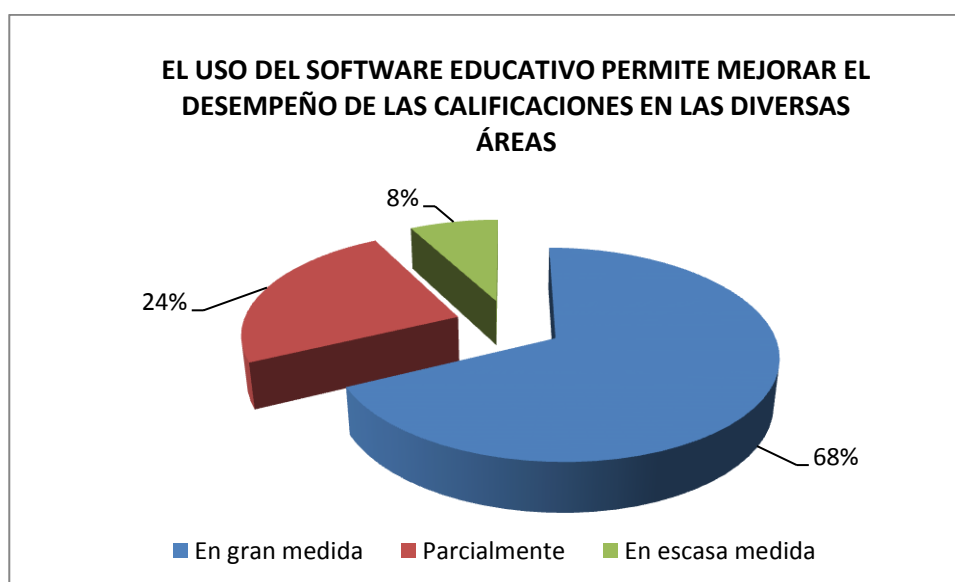
#### Interpretación

Con respecto a los resultados obtenidos del uso del software educativo en el proceso de enseñanza aprendizaje, 72% respondió que son muy positivos, el 24% respondió que son regulares y el 5% respondió que nada positivos. **Indicándonos estos resultados que la construcción del software educativo en el proceso de enseñanza aprendizaje son muy positivos.**

## CUADRO N°2

### EL USO DEL SOFTWARE EDUCATIVO PERMITE MEJORAR EL DESEMPEÑO DE LAS CALIFICACIONES EN LAS DIVERSAS ÁREAS

Respuestas	Nro.	Porcentaje
En gran Medida	78	68%
Parcialmente	28	24%
En escasa medida	9	8%
<b>Total</b>	<b>115</b>	<b>100%</b>



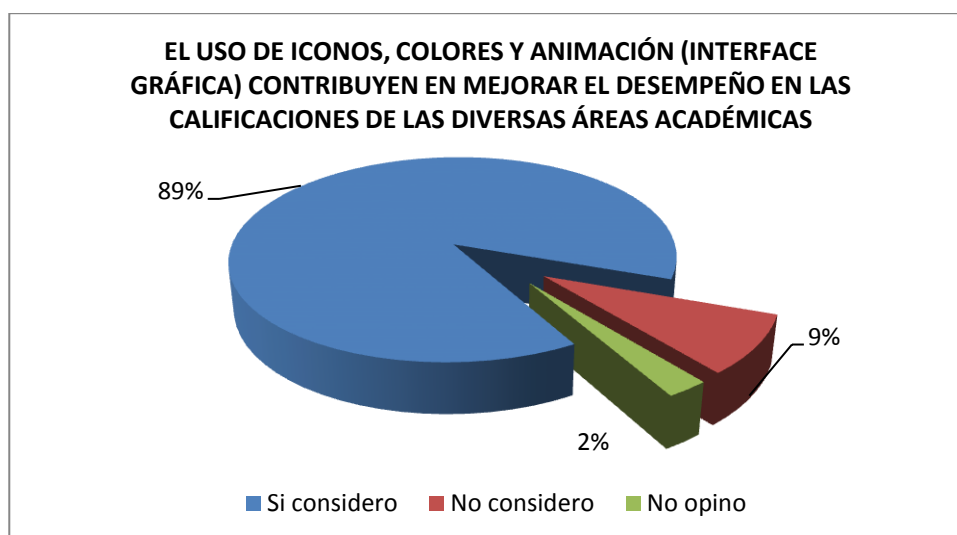
#### Interpretación

Con respecto a los resultados obtenidos sobre el uso del software educativo, el cual permite mejorar el desempeño de las calificaciones en las diversas áreas, el 68% respondió en gran medida, el 24% respondió parcialmente y el 8% respondió en escasa medida. **Indicándonos estos resultados que el uso del software educativo permite mejorar en gran medida el desempeño de las calificaciones en las diversas áreas.**

### CUADRO N°3

#### EL USO DE ICONOS, COLORES Y ANIMACIÓN (INTERFACE GRÁFICA) CONTRIBUYEN EN MEJORAR EL DESEMPEÑO EN LAS CALIFICACIONES DE LAS DIVERSAS ÁREAS ACADÉMICAS

Respuestas	Nro.	Porcentaje
En gran Medida	102	89%
Parcialmente	10	9%
En escasa medida	3	2%
<b>Total</b>	<b>115</b>	<b>100%</b>



#### Interpretación

Con respecto a los resultados obtenidos sobre el uso de iconos, colores y animación (interface gráfica) contribuyen en mejorar el desempeño en las calificaciones de las diversas áreas académicas, el 89% respondió que si considera, el 9% respondió no considera y el 8% respondió que no opina. **Indicándonos estos resultados que si considera uso de iconos, colores y animación (interface gráfica) contribuyen en mejorar el desempeño en las calificaciones de las diversas áreas académicas.**

#### CUADRO N°4

### EL CONTENIDO DE LAS DESTREZAS DEL ÁREA EN EL USO DEL SOFTWARE EDUCATIVO, TE PERMITIERON MEJORAR EL DESEMPEÑO EN LAS CALIFICACIONES DE LAS DIVERSAS ÁREAS ACADÉMICA

Respuestas	Nro.	Porcentaje
Si considero	68	59%
Parcialmente	34	30%
No considero	13	13%
<b>Total</b>	<b>115</b>	<b>100%</b>



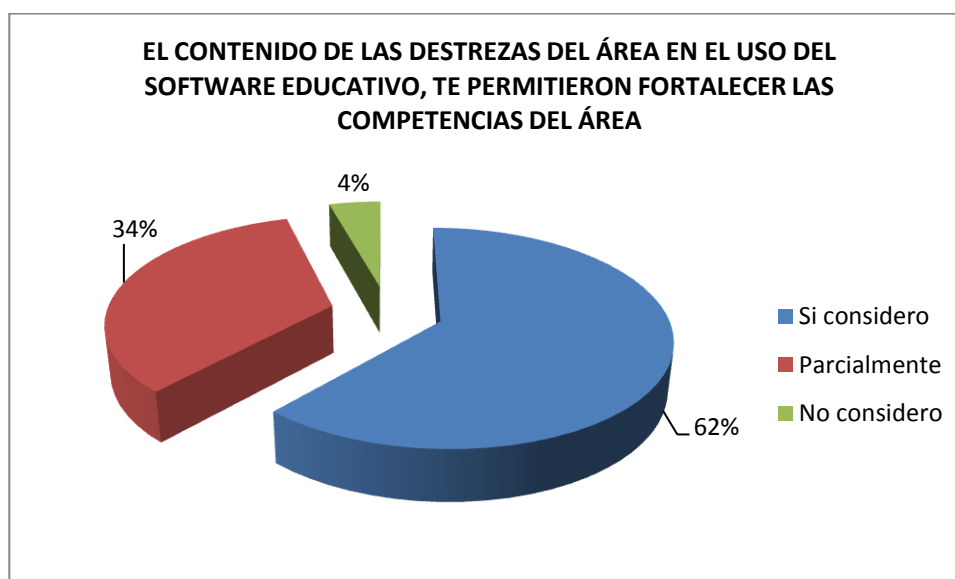
#### Interpretación

Con respecto a los resultados obtenidos sobre el contenido de las destrezas del área en el uso del software educativo, te permitieron mejorar el desempeño en las calificaciones de las diversas áreas académicas, el 59% respondió que si considera, el 30% respondió parcialmente y el 11% respondió que no considera. **Indicándonos estos resultados que los estudiantes consideran el contenido de las destrezas del área en el uso del software educativo permite mejorar en el desempeño de sus calificaciones.**

## CUADRO N°5

### EL CONTENIDO DE LAS DESTREZAS DEL ÁREA EN EL USO DEL SOFTWARE EDUCATIVO, TE PERMITIERON FORTALECER LAS COMPETENCIAS DEL ÁREA

Respuestas	Nro.	Porcentaje
Si considero	71	62%
Parcialmente	39	34%
No considero	5	4%
<b>Total</b>	<b>115</b>	<b>100%</b>



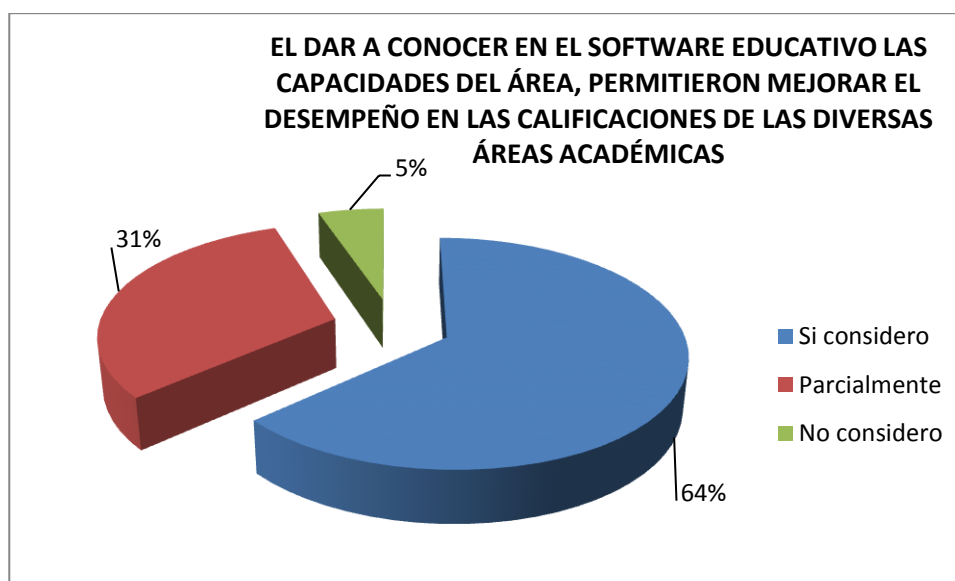
#### Interpretación

Con respecto a los resultados obtenidos sobre el contenido de las destrezas del área en el uso del software educativo, permitieron fortalecer las competencias del área, logrando mejorar el desempeño en las calificaciones de las diversas áreas académicas, el 62% respondió que si considera, el 34% respondió parcialmente y el 4% respondió que no considera. **Indicándonos estos resultados que los estudiantes consideran el contenido de las destrezas del área en el uso del software educativo fortalecer las competencias del área.**

CUADRO N°6

**DAR A CONOCER EN EL SOFTWARE EDUCATIVO LAS CAPACIDADES DEL ÁREA,  
PERMITIERON MEJORAR EL DESEMPEÑO EN LAS CALIFICACIONES DE LAS  
DIVERSAS ÁREAS ACADÉMICAS**

Respuestas	Nro.	Porcentaje
Si considero	73	64%
Parcialmente	36	31%
No considero	6	5%
<b>Total</b>	<b>115</b>	<b>100%</b>



**Interpretación**

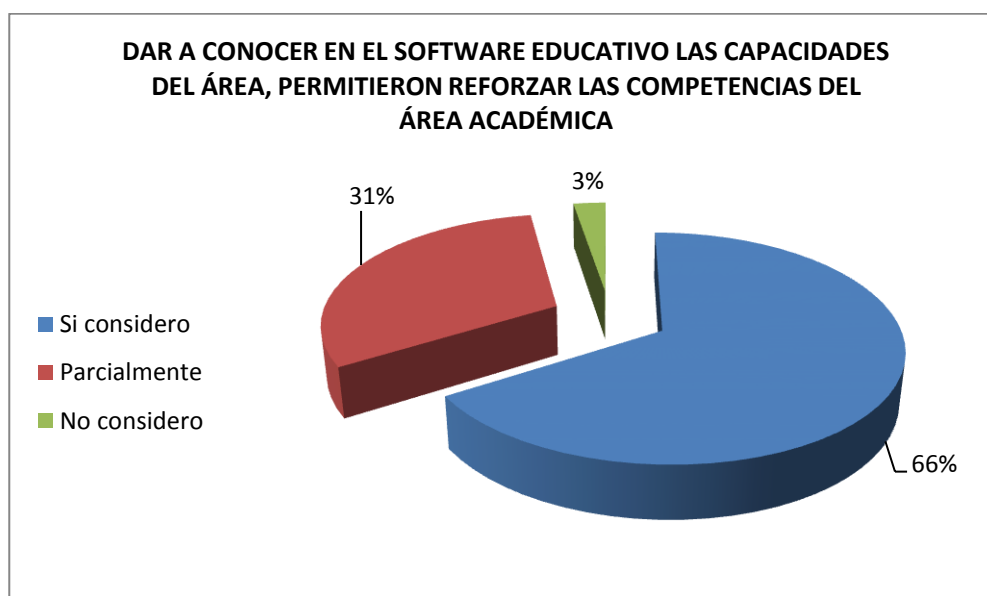
Con respecto a los resultados obtenidos sobre el dar a conocer en el software educativo las capacidades del área, permitieron mejorar el desempeño en las calificaciones de las diversas áreas académicas, el 64% respondió que si considera, el 31% respondió parcialmente y el 5% respondió que no considera. **Indicándonos estos resultados que los estudiantes consideran el dar a conocer en el software educativo las capacidades del área, permitiendo con ello mejorar el desempeño en las calificaciones de las diversas áreas académicas.**



## CUADRO N°7

### DAR A CONOCER EN EL SOFTWARE EDUCATIVO LAS CAPACIDADES DEL ÁREA, PERMITIERON REFORZAR LAS COMPETENCIAS DEL ÁREA

Respuestas	Nro.	Porcentaje
Si considero	76	66%
Parcialmente	36	31%
No considero	3	3%
<b>Total</b>	<b>115</b>	<b>100%</b>



#### Interpretación

Con respecto a los resultados obtenidos sobre el dar a conocer en el software educativo las capacidades del área, permitieron reforzar las competencias del área académica, el 66% respondió que si considera, el 31% respondió parcialmente y el 3% respondió que no considera. **Indicándonos estos resultados que los estudiantes consideran el dar a conocer en el software educativo las capacidades del área, permitiendo con ello reforzar las competencias del área académica.**

#### 4.4 CONTRASTACIÓN DE HIPOTESIS

Para proceder a contrastar la hipótesis principal, se utilizó como prueba estadística chi cuadrado, la cual responde a una estadística no paramétrica con el uso de dos variables como prueba independiente.

Se presenta a continuación los resultados obtenidos de la aplicación de la mencionada prueba mediante una tabla y el procedimiento propio de la prueba estadística.

#### PLANTEAMIENTO DE LA HIPÓTESIS PRINCIPAL

H1: La construcción de un modelo para el desarrollo de software educativo influye positivamente en el proceso de enseñanza - aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”.

Ho: La construcción de un modelo para el desarrollo de software educativo no influye positivamente en el proceso de enseñanza - aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”. (Independientes)

Para probar la hipótesis planteada seguiremos los siguientes procedimientos: [8]

- **FRECUENCIA OBSERVADA**

La construcción de un modelo para el desarrollo de software educativo	Proceso de enseñanza aprendizaje en la etapa escolar			Total
	Gran Medida	Parcialmente	En escasa Medida	
Muy Positivo	57	22	4	83
Regular	21	3	3	27
Nada Positivo	0	3	2	5
<b>Total</b>	<b>78</b>	<b>28</b>	<b>9</b>	<b>115</b>

- **FRECUENCIA ESPERADA**

La construcción de un modelo para el desarrollo de software educativo	Proceso de enseñanza aprendizaje en la etapa escolar			Total
	Gran Medida	Parcialmente	En escasa Medida	
Muy Positivo	56.30	20.20	6.50	83.00
Regular	18.31	6.58	2.11	27.00
Nada Positivo	3.39	1.22	0.39	5.00
<b>Total</b>	<b>78.00</b>	<b>28.00</b>	<b>9.00</b>	<b>115.00</b>

- **ESTADÍSTICA DE PRUEBA**

$$x^2 = \sum \frac{(O - E)^2}{E}$$

**Dónde:**

O = Frecuencia observada en cada celda.

E = Frecuencia esperada en cada celda.

- **DISTRIBUCIÓN DE LA ESTADÍSTICA DE PRUEBA**

En esta parte del proceso observamos que cuando  $H_0$  es verdadero,  $x^2$  sigue una distribución aproximada de chi cuadrado con la expresión  $(f - 1)(c - 1)$  grados de libertad. Es decir  $(3-1)(3-1) = 4$  grados de libertad.

- **NIVEL DE SIGNIFICANCIA O RIESGO**

Si se determina que  $\alpha$  es 5% es decir 0.05, entonces podemos decir según la tabla de Chi cuadrado  $x^2_{0.05, 4} = 9.488$ .

- **REGLA DE DECISIÓN**

Rechazar la hipótesis nula  $H_0$  si el valor de  $x^2$  es mayor o igual a 9.488.

- **CALCULO DE CHI CUADRADO**

Rechazar la hipótesis nula  $H_0$  si el valor de  $\chi^2$  es mayor o igual a 9.488.

$$\chi^2 = \sum \frac{(O-E)^2}{E} = \frac{(57-56.30)^2}{56.30} + \frac{(22-20.20)^2}{20.20} + \frac{(4-6.5)^2}{6.5} + \frac{(21-18.31)^2}{18.31} \dots\dots\dots + \frac{(2-0.39)^2}{0.39} = \mathbf{16.48}$$

- **CONCLUSIÓN DEL PLANTEAMIENTO DE LA HIPÓTESIS**

La hipótesis nula se rechaza, es decir existe una evidencia de una relación o influencia positiva entre la construcción del modelo para el desarrollo de software educativo y el proceso de enseñanza aprendizaje en la etapa escolar.

## 4.5 DISCUSIÓN DE LOS RESULTADOS

Sin lugar a duda la educación en nuestro país es un tema especial toda vez que el estado ha venido dictando medidas por mejorarla y en ese sentido el esfuerzo que realiza los diversos sectores no han sido los suficientes. En ese sentido, se deben aplicar otras estrategias para que esto mejore. El uso del software educativo como estrategia de aprendizaje por parte del docente ha ido acrecentándose en nuestro país y juega un papel importante para el proceso de enseñanza- aprendizaje.

Los profesores alemanes como Graf Werner, Kurt Mocker y Gunter Wesiman, consideran al software educativo “cómo medios de enseñanza y materiales didácticos en general, a los medios que se crearon conscientemente para el proceso pedagógico sobre la base de los documentos de enseñanza, aprovechando los conocimientos pedagógicos, y que sirven al que enseña y al que aprende a realizar procesos didácticos”. [28]

En la tesis, se ha podido demostrar a partir de la aplicación de la estadística no paramétrica chi cuadrado con el uso de dos variables como prueba independiente; de decir las variables de construcción de un modelo para el desarrollo de software educativo y el proceso de enseñanza - aprendizaje en la etapa escolar; influyen en forma positiva; es decir existe una relación entre las dos variables que permite que el estudiante mejore su rendimiento académico.

En las encuestas realizadas a los estudiantes de 4to. y 5to. grado de educación secundaria de la Institución San Francisco de Borja, se observan la aceptación de los estudiantes en el uso de un software educativo con enfoque pedagógico (cuadro N°1) mejora en el desempeño de sus áreas académicas en un 72% en forma positiva. En ese sentido, el uso del software educativo permite mejorar el desempeño de los estudiantes en un 68% en gran medida las calificaciones de las diversas áreas (cuadro N°2).

Por otro lado, la autora Sara Noemi Neiret, señala que “el objetivo de una interfaz gráfica es un medio y no un fin, ya que el fin es la comunicación de los contenidos para el aprendizaje, es importante que este medio permita ser impactante toda vez que los contenidos en un software educativo se convertirían en inútiles” [29]. En ese sentido, se muestra la influencia positiva que tiene el uso de interfaces gráficas en el software educativo (cuadro N° 3); evidenciando con ello un porcentaje de 89% que señalan en gran medida la contribución en la mejora del desempeño de las calificaciones en las diversas áreas académicas.

El autor Martiniano Román Pérez, no dice que “las destrezas son las habilidades específicas que emplean los estudiantes para aprender” [10]. En ese sentido en la encuesta realizada a los estudiantes señalan que un 59% si considera el contenido de las destrezas del área en el uso del software educativo (cuadro N° 4); toda vez que con ello se potencia las capacidades del área, mejorando con las calificaciones. En ese sentido, un 62% si considera el fortalecimiento de las competencias del área (cuadro N° 5).

El mismo autor nos dice que “la capacidad representa una habilidad general, que utiliza el estudiante para aprender y son importantes para los objetivos fundamentales de toda área académica” [10]. En ese sentido, en la encuesta se señala que un 64% si considera dar a conocer en el software educativo las capacidades del área (cuadro N° 6), permitieron mejorar el desempeño en las calificaciones de las diversas áreas académicas. En ese sentido, un 66% si considera el fortalecimiento de las competencias del área (cuadro N° 7).

Finalmente, los estudiantes de esta institución son conocedores de los términos capacidades y destrezas que se desarrollan en las diversas áreas académicas, se evidencia la aceptación de que estos se consideren en el software educativos y puedan ser trabajados con los programas académicos a través de las sesiones de aprendizaje que los docentes preparan en la planificación de sus clases como estrategias de aprendizaje.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- a. Los resultados obtenidos como consecuencia de aplicar las medidas estadísticas nos señalan que la hipótesis nula es rechazada toda vez que el valor de chi cuadrado  $\chi^2 = 16.48$  es mayor que 9.488, deduciendo con ello que la hipótesis nula es rechazada.
- b. Por otro lado, se encontró que 89% de los estudiantes encuestados prefieren interfaces graficas toda vez que permiten mejorar el desempeño en las calificaciones de las diversas áreas académicas.
- c. Un 62% de los estudiantes encuestados consideran como fortaleza el contenido de las destrezas del área en el uso del software educativo para mejorar las competencias del área académica,
- d. Se encontró que un 64% de los estudiantes encuestados consideran conocer en el software educativo las capacidades del área, permitiendo con ello mejorar el desempeño de las calificaciones de las diversas áreas académicas.
- e. Un 66% de los estudiantes encuestados consideran como fortaleza conocer en el software educativo las capacidades del área, permitiendo con ello mejorar las competencias de las áreas académicas.
- f. En líneas generales, la hipótesis sobre la aplicación de un modelo para el desarrollo de software educativo influye positivamente en el proceso de enseñanza aprendizaje en la etapa escolar de la Institución Educativa “San Francisco de Borja”.
- g. La aplicación de este modelo buscará que profesionales del área de computación desarrollen programas educativos que tomen en cuenta procesos de ingeniería de software y procesos pedagógicos que actualmente se viene planteando.

## 5.2 RECOMENDACIONES

- a. El gobierno necesita fortalecer el desarrollo de software educativo toda vez que es una herramienta que debe ser usada por el docente como estrategia para el aprendizaje.
- b. El ministerio de educación a través de sus programas educativos, debe fortalecer el desarrollo de software para el uso en instituciones educativas, esto será una herramienta poderosa que el docente tendrá para el desarrollo de sus sesiones de aprendizaje.
- c. Fortalecer a los docentes con capacitaciones constantes en el uso de las tecnologías de la información y comunicaciones.
- d. Establecer una trilogía conformada por el Ministerio de Educación, Instituciones educativas y docentes, de tal forma que este último participe constantemente en talleres de capacitación de uso de tecnologías de la información y comunicaciones permitiendo no solo afianzar sus conocimientos, sino emplear nuevas estrategias para el aprendizaje.
- e. Promover en las Universidades e Institutos de computación el desarrollo de software educativo, toda vez que es un área muy poco desarrollada, por parte de los estudiantes que vuelcan su esfuerzo en desarrollar soluciones comerciales. Permitiendo con ello el crecimiento del sector educativo, el cual ha venido a menos según las últimas encuestas.
- f. Preparar a los programadores de software en el desarrollo de software educativo, contemplando con ello el uso de modelos orientados al aspecto pedagógico.
- g. Se recomienda ampliar la investigación en Instituciones educativas del sector estatal y con ello medir el uso de software educativo como estrategia de docentes en el desarrollo de sus sesiones de aprendizaje.



## REFERENCIAS BIBLIOGRAFICAS

1. Román P. Martiniano. (2004). Estrategias de Aprendizaje en el Aula: Diseño y Evaluación. Grupo Editorial Norma.
2. Pere Marques (1995). Metodología para la elaboración de software educativo. Guía de uso y metodología de diseño. Barcelona. Editorial Estel. ISBN 84-605-2573-2
3. Piattini M (1996). Análisis y diseño detallado de aplicaciones informáticas de gestión. Rama. Madrid.
4. Cataldi, Zulma. (2000). Una metodología para el diseño, desarrollo y evaluación de software educativo. Tesis. Universidad Nacional de La Plata-Facultad de Informática.
5. Díaz-Antón (2002). Propuesta de una Metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica. IV Congreso Multimedia Educativo, Universidad de Barcelona, España.
6. Galvis, A.; Gómez, R. y Mariño O.(1998) . Ingeniería de Software Educativo con Modelaje Orientado a Objetos: Un medio para desarrollar Micromundos Interactivos”. UNIANDES-LIDIE Informática Educativa. Vol.11, Nro. 1, pp 9-30.
7. Román, M. (2004). Sociedad del Conocimiento y Refundación de la Escuela desde el Aula. Editorial Eos – Madrid Ada. Reina Victoria 8 28.003 Madrid.
8. Webster, Allen L (2000). Estadística aplicada a los negocios y la economía. Editorial McGraw Hill Internacional S.A. Santa Fé de Bogotá, Colombia.
9. Alberto Prieto, Antonio Lloris, Juan Carlos Torres (2001). Introducción a la Informática, McGrawHill, 3ra. Edición.
10. Román Pérez, Martiniano (2004). Estrategias de aprendizaje en el aula: diseño y evaluación. Grupo Editorial Norma. Lima-Perú.
11. Bernardo Carraco, José (2007). Estrategias de aprendizaje, para aprender más y mejor. Ediciones Rialp, S.A. Madrid-España.
12. Cataldi Z., Lage F., et al. (2000). Methodology of design and development of educational software from a pedagogical perspective. Anales de ICECE 2000 International Conference on Engineering and Computer Education. 27-30 de agosto. San Pablo. Brasil.
13. Hinostraza, E., Hepp, P., Mellar, H., Preston, C., Rehbein, L. (1997). ¿Diseño de Software Educativo o de Software Escolar? Revista Informática Educativa, Proyecto SIIE, Colombia, Volumen (Nº\_10 (1)), 57-73.
14. Hinostraza E. and Mellar Harvey. Pedagogy embedded in educational software design: report of a case study. Elsevier. Computers & Education, Volume 37, Issue 1, August 2001, Pages 27-40
15. Ortega, C (2002). Tecnología y Educación: el software como apoyo al proceso de aprendizaje. Revista panamericana de Pedagogía: Saberes y quehaceres del pedagogo. Revista Semestral 3. Nueva Época. Universidad Panamericana. Facultad de Pedagogía. México. Pp. 189-196.

16. Pressman Roger S. (2005). Software Engineering: A Practitioner's Approach. 6º Edición McGraw-Hill.
17. Díaz-Antón (2002). Uso de software educativo de calidad como poderosa herramienta de apoyo al aprendizaje. Encuentro educativo: La escuela como instrumento de cambio, Institutos Educativos Asociados (IEA), Marzo, 2002, Caracas, Venezuela.
18. Díaz-Antón, M.A. Pérez, A.C., Grimán, L. Mendoza.(2002). Instrumento de evaluación de software educativo bajo un enfoque sistémico. Presentado en el 6to. Congreso Iberoamericano, 4to. Simposio Internacional, 7mo. Taller Internacional de Software Educativo: Universidad de Vigo, España.
19. Ausubel, D.; Novak, J. y Hanesian, H.(1978). Psicología educativa. Un punto de vista cognitivo. Trillas. Ediciones 1978, 1997.
20. Hernández, Eugenio (2005). Diseño instruccional aplicado al desarrollo de software educativo. Fundación Arturo Rosenblueth. Tecnología educativa Galileo. México DF.
21. Gros, B. (1997). Diseños y programas educativos. Pautas pedagógicas para la elaboración de software. Barcelona. Editorial Ariel Educación.
22. Información general de la página oficial del Ministerio de Educación. Fecha de consulta 06/01/15  
<http://www.minedu.gob.pe/institucional/>.
23. Normatividad del Ministerio de Educación. Reglamento de Organización y funciones del Ministerio de Educación. Fecha de consulta 06/01/15  
<http://www.minedu.gob.pe/normatividad/>
24. Dirección general de educación básica regular. Página web del Ministerio de Educación. Fecha de consulta 06/01/15  
<http://www.minedu.gob.pe/digebr/> y <http://www.minedu.gob.pe/digebr/des/>
25. Grupo de Análisis para el Desarrollo (GRADE). Fecha de consulta 01/01/15  
<http://www.grade.org.pe/>
26. Diseño instruccional. Fecha de consulta 01/01/15  
<http://disenoinstruccional.wordpress.com/category/definicion-de-di/>
27. ESCALE. Unidad de estadística educativa- Ministerio de Educación. Fecha de consulta 07/01/15  
<http://escale.minedu.gob.pe/indicadores2011>
28. GRAF, W, KURT MOCKER Y GUNTER WESIMAN, Función, estructura y clasificación de los medios didácticos de enseñanza audiovisuales en la educación socialista de las escuelas superiores. Universidad Alejandro de Humboldt, (S.A.). Berlín.
29. Neiret, Sara (2008). Diseño de Interfaz en el Software Educativo Infantil. Diseño de Emociones. Universidad Nacional de La Plata Facultad de Informática.
30. EUMED.Net, Enciclopedia Virtual. Ministerio de Educación. Fecha de consulta 08/01/15  
<http://www.eumed.net/libros-gratis/2013/1241/ministerio-de-educacion-del-peru.html>

## ANEXO

### ENCUESTA

La presente encuesta es anónima y es con la finalidad de realizar un trabajo de investigación; en ese sentido te pedimos que te expreses con la mayor libertad posible y selecciones una alternativa de las presentadas:

1. De acuerdo al uso de los programas educativos ¿cuáles son los resultados obtenidos a través del uso del software educativo en tu proceso de enseñanza aprendizaje?
  - a. Muy positivo
  - b. Regular
  - c. Nada positivo
2. ¿En qué medida consideras que el uso del software educativo permite mejorar el desempeño de las calificaciones en las diversas áreas?
  - a. En gran medida
  - b. Parcialmente
  - c. En escasa medida.
3. ¿Consideras que el uso de iconos, colores y animación (interfaces gráficas) contribuyen en mejorar el desempeño en las calificaciones de las diversas áreas académicas?
  - a. Si considero
  - b. No considero
  - c. No opino
4. ¿Consideras que el contenido de las destrezas del área en el uso del software educativo, te permitieron mejorar el desempeño en las calificaciones de las diversas áreas académicas?
  - a. Si considero
  - b. Parcialmente
  - c. No considero
5. ¿Consideras que el contenido de las destrezas del área en el uso del software educativo, te permitieron fortalecer las competencias del área?
  - a. Si considero
  - b. Parcialmente
  - c. No considero
6. ¿Consideras que el dar a conocer en el software educativo las capacidades del área, te permitieron mejorar el desempeño en las calificaciones de las diversas áreas académicas?
  - a. Si considero
  - b. Parcialmente
  - c. No considero
7. ¿Consideras que el dar a conocer en el software educativo las capacidades del área, te permitieron reforzar las competencias del área académica?
  - a. Si considero
  - b. Parcialmente
  - c. No considero

Gracias por su encuesta..!