



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA.

PROCESO DE TITULACIÓN

DICIEMBRE 2021 – ABRIL 2022

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PRUEBA PRÁCTICA

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN SISTEMAS DE INFORMACIÓN

TEMA:

**ANÁLISIS COMPARATIVO ENTRE PARADIGMAS DE
PROGRAMACIÓN Y SU INFLUENCIA EN UN PROYECTO DE
DESARROLLO DE SOFTWARE**

ESTUDIANTE:

HEIDY BEATRIZ MARQUEZ MORA

TUTOR:

ING. MALIZA CRUZ WELLINGTON ISAAC

AÑO 2022

RESUMEN

El presente estudio de caso aborda la temática “análisis comparativo entre paradigmas de programación y su influencia en un proyecto de desarrollo de software” se tiene como objetivo principal determinar la relación que existe entre los paradigmas de programación y la influencia en el desarrollo de un proyecto de software ya que durante el estudio dentro de la universidad técnica de Babahoyo se identificó falta de conocimiento sobre los diferentes paradigmas de programación por parte de los estudiantes de la carrera de ingeniería en sistemas y esto genera inconvenientes a la hora de crear programas de software. Para tener un mejor enfoque de la temática abordada se realizó una encuesta a los estudiantes de la carrera mencionada y se desarrolló el presente estudio de caso mediante el método cualitativo y la investigación descriptiva mismos que ayudan en la identificación y definición de las variables puestas en estudio para de esta manera generar conclusiones y recomendaciones específicas que permitan dar solución a la problemática planteada.

Palabras claves: paradigmas, lenguajes de programación, software, hardware

ABSTRACT

This case study addresses the theme "comparative analysis between programming paradigms and their influence on a software development project" whose main objective is to determine the relationship between programming paradigms and the influence on the development of a project. of software since during the study within the technical university of Babahoyo a lack of knowledge about the different programming paradigms was identified by the students of the systems engineering career and this generates inconveniences when creating software programs. In order to have a better approach to the topic addressed, a survey was carried out on the students of the aforementioned career and the present case study was developed through the qualitative method and descriptive research, which help in the identification and definition of the variables put into study. in order to generate specific conclusions and recommendations that allow a solution to the problem raised.

Keywords: paradigms, programming languages, software, hardware

CONTENIDO

PLANTEAMIENTO DEL PROBLEMA.....	5
JUSTIFICACIÓN.....	7
OBJETIVOS.....	8
LÍNEAS DE INVESTIGACIÓN	9
MARCO CONCEPTUAL.....	10
MARCO METODOLOGICO	22
RESULTADOS	24
DISCUSIÓN DE RESULTADOS.....	25
CONCLUSIONES.....	27
RECOMENDACIONES	28
REFERENCIAS	29
Referencias	29
ANEXOS.....	32

PLANTEAMIENTO DEL PROBLEMA

Dentro del proceso de estudio en la carrera de sistemas de información de la universidad técnica de Babahoyo se identificó ciertos inconvenientes por parte de los estudiantes al momento de la elaboración o creación de programas de software, por lo que se realizó un análisis minucioso determinando que la posible causa de falla en dichos procesos es la mala elección del paradigma de programación lo que genera fracasos y en ciertas ocasiones los programas ni siquiera llegan a funcionar, esto ya sea por falta de información o conocimiento al momento de decidir el paradigma lo que provoca de diferentes maneras que el programa no funcione o se desarrolle con falencias.

Se dice que el paradigma de programación influye en la complejidad del código y la agilidad para implementar cambios, por lo tanto, también afecta a la mantenibilidad del sistema que está directamente relacionada con la facilidad de implementación de nuevas funcionalidades convirtiéndose en un factor crítico para el éxito o fracaso de un proyecto. En la elaboración de software la parte de diseño y análisis es una de las más importantes, pues de ahí se presentará el desarrollo del programa, para este proceso es importante la selección del paradigma de programación adecuado.

La constante demanda de proyectos de software ha causado que cada vez sean creados con una mayor calidad por lo que existen normas y ciertos lineamientos para guiar los procesos de creación debido a la complejidad que estos puedan tener y de esta manera evitar los fracasos y la pérdida de tiempo de ingenieros preparados en el área. También se ha podido observar una evolución en los diferentes paradigmas de programación ya que los lenguajes están orientados a procesos que deben estar previamente explicados y comprendidos por parte del creador.

Un paradigma de programación permite generar un estilo de software lo que también lleva de la mano el lenguaje de programación que define la manera de trabajar para poder obtener resultados siguiendo lineamientos y parámetros que hacen más fácil el trabajo, sin embargo, es necesario tener el conocimiento adecuado para poder implementar las pautas necesarias dentro de los sistemas de información y de esta manera lograr un programa de calidad para la comunidad.

Los distintos lenguajes de programación están relacionados directamente al paradigma para poder implementar herramientas que permitan concretar ideas y desarrollar códigos que faciliten el trabajo del usuario al momento de utilizar un programa. Considerando que un paradigma es un conjunto de ideas para la creación de un programa es importante tener en cuenta que el programa va a permitir al computador realizar una tarea determinada.

Es muy importante analizar la correcta elección entre los paradigmas de programación y permitir evaluar los beneficios reales de cada uno. Los paradigmas de programación son los principios fundamentales en el desarrollo de un proyecto de software. Lo más fácil es planteárselos como estilos de programación fundamentalmente diferenciados que, en consecuencia, generan códigos de software que están estructurados de forma distinta.

Relacionar una conceptualización determinada resulta fácil, sin embargo, al unir diferentes definiciones o conclusiones puede causar cierta confusión o estrés por lo que se considera importante plantear un resumen específico de las diferencias puntuales entre las características del presente caso de estudio para allí llegar a conclusiones más rápidas y de esta manera elegir el paradigma adecuado en la creación de programas de software.

Para los estudiantes de la carrera de sistemas de información uno de los mayores retos en el diseño de programas lo que tienes completa relación con el paradigma de programación que se ha utilizado. Por ello resulta importante el estudio de los mismos y para su mejor comprensión resulta factible plantear las relaciones existentes y así determinar cuál es mejor al momento de la implementación.

JUSTIFICACIÓN

Se dice que el éxito o fracaso de un proyecto de desarrollo de software está influenciado directa o parcialmente en la elección del paradigma de programación que se ha utilizado o el lenguaje implementado, por lo que conocer la relación entre ellos y los detalles relevantes que los relacionan con la informática va a permitir seleccionar el paradigma adecuado que conlleve sin duda alguna al éxito y correcta ejecución del proyecto o programa a desarrollar y de esta manera ahorrar tiempo y recursos.

Un paradigma de programación es un conjunto de principios en programación de software o básicamente un estilo de programación mediante el cual es posible generar códigos diferentes con una estructura previamente establecida para así desarrollar un software que sea una ayuda para la sociedad, eficiente y completo con el que se cumpla con las demandas que el mundo actual exige.

La programación permite a la sociedad un desarrollo de actividades de calidad por lo que la elaboración de programas resulta en ciertos casos un trabajo complejo debido a la importancia que tienen los resultados frente a la sociedad. Una persona que debe elaborar o modificar un programa debe tener un alto conocimiento en los diferentes aspectos del sistema informático motivo por el cual se desarrolla el presente estudio de caso para dar un mejor enfoque desde los paradigmas de programación que resultan clave esencial al momento de la creación de un programa.

En el presente estudio de caso se tiene como objetivo determinar la relación que existe entre los paradigmas de programación y su influencia en el desarrollo de un proyecto de software, ya que al definir y establecer las diferencias y relaciones existentes entre ellos es posible ejecutar de manera acertada un proyecto de desarrollo de software lo que va a permitir evitar tantas pruebas, cambios y fracasos, también se desarrolla con el fin de orientar sobre la correcta elección del paradigma y así se obtengan programas de software productivos.

OBJETIVOS

Objetivo general

Determinar la relación que existe entre los paradigmas de programación y la influencia en el desarrollo de un proyecto de software.

Objetivos específicos

- Identificar las dificultades que se presentan al momento de desarrollar un proyecto de software.
- Determinar lo que influyen los paradigmas de programación en el mantenimiento de un desarrollo de proyecto de software.
- Indicar el paradigma orientado a objetos para el software que puede producir códigos más fáciles de entender.

LÍNEAS DE INVESTIGACIÓN

La elaboración del presente estudio de caso se desarrolló basándose en las líneas de investigación de la universidad técnica de Babahoyo reconociendo como pertinente la línea de “sistemas de información y comunicación, emprendimiento e innovación y a su vez la sub línea “redes de tecnologías de software y hardware” que se relacionan directamente con el tema seleccionado análisis comparativo entre paradigmas de programación y su influencia en un proyecto de desarrollo de software

- **Línea de investigación**

Sistema de información y comunicación, emprendimiento e innovación.

- **Sub línea de investigación**

Redes y tecnologías inteligentes de software y hardware.

MARCO CONCEPTUAL

Paradigma de programación

Un paradigma de programación puede ser definido como un estilo o manera de programación de software, se trata de un conjunto de técnicas o métodos que pueden ser aplicados en todos los niveles de un diseño de programación que ayuden a resolver problemas computacionales. Existen muchas formas de diseñar un mensaje de programación y varios modelos para trabajar estos resultados. Un paradigma no es un simple modelo como copia, es mucho más que eso, se puede decir que es un modelo ejemplar, algo perfecto de tal manera que pueda considerarse digno de ser imitado y seguido.

Según Guisti (2019) más allá de todos los aspectos instrumentales que tiene cada paradigma, el creador o ejecutor de un programa está obligado a desarrollar técnicas y habilidades de manera general para poder entender el problema en específico, así como sus restricciones y las posibles soluciones que puedan plantearse al presentarse un problema. Pueden también pensar en paradigmas alternativos para dar solución a un problema, en este aspecto formativo es necesaria una gran preparación de informática (p.11).

Un paradigma de programación es explicado también como un estilo de desarrollo de programas o más bien como un modelo que nos ayuda a resolver problemas computacionales. Aquí también tienen relación los lenguajes de comunicación, que generalmente incluye uno o varios paradigmas que van a permitir implementar las órdenes necesarias para poder desarrollar el programa deseado.

La programación generativa también puede ser un paradigma de ingeniería y se puede basar en el desarrollo de una familia, lo cual tiene como producto final crear un modelo que sea capaz de sintetizar varios de los programas de una misma familia y a partir de esto especificar un alto nivel de abstracción. El resultado de esta programación

no siempre será un programa completo, pues también puede generarse solo un programa parcial o ciertos componentes para cualquier software. (Heredia, 2018)

La programación y su proceso de desarrollo se ha dado de acuerdo a etapas evolutivas, las cuales son cuatro etapas que se caracterizan en la aplicación de herramientas y teorías de cada momento histórico y avances la ciencia al que pertenezca según los años de evolución: Dentro de la primera etapa se puede decir que la programación en sus inicios se da como un arte cuyas principales herramientas fueron el lenguaje seleccionado y las habilidades personales, por lo que un software no tenía respaldo ni de teorías, ni de metodología que diera una explicación a su funcionamiento y al de los programas. Dentro de la segunda etapa se puede identificar las dificultades que se dan al momento de programar, así como la evolución de los programas y los mecanismos de los mismos.

La siguiente etapa es decir la tercera se caracteriza en una aparición de las importantes tecnologías para el desarrollo de software, los estándares y modelos de software. Con esta etapa se da el inicio del paradigma orientado a objetos para volverse reutilizable. Por último, tenemos la cuarta etapa en dónde se propone una alternativa para los modelos de programación que fueron utilizados en los inicios y corresponden también a una utilización de modelos de programación declarativa.

Según (Dominguez) Existen métodos de diseño en los que influyen los lenguajes de programación indicando los pasos a cumplir para poder desarrollar un programa. Los paradigmas muestran una constante creciente y evolución que con el tiempo ha tenido la incorporación de elementos que permiten crear programas cada vez más eficientes y sólidos, con el fin de facilitar el trabajo de programador para su mantenimiento desarrollo y adaptación. Sin embargo, las características de los lenguajes de programación fueron puestas en duda y criticadas de tal manera que se han dado diferentes teorías y grupos que postulan formas diferentes para la implementación de un paradigma.

Clasificación de los paradigmas

Los paradigmas pueden ser clasificados de diferentes maneras desde un punto de vista de criterios en prioridad. Pero si se parte desde principios fundamentales de cada paradigma son clasificados en dos; paradigma de programación imperativa y paradigma de programación declarativa. Los que a su vez tiene otra clasificación dentro de cada grupo; imperativa (programación estructurada, programación procedimental y programación modular); declarativa (programación lógica y programación funcional) (Herrera & Alexander, 2020)

Paradigmas de programación imperativa

En uno de sus documentos (Ionos, 2020) menciona que “entre todos los paradigmas de programación, la imperativa es considerada el paradigma clásico, ya que los principales lenguajes de programación y los primeros programas informáticos están basados en este enfoque” lo que da una secuencia regularizada de instrucciones determinadas para así llegar a la elaboración y ejecución de un programa. En esta programación se radica el trabajar lo más cerca posible del sistema para que así la programación resulte fácil de comprender, abarcable e incluso reutilizable.

Es el grupo más conocido y utilizado dentro del proceso de programación, aquí se desarrollan los programas a través de procedimientos e intervienen tres lenguajes imperativos. Este paradigma tiene su inicio en el principio del año 1950 en donde los diseñadores de programas reconocieron varias variables y comandos que constituían una útil abstracción el acceso a memoria e instrucciones de las máquinas.

Este paradigma tiene un elevado dominio en la actualidad ya que cierta parte del software actual se ha desarrollado en lenguajes imperativos por lo que resulta mucho más beneficioso el uso e implementación de este paradigma. La mayor parte de los

programadores profesionales son exclusivamente imperativos por lo que resulta uno de los paradigmas más importantes.

Este paradigma tiene los tres conceptos básicos que pueden generar secuencias basándose desde una acción, es decir que el lenguaje imperativo se diseña para transformar estados mediante ciertas asignaciones de manera ordenada, también se sostiene una ventaja muy importante de este paradigma que es el hecho de la facilidad que posee para representar conceptos cotidianos de manera algorítmica, esto ha dado como resultado una importancia atractiva para ser usado en todo tipo de proyectos de software. (Dimaggio, 2019)

El paradigma de programación imperativo tiene tres enfoques adicionales para escribir y estructurar un código software estos son: programación estructurada procedimental y modular

Programación estructurada

La programación estructurada es una forma simplificada de la programación imperativa pues su modificación radica en qué en lugar de tener instrucciones absolutas prevé el uso de bucles y ciertas estructuras de control para de esta manera realizar de forma automática la instrucción siempre y cuando se de una condición determinada. (Ionos, 2020) Se puede decir también que la programación estructurada orienta a una mejora en la claridad y calidad del trabajo lo que también se basa en el teorema de cada programa estructurado que ha sido propuesto por lo que ha permitido desarrollar mayor cantidad de software con una fácil comprensión.

Un programa estructurado es la base teórica sobre la que se construye una nueva forma de programar y esta es una característica fundamental de esta programación. Los

programas desarrollados en base a la programación estructurada son mucho más sencillos de comprender debido a que tienen una estructura secuencial. Una ventaja de estos programas es que resulta poseer una estructura clara debido a que las secuencias están ligadas entre sí. Durante la fase de prueba y depuración cada programa es optimizado lo que permite hacer más sencillo el seguimiento de errores y fallas para poder corregirlos a tiempo. (Angarita et all, 2020)

Si hablamos del ámbito económico el costo del mantenimiento de estos programas es reducido ya que mantener o modificar estos programas es más fácil en diferencia a los otros programas existentes. Una forma virtual de incluirse en la participación del módulo de la programación estructurada es conocer las funciones que permiten estar en este determinado conjunto punto la mente humana utiliza ciertos patrones o modelos con los que se puede facilitar la comprensión en el mundo real así estos patrones mentales guían el comportamiento y aquellas características de un elemento o situación ayudando a la comprensión y eliminando detalles irrelevantes.

Programación procedimental

El paradigma procedimental surge de la programación imperativa esto con el fin de disminuir la dificultad del código que se genera por paradigmas no estructurados, mismos que tenían la finalidad de favorecer el uso de sentencias como “go to” lo que provocaba cierta cantidad de saltos dentro de este código provocando que las operaciones de actualización y almacenamiento se vuelvan complicadas.

Esta programación organiza cada código desde una base de procedimiento que no resultan más que funciones reutilizables y pequeñas, organizadas en módulos que agrupan funciones relacionadas un mismo objetivo. Está enfocado también a construir un sistema basado desde un punto de vista en donde las operaciones y procedimientos deben ser realizados en base a un flujo de datos.

Se establecen objetivos desde un programador organizado en códigos y módulos de función con el fin de dividir la complejidad que posee un sistema grande en unidades y convertirlas en pequeños grupos, pero utilizando códigos y estableciendo ciertas divisiones claras para cada sección del programa lo que nos dé como resultado facilitar las nuevas funcionalidades. Dentro de este paradigma no se conoce que exista una protección clara de la modificación de los procesos y es por esto que se da la mutación de variables, esto genera un problema de inseguridad en los programas, esto se puede evitar manteniendo los conceptos claros y evitando el uso de las variables de manera global.

Para desarrollar la aplicación del paradigma procedimental se pueden encontrar varios lenguajes de programación para este paradigma, el más utilizado es Python que es un lenguaje de programación multipropósito creado a finales de la década de 1980, su creación está centrada en la legibilidad del código y posee ciertas características importantes como: Sintaxis limpia, tipado dinámico, resolución dinámica de nombres y ejecución de código interactivo.

Este multiparadigma soporta diferentes estilos y modificación de código tales como; procedimental, estructurado, funcional y orientado a objetivos, pero Python es principalmente bueno en la implementación del paradigma procedimental ya que su sintaxis y flexibilidad da crecimiento notorio a la industria global.

Programación modular

La programación modular es clasificada como un subtipo del paradigma de programación imperativo y es muy similar al enfoque procedimental, incluso se puede decir que se adapta a sus requerimientos en los proyectos de software más amplios ya que el código fuente de esta se da específicamente en bloques lógicos y parciales y a su vez son independientes unos de otros para de esta manera se pueda proporcionar una transparencia que nos permita hacer más fácil el proceso de debugging. Los bloques

parciales de esta programación son denominados módulos, al poderse probar de manera separada antes de ser vinculados en una aplicación. (Silva, 2019)

La programación modular es la descomposición de un programa en partes pequeñas que son denominadas módulos o subprogramas y cada uno de ellos es encargado de llevar a cabo una tarea en específica que se agrupa según su funcionalidad cada uno de estos módulos es analizada y codificado de manera separada. La estructura de un programa modular consta de un módulo principal desde dónde se llama el resto de módulos ese módulo principal tiene el control en el inicio de la ejecución del programa hasta que se implemente el módulo concreto.

Paradigmas de programación declarativa

Este paradigma es una aproximación a la resolución de diferentes problemas desde una programación imperativa. Dentro de este paradigma un programa es una teoría de un espacio lógico apropiado, en dónde la programación declarativa nos da un nivel de abstracción mayor que nos deja como resultado programadores que se concentran a declarar aquello que se debe calcular sin necesidad de detallar cómo debe ser calculado.

Los paradigmas declarativos están basados en desarrollar programas dando un enfoque específico de un conjunto de proposiciones o afirmaciones que caractericen el problema y así plantear una solución. Desde esta conceptualización se puede decir que el paradigma declarativo utiliza mecanismos internos de control con el que se evalúan y relacionan las especificaciones planteadas a fin de obtener la solución. No necesariamente se hace uso de los pasos a seguir para dar una respuesta a la problemática, pero es importante que se cumplan ciertos algoritmos.

Se puede decir que la característica fundamental de este paradigma es que no existe una asignación o cambio del estado de un programa ya que las variables pueden

ser “identificadores con valores que no cambian o modifican toda la evaluación ya que solo existen valores y expresiones que nos devuelven nuevos valores a partir de los que ya existen” (Mitchel, 2018). Los paradigmas permiten utilizar variables que dejen almacenar valores intermedios. Con implementación de estos paradigmas al principio es necesario eliminar la necesidad de probar Buscar el ¿Qué? es necesario para dar la solución.

Este paradigma consiste básicamente en decirle a un programa lo que tiene que hacer, más no como debería hacerlo. Esto mediante la implementación de un lenguaje específico de dominio que permite expresar lo que el usuario quiere, al mismo tiempo que materializa el resultado final deseado, a pesar de que la programación declarativa tiene ventajas en un enfoque imperativo no es tan directa como parece pues al analizar las herramientas declarativas podemos encontrar las dificultades de esta. (Pereiro, 2020)

Programación lógica

La programación lógica está enmarcada como una programación predicativa basada en lógica matemática, lo cual en vez de degenerar una sucesión de instrucciones este software contiene un conjunto de principios con los que se pueden entender o definir como una recopilación de suposiciones y hechos. Todas las solicitudes que se desarrolla en este programa son procesadas de manera que el intérprete o usuario recurre a los principios de aplicar las reglas previamente definidas y de esta manera podrán alcanzar el resultado deseado.

Esta programación se basa en la lógica de primer orden punto las cosas son declaradas en lo que hay que hacer y según la ejecución se define la mejor manera de realizarlas también se organiza de manera coherente en todas las instituciones que permitan alcanzar un objetivo que se ha fijado de la manera más rápida y eficiente. (Castro & Lozano, 2018) Esta programación es la menos utilizada actualmente ya que se ha

impuesto el modelo de programación orientada a objetivos dónde están basados los lenguajes de programación actual.

Programación funcional

La programación funcional está integrada por la función concatenada en dónde cada parte del programa es definido como una función en específico en la que se puede adoptar diferentes estructuras. Es decir, pueden ser vinculadas entre sí como datos o ser utilizadas en forma de parámetros. Es muy importante para la informática en general y también cuenta con aplicaciones versátiles en donde el tratamiento de las funciones de las aplicaciones permite a los programadores trabajar de manera funcional para comprender y aplicar diferentes normas de cálculo para alcanzar nuevas funciones.

La programación funcional es una fórmula cual se puede resolver diferentes problemáticas este paradigma permite desarrollar funciones para evitar que los datos sean mutables también las mismas funciones serán tratadas con total cuidado para que sean asignadas a variables que permitan ser utilizadas como entrada y salida de muchas funciones. Este paradigma de programación se enfoca en el que se está haciendo y verifica lo que se está expresando desde la lógica para no utilizar ciclos o condicionales

Relación y comparación entre paradigmas

La programación imperativa es una contraposición a la programación declarativa ya que presenta una aproximación diferente a la resolución de problemas dejando así un menor nivel de abstracción y un elevado número de libertades en el programador también es considerada como una de las escuelas de pensamiento mucho más importante pues según su clasificación es posible que sea uno de los paradigmas dominantes tomando en cuenta todos los resultados de las encuestas en base a el uso de paradigma más frecuente.

Mediante la relación de los dos paradigmas principales se puede elaborar una comparación estructurada en base a los argumentos de (Ionos, 2020);

Paradigma de programación imperativa	Paradigma de programación declarativa
<ul style="list-style-type: none"> • Clásico • ¿Cómo? • Su nombre proviene del latín imperare, lo cual significa ordenar. • Se dice que los lenguajes de programación típicos son Pascal, ALGOL y Fortan. • En este paradigma se busca determinar la vía de solución deseada. • En este paradigma el programa consta de instrucciones sucesivas que va a indicar al ordenador todo lo que debe computar y ejecutarlo en secuencia 	<ul style="list-style-type: none"> • Nueva corriente • ¿Qué? • Su nombre proviene del latín declarare que significa denominar. • Se dice que los lenguajes de programación típicos son Lisp, Haskekk, ML, Prolog y Oz. • En este paradigma se predetermina aquel resultado deseado. • El programa creado en base a este paradigma va a constar de instrucciones y el cual se Determine la manera de proceder el programa Tuvo una petición por los cálculos pueden ser llevados a cabo mediante una manipulación de valores y un proceso de control

Tabla 1-Heidy Márquez

Desarrollo de software

Según (Montero et all, 2018) El desarrollo de software es definido como un conjunto de actividades informáticas que se dedican al proceso de diseño, creación, despliegue y compatibilidad de un software. Por otro lado, el software es el conjunto de programas o instrucciones mediante el cual la computadora va a saber qué hacer. El

proceso de desarrollo de software es también conocido como el ciclo de vida de un software que se estructura de manera aplicada en el desarrollo de un producto en el cual hay diferentes modelos a seguir que permiten el establecimiento de un proceso con los cuales se describe su información que difiere en actividades que tengan lugar en el proceso

Dentro del proceso de desarrollo de software se dan diferentes actividades, la planificación es una de ellas lo que es una importante tarea en el momento de crear un producto que permite obtener requisitos y el análisis de los mismos. Una vez que se ha recopilado los requisitos del cliente en el software deseado se le llama a este documento especificación funcional. Otra actividad es la implementación y pruebas lo cual conlleva un proceso en el que los ingenieros programan el código de trabajo en relación a las demandas requeridas estas pruebas realizadas son de esencial importancia dentro del proceso de desarrollo del software porque va a permitir detectar los errores y corregirlos a tiempo. Está también influye en la elaboración de otros proyectos.

El despliegue y mantenimiento es una actividad que se da cuando el código está probado de la manera adecuada y ha sido aceptado de acuerdo a su producción. El mantenimiento con problemas recientes no requiere de más tiempo en el desarrollo inicial del proceso y es posible que se puede incorporar el código para hacer que se ajuste a un diseño original a fin de evitar costes demasiado altos. Es importante conocer que un modelo de desarrollo de software puede ser definido como una representación abstracta en una manera particular que no representa encía un software sino un enfoque general y como que evidentemente puede ser modificado y presentado en adaptaciones de acuerdo a las necesidades en el desarrollo de un software punto existen muchos modelos, pero pues necesito saber cuál es el más adecuado.

El desarrollo de software tiene la intervención de varias personas cómo es el cliente que presenta el requerimiento el analista de sistema y el profesional en el área que es quién va a desarrollar el sistema de software para dar solución a la problemática presentada por el cliente. Es un desarrollo complejo por lo que hoy en día se presenta

mucha atención para ejecutarse aun cuando se puede tratar de proyectos de menor categoría.

Fases del desarrollo de software

Para desarrollar un programa de software es necesario seguir ciertos procesos y fases que van a permitir llevar un orden adecuado;

Análisis de requisitos

Permite traer los requisitos del producto dentro de la primera etapa para poder crearlo punto consiste en analizar los requerimientos del usuario para poder dar solución a su problema y en base a los resultados del Análisis se plasma el documento en dónde se va a especificar el tipo de Software que necesita. Dentro de esta área se han ideado modelos para una adecuada elección y arquitectura esta parte se refiere a determinar la manera en la que funciona los programas para considerar así la implementación tecnológica la red el Hardware entre otros. También Define los casos para cubrir las funciones y transformar las entidades definidas en el análisis de requisitos.

Pruebas

Está consiste en realizar las pruebas más adecuadas y suficientes para identificar los errores y corregirlos a tiempo. Generalmente hay dos formas de organizar las áreas de pruebas una que está compuesta de personal inexperto para que va lo de la documentación y otra que esté en un área conformada por programadores con experiencia para verificar que no existan fallas.

MARCO METODOLOGICO

Metodología de la investigación

La metodología de investigación aplicada para la obtención de resultados y análisis fue; investigación deductiva que nos permitió tener un mejor enfoque y análisis de la temática abordada, así como llegar a conclusiones mediante el análisis de datos; se aplicó el método cualitativo.

Método cualitativo

El método cualitativo dentro del presente estudio de caso permite recoger y trabajar los datos obtenidos en diferentes fuentes como lo son tesis y revistas o trabajos presentados desde fuentes oficiales dicha información ha sido interpretada para poder presentar los datos que lleguen a una comprensión correcta del estudio puesto en consideración. En este método se realiza un análisis de texto y la comparación entre diferentes conceptualizaciones encontradas en la investigación.

Modalidad de la investigación

La presente investigación se da mediante la recolección de datos en donde los sujetos participantes (investigadores) ponen en consideración la información obtenida, analizándola y relacionándola con la temática abordada para de esta manera dar desarrollo al tema y cumplimiento a los objetivos planteados. La recolección y análisis de información se da mediante fuentes oficiales en las que se ha dado desarrollo de manera breve a lo que se desea investigar. Es importante tener en cuenta que el presente estudio de caso tiene como finalidad, determinar la relación entre los paradigmas existentes y su influencia en los programadores de software por lo que en el sustento teórico se desarrollan cada uno de estos temas.

Tipo de investigación

Investigación descriptiva

La investigación descriptiva nos permite realizar un informe sobre la temática abordada, en este caso los paradigmas y la influencia que tienen en el desarrollo de programas de software, así como sus características y de esta manera poder llegar a obtener información clara sobre nuestro objetivo de estudio. También va a permitir puntualizar el contexto de estudio y describir sus procesos.

Métodos, técnicas e instrumentos

Método

La aplicación del método hace referencia a los procedimientos que se llevan a cabo en la investigación de tal manera que se recurre a la atención de necesidades descritas en el proceso del presente estudio de caso, es decir se indaga la realidad en la que se plantea el trabajo. En el presente proyecto de investigación se aplica el método analítico en donde se definirán conceptos a partir de la observación de los hechos que se puedan detallar mediante la búsqueda de información en fuentes confiables, tesis, revistas, entre otras.

Técnica

Encuesta

Mediante la elaboración de una encuesta se ha podido explorar de manera directa con los sujetos en donde se desarrolla la investigación, a fin determinar la relación que existe entre los paradigmas de programación y la influencia en el desarrollo de un proyecto de software.

Instrumento

cuestionario

En el cuestionario se ha analizado la información obtenida para plantearnos interrogativas que den un mejor alcance de los paradigmas de programación y la influencia que tienen en el desarrollo de un proyecto de software.

RESULTADOS

En la elaboración del presente estudio de caso se ha utilizado la investigación deductiva que permite tener un mejor enfoque de la temática abordada, también se ha hecho uso del método cualitativo con el que se pueden analizar todos los datos adquiridos en las diferentes fuentes de investigación. Se ha utilizado un tipo de investigación descriptiva para poder desarrollar los conceptos necesarios dentro del estudio de caso también se realizó una encuesta mediante un cuestionario con el fin de obtener información directa de los sujetos en dónde se desarrolla la investigación y así obtener conclusiones concretas y verificables

Con la implementación de la metodología mencionado y la aplicación de la encuesta se pudo evidenciar que el mayor porcentaje considera el paradigma de programación como un tema relevante dentro de la carrera de ingeniería en sistemas por lo que se considera que la temática abordada es de interés para la comunidad educativa. Los paradigmas existentes dentro de la programación son de Gran importancia sin embargo no todos tienen conocimiento de su existencia o definición

Uno de los principales pasos para la creación de un software es escoger adecuadamente el paradigma, en encuesta realizada se evidenció que el mayor porcentaje de estudiantes encuestados no es capaz de identificar el programa adecuado para escoger en la creación de un Software. A pesar de ser estudiantes de la carrera de ingeniería se evidenció un alto índice de estudiantes que no se creen capaces de conocer los tipos de paradigmas existentes por lo que dentro de un proceso de enseñanza y aprendizaje se debe tener en cuenta los conocimientos que poseen los estudiantes para poder implementar medidas que es regresión en el sentido de la carrera

DISCUSIÓN DE RESULTADOS

Es importante tener en cuenta toda la información recopilada durante la elaboración presente estudio de caso y los resultados obtenidos mediante la aplicación del marco metodológico

Se puede decir que los estudiantes de la carrera de ingeniería sistema tienen el total conocimiento de la importancia que tienen los paradigmas de programación en la creación de un software sin embargo no poseen los conocimientos necesarios sobre el tema esto debido a múltiples factores tales como poca importancia, poco interés por parte de los docentes, falta de tiempo o las medidas tomadas actualmente por el covid-19. Aun teniendo en cuenta todas las posibilidades Dentro de esta problemática se considera responsabilidad por parte del estudiante para no buscar información que le sea relevante al aspecto a los paradigmas de programación para de esta manera generar en ellos mismos un aprendizaje más práctico que le permita enfrentarse al mundo laboral sin inconvenientes

Cuando una persona tiene como finalidad crear un software y a la vez no es capaz de elegir el adecuado paradigma va directamente en su cado hacia el fracaso de su proyecto, por lo que se debe analizar todas las opciones antes de tomar medidas precipitadas. Actualmente la ampliación y creación de programa de software es una de las actividades más relevantes dentro del campo de sistemas puesto que el usuario generalmente solicita ciertos detalles son relevantes y facilitaría sus actividades por lo que le son necesarias dentro de un programa para así llegar a un buen análisis de la información.

Los programas de software en la actualidad son de Gran relevancia tanto para ámbitos educativos o de trabajo por lo que es necesario que la comunidad tenga conocimiento de su manejo y cada vez más se creen profesionales especiales en el área con el fin de ofrecer al mercado servicios de calidad en menor tiempo tal como lo es un servicio de software.

Dentro de Universidad Técnica de Babahoyo en la carrera de ingeniería en sistemas se puede evidenciar la constante enseñanza sobre los paradigmas de programación ya que se convierten en un Pilar fundamental para la creación de un programa por lo que es de suma importancia de la carrera y mientras más fácil y rápido se tenga conocimiento esta información será más fácil obtener nuevos contenidos.

Un programa de software posee lenguajes complejos por lo que se convierte en una temática un poco difícil de comprender sin embargo es necesario realizar las investigaciones necesarias para así llegar a conceptualizaciones que permitan un enfoque específico

CONCLUSIONES

En relación a la encuesta realizada a los estudiantes de la carrera de ingeniería en sistemas y abordando en la temática del paradigma de programación como un tema relevante a tratar en esta carrera se obtuvo como resultado una totalidad que si estaba de acuerdo ya que no sé presentaban dentro de las clases las temáticas necesarias para llegar a una buena comprensión del tema.

Se pudo identificar que el mayor porcentaje de estudiantes encuestados no son capaces de escoger el paradigma adecuado para desarrollar un programa de software lo que se convierte en un gran problema siendo está la carrera encargada de explicar esos procesos.

El mayor porcentaje de personas encuestadas afirmó que los programas de software contienen códigos y lenguajes complejos por lo que resulta una parte complicada del aprendizaje de los estudiantes en esta temática sin embargo se convierte en una de las más importantes para poder desarrollar un programa de software.

RECOMENDACIONES

Cómo medida pertinente para dar solución a la problemática detectada mediante una encuesta realizada a estudiantes de la carrera de ingeniería en sistemas se plantea medidas que integre en los sílabos de los docentes la aplicación de la temática de paradigmas de programación como uno de los temas más relevantes dentro de cierta unidad para que de esta manera los estudiantes tengan una mejor comprensión de la temática.

Se recomienda implementar medidas por parte de los directivos de la carrera para que gestionen una evaluación continua respecto a los conocimientos de los estudiantes en los paradigmas de programación para así asegurarse de que ellos son una necesidad de aprender y comprender dicha temática que les permitirá desarrollar programas de software y reducir la medida de fracasos de los mismos.

Se recomienda al personal competente implementar medidas que permitan a los estudiantes una mejor comprensión y aprendizaje de los códigos y lenguajes que se utilizan en los programas de software utilizando estrategias de enseñanza para que mediante la práctica y estudio se obtengan estudiantes capaces de manejar estas herramientas de software con facilidad.

REFERENCIAS

- Angarita, L. B., Taborda, A. P., & Márquez, J. D. (2020). *videojuego para el aprendizaje de logica* . Obtenido de Revista educacion en ingenieria : <https://educacioneningenieria.org/index.php/edi/article/view/1141/1002>
- Castro, M., & Lozano, I. (2018). *Programación con logica de terminos*. Obtenido de UPAEP Mexico: https://res.cic.ipn.mx/2018_147_6/TFLPL_%20Programacion%20con%20logica%20de%20terminos.pdf
- Dimaggio, M. (2019). *¿que son paradigmas de programacion?* Obtenido de <https://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>
- Dominguez, I. (Abril de 2005). *Fundamentos teóricos de los paradigmas de programacion*. Obtenido de academia Accelerating the world's reserve: <https://d1wqtxts1xzle7.cloudfront.net/42704971/FundamTeoricosParadigProg-with-cover-page-v2.pdf?Expires=1646157092&Signature=C7P~Vy8cWU~AUuZPXI~NL-rR9TAfHXpRHs4d-JZqc-UimA0kKphov6NoWkoZ5y4-4zuvG8SRBGqgppxhnCcNN-zo61L-FindRUPpB3Ua-jgPILh3U49WALtfvoEAW~cjIA>
- Giusti1, L., Sanz, V., & Giusti, A. (25 de noviembre de 2019). *Evolución de un curso inicial de programación a un enfoque multiparadigma. Análisis y resultados*. Obtenido de Universidad Nacional de La Plata, La Plata, Argentina: <https://teyet-revista.info.unlp.edu.ar/TEyET/article/view/1308/975>

- Heredia, S. (2018). *Generador de Código Utilizando el Paradigma de Líneas de Producto Software*. Obtenido de Revista Científica Hallazgos 21, 3,(2), 190- 212. :
file:///C:/Users/LAPTO/Downloads/Dialnet-
GeneradorDeCodigoUtilizandoElParadigmaDeLineasDePr-7148203.pdf
- Herrera, B., & Alexander, W. (2020). *estudio comparativo entre paradigmas de programacion y su influencia en un proyecto de desarrollo de software*. Obtenido de Universidad de las fuerzas armadas:
<http://repositorio.espe.edu.ec/bitstream/21000/22123/1/T-ESPE-043669.pdf>
- Ionos. (2020). *Paradigmas de programacion, principios basicos de programacion*. Obtenido de Digital Guide: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/paradigmas-de-programacion/#:~:text=El%20paradigma%20de%20programaci%C3%B3n%20procedimental,como%20subprogramas%2C%20rutinas%20o%20funciones>.
- Mitchel, J. (2018). *Características de la programacion declarativa* . Obtenido de Universidad de Alcalá:
<https://rua.ua.es/dspace/bitstream/10045/4032/1/tema02.pdf>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). *Metodologías ágiles frente a las tradicionales*. Obtenido de Revista multidisciplinaria:
https://www.researchgate.net/profile/Harry-Vite-Cevallos/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software/links/5b942061a6fdccfd542a2b13/Metodologias-agiles-frente-a-las-tradicionales-en-el-proce
- Pereiro, F. (2020). *La Programación Declarativa ¿De Verdad Existe?* Obtenido de Developers: <https://www.toptal.com/software/la-programacion-declarativa-de-verdad-existe>

Silva, C. V. (2019). *En Búsqueda de un Procedimiento de Desarrollo de Software*.

Obtenido de Universidad Villa del Mar:

<https://scielo.conicyt.cl/pdf/infotec/v30n3/0718-0764-infotec-30-03-00095.pdf>

ANEXOS

TABULACIÓN TOTAL DE LOS RESULTADOS

Tabla 2

¿Cree usted que el paradigma de programación es un tema relevante a tratar en la carrera de ingeniería en sistemas?

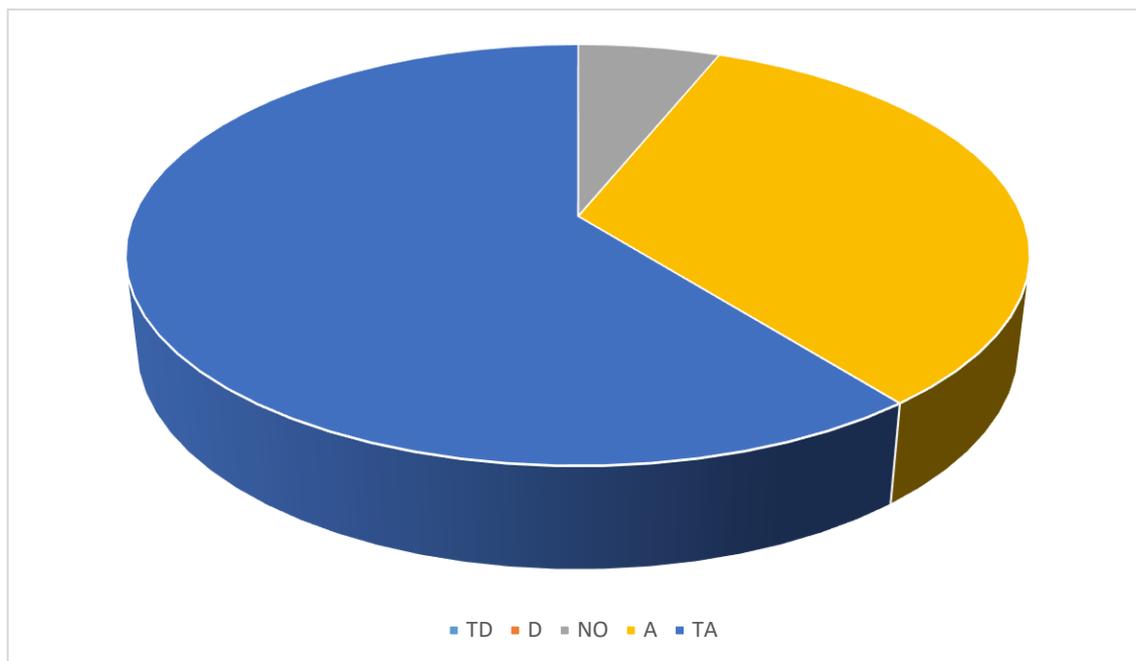
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	0	0%
D	0	0%
NO	2	6%
A	10	33%
TA	18	60%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 1

¿Cree usted que el paradigma de programación es un tema relevante a tratar en la carrera de ingeniería en sistemas?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 2

¿conoce Ud. los tipos de paradigmas existentes?

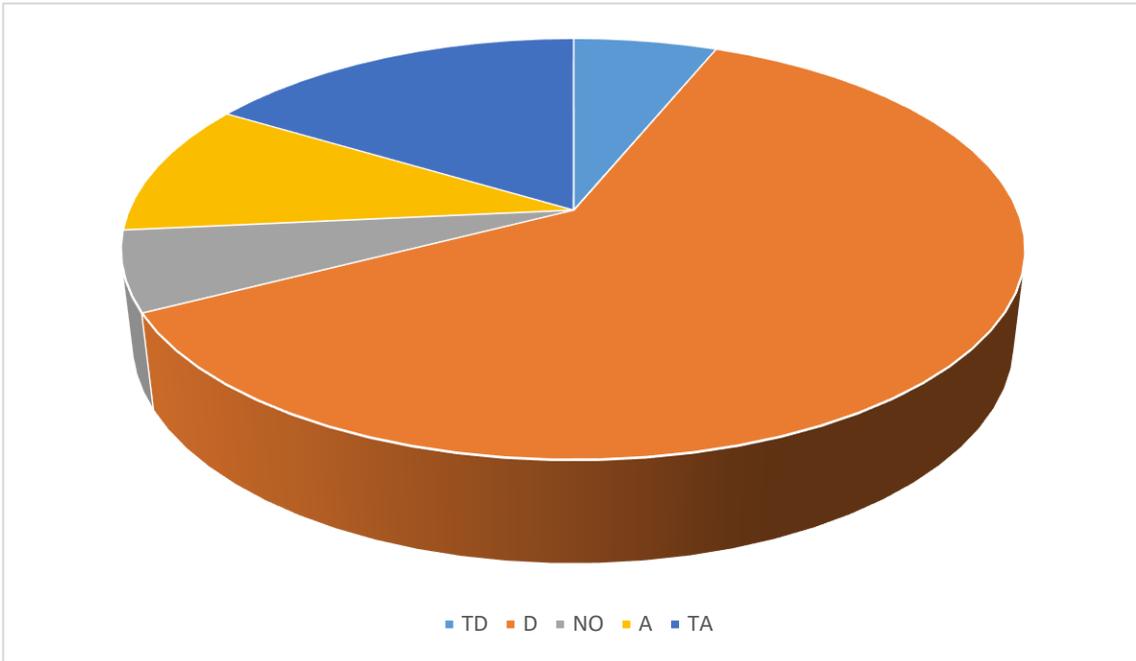
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	2	6%
D	18	60%
NO	2	6%
A	3	10%
TA	5	16%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 2

¿conoce Ud. los tipos de paradigmas existentes?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 3

¿Cree usted que puede identificar que paradigma escoger en la creación de un programa de software?

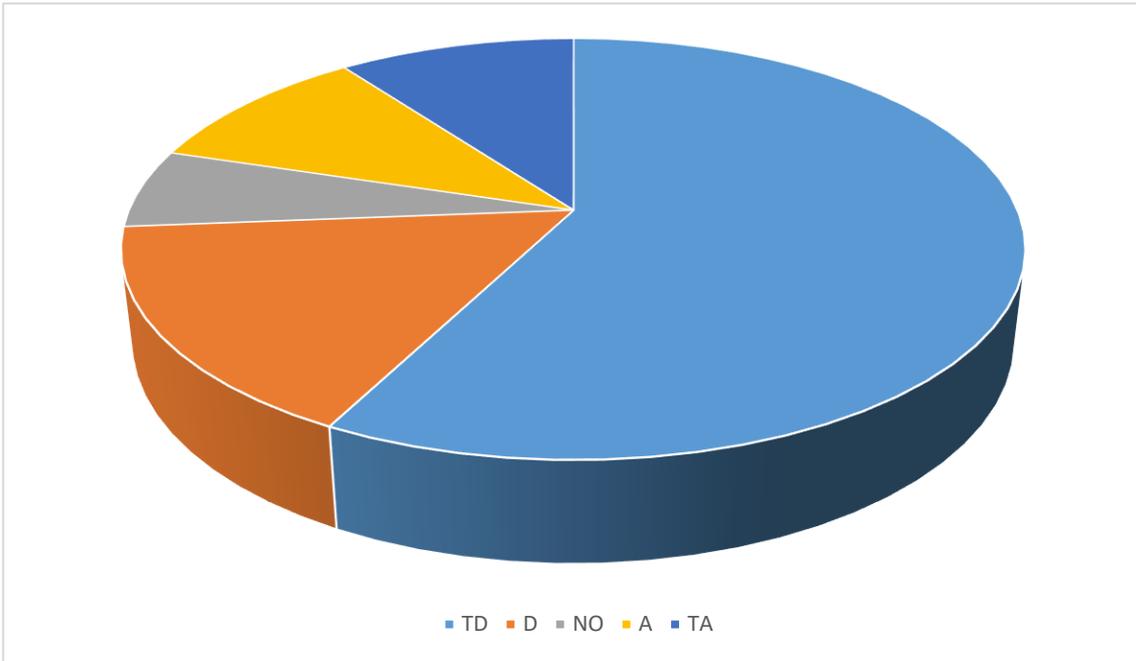
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	17	57%
D	5	16%
NO	2	6%
A	3	10%
TA	3	10%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 3

¿Cree usted que puede identificar que paradigma escoger en la creación de un programa de software?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 4

¿considera usted que posee el conocimiento adecuado sobre los tipos de paradigmas existentes?

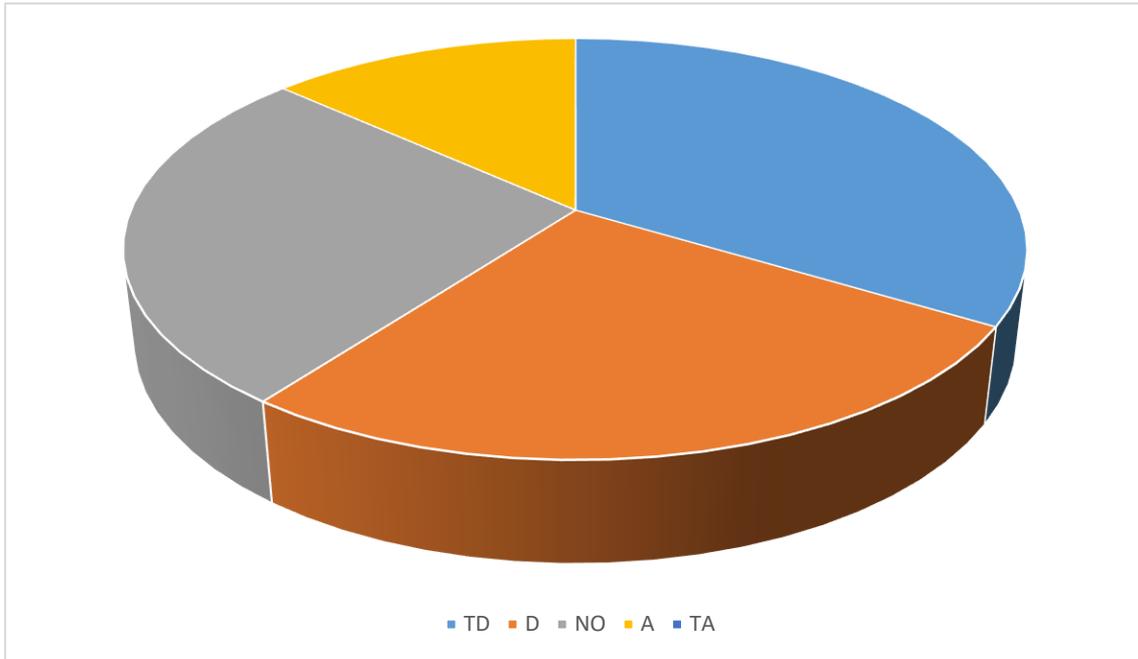
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	10	33%
D	8	26%
NO	8	26%
A	4	13%
TA	0	0%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 4

¿considera usted que posee el conocimiento adecuado sobre los tipos de paradigmas existentes?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 5

¿está usted de acuerdo en que los programas de software son relevantes en la vida cotidiana?

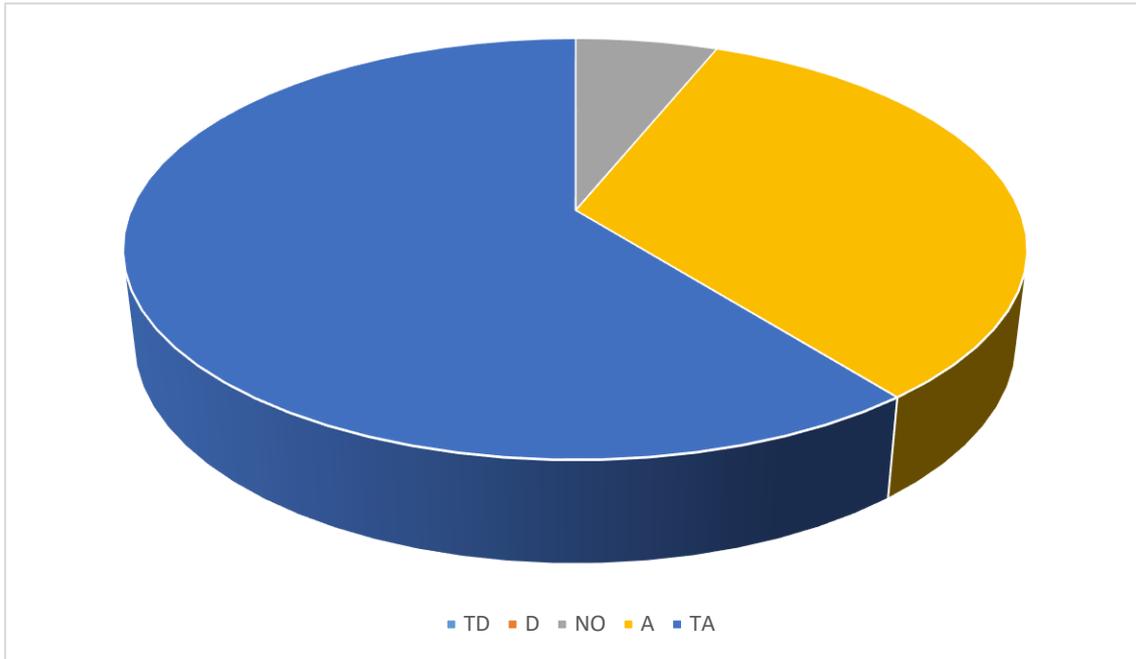
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	0	0%
D	0	0%
NO	2	6%
A	10	33%
TA	18	60%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 5

¿está usted de acuerdo en que los programas de software son relevantes en la vida cotidiana?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 6

¿está usted de acuerdo en que dentro de los programas de estudio de la universidad técnica de Babahoyo se planteen las diferencias de los tipos de paradigmas?

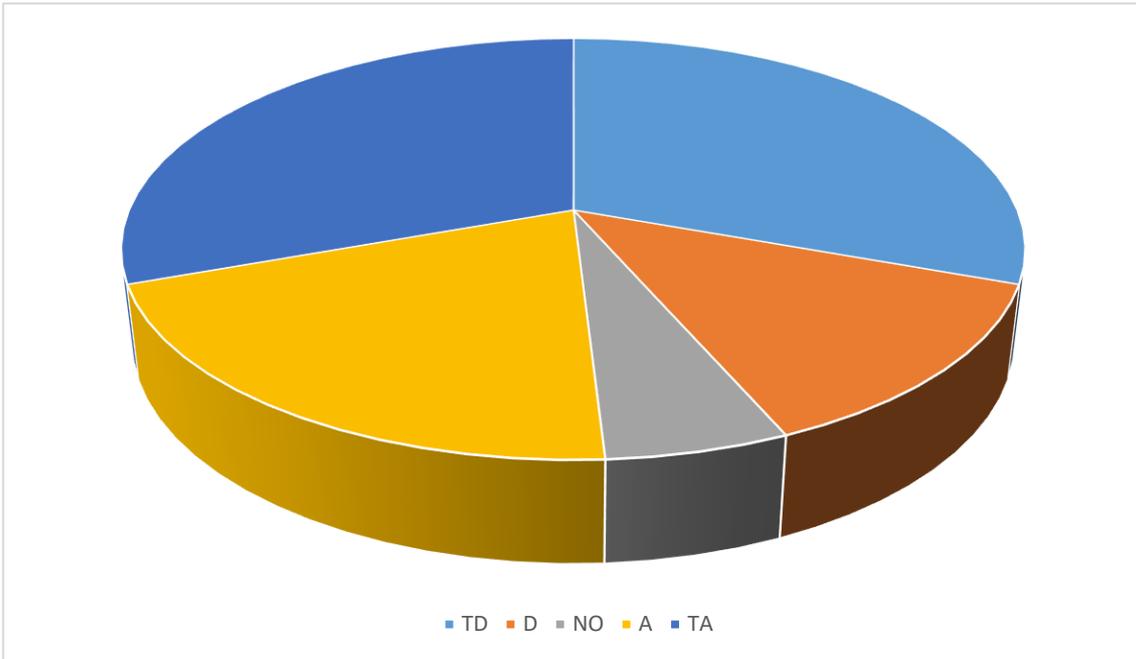
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	10	33%
D	4	14%
NO	2	6%
A	10	22%
TA	4	33%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 6

¿está usted de acuerdo en que dentro de los programas de estudio de la universidad técnica de Babahoyo se planteen las diferencias de los tipos de paradigmas?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 7

¿cree usted que el programa de software tiene códigos y lenguajes complejos?

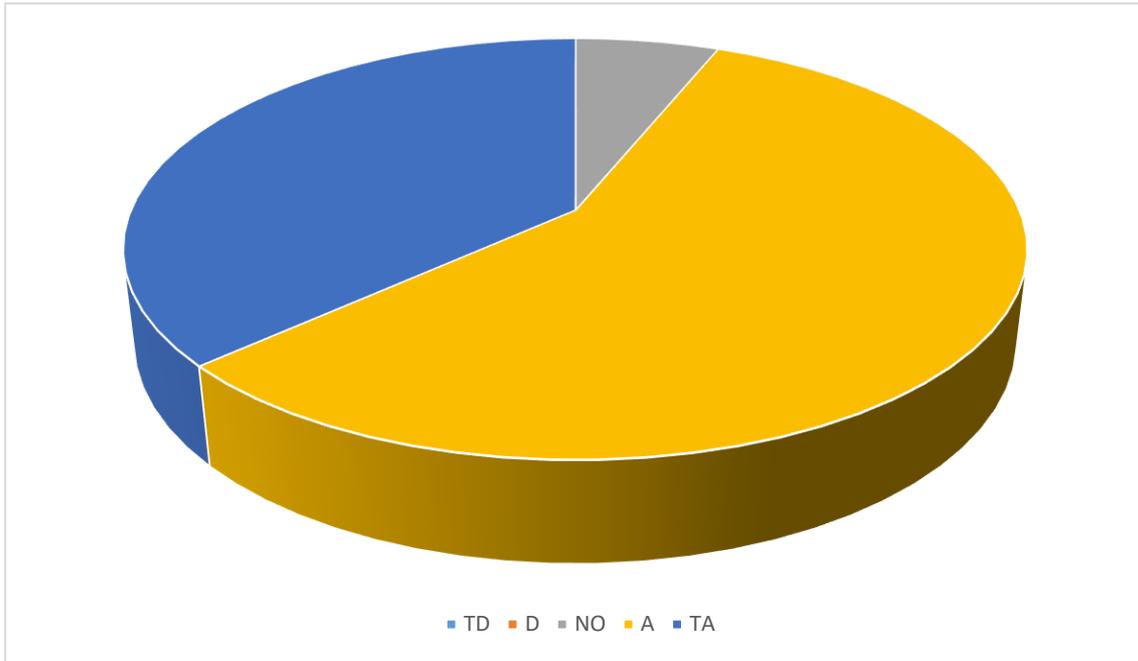
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	0	0%
D	0	0%
NO	2	6%
A	17	56%
TA	11	36%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 7

¿cree usted que el programa de software tiene códigos y lenguajes complejos?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Tabla 8

¿esta usted de acuerdo en asistir a conferencias con el fin de recibir capacitaciones sobre los paradigmas y su relacion con los programas de software?

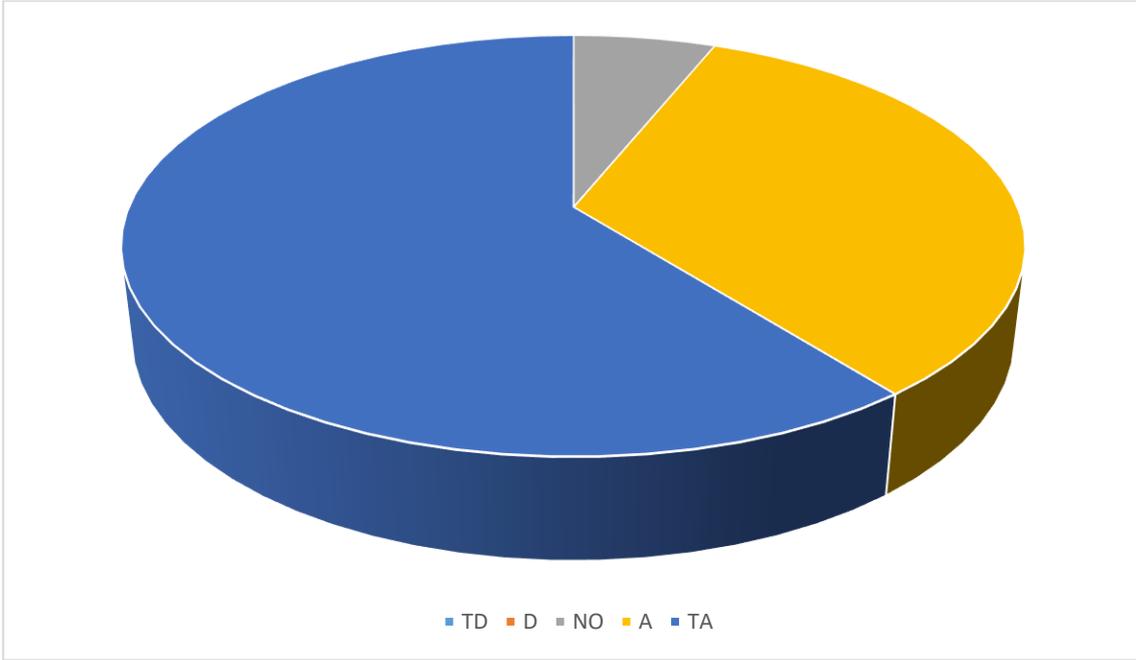
ALTERNATIVA	FRECUENCIA	PORCENTAJE
TD	0	0%
D	0	0%
NO	2	6%
A	10	33%
TA	18	60%
Total	30	100%

Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Figura 8

¿esta usted de acuerdo en asistir a conferencias con el fin de recibir capacitaciones sobre los paradigmas y su relacion con los programas de software?



Elaborado por: Márquez Heidi.

Fuente: Encuesta a estudiantes de la carrera de ingeniería en sistemas

Formato de encuesta

PREGUNTAS	TD	D	NO	A	TA
	1	2	3	4	5
1. ¿Cree usted que el paradigma de programación es un tema relevante a tratar en la carrera de ingeniería en sistemas ?					
2. ¿conoce Ud. los tipos de paradigmas existentes?					
3. ¿Cree usted que puede identificar que paradigma escoger en la creación de un programa de software?					
4. ¿considera usted que posee el conocimiento adecuado sobre los tipos de paradigmas existentes?					
5. ¿está usted de acuerdo en que los programas de software son relevantes en la vida cotidiana?					
6. ¿está usted de acuerdo en que dentro de los programas de estudio de la universidad técnica de Babahoyo se planteen las diferencias de los tipos de paradigmas?					
7. ¿cree usted que el programa de software tiene códigos y lenguajes complejos?					
8. ¿está usted de acuerdo en asistir a conferencias con el fin de recibir capacitaciones sobre los paradigmas y su relación con los programas de software?					

Encuesta aplicada mediante google formulario

ENCUESTA DIRIGIDA A LOS ESTUDIANTES DE INGENIERIA EN SISTEMAS

Descripción del formulario

1. ¿Cree usted que el paradigma de programación es un tema relevante a tratar en la carrera de ingeniería en sistemas ?

- TOTALMENTE EN DESACUERDO
- DESACUERDO
- NO OPINA
- ACUERDO
- TOTALMENTE DE ACUERDO

2. ¿conoce ud los tipos de paradigmas existentes?

- TOTALMENTE EN DESACUERDO
- DESACUERDO
- NO OPINA
- ACUERDO
- TOTALMENTE DE ACUERDO



Evidencia de tutorías

