Diferencias entre el paradigma orientado a objetos y el paradigma funcional

Maximiliano Moraga v.

Informe para DHC

Contenido

[Introducción 3](#_Toc10449408)

[Historia 4](#_Toc10449409)

[Paradigma de la programación funcional. 4](#_Toc10449410)

[Paradigma de la programación orientada a objetos. 5](#_Toc10449411)

[Diferencias 7](#_Toc10449412)

[Conclusión 8](#_Toc10449413)

[Referencias 9](#_Toc10449414)

# Introducción

En el centro de cómputo en Oslo, se establecen los primeros conceptos de POO (Programación Orientada a Objetos). En 1962, Simula es el primer lenguaje de POO. Varios años después gran parte de los lenguajes comienzan a hacer uso de este estándar emergente.

POO se fue convirtiendo poco a poco en un estándar en el ámbito de la programación, a mediados de los años 80, C++ uno de los lenguajes más utilizados para el desarrollo de software adopta este paradigma.

El presente Informe pretende realizar una comparativa directa entre el paradigma orientado a objetos y el paradigma funcional.

La finalidad es dar a conocer al estudiante, que se encuentra en la transición entre estos paradigmas, dando una perspectiva a este que no se menciona a menudo.

Este trabajo se dividirá en cuatro partes principales.

En la primera parte se presentarán antecedentes históricos.

En la segunda parte se describirán las diferencias en una tabla de comparación directa.

Finalmente, en la tercera parte se presentarán las conclusiones del informe más anexos.

El propósito de este informe es esclarecer las diferencias entre estas dos formas de programar.

En el informe se presentan las características de ambos paradigmas, con el objetivo de que el estudiante se sienta más cómodo con el cambio en la forma de programar, para que sea capaz de utilizar la forma de programar.

# Historia

## Paradigma de la programación funcional.

A mediados de los años 50s, cuando aún no existían los computadores como los conocemos hoy, existía un auge por la Inteligencia Artificial (IA), como conceptualización, la cual se interesaba por el campo de la psicología, la lingüística, la matemática entre otros.

Al hablar de lenguajes de programación, se debe considerar lambda como uno de los primeros lenguajes funcionales desarrollado a principio de los años 30s, el cual consistía en el cálculo aritmético de procesos matemáticos, que se utilizaba principalmente en el área del análisis numérico.

Cuando hablamos de un paradigma funcional es necesario conocer como funciona un proceso directamente en el procesador, por esto es por lo que los lenguajes funcionales evolucionaron, y pasaron a ser un tanto mas declarativos, en el proceso de ejecución, los cuales incorporaron variables que son funcionalidades dentro del lenguaje, que puede variar su estado o valor según el momento en el tiempo de ejecución de la aplicación. Para poder implementar el concepto de variable en tiempo de ejecución fue necesario incorporar tipos de datos que identificaban como se comportaría esta variable que podía cambiar su estado.

De ese momento en adelante ocurrió un cambio y la programación funcional cambio su objetivo, y se comenzó a llamar programación estructural, la cual usaba los algoritmos o pseudo lenguaje para una finalidad mas educativa, que tiene que ver mas en como funcionan los procesos, estructuras de programación, variables, para que los estudiantes comprendan y adquieran herramientas o habilidades para poder desarrollar software.

La programación estructural define una aplicación como una seria de pasos lógicos y finitos, para poder generar una solución a un problema, pero esto tiene restricciones en ocasiones no satisfacen las necesidades de la solución, como por ejemplo cuando se requieren estructuras mas apegadas a la realidad como los objetos que poseen atributos, tenemos por un lado los lenguajes estructurales que pueden manejar datos del tipo enteros, letras, listas, entre otras. En cambio, con el objeto podemos tener un único objeto que contiene diferentes datos como una persona, que posee datos abstractos como lo puede ser un nombre o un dato funcional como lo puede ser caminar.

## Paradigma de la programación orientada a objetos.

Bien como lo mencionamos en la introducción, Simula fue el primer lenguaje orientado a objetos, el cual se desarrolló con la necesidad de generar una solución a la simulación de naves aéreas, las cuales usan demasiados factores agrupados en nodos para poder funcionar. Generar una solución en el paradigma funcional se hace un infierno al intentar trabajar tantos datos simultáneamente.

(DEITEL, 2008) “(…) la orientación a objetos es una manera natural de pensar acerca del mundo real y describir programas de cómputo”. (p.16)

A principio de los años 70 teníamos un problema, ya que, la mayoría de los sistemas que se planificaban no se llevaban a cabo. Este alto grado de planificación es contrario a la propia forma que tiene el ser humano al momento de aprender y crear que consiste en el ensayo y error. La programación orientada a objetos puede salta el proceso de planificación, al generar un código modularizado (segmentado por partes), no es necesario saber cómo quedara el producto final.

Aun que se hable de paradigma orientado a objetos, en realidad este tipo de programación estaba en pañales con solo sus clases y objetos. Estas clases consistían en crear una especie de plantilla del objeto que se deseaba utilizar, definiendo lo que hacia el objeto y los atributos que tenía.

En los años 70 se incorpora al paradigma el concepto de encapsulación, que consistía en ocultar la información del objeto, restringiendo el acceso a los datos propios de este, a menos que se definiera un método que accediera al dato y lo entregue o por el contrario un método que asignara un valor al dato propio del objeto. Esto genera una seguridad en los datos ya que tenemos que acceder a ellos por un tercero.

En los años 80 AT&T amplio el lenguaje C, creando C++, que incorporaba toda la conceptualización teórica que existía hasta el momento. El desarrollo técnico del hardware y su disminución del costo fue el detonante final. Con más computadoras al alcance de más personas más programadores, más problemas y más algoritmos surgieron.

En los 90 se logra asentar la Programación Orientada a Objetos (POO de aquí en adelante), pasando a ser un estándar de la programación, siendo el mejor paradigma para resolver problemas, generar prototipos de programación eficientes y rápidos. Con esto nace el concepto de RAD (Rapid Aplication Developments) aplicación rápida para desarrolladores.

En 1996 surge una herramienta de programación llamada Java que revoluciona completamente, en entorno de programación. Su filosofía es aprovechar el software existente. Facilitar la adaptación de este a otros usos diferentes a los originales sin necesidad de modificar el código ya existente. Java incorpora POO como base de arquitectura para desarrollar aplicaciones, haciendo imposible el desarrollo de aplicación, sin conocer los conceptos teóricos de POO.

# Diferencias

En primer lugar, no existe un estándar internacional cuando se trata de definir un paradigma, por ende, estas diferencias fueron extraídas en base a los libros de la bibliografía.

La gran diferencia entre estos dos paradigmas apunta directamente en la forma en la que se manejan los datos. La programación estructural por un lado se refiere a datos como algo abstracto, que puede variar desde un numero cualquiera hasta un objeto complejo que contiene cualidades o características propias tanto como acciones que son métodos. La programación estructural por otro lado solo controla un flujo de datos en tiempo de ejecución paso a paso.

Una de las características mas grandes entre estos dos paradigmas es el concepto de herencia, ya que es una forma natural de reutilizar código.

Programación Orientada a Objetos Programación secuencial

|  |  |
| --- | --- |
| La programación orientada a objetos es más moderna, es una evolución de la programación estructurada que plasma en el diseño de una familia de lenguajes conceptos que existían previamente con algunos nuevos. | Es compleja la coordinación y organización entre programadores para la creación de aplicaciones de media y gran envergadura. |
| La programación orientada a objetos se basa en lenguajes que soportan sintáctica y semánticamente la unión entre los tipos abstractos de datos y sus operaciones (a esta unión se la suele llamar clase). | Es difícil modificar y extender los programas, pues suele haber datos compartidos por varios subprogramas, que introducen interacciones ocultas entre ellos. |
| La programación orientada a objetos incorpora en su entorno de ejecución mecanismos tales como el polimorfismo y el envío de mensajes entre objetos. | Es difícil mantener los programas. Casi todos los sistemas informáticos grandes tienen errores ocultos, que no surgen a la luz hasta después de muchas horas de funcionamiento. |

# Conclusión

Anteriormente se presentó una comparación directa entre dos paradigmas, que son los principales estándares, en el área de desarrollo de software, el paradigma funcional y el paradigma orientado a objetos.

El presente informe tenía como finalidad, lograr que el estudiante que se encuentra en el proceso de transición entre estos dos paradigmas logre entender las diferencias entre estas dos formas de desarrollar software.

Este trabajo se dividió en cuatro partes las cuales convergen en un solo objetivo, que es dar a conocer las diferencias entre los dos paradigmas mas importantes en el desarrollo de software.

En la primera parte se presentaron los antecedentes históricos, que conllevaron a que estos dos paradigmas sean los mas importantes en esta área de la informática.

En la segunda parte se presentó una tabla comparativa de las funcionalidades de cada paradigma.

Finalmente, en la tercera parte se expone la conclusión, bibliografía y el lenguaje técnico.

Actualmente la orientación a objetos parece ser el mejor paradigma, no obstante, no es una solución a todos los problemas. Trata de eliminar la crisis del software

# Referencias

Aguilar, L. J. (2008). *FUNDAMENTOS DE PROGRAMACIÓN. Algoritmos, estructura de datos.* Madrid.: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

DEITEL, H. (2008). *Como Programar en Java* (Séptima Edición. ed.). Mexico: PEARSON EDUCATION.