Paradigmas de programación

Programming paradigms

Autor 1: brayan de jesus Orrego Autor 2:Angie paola Ramirez Villada

Riisaralda,universidad tecnológica,Pereira
Correo-e: b.orrego1@utp
Correo-e: paola.ramirez2@utp.edu.co

***Resumen*— Un paradigma de programación es un estilo de desarrollo de programas**. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.

¿Cuáles son los principales paradigmas de programación?

- **Imperativo**. Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.
- **Declarativo**. Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.
- **Lógico**. El problema se modela con enunciados de lógica de primer orden.
- **Funcional**. Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.
- **Orientado a objetos.** El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

Palabras claves:
- **algoritmos**
- **computadores**
- **códigos fuentes**
- **lenguajes**

• algorithms

• computers

• source codes

• idioms

I. INTRODUCCIÓN

Hablaremos de los paradigmas de programación ¿ QUE ES ? SUS VENTAJAS, desventajas, lenguajes y daremos algunos ejemplos

Fecha de Recepción: (Letra Times New Roman de 8 puntos)
Fecha de Aceptación: Dejar en blanco

¿ QUE ES ?

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con un conjunto de normas específicas, tales como: estructura modular, fuerte cohesión, alta rentabilidad, etc.

Los paradigmas pueden ser considerados como patrones de pensamiento para la resolución de problemas. Desde luego siempre teniendo en cuenta los lenguajes de programación, según nuestro interés de estudio.

No es mejor uno que otro sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro.

Hay multitud de ellos atendiendo a alguna particularidad metodológica o funcional Cuando un lenguaje refleja bien un paradigma particular, se dice que soporta el paradigma, y en la práctica un lenguaje que soporta correctamente un paradigma, es difícil distinguirlo del propio paradigma, por lo que se identifica con él.[1]

Ventajas

- Reducción de los costos de mantenimiento
- Reducción del esfuerzo en las pruebas y depuración
- Los bloques de código son casi auto-explicativos, lo que reduce y facilita la documentación.
- Se incrementa el rendimiento de los programadores
- Ausencia de efectos colaterales
- Proceso de depuración menos problemático
- Pruebas de unidades más confiables
- Mayor facilidad para la ejecución concurrente
-

Desventajas

- El principal inconveniente de este método de programación es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático el manejo de su código fuente
- Se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar difícil su manejo
- Falta de estandarización
- Bajo rendimiento de los programas
- [1]

Un **lenguaje de programación** es un lenguaje formal (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico y/o lógico de una computadora, de manera que se puedan obtener diversas clases de datos. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa.[1]

Por tanto, programar viene a ser el proceso de crear un software fiable mediante la escritura, prueba, depuración, compilación o interpretación, y mantenimiento del código fuente de dicho programa informático.

II. CONTENIDO

En informática, se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Dicho lenguaje está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de estos procesos creativos.

La implementación de lenguajes de programación permite el trabajo conjunto y coordinado, a través de un conjunto afín y finito de instrucciones posibles, de diversos programadores o arquitectos de software, para lo cual estos lenguajes imitan, al menos formalmente, la lógica de los lenguajes humanos o naturales.

angularjs Aplicaciones en Facebook aplicaciones mobile behavioral targeting Botones Call-to-action breadcrumbs breadcrumbs web búsqueda de personal búsqueda facetada Call-to-action buttons Client-side cliente-destacado cms a medida coding comercio electrónico content marketing CSS3 PIE cuanto debe pesar un sitio data-driven web design Datos estructurados Defacement Denegación de servicio Desarrollar una aplicación web desventajas de PhoneGap diseñar newsletters diseño web diseño web argentina diseño web esqueuomórfico Diseño web responsive Diseño web responsivo diseño web santa fe diseño web Smart TV diseño web televisores DOM desde PHP enlaces rotos filtros de búsqueda flash flat web design formularios sitio web fragmentos enriquecidos función de autocompletar futuro de la realidad aumentada html HTML5 html5shiv inbound marketing Initializr interfaces Web para televisores javascript jobs jQuery Mobile Mapbox maquetado html/css maquetador web

..

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

Lenguajes:

- Lenguajes de bajo nivel. Se trata de lenguajes de programación que están diseñados para un hardware específico y que por lo tanto no pueden migrar o exportarse a otros computadores. Sacan el mayor provecho posible al sistema para el que fueron diseñados, pero no aplican para ningún otro.
- Lenguajes de alto nivel. Se trata de lenguajes de programación que aspiran a ser un lenguaje más universal, por lo que pueden emplearse indistintamente de la arquitectura del hardware, es decir, en diversos tipos de sistemas. Los hay de propósito general y de propósito específico.
- Lenguajes de nivel medio. Este término no siempre es aceptado, que propone lenguajes de programación que se ubican en un punto medio entre los dos anteriores: pues permite operaciones de alto nivel y a la vez la gestión local de la arquitectura del sistema.

Otra forma de clasificación a menudo es la siguiente:

- Lenguajes imperativos. Menos flexibles, dada la secuencialidad en que construyen sus instrucciones, estos lenguajes programan mediante órdenes condicionales y un bloque de comandos al que retornan una vez llevada a cabo la función.
- Lenguajes funcionales. También llamados procedimentales, estos lenguajes programan mediante funciones que son invocadas conforme a la entrada recibida, que a su vez son resultado de otras funciones[1] http://revistas.utp.edu.co/index.php/revistaciencia/pages/view/formatos.

Conclusiones :

Es decir, un modelo **para** resolver problemas computacionales. Los lenguajes de **programación**, necesariamente, se encuadran en uno o varios **paradigmas** a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.

RECOMENDACIONES:

1) Pica código: picar y picar y picar…

¿Por qué situo picar código en el primer lugar de mi lista de consejos? Pues porque es lo más complicado y a la vez es el aspecto clave de la programación. Al picar código te das cuenta de los errores que cometes a la hora de diseñar, en la gestión de bugs, en la creación de hilos y luego vuelves sobre cada una de esas destrezas para mejorar. No puedes solo dedicarte al diseño de soluciones, tienes que programar para producir una aplicación, lo cual es importante dominar para tener éxito. Y ya que estás no pares después de solucionar un problema del código, siempre es mejor desechar tu primer desarrollo, que casi siempre será un prototipo, y es tu segundo desarrollo el que debe apuntalar los problemas y funcionalidades que perdiste de vista al programar dicho prototipo.

2) Lee libros de programación

Picar código es más fácil dicho que hecho, y existe una gran diferencia entre un buen código y un código pobre. Esto es obvio, pero, ¿cómo puedes distinguirlos? Hasta que no ves con tus propios ojos un buen código es difícil de entender realmente la diferencia… Y es aquí cuando los libros resultan muy útiles ya que la mayoría de la veces los autores son grandes programadores. Ofrecen toda su experiencia en forma de libro. Yo adoro los libros, pero hay uno que me ha ayudado especialmente, Clean Code de Uncle Bob. Al leer este libro he descubierto defectos en mi código y la forma de solucionarlo gracias a los consejos que ofrece. Mi consejo personal es que si puedes hacerte con libros de programación, no lo dudes. También recomiendo la lectura de libros más clásicos y usarlos como libros de consulta. Otro libro muy útil es Effective Java de Joshua Bloch, que está repleto de buenos consejos. Está en mi lista de libros imprescindibles para programadores de Java. Además, al leer, estás aprendiendo de la experiencia de otra persona, y solo hay dos formas de mejorar: o mediante tu propia experiencia (que es limitada) o mediante la experiencia de otros (que es ilimitada). Recuerda que el que mucho abarca poco aprieta, en vez de leerte 5 libros, es mejor leerse 2 que realmente te sirvan varias veces.[2] http://revistas.utp.edu.co/index.php/revistaciencia/pages/view/formatos.

Observaciones generales:

En el proceso de selección de artículos para publicar, se realiza una evaluación inicial para determinar si el trabajo cumple con los términos y observaciones presentadas en este documento. En la segunda evaluación se evalúa su contenido y aporte por parte de evaluadores calificados de acuerdo al área correspondiente.

Los artículos que no llenen los requisitos de la convocatoria en cuanto a formato, no serán tenidos en cuenta para su publicación y serán descartados en la evaluación inicial.

Este documento de ejemplo, en Microsoft Word, para la elaboración de artículos para la revista SCIENTIA ET TECHNICA podrá ser descargado de la página:

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

http://revistas.utp.edu.co/index.php/revistaciencia/pages/view/formatos.

Haciendo clic en la pestaña *Formatos*.

Presentación de trabajos:
Los artículos deben venir acompañados por los formatos de datos del autor, el cual se puede descargar en la página *web* de la revista http://revistas.utp.edu.co/index.php/revistaciencia/pages/view/formatos. haciendo clic en la pestaña *Formatos*. Estos formatos deben ser cargados en la plataforma Open Journal Systems. Los datos allí consignados serán incorporados en la Base Bibliográfica *Publindex* de Colciencias.

Los artículos deben estar presentados en el formato de la revista, el cual se puede descargar en la página *web* de la revista http://revistas.utp.edu.co/index.php/revistaciencia/pages/view/formatos haciendo clic en la pestaña *Formatos*. El no uso de este formato descalifica el artículo y no será tenido en cuenta en la convocatoria.

Envío de artículos

La recepción de artículos se realizará por medio de Open Journal Systems - OJS en las fechas en que están abiertas las convocatorias.

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos