Paradigmas de Programación

Maria L Zamora y Ali Limon

Agosto 13, 2018

I. Paradigmas de Programación

Imperativo

Programación Imperativa supone que la computadora puede mantener por medio de entorno de variables cualquier cambio en procesos computacionales.

Características

- Los cálculos se realizan a través de una secuencia guiada de pasos
- Variables se acceden o cambian
- El orden de los pasos es crucial

Ventajas

- Eficiente
- Asemeja a la máquina
- Popular
- Familiar

Desventajas

- El orden es crucial en el programa
- Programas difíciles de analizar de manera independiente
- Se requiere un mejor diseño de funciones

```
a <- 1
b <- 2
c <- a + b
print(c)
```

```
## [1] 3
```

En este curso se utilizará un paradigma imperativo, ya que es el más sencillo de entender.

Lenguajes de Programación

- R
- Python
- Visual Basic Excel

Orientado a Objetos

La Programación orientada a Objetos es un paradigma de programación que tiene como objetivo la implementación basada en una colección de objetos que están estructurados en clases.

Características

- La implementación y desarrollo del paradigma está fundamentado en los objetos.
- Dar prioridad a los objetos y su abstracción como una parte fundamental en la solución de problemas.
- Definir los métodos, propiedades y características de los objetos así como su relación (interacción).

Ventajas

- La implementación de clases y objetos proporciona una relación más directa con la realidad al implementar funciones y métodos como comportamientos de las entidades.
- Bajo acoplamiento y alta cohesión: Gracias a la modularidad, cada componente o módulo de un desarrollo tiene independencia de los demás componentes.
- Facilidad de desarrollo y el mantenimiento debido a la filosofía del paradigma

Desventajas

- El uso de tareas simples termina siendo improductivo
- Velocidad de ejecución
- Se hereda código no usable a la nueva clase

```
myString <- "Objecto String"
print(class(myString))
```

```
## [1] "character"
```

```
print(tolower(myString))
```

```
## [1] "objecto string"
```

```
print(toupper(myString))
```

```
## [1] "OBJECTO STRING"
```

Lenguajes de Programación

- Java
- Python

Funcional

La programación funcional es un paradigma de programación declarativa basado en el uso de funciones matemáticas, en contraste con la programación imperativa, que enfatiza los cambios de estado mediante la mutación de variables.

Características

- Lenguajes expresivos y matemáticamente elegantes
- Se evita el concepto de estado del computo

Ventajas

- Altos niveles de abstracción
- Elegancia, ligibilidad y flexibilidad
- Las características del paradigma, en especial la utilización de funciones puras, permiten realizar ciertas optimizaciones particulares como paralelización

Desventajas

- Dificultad inicial para producir buen código
- Falta de Recursos (librerias y software)

```r
recur_factorial <- function(n) {
  if(n <= 1) {
    return(1)
  } else {
    return(n * recur_factorial(n-1))
  }
}
```

Lenguajes de Programación

- R
- Scala
- Python