

## Estudiantes

10-10264 - Maria Lourdes García

10-10517 - Daniela Ortiz

---

## Respuestas

Los lenguajes de programación orientados a objetos que poseen herencia simple están limitados a incorporar comportamientos de un solo ancestro al momento de definir una clase.

Depende. En lenguajes de POO con herencia simple, al momento de definir una clase, esta puede heredar comportamientos de su padre y si su padre hereda de su abuelo, podría estar heredando comportamientos de ambos ancestros. En Ruby todas las clases heredan de la clase madre `Object()`, por lo que basta tener una clase que herede de otra, para incorporar comportamientos de más de 1 ancestro.

Lenguajes de POO con un sistemas de tipos estático (C++, Java, C#) no tienen la posibilidades de elegir la implementación de un método a tiempo de ejecución (despacho dinámico).

No, existen múltiples lenguajes de POO con despacho dinámico, el cual se realiza mediante virtual tables. En C++ un método puede ser declarado como virtual, en Java todos los métodos son virtuales y en C# puedes crear virtual tables.

La introspección y reflexibilidad son conceptos que se manejan en la POO pero no guardan ninguna relación entre sí.

No, la introspección depende de la reflexividad, ya que no se puede modificar un valor (reflexividad) si no se sabe de qué tipo es(introspección)

En un lenguaje con un sistema de tipos dinámico la sobrecarga de métodos es innata y representa una comodidad dado que permite implementar un mismo método para distintos tipos.

Depende del lenguaje de programación, ya que algunos lenguajes tienen innata la redefinición de los métodos (por ejemplo Ruby), mientras otros como Python no sobrecargan los métodos, sino que los sobrescriben.

En los lenguajes POO existen los términos interfaz, módulo, clase abstracta, rol, etc; definidos como objetos que pueden encapsular definiciones de clases o implementaciones concretas de métodos.

Si, ya que, en el caso de las interfaces, se definen solamente las firmas de los métodos de las clases y, en el caso de los módulos, se implementan los métodos.

Los métodos virtuales permiten asociar, al momento de compilar, una implementación de un método sobrecargado con una llamada del mismo; eliminando el **overhead** del despacho dinámico.

No, la invocación de los métodos sobrecargados en un método virtual se realiza con despacho dinámico.

Cuando un lenguaje de POO tiene herencia simple no ocurre el problema del diamante pero de igual forma pueden existir llamadas ambiguas de métodos, dado que incorporar interfaces, módulos, protocolos, etc, no evita colisión de nombres.

No, ya que depende de si el lenguaje está implementado con despacho estático o dinámico para poder realizar la llamada, buscando los métodos en sus respectivos ancestros.

El paso de mensaje es un término que se maneja en modelos concurrentes, también de POO y es equivalente a la llamada de una función.

Depende. Si el tipo de paso de mensaje es síncrono, resulta equivalente a la llamada de una función, pero si es asíncrono no es equivalente.

Sin importar la herencia del lenguaje de POO, una clase podría tener más de un ancestro.

Si, una clase puede tener más de un ancestro independientemente del tipo de herencia del lenguaje de POO (simple o múltiple). El tipo de herencia solo limita la cantidad de padres que puede tener una misma clase. Sin embargo, cada clase puede heredar de una clase padre, la cual a su vez puede tener una clase padre de la cual hereda y así sucesivamente.