Conceptos Básicos de POO

Daniel González Duque

Instancia vs Clase

- La clase es la generalidad de los objetos, estas crean las instancias
- Las instancias son objetos particulares

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Conceptos Básicos de POO

En la Programación Orientada Objetos (POO) las clases deben tener las siguientes características:

- Encapsulamiento
- Abstracción
- Herencia
- Polimorfismo

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Encapsulamiento

- Permite que el código sea modular.
- Permite tener una organización en el código.
- Permite la reserva de información y la integridad de los datos que contiene el objeto.
- El cambio de la implementación interna de una clase no debería generar errores o cambios en la funcionalidad del sistema.
- Requiere de una interfaz para poder realizar modificaciones específicas.

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Encapsulamiento

Formas para encapsular los atributos en Python:

- Publico: self.<variable> = <valor>
- Protegido: self._<variable> = <valor>
- Privado: self.__<variable> = <valor>

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Herencia

Una clase puede heredar otra clase.

- Esto permite utilizar los métodos y atributos de otras clases.
- Permite crear subclases a partir de clases.
- Permite crear una estructura jerárquica de clases cada vez más especializada.

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Polimorfismo

Objetos de diferentes clase que tienen métodos y atributos con el mismo nombre, pero pueden tener diferentes resultados.

- No es esencial en Python ya que cualquier variable puede almacenar cualquier tipo de datos.

(Chappel, 2008; Norish, 2015; WordPress.com, 2008)

Diagrama UML

Daniel González Duque

¿Qué es un Diagrama UML?

- UML (Unified Modeling Language) es una forma de visualizar un programa de software usando una colección de diagramas.
- Es una guía para desarrollar un software específico de una manera ordenada.

(Lucid Software Inc., 2018)

Tipos de diagramas

- **Diagramas UML estructurales**
  - Clases
  - Paquetes
  - Componentes
  - …

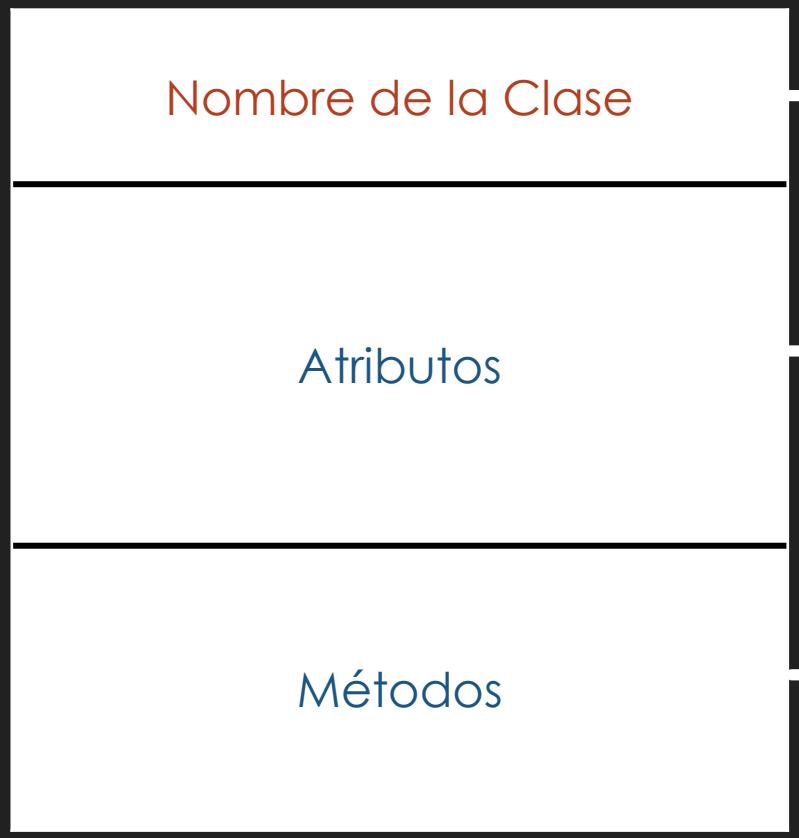
(Lucid Software Inc., 2018)

Tipos de diagramas

- **Diagramas UML de comportamiento**
  - Actividad
  - Secuencia
  - Casos
  - Comunicación
  - Interacción
  - …

(Lucid Software Inc., 2018)

Diagramas de Clase



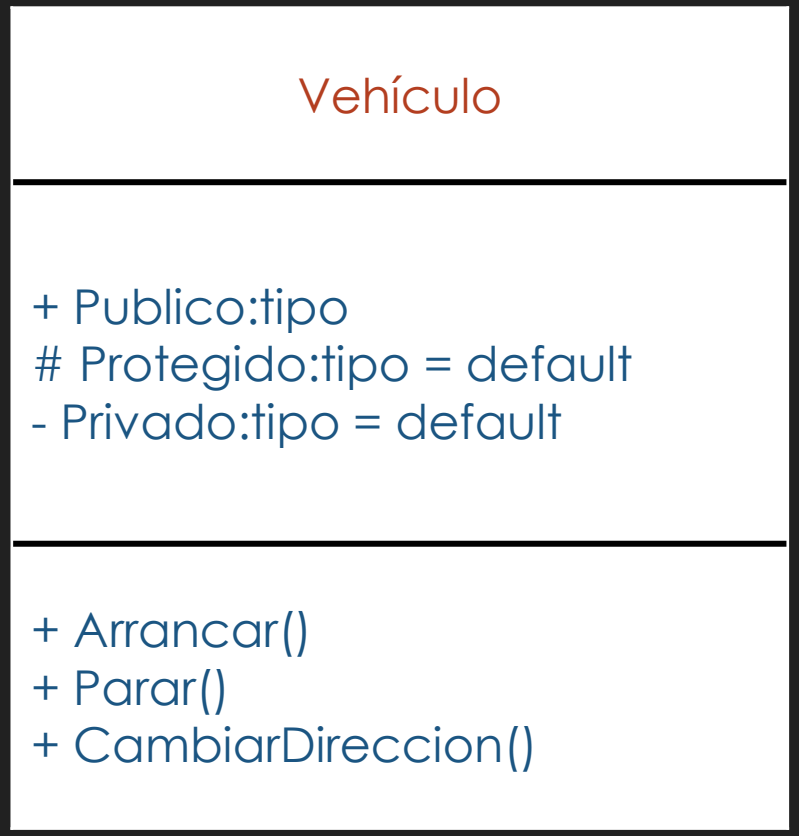
→ Nombre que llevará el objeto dentro del programa

→ Nombre que llevará cada atributo dentro de la clase

→ Nombre que llevará cada método dentro de la clase

(Lucid Software Inc., 2018)

Como se ingresa la información



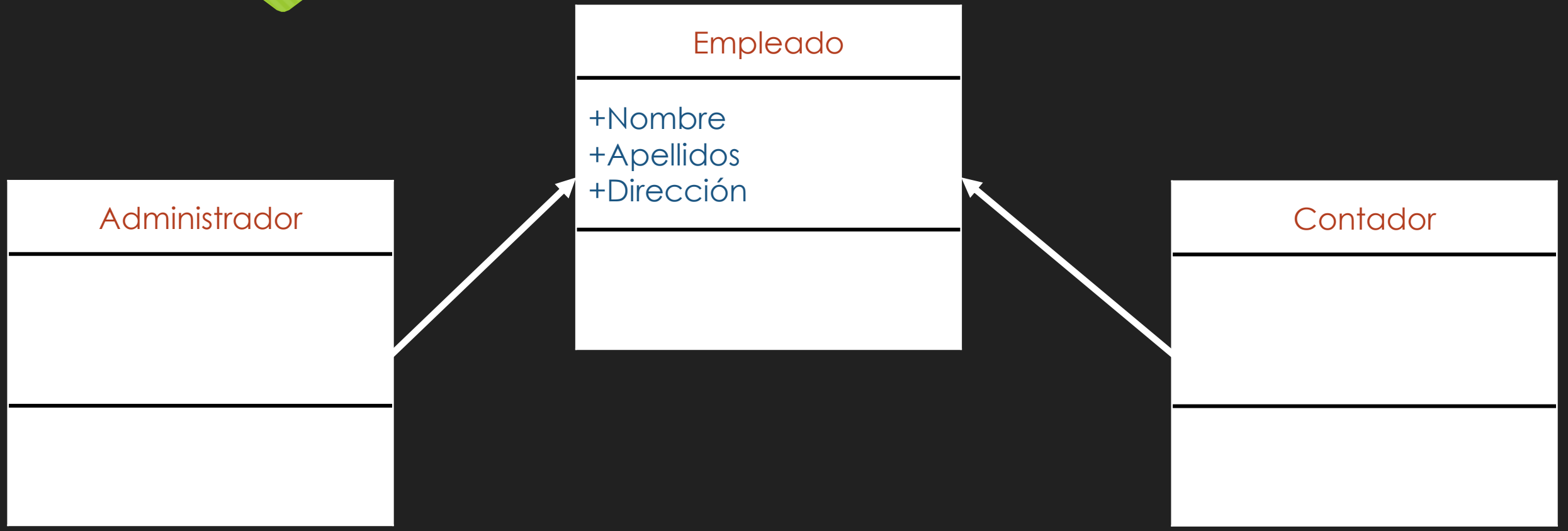
(Lucid Software Inc., 2018)

Ejemplo

Vehículo
+ Velocidad:int = 0 + Pasajeros:int = 1 + TipoCombustible:str = 'Corriente'
+ Arrancar() + Parar() + CambiarDireccion()

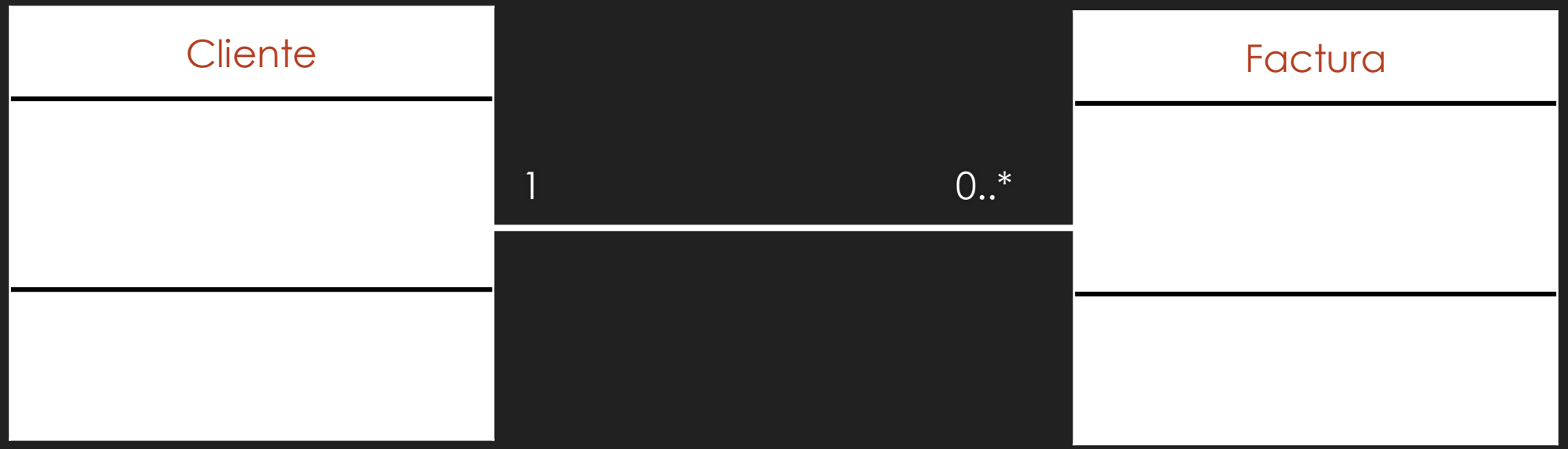
(Lucid Software Inc., 2018)

Herencia



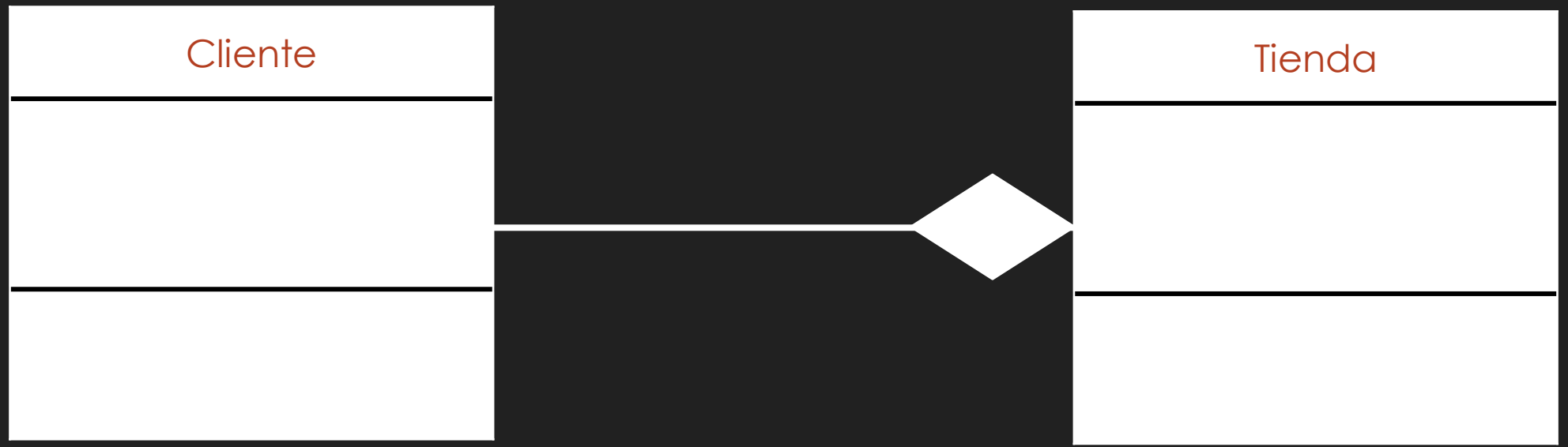
(Lucid Software Inc., 2018)

Asociaciones



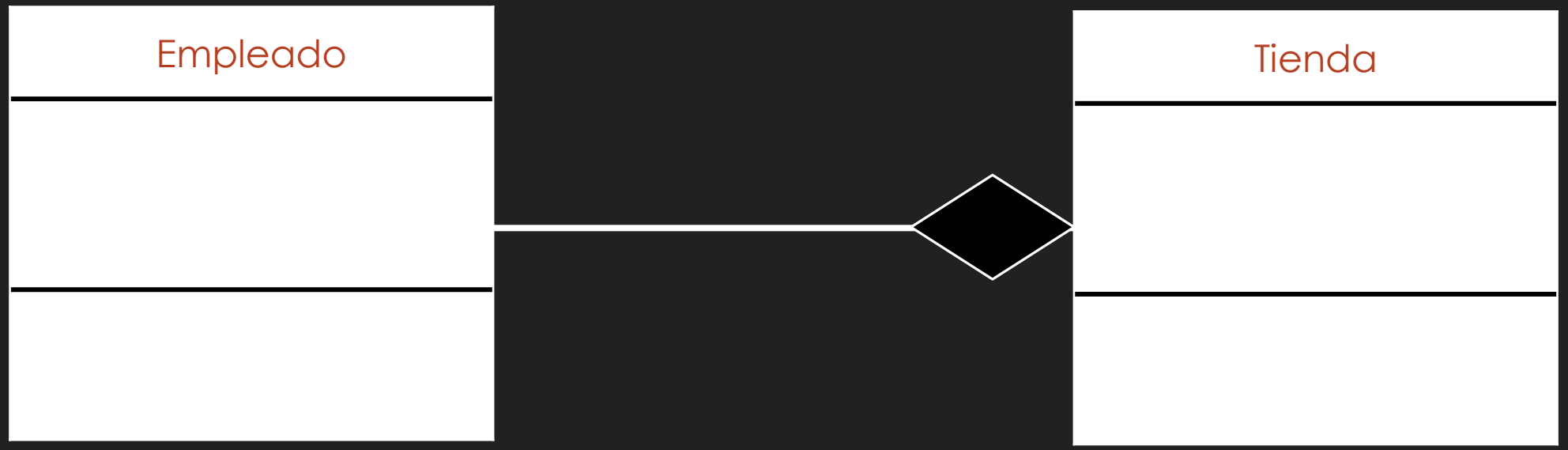
(Lucid Software Inc., 2018)

Agregación



(Lucid Software Inc., 2018)

Composición



(Lucid Software Inc., 2018)

Dependencia de uso

(Lucid Software Inc., 2018)

Referencias

- Chappel, D. (2008). Introducción a la Programación Orientada a Objetos. Recuperado 1 de marzo de 2018, a partir de http://es.ccm.net/contents/414-introduccion-a-la-programacion-orientada-a-objetos
- Lucid Software Inc. (2018). Qué es el lenguaje unificado de modelado (UML). Recuperado 1 de marzo de 2018, a partir de https://www.lucidchart.com/pages/es/qué-es-el-lenguaje-unificado-de-modelado-uml
- Norish. (2015). Introduction to Object Oriented Programming Concepts (OOP) and More. Recuperado 1 de marzo de 2018, a partir de https://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep
- WordPress.com. (2008). Características de P.O.O. Recuperado 1 de marzo de 2018, a partir de https://algonzalezpoo.wordpress.com/caracteristicas-de-poo/