Universidad Corporativa | Programas Académicos

Ingeniería en Desarrollo de Software

# Actividad | 2 # | Diagramas de Paradigma Orientado a Objetos

## Análisis y Diseño de Sistemas

Ingeniería en Desarrollo de Software

academia global

TUTOR: Eduardo Israel Castillo García

ALUMNO: Casandra Montserrat Ortiz Cortes G-1

FECHA:06/02/2024

# Índice

## INTRODUCCIÓN

La programación orientada a objetos consiste en crear entidades responsables de la información que contienen y que guardan un comportamiento y responsabilidades dentro del sistema comunicándose a través de mensajes.

Por las características que guarda, se establecen criterios que definen la creación adecuada de las entidades u objetos, los cuales se agrupan por sus características en clases que mantienen el **estado** de cada elemento (información del objeto en **atributos**), el comportamiento o los mecanismos de interacción con el exterior (**métodos**) de las partes pertinentes dentro del problema que se desea modelar (**abstracción**).

La información almacenada en el objeto debe estar celosamente resguardada y no debe ser vista o modificarse si no es a través de métodos que garanticen su integridad, a lo que se conoce como **encapsulamiento**.

Además, se puede reutilizar el código y especializarlo a partir de una estructura o jerarquía que permite agregar información, especializar el comportamiento .

## DESCRIPCIÓN

En este artículo, exploraremos la Programación Orientada a Objetos (POO) y los beneficios que ofrece. La POO es un paradigma de programación que se centra en la creación de objetos que representan entidades del mundo real, y en la interacción entre estos objetos a través de métodos y propiedades.

La POO se ha convertido en una de las metodologías de programación más utilizadas en la actualidad, debido a su capacidad para organizar y estructurar el código de manera más eficiente. Además, ofrece una mayor reutilización de código, lo que resulta en un desarrollo más rápido y mantenible.

A lo largo de este artículo, también veremos ejemplos prácticos de cómo se implementa la POO en diferentes situaciones, lo que te ayudará a comprender mejor sus conceptos y aplicaciones.

La Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en la creación de objetos que interactúan entre sí.

## JUSTIFICACIÓN

La Programación Orientada a Objetos (POO) ofrece una serie de beneficios que hacen que sea una metodología muy utilizada en el desarrollo de software. Al utilizar la POO, los programadores pueden organizar su código de una manera más estructurada y modular, lo que facilita su mantenimiento y reutilización.

A continuación, se presentan algunos de los principales beneficios de utilizar la Programación Orientada a Objetos:

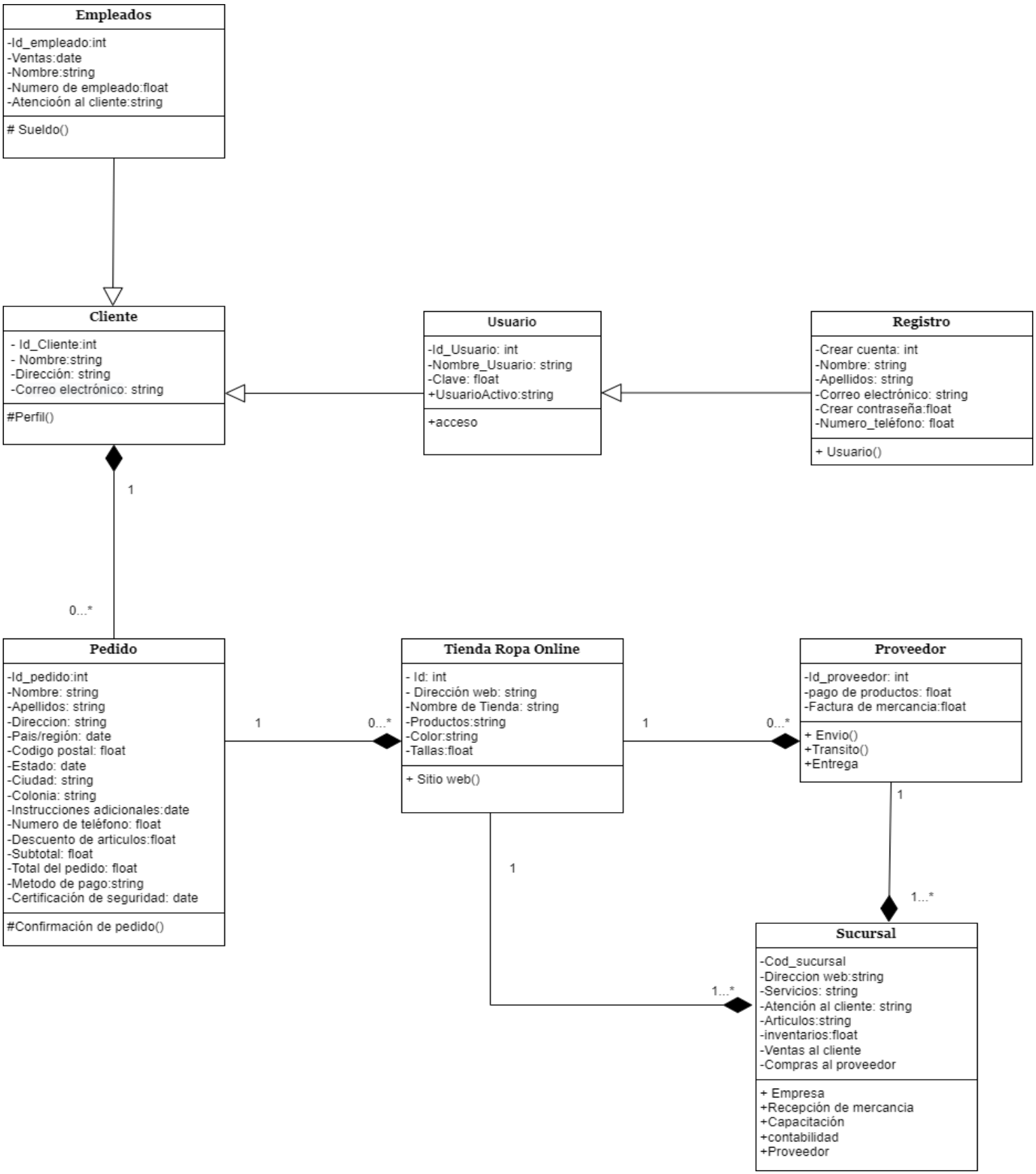
Reutilización de código: Uno de los beneficios más importantes de la POO es la posibilidad de reutilizar código. Al utilizar clases y objetos, es posible crear componentes que se pueden

utilizar en diferentes partes del programa, lo que ahorra tiempo y esfuerzo en el desarrollo.

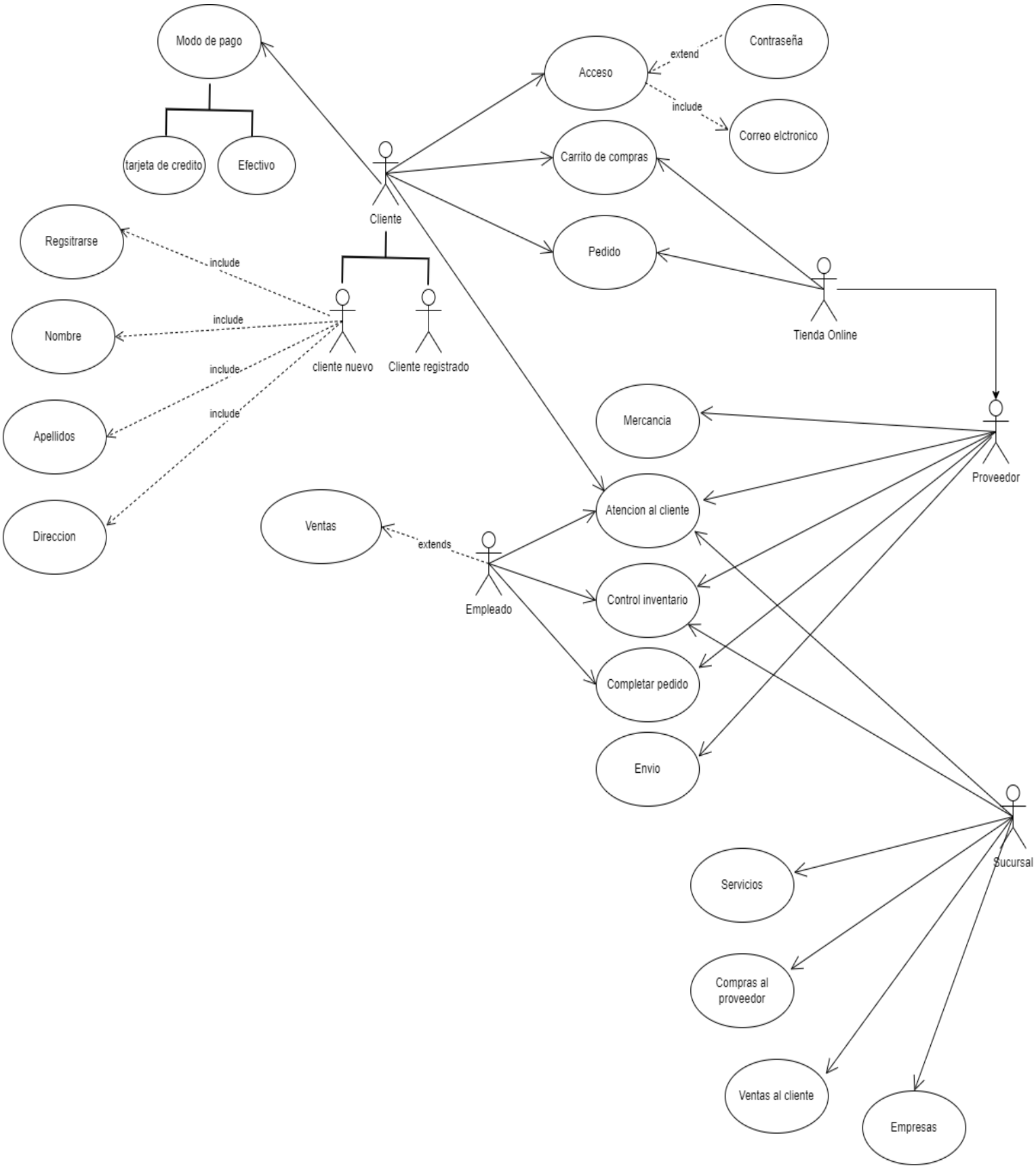
Modularidad: La POO permite organizar el código en módulos independientes, conocidos como clases. Cada clase tiene su propia funcionalidad y puede ser modificada o reemplazada sin afectar al resto del programa. Esto facilita el mantenimiento y la escalabilidad del software .

## DESARROLLO

● Diagramas de clases



● Diagrama de casos de uso



## CONLUSIÓN

La Programación Orientada a Objetos es una metodología de programación que ofrece numerosos beneficios a los desarrolladores. Al utilizar este enfoque, los programadores pueden organizar su código de manera más eficiente, reutilizar componentes y simplificar el mantenimiento de sus aplicaciones.

Algunos de los beneficios más destacados de la Programación Orientada a Objetos incluyen la modularidad, la encapsulación, la herencia y el polimorfismo. Estos conceptos permiten a los desarrolladores crear programas más flexibles, escalables y fáciles de entender.

En cuanto a los ejemplos de Programación Orientada a Objetos, hemos visto cómo podemos crear una clase «Persona» para representar a una persona en nuestro programa. También hemos explorado cómo implementar la herencia en una clase «Animal» para aprovechar la reutilización de código. Por último, hemos analizado el uso del polimorfismo en una clase «Figura» para permitir diferentes comportamientos en función del tipo de figura.

Es importante tener en cuenta que estos ejemplos.

## Referencias

*Programación orientada a objetos*. (14 de febrero de 2015). Obtenido de https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos