

# UML AND CLASS DIAGRAMS

## Advanced Programming

Author: Eng. Carlos Andrés Sierra, M.Sc.  
[cavirguezs@udistrital.edu.co](mailto:cavirguezs@udistrital.edu.co)

Computer Engineer  
Lecturer  
Universidad Distrital Francisco José de Caldas

2024-III



# Outline

1 Unified Modeling Language (UML)

2 UML Diagrams

3 UML Class Diagrams



# Outline

1 Unified Modeling Language (UML)

2 UML Diagrams

3 UML Class Diagrams



# Basics of UML



Figure: Prompt: Draw a software architect drawing some software designs.



- The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.  
*independently of programming language*



# Basics of UML



**Figure:** Prompt: Draw a software architect drawing some software designs.



- The **Unified Modeling Language (UML)** is a **general-purpose, developmental, modeling language** in the field of **software engineering**, that is intended to provide a **standard way** to visualize the **design** of a system.
- The **UML** represents a **collection of best engineering practices** that have proven **successful** in the modeling of **large and complex systems**.

# Importance of UML

- The UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
- It is a powerful and flexible graphical language that is used to model systems in an object-oriented way.



# Importance of UML

- The **UML** is a **standard language** for specifying, visualizing, constructing, and documenting the artifacts of software systems.
- It is a **powerful** and **flexible graphical language** that is used to **model** systems in an **object-oriented way**.



# Outline

1 Unified Modeling Language (UML)

2 UML Diagrams

3 UML Class Diagrams



# List of UML Diagrams

- UML has **14 types** of diagrams, which can be divided into two categories:
  - **Structural Diagrams:** These diagrams represent the **static** aspects of the system. Here are some examples: **Class Diagram**, **Object Diagram**, **Component Diagram**, **Deployment Diagram**, among others.
  - **Behavioral Diagrams:** These diagrams represent the **dynamic** aspects of the system. Here are some examples: **Activity Diagram**, **Sequence Diagram**, **State Diagram**, among others.

*Code → General Definitions*

*Production* *YAML* *DevOps*



# List of UML Diagrams

- UML has **14 types** of diagrams, which can be divided into two categories:
  - **Structural Diagrams:** These diagrams represent the **static** aspects of the system. Here are some examples: Class Diagram, Object Diagram, Component Diagram, Deployment Diagram, among others.
  - **Behavioral Diagrams:** These diagrams represent the **dynamic** aspects of the system. Here are some examples: Activity Diagram, Sequence Diagram, State Diagram, among others.

Objects

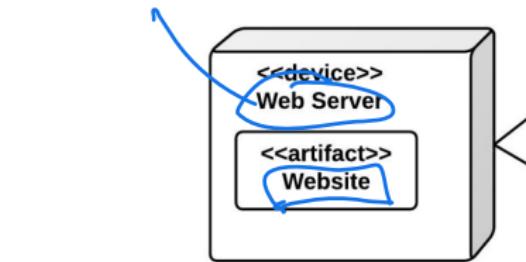
classes  
communication



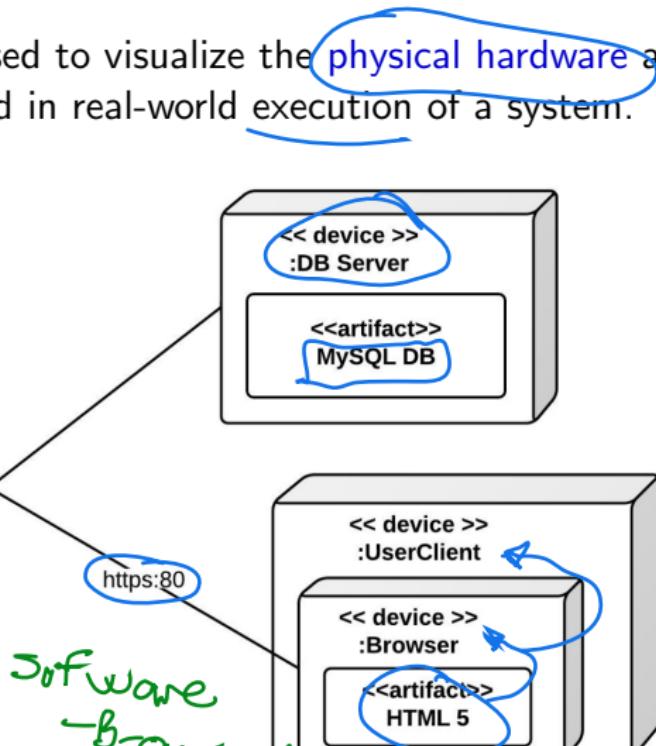
# UML Deployment Diagrams I

Deployment diagrams are used to visualize the physical hardware and software expected to be used in real-world execution of a system.

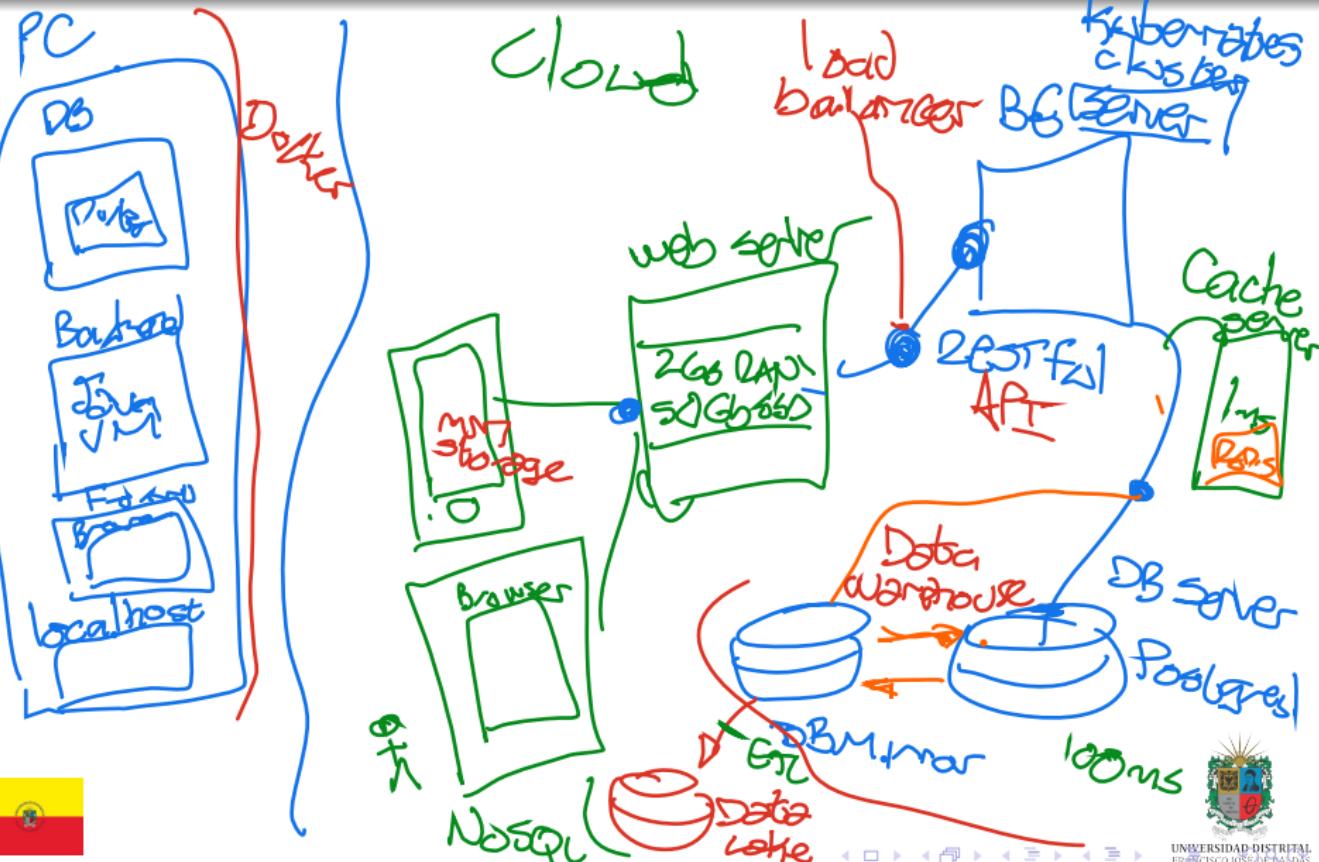
On-premises  
cloud



Physical  
2 servers  
↓ client  
Internet

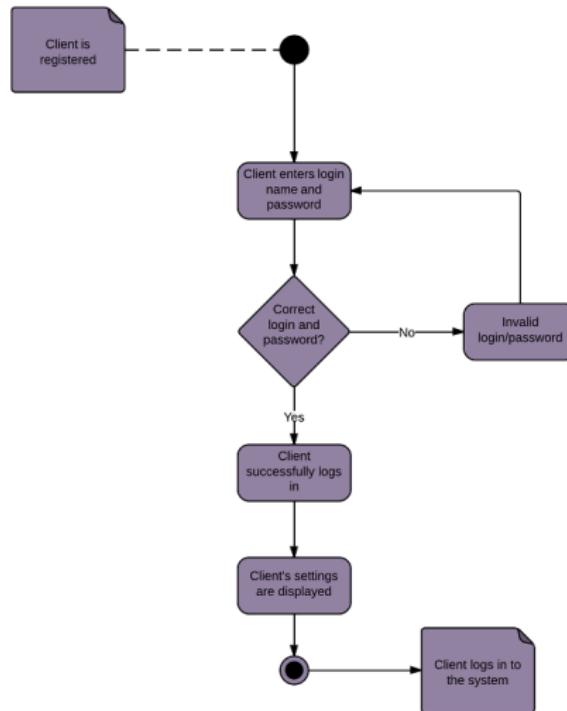


# UML Deployment Diagrams II → Monolith



# UML Activity Diagrams I

Activity diagrams are used to model **workflow** and **business processes**.

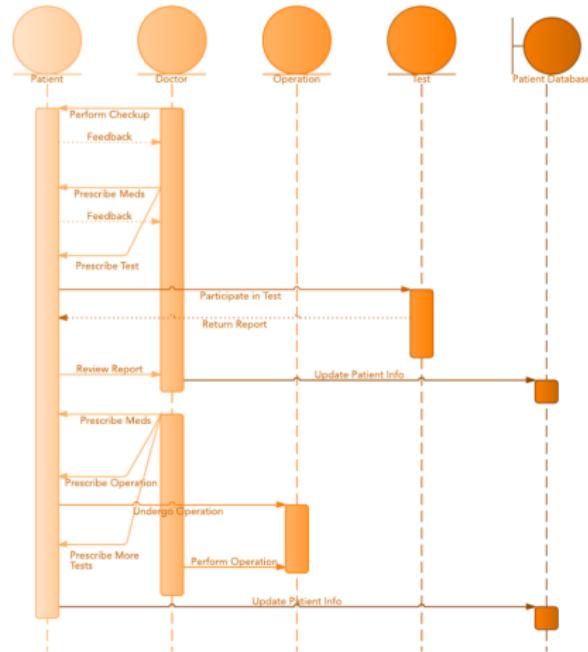


# UML Activity Diagrams II



# UML Sequence Diagrams I

Sequence diagrams are used to model **interactions** between **objects** in a system.

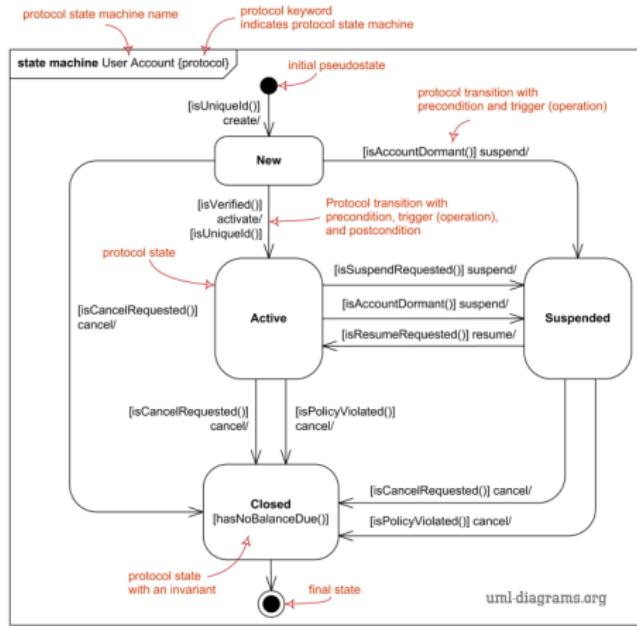


# UML Sequence Diagrams II



# UML State Diagrams I

State diagrams are used to model the **dynamic behavior** of an object in a system.

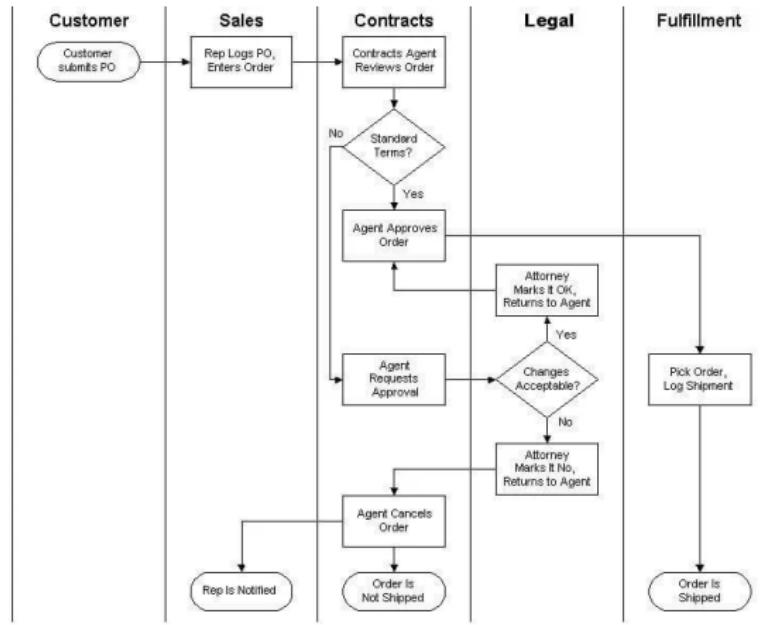


# UML State Diagrams II



# Business Process Diagrams I

Business process diagrams are used to model the **workflow** and **business processes** of an **organization**.



# Business Process Diagrams II



# Outline

1 Unified Modeling Language (UML)

2 UML Diagrams

3 UML Class Diagrams



# Basic Concepts of UML Class Diagrams

- A **class diagram** is a type of **static structure diagram** that describes the structure of a system by showing the system's **classes**, their **attributes**, **operations**, and the **relationships** among the classes.
- A **class diagram** is a collection of **classes** and **interfaces** that are used to model the objects in a system.



# Basic Concepts of UML Class Diagrams

- A **class diagram** is a type of **static structure diagram** that describes the structure of a system by showing the system's **classes**, their **attributes**, **operations**, and the **relationships** among the classes.
- A **class diagram** is a collection of **classes** and **interfaces** that are used to **model** the **objects** in a system.



# Types of Objects Relations

- Association: A **relationship** between two classes that is used to represent a **connection** between the classes.
- Aggregation: A **relationship** between two classes that is used to represent a **part-whole** relationship between the classes.
- Composition: A **relationship** between two classes that is used to represent a **stronger part-whole** relationship between the classes.



# Types of Objects Relations

- Association: A **relationship** between two classes that is used to represent a **connection** between the classes.
- Aggregation: A **relationship** between two classes that is used to represent a **part-whole** relationship between the classes.
- Composition: A **relationship** between two classes that is used to represent a **stronger part-whole** relationship between the classes.



# Types of Objects Relations

- Association: A **relationship** between two classes that is used to represent a **connection** between the classes.
- Aggregation: A **relationship** between two classes that is used to represent a **part-whole** relationship between the classes.
- Composition: A **relationship** between two classes that is used to represent a **stronger part-whole** relationship between the classes.



# Good practices in UML Class Diagrams

- Use **singular nouns** for **class names**.
- Use **Camel Case** for **class names**.
- Use **singular nouns** for **attributes**.
- Use **some case** for **attributes**.
- Use **verbs** for **operations as methods**.
- Use **some casee** for **operations**.



# Class Diagram Example



# Outline

1 Unified Modeling Language (UML)

2 UML Diagrams

3 UML Class Diagrams



# Thanks!

## Questions?



Repo:

 [github.com/engandres/ud-public/courses/advanced-programming](https://github.com/engandres/ud-public/courses/advanced-programming)

