Diseño de Software

Ingeniería de Software

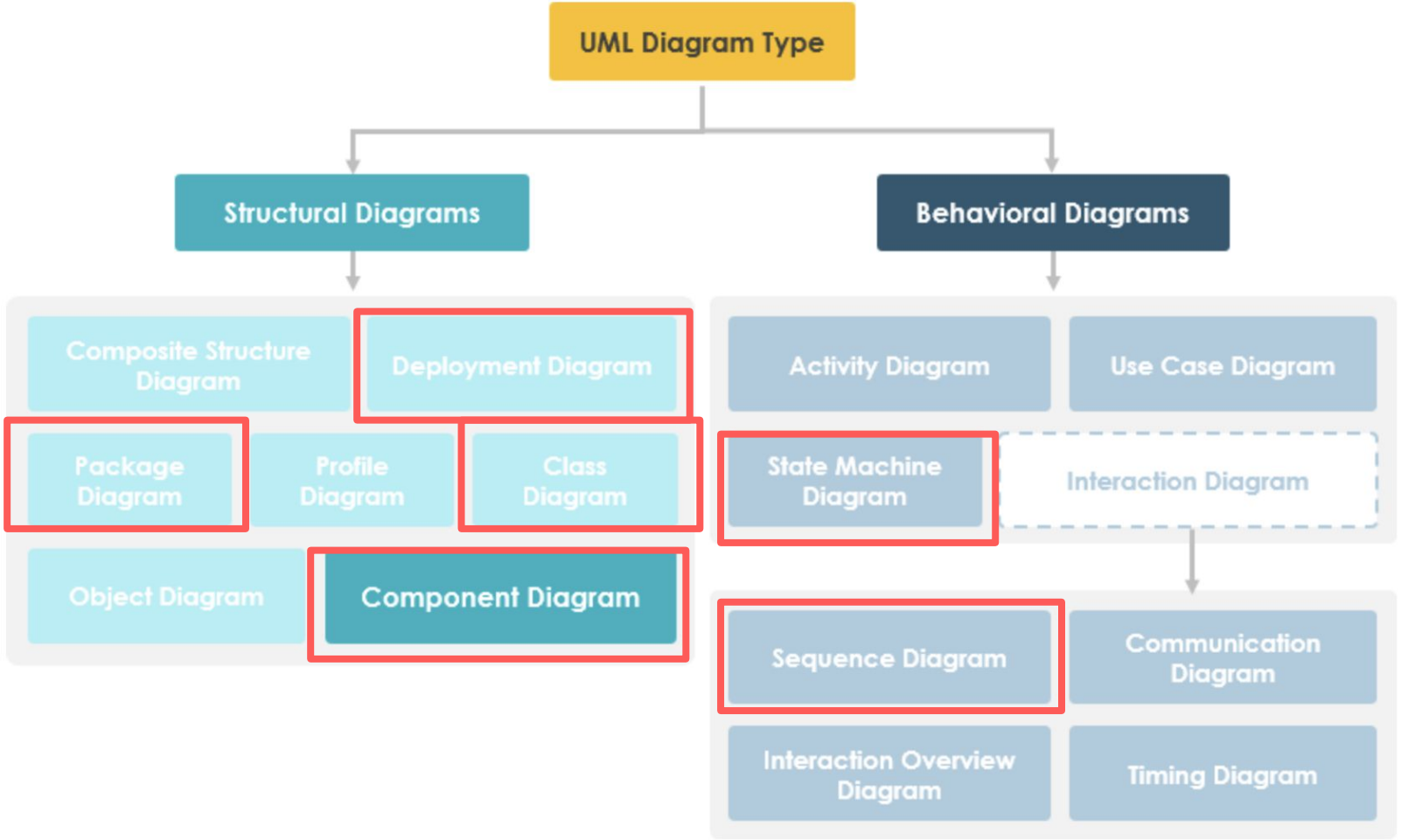

UML Diagram Type
Structural Diagrams
Behavioral Diagrams
Composite Structure Diagram
Deployment Diagram
Package Diagram
Profile Diagram
Class Diagram
Object Diagram
Component Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Interaction Overview Diagram
Timing Diagram

Diagrama de Clases

They clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects.

- Illustrate data models for information systems, no matter how simple or complex.
- Better understand the general overview of the schematics of an application.
- Visually express any specific needs of a system and disseminate that information throughout the business.
- Create detailed charts that highlight any specific code needed to be programmed and implemented to the described structure.
- Provide an implementation-independent description of types used in a system that are later passed between its components.

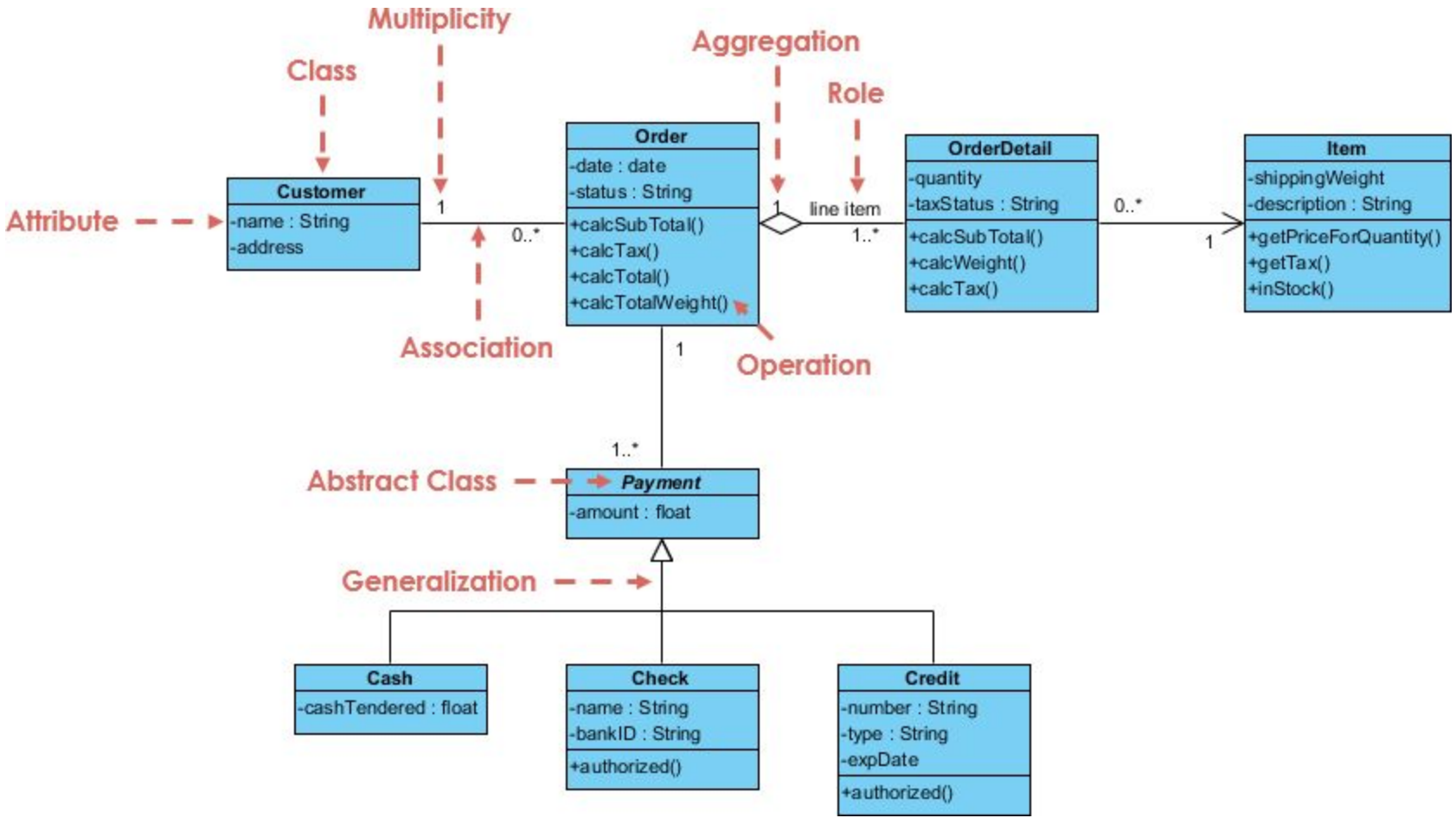
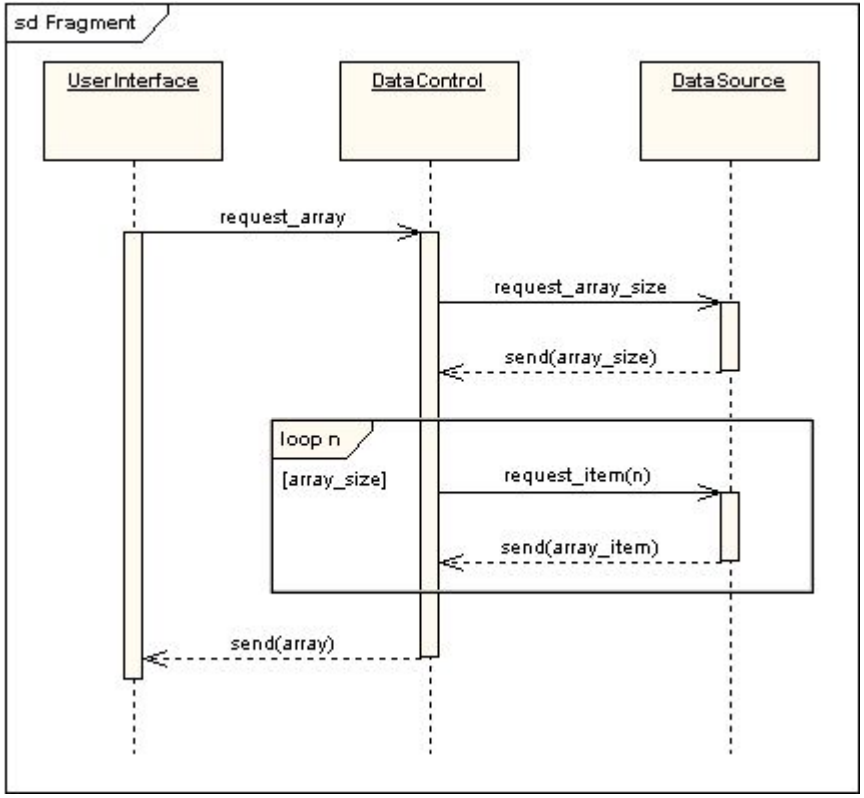

Multiplicity
Class
Aggregation
Role
Attribute
Customer
-name : String
-address
1
0..*
Order
-date : date
-status : String
+calcSubTotal()
+calcTax()
+calcTotal()
+calcTotalWeight()
1
line item
1..*
OrderDetail
-quantity
-taxStatus : String
+calcSubTotal()
+calcWeight()
+calcTax()
0..*
1
Item
-shippingWeight
-description : String
+getPriceForQuantity()
+getTax()
+inStock()
Association
Operation
1
1..*
Abstract Class
Payment
-amount : float
Generalization
Cash
-cashTendered : float
Check
-name : String
-bankID : String
+authorized()
Credit
-number : String
-type : String
-expDate
+authorized()

Diagrama de Secuencia

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.


sd Fragment
UserInterface
DataControl
DataSource
request_array
request_array_size
send(array_size)
loop n
[array_size]
request_item(n)
send(array_item)
send(array)

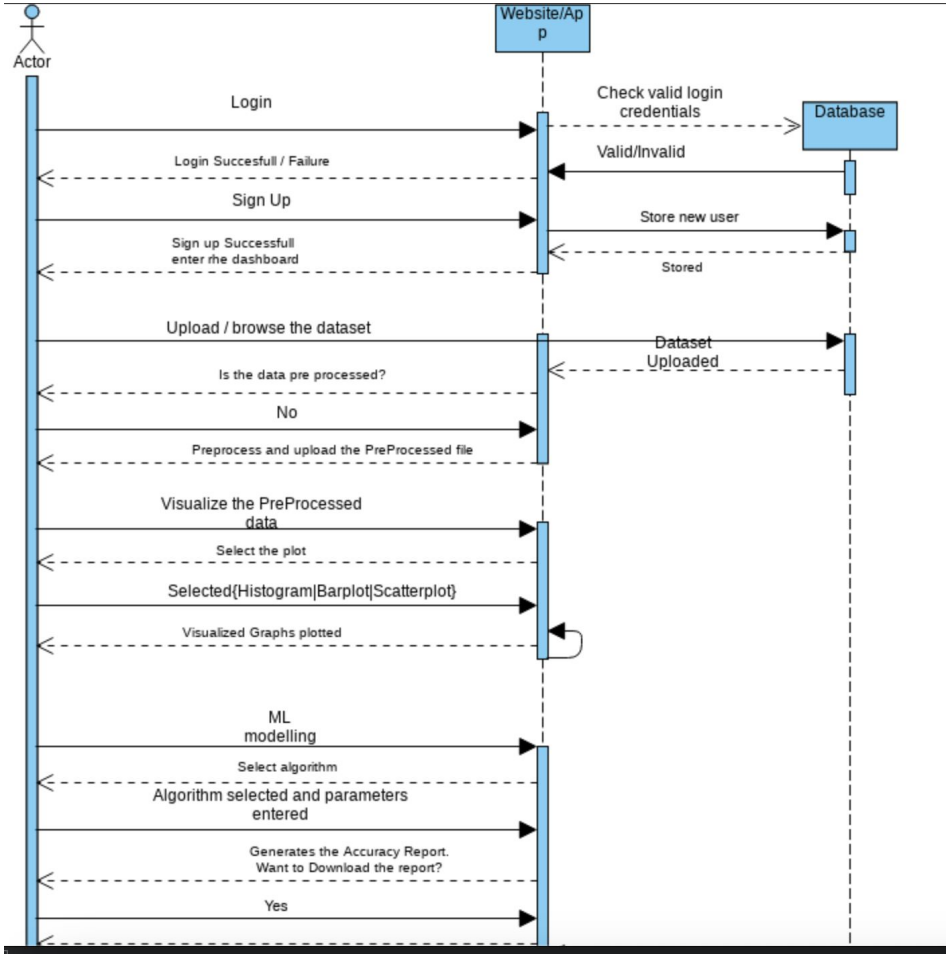
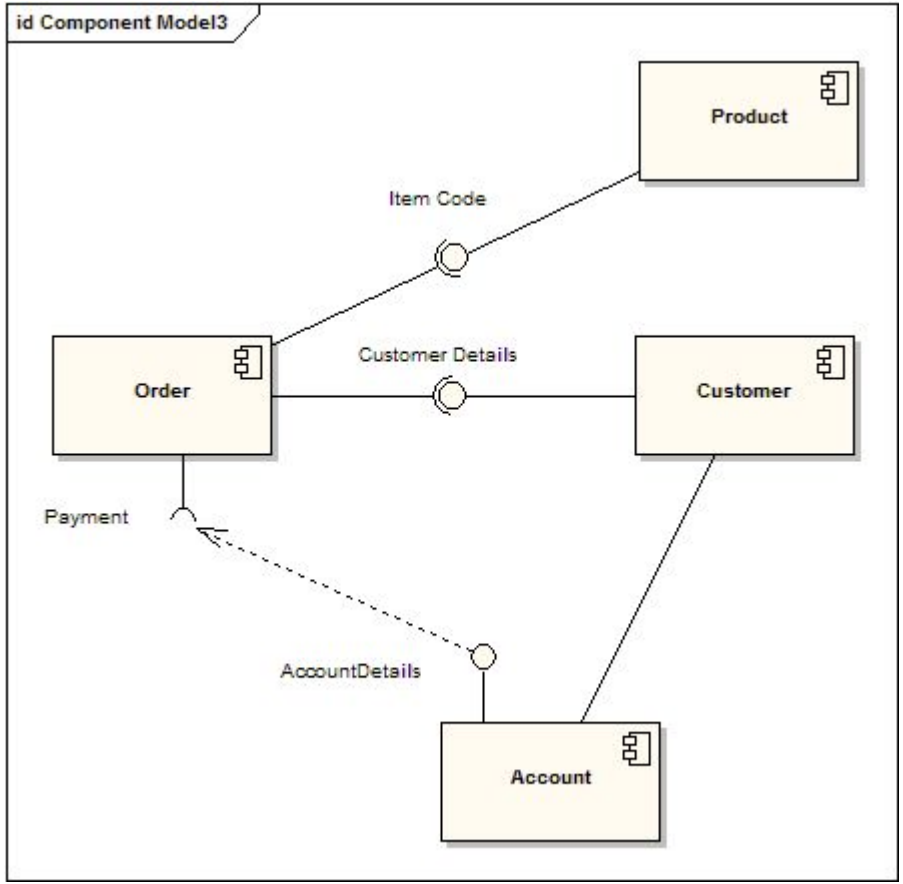

Actor
Website/App
Database
Login
Check valid login credentials
Login Succesfull / Failure
Valid/Invalid
Sign Up
Store new user
Sign up Successfull enter rhe dashboard
Stored
Upload / browse the dataset
Dataset Uploaded
Is the data pre processed?
No
Preprocess and upload the PreProcessed file
Visualize the PreProcessed data
Select the plot
Selected{Histogram|Barplot|Scatterplot}
Visualized Graphs plotted
ML modelling
Select algorithm
Algorithm selected and parameters entered
Generates the Accuracy Report. Want to Download the report?
Yes

Diagrama de Componentes

Used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.


id Component Model3
Product
Item Code
Order
Customer Details
Customer
Payment
AccountDetails
Account

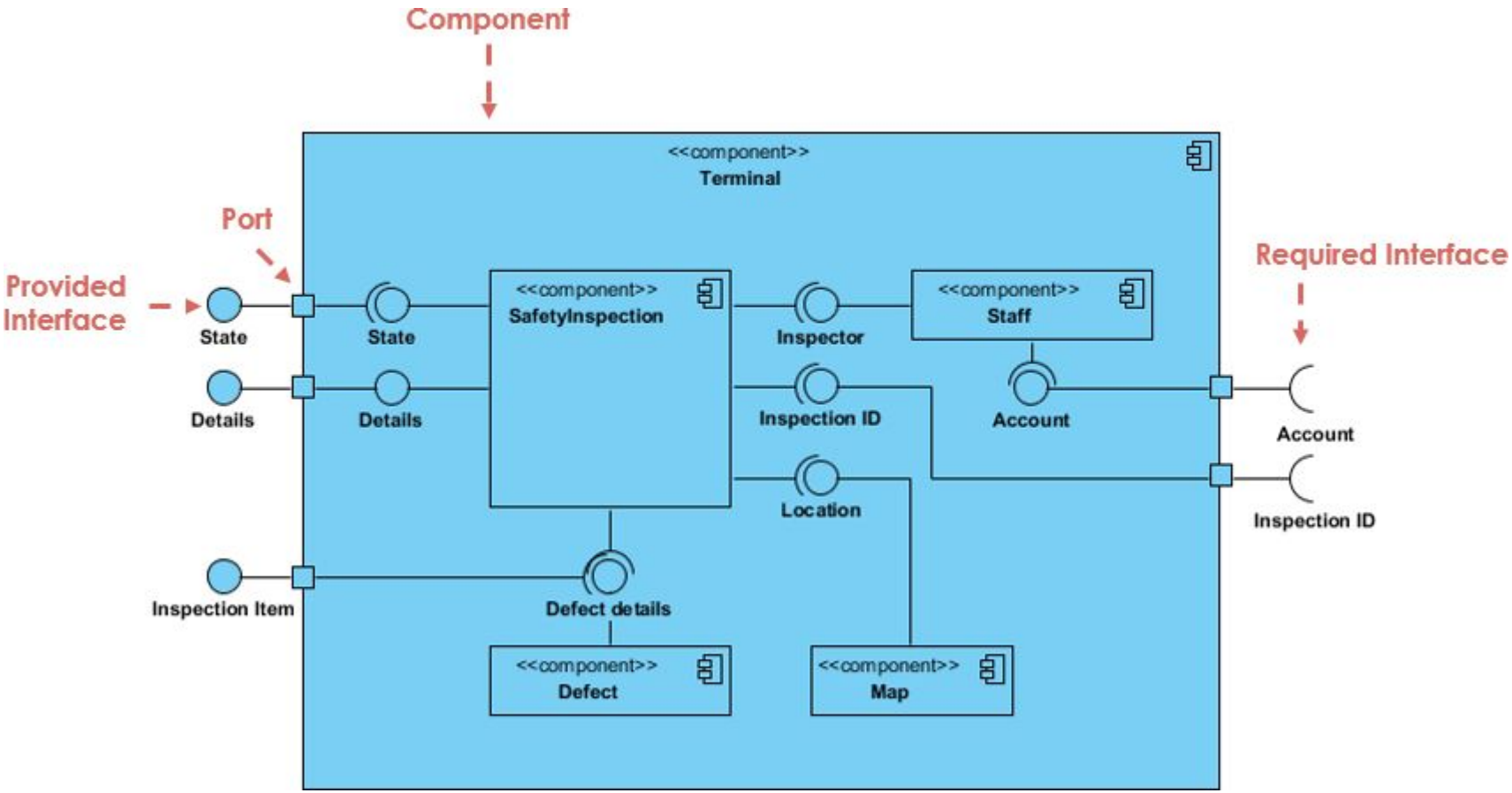
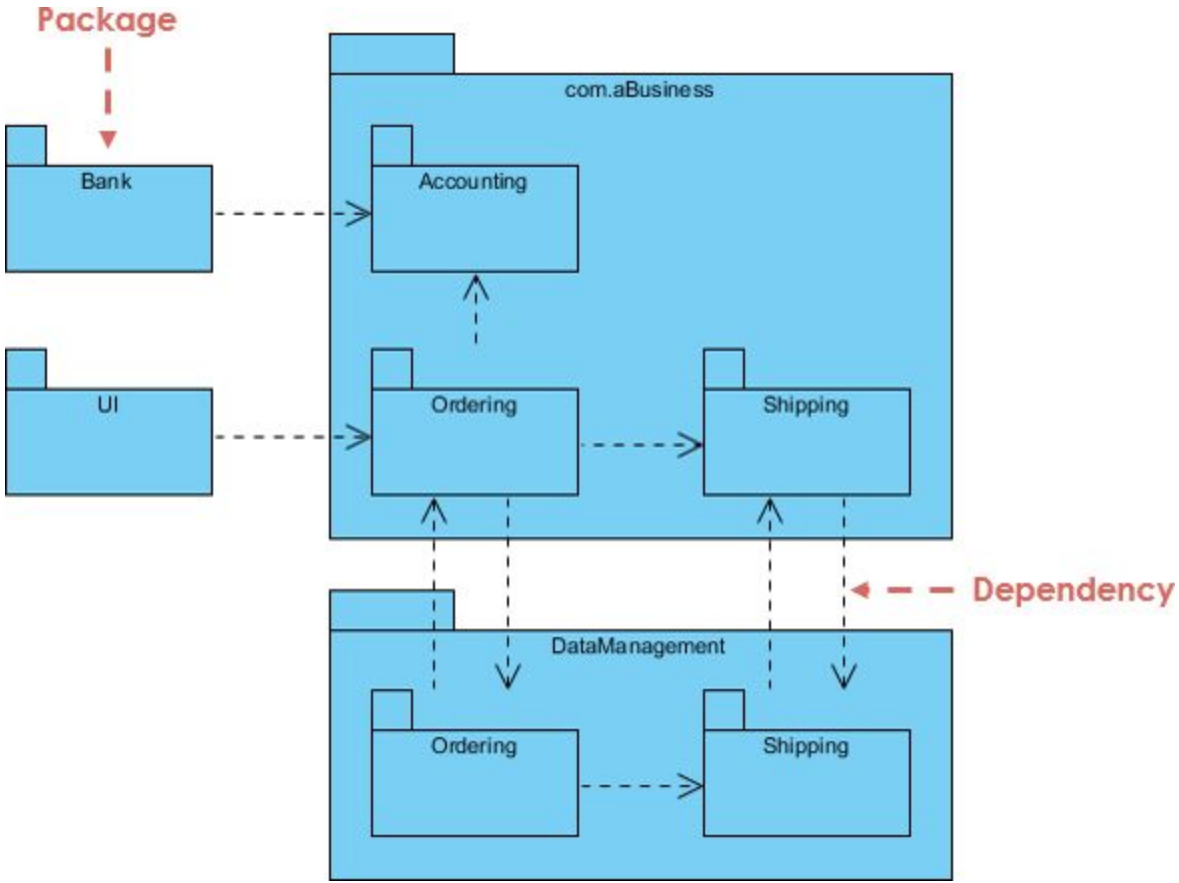

Component
<<component>>
Terminal
Port
Provided Interface
Required Interface
State
State
<<component>>
SafetyInspection
Inspector
<<component>>
Staff
Details
Details
Inspection ID
Account
Account
Location
Inspection ID
Inspection Item
Defect details
<<component>>
Defect
<<component>>
Map

Diagrama de Paquetes

Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.

- Package Diagram can be used to simplify complex class diagrams, it can group classes into packages.
- A package is a collection of logically related UML elements.
- Packages are depicted as file folders and can be used on any of the UML diagrams.


Package
com.aBusiness
Bank
Accounting
UI
Ordering
Shipping
Dependency
DataManagement
Ordering
Shipping

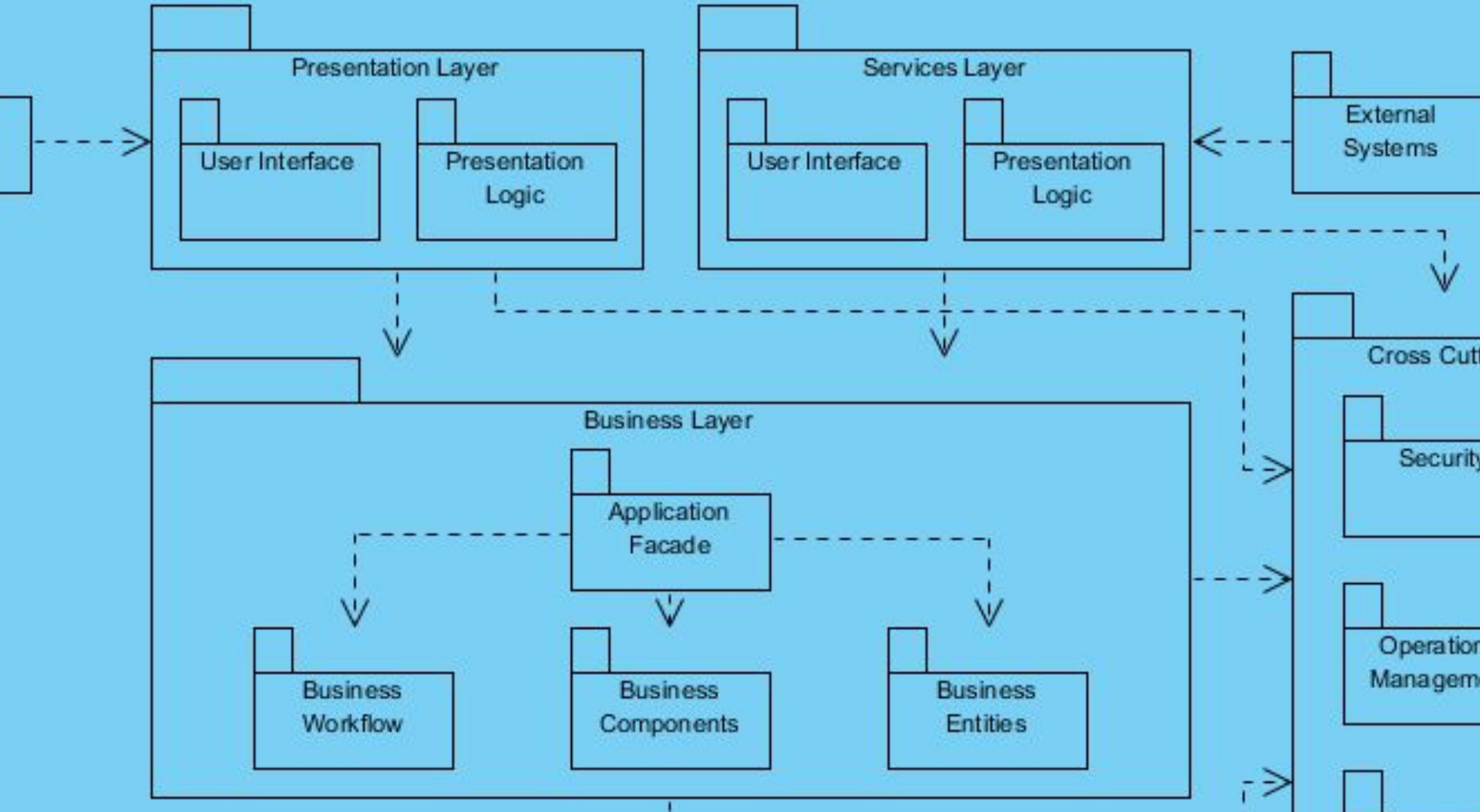

Presentation Layer
User Interface
Presentation Logic
Services Layer
User Interface
Presentation Logic
External Systems
Cross Cutt
Security
Operation
Managem
Business Layer
Application Facade
Business Workflow
Business Components
Business Entities

Diagrama de Estados

Used to represent the condition of the system or part of the system at finite instances of time. It's a **behavioral** diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as **State machines** and **State-chart Diagrams**.

- We use it to state the events responsible for change in state (we do not show what processes cause those events).
- We use it to model the dynamic behavior of the system .
- To understand the reaction of objects/classes to internal or external stimuli.

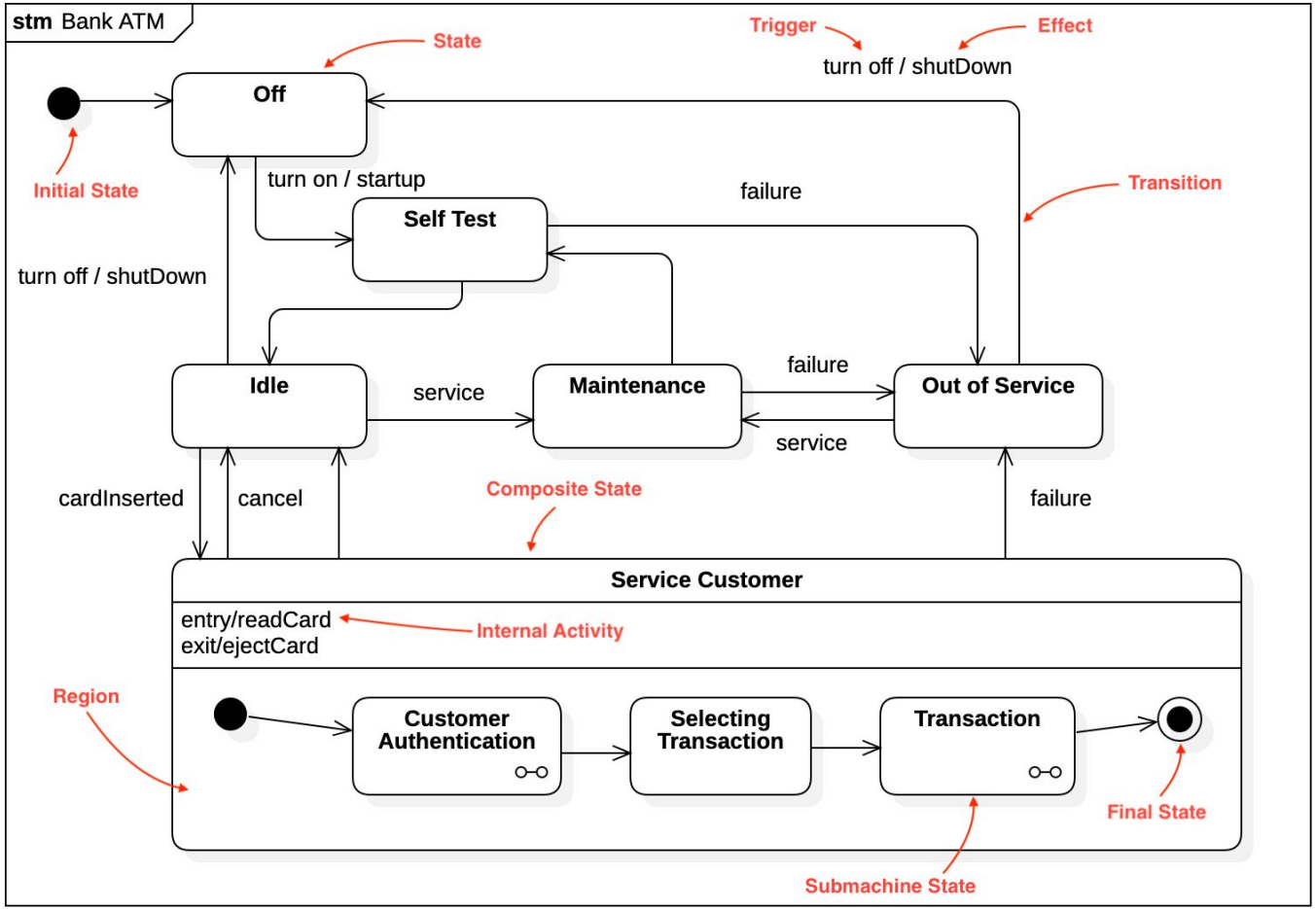

stm Bank ATM
State
Trigger
Effect
turn off / shutDown
Off
Initial State
turn on / startup
failure
Transition
Self Test
turn off / shutDown
Idle
service
Maintenance
failure
Out of Service
service
cardInserted
cancel
Composite State
failure
Service Customer
entry/readCard
exit/ejectCard
Internal Activity
Region
Customer Authentication
Selecting Transaction
Transaction
Final State
Submachine State

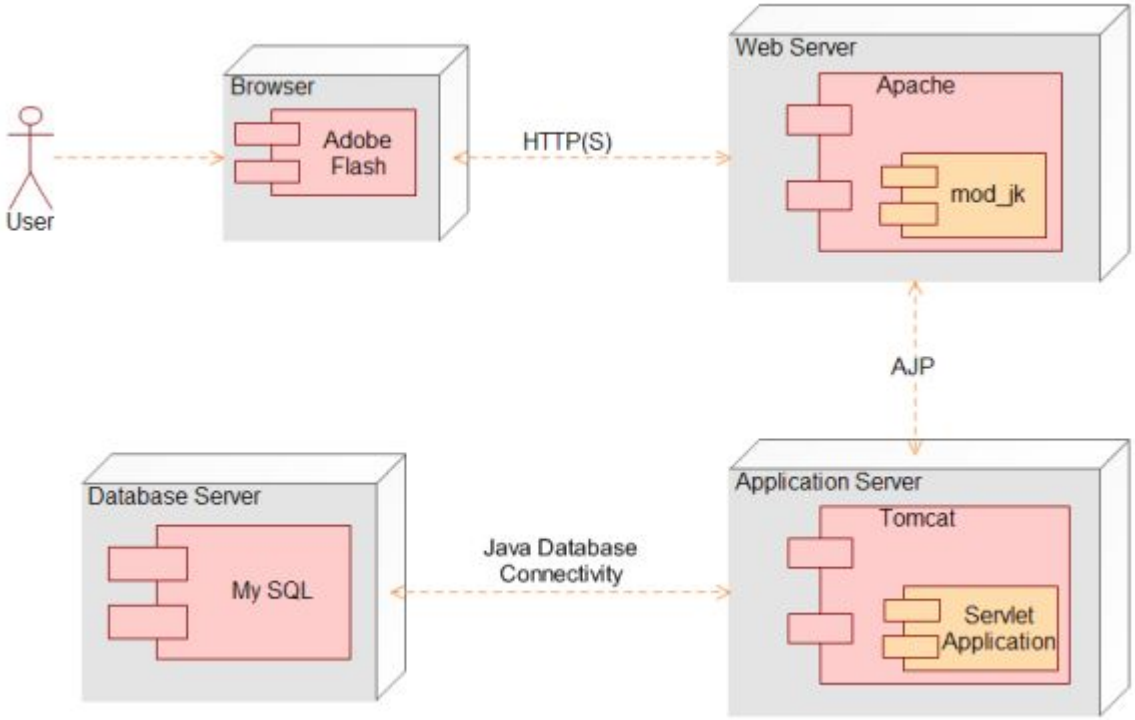
Diagrama de Despliegue

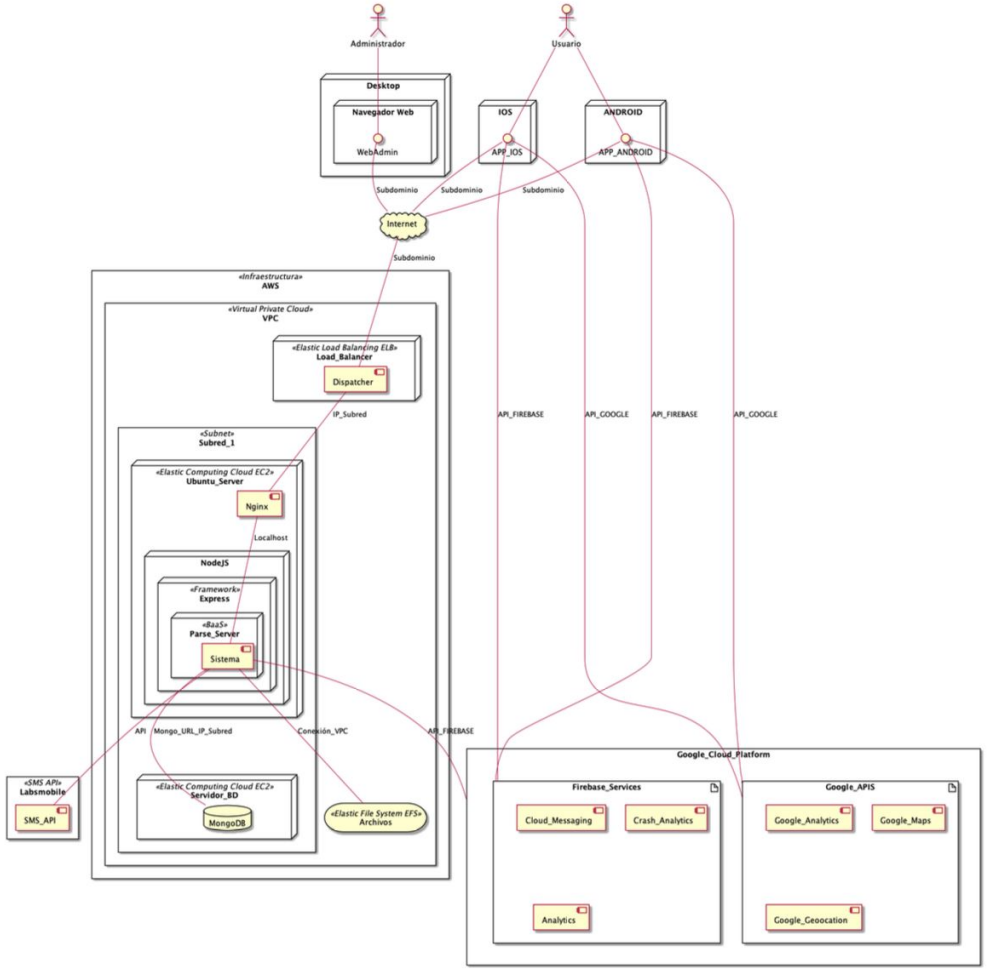
The deployment diagram describes the physical deployment of information generated by the software program on hardware components.

- Show which software elements are deployed by which hardware elements.
- Illustrate the runtime processing for hardware.
- Provide a view of the hardware system’s topology

Diagrama de Despliegue

- Have you identified the scope of your system? For example, you should know whether you are diagramming a single application or the deployment to a whole network of computers.
- What are the limitations of your physical hardware? What legacy systems will you need to interact with? Make sure that you know the operating software and protocols you will be working with and what monitoring you will be putting into place.
- Which distribution architecture are you using? You should know how many tiers your application will have and what application you will deploy to.
- Do you have all the nodes you need? Do you know how they are all connected?
- Do you know which components are going to be on which nodes?


User
Browser
Adobe Flash
HTTP(S)
Web Server
Apache
mod_jk
AJP
Database Server
My SQL
Java Database Connectivity
Application Server
Tomcat
Servlet Application


Administrador
Usuario
Desktop
Navegador Web
WebAdmin
IOS
APP_IOS
ANDROID
APP_ANDROID
Subdominio
Subdominio
Subdominio
Internet
Subdominio
«Infraestructura»
AWS
«Virtual Private Cloud»
VPC
«Elastic Load Balancing ELB»
Load_Balancer
Dispatcher
IP_Subred
API_FIREBASE
API_GOOGLE
API_FIREBASE
API_GOOGLE
«Subnet»
Subred_1
«Elastic Computing Cloud EC2»
Ubuntu_Server
Nginx
Localhost
NodeJS
«Framework»
Express
«BaaS»
Parse_Server
Sistema
API
Mongo_URL_IP_Subred
Conexión_VPC
API_FIREBASE
«SMS API»
Labsmobile
SMS_API
«Elastic Computing Cloud EC2»
Servidor_BD
MongoDB
«Elastic File System EFS»
Archivos
Google_Cloud_Platform
Firebase_Services
Cloud_Messaging
Crash_Analytics
Analytics
Google_APIS
Google_Analytics
Google_Maps
Google_Geoocation