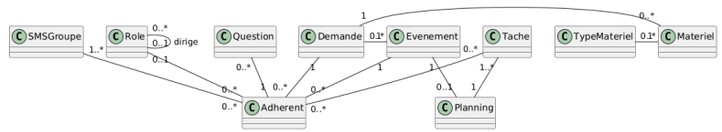
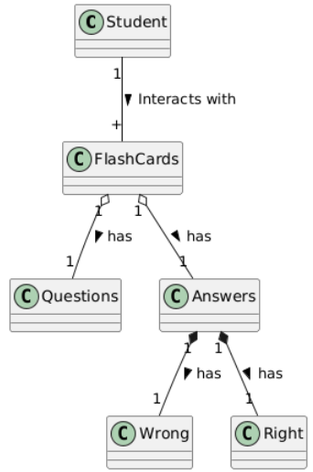
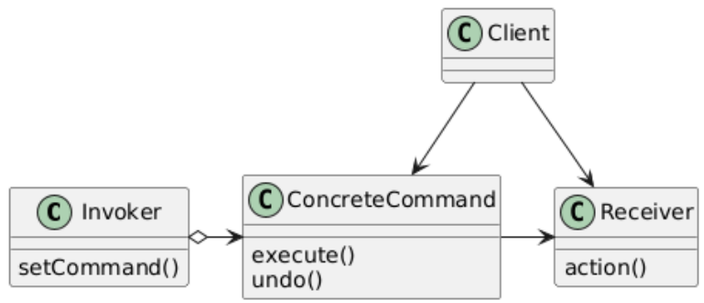
DIAGRAMAS DE CLASES

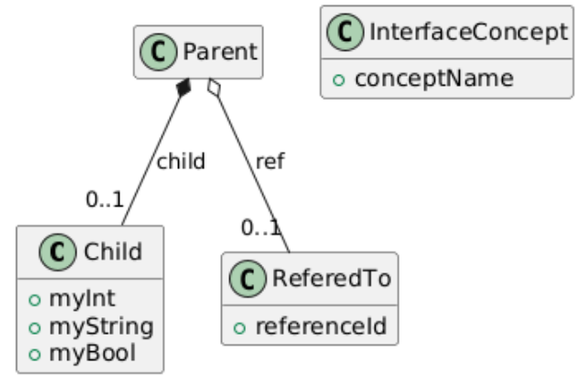
Repositorio: https://real-world-plantuml.com/?type=class

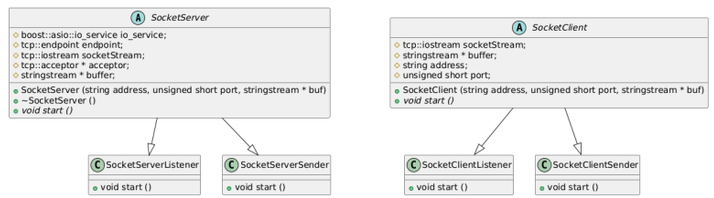
Número de diagramas: 68 Número total: 315

#	HU	HU (INGLÉS)	Código PlantUML	Diagrama
1	**Como** administrador del sistema, **quiero** organizar la planificación de eventos, asignar roles a los participantes y gestionar tareas, materiales, preguntas y solicitudes, **para** mantener un control completo de la logística y la comunicación con los participantes.	As a system administrator, I want to organize event planning, assign roles to participants, and manage tasks, materials, questions, and requests, so that I can maintain full control of logistics and communication with participants.	@startuml class "Role" as role class "Adherent" as adherent class "Materiel" as materiel class "TypeMateriel" as type class "Evenement" as evenement class "Planning" as planning class "Tache" as tache class "Demande" as demande class "Question" as question class "SMSGroupe" as sms evenement "1" -- "0..1" planning evenement "1" -- "0..*" adherent tache "1..*" -- "1" planning tache "0..*" -- "0..*" adherent role "0..1" -- "0..*" adherent question "0..*" -- "1" adherent sms "1..*" -- "0..*" adherent role "0..*" -- "0..1" role : dirige demande "0..*" - "1" evenement adherent "0..*" - "1" demande type "0..*" - "1" materiel demande "1" - "0..*" materiel @enduml	

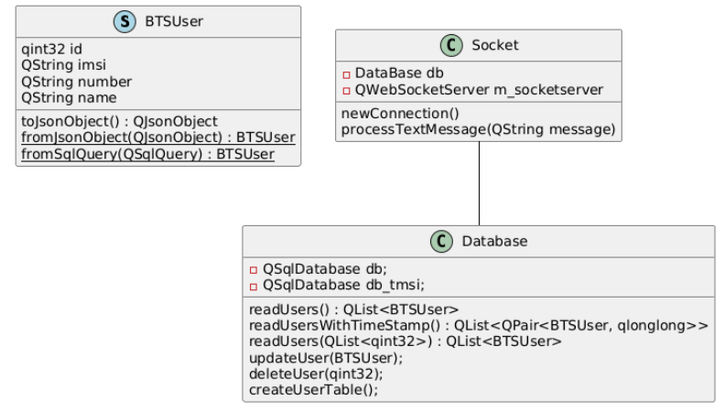
2	**Como** estudiante, **quiero** interactuar con tarjetas de estudio que contienen preguntas y respuestas, **para** poder aprender de manera estructurada, identificar las respuestas correctas y equivocadas, y mejorar mi comprensión del contenido.	As a student, I want to interact with flashcards containing questions and answers, so that I can learn in a structured way, identify correct and incorrect answers, and improve my understanding of the material.	@startuml Class Student Class FlashCards Class Questions Class Answers Student"1" -- "+"FlashCards : Interacts with > FlashCards"1" o-- "1"Questions : has > FlashCards"1" o-- "1"Answers : has > Answers"1" *-- "1"Wrong : has > Answers"1" *-- "1"Right : has > @enduml	

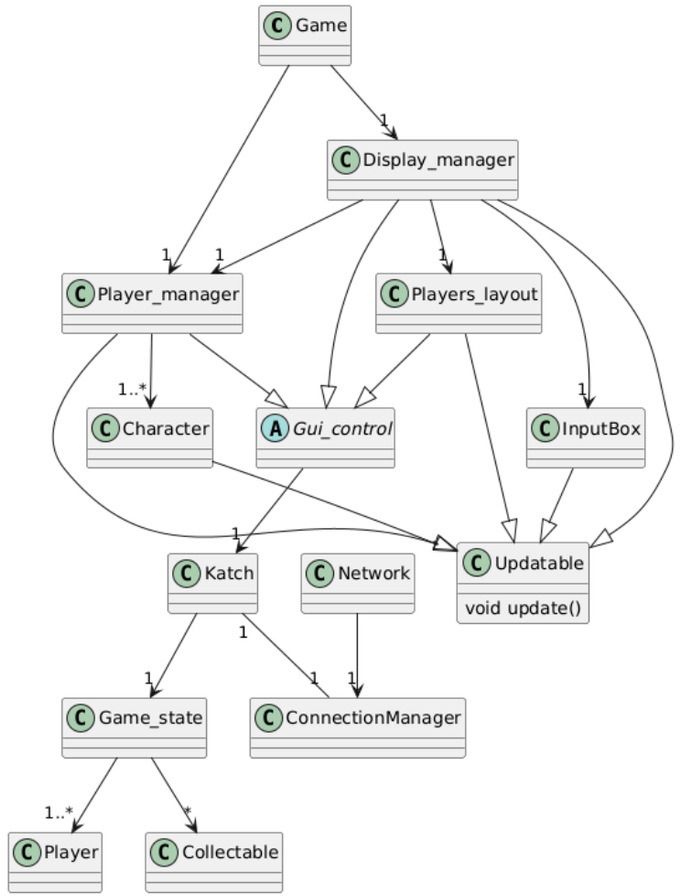
3	**Como** administrador de un sistema de control remoto para dispositivos, **quiero** que el sistema permita configurar un componente ejecutor con diferentes acciones encapsuladas, **para** que pueda reutilizar el mismo ejecutor con distintas operaciones sin modificar su código, y que las acciones puedan revertirse si es necesario.	As an administrator of a remote device control system, I want the system to allow configuring an executor component with different encapsulated actions, so that I can reuse the same executor with different operations without modifying its code, and actions can be undone if necessary.	@startuml
			class Invoker {
			setCommand()
			}
			class Client
			class Receiver {
			action()
			}
			class ConcreteCommand {
			execute()
			undo()
			}
			
			Client -right-> Receiver
			Client --> ConcreteCommand
			Invoker o-right-> ConcreteCommand
			Receiver <-left- ConcreteCommand
			@enduml
				

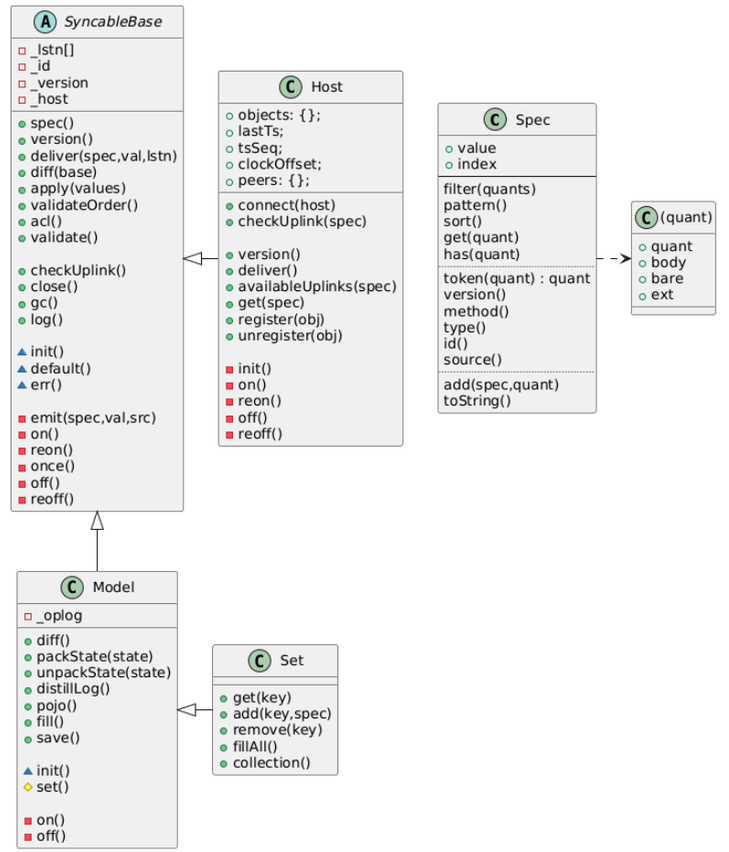
4	**Como** usuario del sistema, **quiero** que los elementos principales puedan relacionarse con elementos referenciados y contener elementos secundarios, **para** organizar la información jerárquicamente, mantener referencias claras y gestionar los datos de manera estructurada.	As a system user, I want parent elements to relate to referenced elements and contain child elements, so that information is organized hierarchically, references are clear, and data can be managed in a structured way.	@startuml
			
			class Child {
			+myInt
			+myString
			+myBool
			}
			hide empty members
			
			class Parent {
			}
			hide empty members
			
			class ReferedTo {
			+referenceId
			}
			hide empty members
			
			class InterfaceConcept {
			+conceptName
			}
			hide empty members
			
			Parent o-- "0..1" ReferedTo : ref
			Parent *-- "0..1" Child : child
			
			@enduml
				

5	**Como** desarrollador de aplicaciones de red, **quiero** implementar una arquitectura cliente-servidor usando clases abstractas y herencia, **para** facilitar la creación de componentes que envíen o reciban datos a través de sockets de manera reutilizable.	As a network application developer, I want to implement a client-server architecture using abstract classes and inheritance, so that I can facilitate the creation of components that send or receive data through sockets in a reusable way.	@startuml abstract class SocketServer { #boost::asio::io_service io_service; #tcp::endpoint endpoint; #tcp::iostream socketStream; #tcp::acceptor * acceptor; #stringstream * buffer; +SocketServer (string address, unsigned short port, stringstream * buf) +~SocketServer ()	

			+{abstract} void start ()
			}
			
			class SocketServerListener {
			+void start ()
			}
			
			class SocketServerSender {
			+void start ()
			}
			
			SocketServer -down-|>
			SocketServerListener
			SocketServer -down-|>
			SocketServerSender
			
			abstract class SocketClient {
			#tcp::iostream socketStream;
			#stringstream * buffer;
			#string address;
			#unsigned short port;
			
			+SocketClient (string address, unsigned
			short port, stringstream * buf)
			+{abstract} void start ()
			}
			
			class SocketClientListener {
			+void start ()
			}
			
			class SocketClientSender {
			+void start ()
			}
			
			SocketClient -down-|>
			SocketClientListener
			SocketClient -down-|> SocketClientSender
			
			@enduml	

6	**Como** administrador del sistema, **quiero** que el sistema gestione usuarios BTS mediante conexiones de socket y una base de datos, permitiendo crear, leer, actualizar y eliminar usuarios, así como procesar mensajes en tiempo real, **para** mantener la información de los usuarios actualizada y asegurar la comunicación eficiente con el servidor.	As a system administrator, I want the system to manage BTS users through socket connections and a database, allowing me to create, read, update, and delete users, as well as process messages in real time, so that user information is kept up to date and communication with the server is efficient.	@startuml
			
			class BTSUser << (S,lightblue) >> {
			qint32 id
			QString imsi
			QString number
			QString name
			
			toJsonObject() : QJsonObject
			{static} fromJsonObject(QJsonObject) : BTSUser
			{static} fromSqlQuery(QSqlQuery) : BTSUser
			}
			
			class Socket {
			newConnection()
			processTextMessage(QString message)
			
			-DataBase db
			-QWebSocketServer m_socketserver
			}
			
			class Database{
			readUsers() : QList<BTSUser>
			readUsersWithTimeStamp() :
			QList<QPair<BTSUser, qlonglong>>
			readUsers(QList<qint32>) :
			QList<BTSUser>
			updateUser(BTSUser);
			deleteUser(qint32);
			createUserTable();
			
			-QSqlDatabase db;
			-QSqlDatabase db_tmsi;
			}
			
			Socket -- Database
			@enduml
				

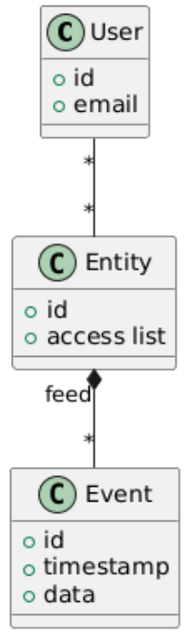
7	**Como** jugador de un videojuego multijugador, **quiero** que la aplicación gestione en todo momento a los jugadores, personajes, elementos coleccionables y la comunicación en red, **para** que pueda interactuar en una partida en tiempo real, visualizar correctamente la información en pantalla y mantener sincronizado mi progreso con los demás participantes.	As a multiplayer video game player, I want the application to manage players, characters, collectibles, and network communication at all times, so that I can interact in a real-time match, correctly visualize information on screen, and keep my progress synchronized with other participants.	@startuml
			
			Game --> "1" Player_manager
			Game --> "1" Display_manager
			
			abstract Updatable
			
			Players_layout --|> Updatable
			Player_manager --|> Updatable
			Display_manager --|> Updatable
			Character --|> Updatable
			InputBox --|> Updatable
			
			abstract Gui_control
			
			Players_layout --|> Gui_control
			Player_manager --|> Gui_control
			Display_manager --|> Gui_control
			
			Display_manager --> "1" InputBox
			Display_manager --> "1" Player_manager
			Display_manager --> "1" Players_layout
			
			Player_manager --> "1..*" Character
			
			Network --> "1" ConnectionManager
			
			Game_state --> "1..*" Player
			Game_state --> "*" Collectable
			
			Katch --> "1" Game_state
			Katch "1" -- "1" ConnectionManager
			Gui_control --> "1" Katch
			
			class Updatable {
			void update()
			}
			
			@enduml
				

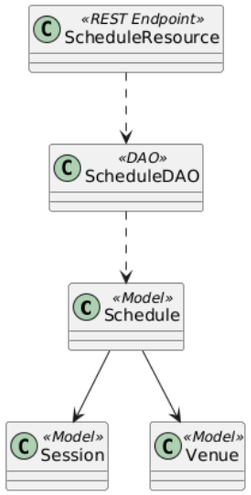
8	**Como** desarrollador de aplicaciones distribuidas, **quiero** un sistema que gestione objetos sincronizables, permitiendo definir su especificación, estado y versiones, **para** asegurar que los datos se mantengan consistentes entre diferentes nodos y se puedan actualizar, consultar o eliminar de forma controlada.	As a developer of distributed applications, I want a system that manages synchronizable objects, allowing the definition of their specification, state, and versions, so that data remains consistent across different nodes and can be updated, queried, or deleted in a controlled manner.	@startuml
			
			class Spec {
			+ value
			+ index
			--
			filter(quants)
			pattern()
			sort()
			get(quant)
			has(quant)
			..
			token(quant) : quant
			version()
			method()
			type()
			id()
			source()
			..
			add(spec,quant)
			toString()
			}
			
			class "Spec.Map" as SpecMap {
			+ map: {}
			
			+ add(versionVector)
			+ covers(version)
			+ toString(trim)
			}
			
			class "(quant)" as qq {
			+quant
			+body
			+bare
			+ext
			}
			
			Spec .> qq
				

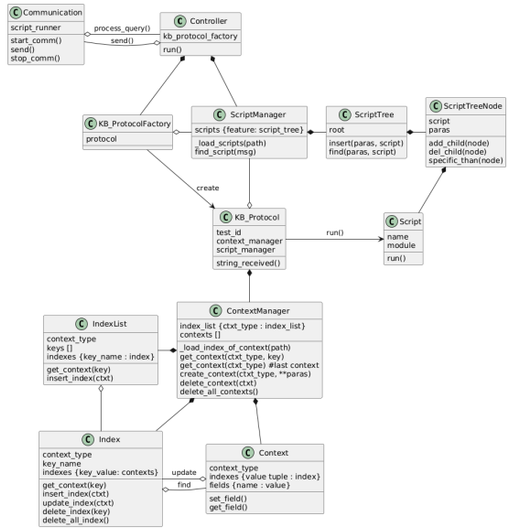
			abstract class SyncableBase {
			- _lstn[]
			- _id
			- _version
			- _host
			
			+ spec()
			+ version()
			+ deliver(spec,val,lstn)
			+ diff(base)
			+ apply(values)
			+ validateOrder()
			+ acl()
			+ validate()
			
			+ checkUplink()
			+ close()
			+ gc()
			+ log()
			
			~ init()
			~ default()
			~ err()
			
			- emit(spec,val,src)
			- on()
			- reon()
			- once()
			- off()
			- reoff()
			}
			
			class Model {
			- _oplog
			
			+ diff()
			+ packState(state)
			+ unpackState(state)
			+ distillLog()
			+ pojo()	

			+ fill()
			+ save()
			
			~ init()
			# set()
			
			- on()
			- off()
			}
			
			class Set {
			+ get(key)
			+ add(key,spec)
			+ remove(key)
			+ fillAll()
			+ collection()
			}
			
			class Host {
			+ objects: {};
			+ lastTs;
			+ tsSeq;
			+ clockOffset;
			+ peers: {};
			
			+ connect(host)
			+ checkUplink(spec)
			
			+ version()
			+ deliver()
			+ availableUplinks(spec)
			+ get(spec)
			+ register(obj)
			+ unregister(obj)
			
			- init()
			- on()
			- reon()
			- off()
			- reoff()	

			} Model -up-|> SyncableBase Set -up|> Model Host -left|> SyncableBase @enduml	

9	**Como** usuario del sistema, **quiero** que se gestionen las entidades y su feed de eventos, **para** poder acceder a la información relevante de cada entidad a la que tengo acceso y mantenerte actualizado sobre los eventos asociados.	As a system user, I want my entities and their event feeds to be managed, so that I can access relevant information for each entity I have access to and stay updated on the associated events.	@startuml
			
			class User {
			+ id
			+ email
			}
			
			class Entity {
			+ id
			+ access list
			}
			
			class Event {
			+ id
			+ timestamp
			+ data
			}
			
			User "*" -- "*" Entity
			Entity "feed" *-- "*" Event
			
			@enduml
				

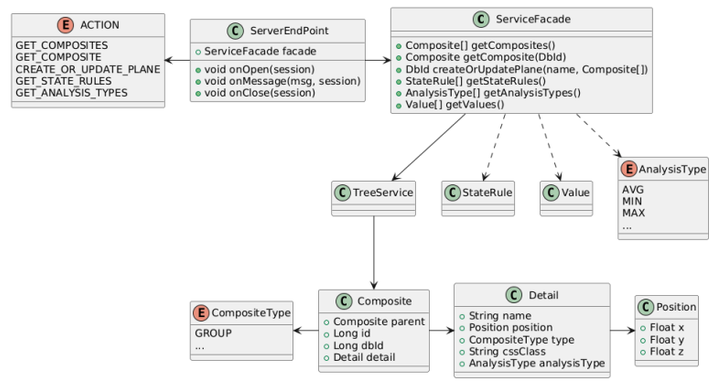
10	**Como** usuario del sistema, **quiero** que se gestione la información de los horarios, sesiones y recintos mediante un servicio REST que interactúe con la base de datos, **para** poder consultar y organizar los eventos de manera estructurada y accesible.	As a system user, I want schedule, session, and venue information to be managed through a REST service that interacts with the database, so that I can view and organize events in a structured and accessible way.	@startuml class Schedule << Model >> class Session << Model >> class Venue << Model >> class ScheduleDAO << DAO >> class ScheduleResource << REST Endpoint >> ScheduleResource ..> ScheduleDAO ScheduleDAO ..> Schedule Schedule --> Session Schedule --> Venue @enduml	

11	**Como** usuario del sistema, **quiero** que el controlador gestione la comunicación y la ejecución de protocolos y scripts, administrando contextos, índices y árboles de scripts, **para** poder procesar consultas, ejecutar acciones automatizadas y mantener la información organizada de manera coherente y eficiente.	As a system user, I want the controller to manage communication and the execution of protocols and scripts, handling contexts, indexes, and script trees, so that I can process queries, perform automated actions, and keep information organized coherently and efficiently.	@startuml
			class Controller {
			kb_protocol_factory
			
			run()
			}
			
			class KB_ProtocolFactory {
			protocol
			}
			
			class KB_Protocol {
			test_id
			context_manager
			script_manager
			
			string_received()
			}
			
			class Communication {
			script_runner
				

			start_comm()
			send()
			stop_comm()
			}
			
			class ScriptManager {
			scripts {feature: script_tree}
			
			_load_scripts(path)
			find_script(msg)
			}
			
			class ScriptTree{
			root
			
			insert(paras, script)
			find(paras, script)
			}
			
			class ScriptTreeNode{
			script
			paras
			
			add_child(node)
			del_child(node)
			specific_than(node)
			}
			
			class Script {
			name
			module
			run()
			}
			
			class ContextManager {
			index_list {ctxt_type : index_list}
			contexts []
			
			_load_index_of_context(path)
			get_context(ctxt_type, key)	

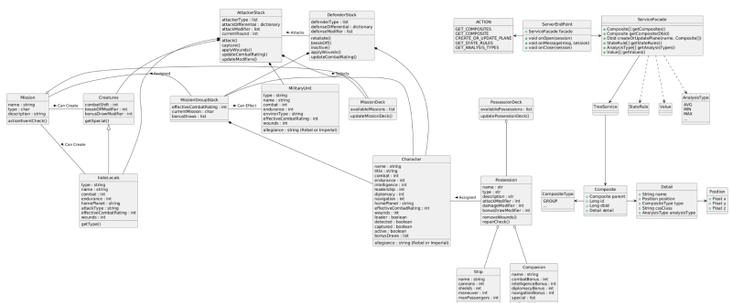
			get_context(ctxt_type) #last context
			create_context(ctxt_type, **paras)
			delete_context(ctxt)
			delete_all_contexts()
			}
			
			class IndexList {
			context_type
			keys []
			indexes {key_name : index}
			
			get_context(key)
			insert_index(ctxt)
			}
			
			class Index {
			context_type
			key_name
			indexes {key_value: contexts}
			
			get_context(key)
			insert_index(ctxt)
			update_index(ctxt)
			delete_index(key)
			delete_all_index()
			}
			
			class Context{
			context_type
			indexes {value tuple : index}
			fields {name : value}
			
			set_field()
			get_field()
			}
			
			Communication o- Controller:
			process_query()
			Communication -o Controller: send()
			Controller *-- KB_ProtocolFactory	

			Controller *-- ScriptManager KB_ProtocolFactory --> KB_Protocol: create KB_ProtocolFactory o- ScriptManager KB_Protocol o-up- ScriptManager KB_Protocol *-- ContextManager KB_Protocol -> Script: run() ScriptManager *- ScriptTree ScriptTree *- ScriptTreeNode ScriptTreeNode *-- Script ContextManager *-left- IndexList ContextManager *-- Context ContextManager *-- Index IndexList o-- Index Index o- Context: find Context o-left- Index: update @enduml	

12	**Como** usuario del sistema, **quiero** interactuar con un servicio que gestione composiciones y detalles asociados, permitiéndole crear o actualizar planos, consultar composiciones, reglas de estado, tipos de análisis y valores, **para** poder organizar y analizar la información de manera estructurada y accesible.	As a system user, I want to interact with a service that manages composites and their associated details, allowing me to create or update planes, query composites, state rules, analysis types, and values, so that I can organize and analyze information in a structured and accessible way.	@startuml
			
			class ServiceFacade {
			+Composite[] getComposites()
			+Composite getComposite(DbId)
			+DbId createOrUpdatePlane(name, Composite[])
			+StateRule[] getStateRules()
			+AnalysisType[] getAnalysisTypes()
			+Value[] getValues()
			}
			class TreeService {
			}
			class ServerEndPoint {
			+ServiceFacade facade
			+void onOpen(session)
				

			+void onMessage(msg, session)
			+void onClose(session)
			}
			class Composite {
			+Composite parent
			+Long id
			+Long dbId
			+Detail detail
			}
			class Detail {
			+String name
			+Position position
			+CompositeType type
			+String cssClass
			+AnalysisType analysisType
			}
			class Position {
			+Float x
			+Float y
			+Float z
			}
			class StateRule {
			}
			class Value {
			}
			enum AnalysisType {
			AVG
			MIN
			MAX
			...
			}
			enum CompositeType {
			GROUP
			...
			}
			enum ACTION {
			GET_COMPOSITES
			GET_COMPOSITE	

			CREATE_OR_UPDATE_PLANE GET_STATE_RULES GET_ANALYSIS_TYPES } ServiceFacade --> TreeService ServiceFacade ..> AnalysisType ServiceFacade ..> StateRule ServiceFacade ..> Value ServerEndPoint -> ServiceFacade TreeService --> Composite Composite -> Detail Detail ->Position ACTION <- ServerEndPoint CompositeType <- Composite @enduml	

13	**Como** jugador del juego, **quiero** gestionar personajes, posesiones, naves, compañeros, misiones y unidades militares, así como interactuar con el servicio del sistema para consultar composiciones y reglas de análisis, **para** poder organizar mis recursos, planificar estrategias y ejecutar acciones de manera efectiva durante el juego.	As a game player, I want to manage characters, possessions, ships, companions, missions, and military units, as well as interact with the system service to query composites and analysis rules, so that I can organize my resources, plan strategies, and perform actions effectively during the game.	@startuml
			Hide circle
			
			' Classes PossessionInfo
			class PossessionDeck {
			availablePossessions : list
			updatePossessionDeck()
			}
			class Possession {
			name : str
			type : str
			description : str
			attackModifier : int
			damageModifier : int
			bonusDrawModifier : int
			removeWounds()
			repairCheck()
			}
			class Ship {
			name : string
			cannons : int
			shields : int
			maneuver : int
				

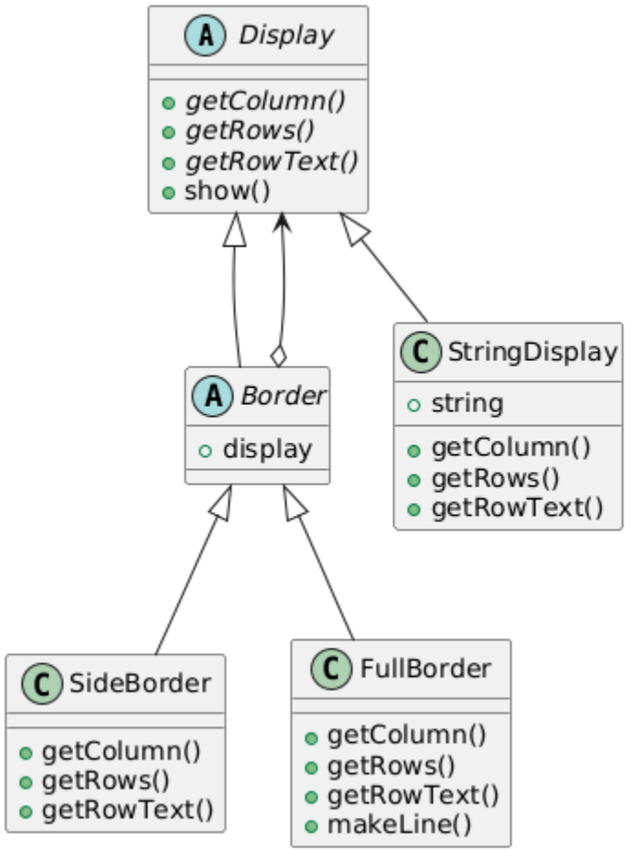
			maxPassengers : int
			}
			class Companion {
			name : string
			combatBonus : int
			intelligenceBonus : int
			diplomacyBonus : int
			navigationBonus : int
			special : list
			getSpecial()
			}
			
			PossessionDeck o-- Possession
			Possession o-- Ship
			Possession o-- Companion
			
			' Classes Units
			class Character {
			allegiance : string (Rebel or Imperial)
			name : string
			title : string
			combat : int
			endurance : int
			intelligence : int
			leadership : int
			diplomacy : int
			navigation : int
			homePlanet : string
			effectiveCombatRating : int
			wounds : int
			leader : boolean
			detected : boolean
			captured : boolean
			active : boolean
			bonusDraws : list
			}
			
			class IrateLocals {
			type : string	

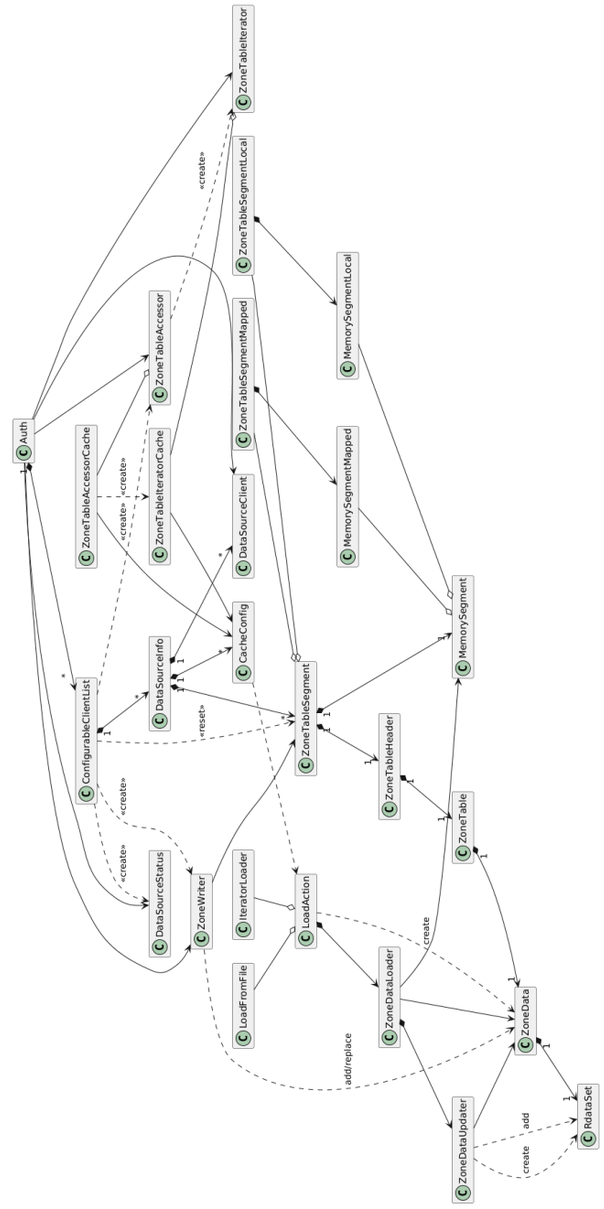
```
        name : string
        combat : int
        endurance : int
        homePlanet : string
        attackType : string
        effectiveCombatRating : int
        wounds : int
        getType()
}

class Creatures {
        combatShift : int
        breakOffModifier : int
        bonusDrawModifier : int
        getSpecial()
}

class MilitaryUnit {
        allegiance : string (Rebel or
Imperial)
        type : string
        name : string
        combat : int
        endurance : int
        environType : string
        effectiveCombatRating : int
        wounds : int
}

Creatures <|-- IrateLocals
Character - Possession : < Assigned

' Classes Stacks
class MissionGroupStack {
        effectiveCombatRating : int
        currentMission : char
        bonusDraws : list
}
class MissionDeck {
        availableMissions : list
```

			updateMissionDeck()
			}
			class Mission {
			name : string
			type : char
			description : string
			actionEventCheck()
			}
			
			class DefenderStack {
			defenderType : list
			defenseDifferential : dictionary
			defenseModifier : list
			retaliate()
			breakOff()
			inactive()
			applyWounds()
			updateCombatRating()
			}
			
			class AttackerStack {
			attackerType : list
			attackDifferential : dictionary
			attackModifier : list
			currentRound : int
			attack()
			capture()
			applyWounds()
			updateCombatRating()
			updateModifiers()
			}
			
			MissionGroupStack *-- Character
			MissionGroupStack - MissionDeck : Selects >
			MissionGroupStack - MilitaryUnit : > Can Effect
			Mission - MissionGroupStack : < Assigned
			Mission o- MissionDeck
			Mission - IrateLocals : Can Create >	

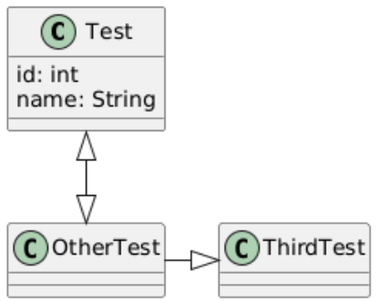
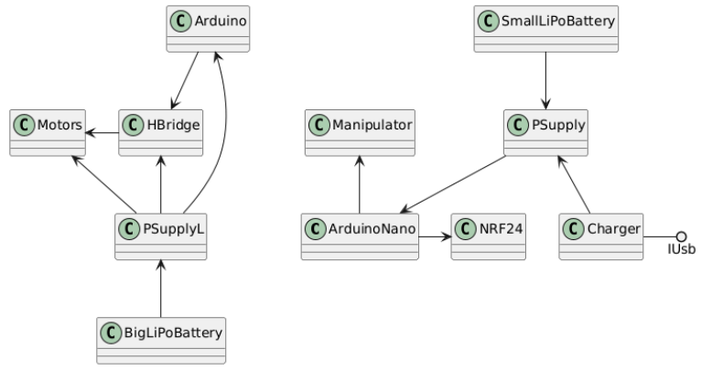
			Mission - Creatures : Can Create >
			DefenderStack *-- Character
			DefenderStack *-- MissionGroupStack
			AttackerStack - DefenderStack : Attacks >
			AttackerStack *-- Character
			AttackerStack *-- IrateLocals
			AttackerStack *-- Creatures
			AttackerStack *-- MilitaryUnit
			
			' Service Classes
			class ServiceFacade {
			+Composite[] getComposites()
			+Composite getComposite(DbId)
			+DbId createOrUpdatePlane(name, Composite[])
			+StateRule[] getStateRules()
			+AnalysisType[] getAnalysisTypes()
			+Value[] getValues()
			}
			class TreeService {
			}
			class ServerEndPoint {
			+ServiceFacade facade
			+void onOpen(session)
			+void onMessage(msg, session)
			+void onClose(session)
			}
			class Composite {
			+Composite parent
			+Long id
			+Long dbId
			+Detail detail
			}
			class Detail {
			+String name
			+Position position
			+CompositeType type
			+String cssClass
			+AnalysisType analysisType
			}	

			class Position {
			+Float x
			+Float y
			+Float z
			}
			class StateRule {
			}
			class Value {
			}
			enum AnalysisType {
			AVG
			MIN
			MAX
			...
			}
			enum CompositeType {
			GROUP
			...
			}
			enum ACTION {
			GET_COMPOSITES
			GET_COMPOSITE
			CREATE_OR_UPDATE_PLANE
			GET_STATE_RULES
			GET_ANALYSIS_TYPES
			}
			
			ServiceFacade --> TreeService
			ServiceFacade ..> AnalysisType
			ServiceFacade ..> StateRule
			ServiceFacade ..> Value
			ServerEndPoint -> ServiceFacade
			TreeService --> Composite
			Composite -> Detail
			Detail ->Position
			ACTION <- ServerEndPoint
			CompositeType <- Composite
			
			@enduml	

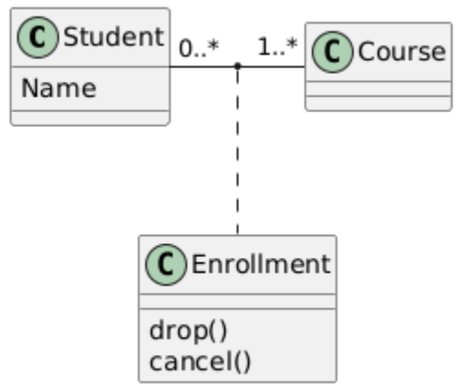
14	**Como** usuario del sistema que visualiza información, **quiero** poder mostrar textos simples y agregar distintos tipos de bordes a las visualizaciones, **para** poder presentar la información de manera clara y estructurada, personalizando su apariencia según mis necesidades.	As a system user visualizing information, I want to display simple texts and add different types of borders to visualizations, so that I can present information clearly and structured, customizing its appearance according to my needs.	@startuml
			
			abstract Display {
			{abstract} +getColumn()
			{abstract} +getRows()
			{abstract} +getRowText()
			+show()
			}
			
			class StringDisplay {
			+string
			+getColumn()
			+getRows()
			+getRowText()
			}
			
			abstract Border {
			+display
			}
			
			class SideBorder {
			+getColumn()
			+getRows()
			+getRowText()
			}
			class FullBorder {
			+getColumn()
			+getRows()
			+getRowText()
			+makeLine()
			}
			
			StringDisplay -up-|> Display
			Border -up-|> Display
			Border o-up->Display
			SideBorder -up-|> Border
			FullBorder -up-|> Border
			
			@enduml
				

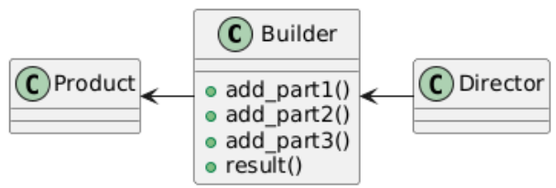
15	**Como** administrador o usuario del sistema de datos, **quiero** poder gestionar y configurar clientes de datos, acceder y actualizar las diferentes zonas de información, y supervisar el estado de las fuentes de datos, **para** asegurar que toda la información se procese, almacene y actualice correctamente, manteniendo la coherencia y el control sobre los distintos segmentos y registros dentro del sistema.	As an administrator or user of a data system, I want to manage and configure data clients, access and update different information zones, and monitor the status of data sources, so that all information is processed, stored, and updated correctly, maintaining consistency and control over the different segments and records within the system.	@startuml hide members Auth "1" *--> "*" ConfigurableClientList Auth --> DataSourceClient Auth --> ZoneWriter Auth --> ZoneTableAccessor Auth --> DataSourceStatus Auth --> ZoneTableIterator ConfigurableClientList "1" *--> "*" DataSourceInfo ConfigurableClientList ..> ZoneTableSegment : <<reset>> ConfigurableClientList ..> DataSourceStatus : <<create>> ConfigurableClientList ..> ZoneWriter : <<create>> ConfigurableClientList ..> ZoneTableAccessor : <<create>> DataSourceInfo "1" *--> "*" DataSourceClient DataSourceInfo "1" *--> "*" CacheConfig DataSourceInfo "1" *--> "*" ZoneTableSegment ZoneTableAccessor ..> ZoneTableIterator : <<create>> ZoneTableAccessorCache --> CacheConfig ZoneTableAccessorCache ..> ZoneTableIteratorCache : <<create>> ZoneTableAccessorCache --o ZoneTableAccessor ZoneTableIteratorCache --o ZoneTableIterator	

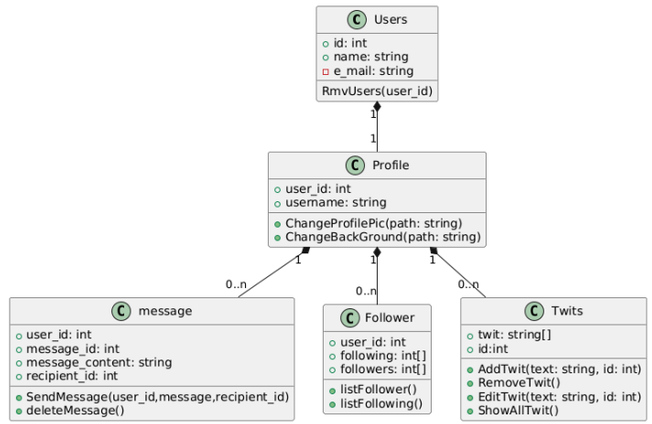
			ZoneTableIteratorCache --> CacheConfig ZoneWriter --> ZoneTableSegment ZoneWriter ..> ZoneData : add/replace ZoneTableSegment "1" *--> "1" ZoneTableHeader ZoneTableSegment "1" *--> "1" MemorySegment CacheConfig ..> LoadAction LoadAction ..> ZoneData : create LoadAction *--> ZoneDataLoader ZoneDataLoader --> ZoneData ZoneDataLoader *--> ZoneDataUpdater ZoneDataLoader --> MemorySegment ZoneDataUpdater --> ZoneData ZoneDataUpdater ..> RdataSet : create ZoneDataUpdater ..> RdataSet : add ZoneTableHeader "1" *--> "1" ZoneTable ZoneTable "1" *--> "1" ZoneData ZoneData "1" *--> "1" RdataSet LoadFromFile --o LoadAction IteratorLoader --o LoadAction MemorySegmentMapped --o MemorySegment MemorySegmentLocal --o MemorySegment ZoneTableSegmentMapped --o ZoneTableSegment ZoneTableSegmentLocal --o ZoneTableSegment	

			ZoneTableSegmentMapped *--> MemorySegmentMapped ZoneTableSegmentLocal *--> MemorySegmentLocal @enduml	
16	**Como** usuario del sistema que gestiona pruebas, **quiero** poder relacionar diferentes tipos de pruebas entre sí, **para** organizar y conectar información de manera que pueda ver cómo se vinculan distintas pruebas y sus derivados dentro del sistema.	As a user of the system that manages tests, I want to be able to relate different types of tests to each other, so that I can organize and connect information and see how different tests and their derivatives are linked within the system.	@startuml class Test Test : id: int Test : name: String class OtherTest Test <|--|> OtherTest OtherTest -|> ThirdTest @enduml	
17	**Como** operador del sistema robótico, **quiero** que los microcontroladores Arduino gestionen los manipuladores, motores y la comunicación vía NRF24, así como el suministro de energía mediante baterías y fuentes de alimentación, **para** controlar de manera eficiente los movimientos y operaciones del robot.	As a robotic system operator, I want the Arduino microcontrollers to manage manipulators, motors, and NRF24 communication, as well as power supply via batteries and power sources, so that I can efficiently control the robot's movements and operations.	@startuml class ArduinoNano class Manipulator class NRF24 class PSupply class Charger class SmallLiPoBattery ArduinoNano -up-> Manipulator ArduinoNano -> NRF24 PSupply --> ArduinoNano PSupply <-up- SmallLiPoBattery Charger -up-> PSupply Charger -() IUsb ' Robot classes class Arduino class NRF24 class Motors	

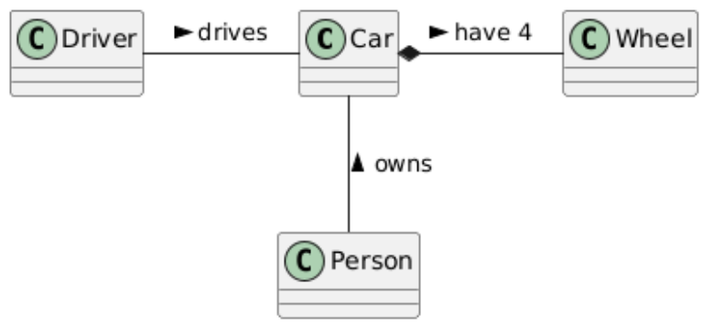
			class HBridge class PSupplyL class BigLiPoBattery Arduino -down-> HBridge HBridge -left-> Motors PSupplyL -up-> Arduino PSupplyL -up-> HBridge PSupplyL -up-> Motors PSupplyL <-down- BigLiPoBattery @enduml	

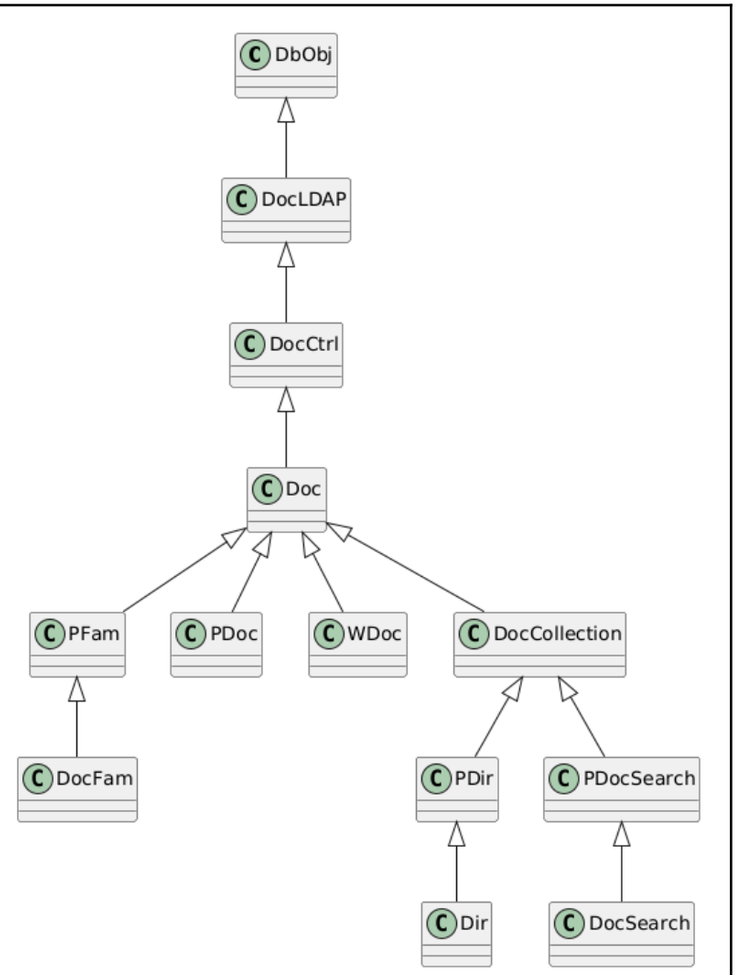
18	**Como** administrador del sistema académico, **quiero** poder inscribir estudiantes en cursos específicos y permitir que cancelen la inscripción cuando sea necesario, **para** asegurar que los registros de cada curso y de cada estudiante estén siempre correctos y actualizados.	As an administrator of the academic system, I want to enroll students in specific courses and allow them to cancel enrollment when necessary, so that the records of each course and each student are always accurate and up-to-date.	@startuml
			class Student {
			Name
			}
			Student "0..*" - "1..*" Course
			(Student, Course) . Enrollment
			
			class Enrollment {
			drop()
			cancel()
			}
			@enduml
				

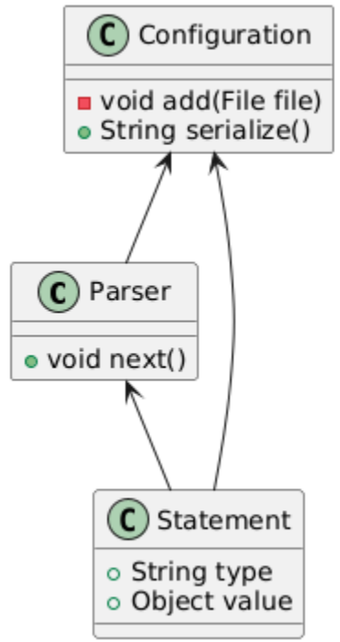
19	**Como** usuario del sistema que diseña productos, **quiero** poder ensamblar un producto de manera guiada, agregando cada parte de forma secuencial, **para** obtener un producto final completo y funcional que cumpla con los requisitos que necesito.	As a user of the product design system, I want to assemble a product step by step, adding each part sequentially, so that I can obtain a complete and functional product that meets the requirements.	@startuml
			
			Builder <--R Director
			Product <--R Builder
			
			class Builder {
			+ add_part1()
			+ add_part2()
			+ add_part3()
			+ result()
			}
			
			@enduml
				

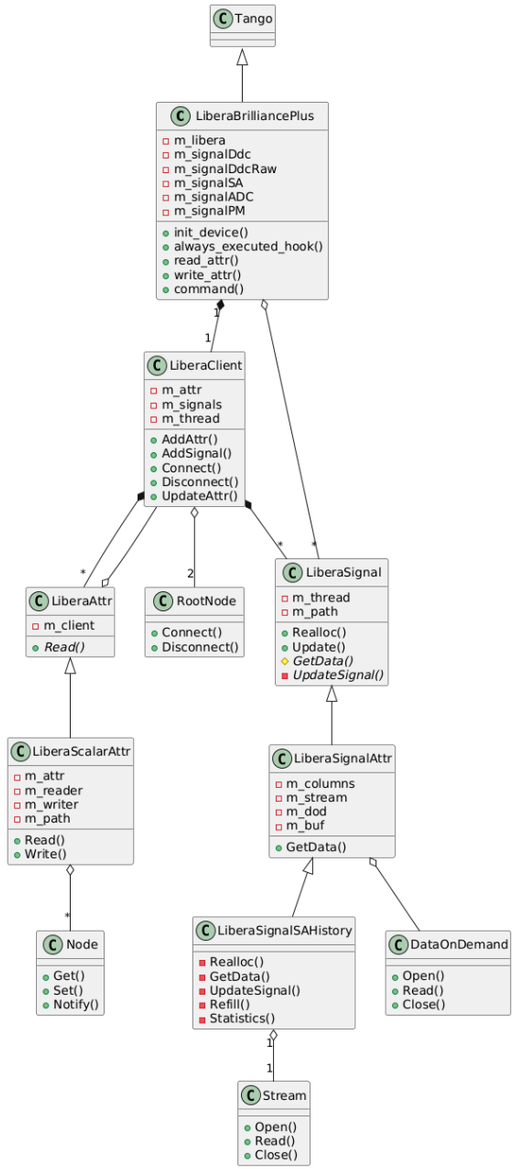
20	**Como** usuario del sistema de red social, **quiero** poder crear y personalizar mi perfil, enviar y eliminar mensajes, seguir a otros usuarios y publicar, editar o eliminar mis twits, **para** interactuar con otros usuarios, compartir información y gestionar mi actividad en la plataforma de manera sencilla y organizada.	As a user of the social network system, I want to create and customize my profile, send and delete messages, follow others, and post, edit, or delete my tweets, so that I can interact with others, share information, and manage my activity on the platform easily and efficiently.	@startuml
			
			Users "1" *-- "1" Profile
			Profile "1" *-- "0..n" message
			Profile "1" *-- "0..n" Follower
			Profile "1" *-- "0..n" Twits
			
			class Users{
			+id: int
			+name: string
			-e_mail: string
			RmvUsers(user_id)
			}
			class Profile{
			+user_id: int
			+username: string
			+ChangeProfilePic(path: string)
			+ChangeBackGround(path: string)
			
			}
			class message{
			+user_id: int
			+message_id: int
			+message_content: string
			+recipient_id: int
			+SendMessage(user_id,message,recipient_id)
			+deleteMessage()
			
			}
			
			class Follower{
			+user_id: int
			+following: int[]
			+followers: int[]
			+listFollower()
			+listFollowing()
				

			}
			
			class Twits{
			+twit: string[]
			+id:int
			+AddTwit(text: string, id: int)
			+RemoveTwit()
			+EditTwit(text: string, id: int)
			+ShowAllTwit()
			
			}
			
			@enduml	

21	**Como** usuario del sistema de gestión de vehículos, **quiero** poder asociar conductores con los autos que manejan, conocer los propietarios de cada auto y verificar los componentes principales como las ruedas, **para** asegurar un control claro sobre quién conduce cada vehículo y el estado de sus elementos esenciales.	As a user of the vehicle management system, I want to associate drivers with the cars they drive, know the owners of each car, and check main components such as the wheels, so that I can maintain clear control over who drives each vehicle and the status of its essential elements.	@startuml class Car Driver - Car : drives > Car *- Wheel : have 4 > Car -- Person : < owns @enduml	

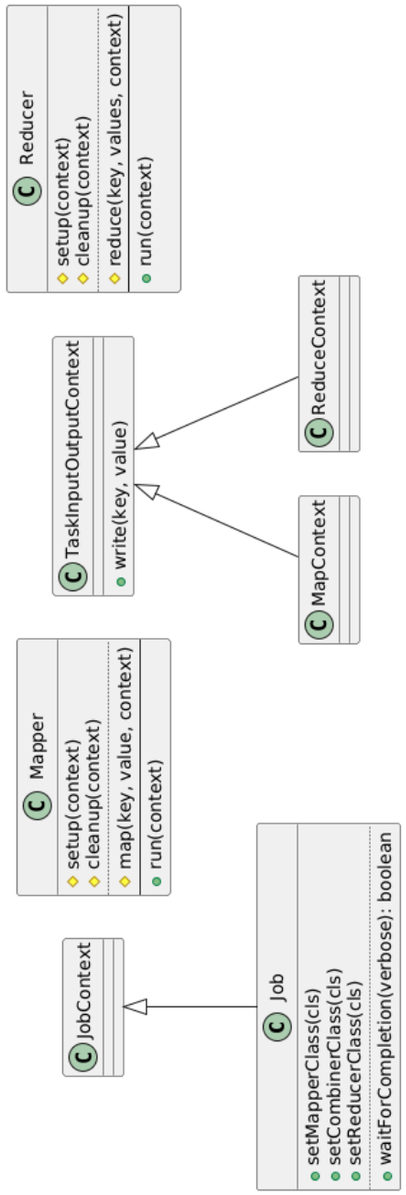
22	**Como** usuario del sistema de gestión de documentos, **quiero** organizar documentos y colecciones en distintas categorías y niveles de jerarquía, **para** poder acceder, controlar y buscar fácilmente cualquier tipo de documento según su clasificación y relación con otros documentos dentro del sistema.	As a user of the document management system, I want to organize documents and collections into different categories and hierarchical levels, so that I can access, control, and search for any type of document easily according to its classification and relationship with other documents within the system.	@startuml DbObj <|-- DocLDAP DocLDAP <|-- DocCtrl DocCtrl <|-- Doc Doc <|-- PFam PFam <|-- DocFam Doc <|-- PDoc Doc <|-- WDoc Doc <|-- DocCollection DocCollection <|-- PDir PDir <|-- Dir DocCollection <|-- PDocSearch PDocSearch <|-- DocSearch @enduml	

23	**Como** usuario del sistema que gestiona configuraciones, **quiero** poder crear y organizar declaraciones o instrucciones, procesarlas secuencialmente mediante un parser y generar una configuración completa que pueda ser almacenada o exportada, **para** asegurar que las configuraciones estén bien estructuradas, sean consistentes y fáciles de utilizar dentro del sistema.	As a user of the configuration management system, I want to create and organize statements or instructions, process them sequentially through a parser, and generate a complete configuration that can be stored or exported, so that configurations are well-structured, consistent, and easy to use within the system.	@startuml
			
			class Parser {
			+ void next()
			}
			
			class Statement {
			+ String type
			+ Object value
			}
			
			class Configuration {
			- void add(File file)
			+ String serialize()
			}
			
			Parser <-- Statement
			Configuration <-- Parser
			Configuration <-- Statement
			
			@enduml
				

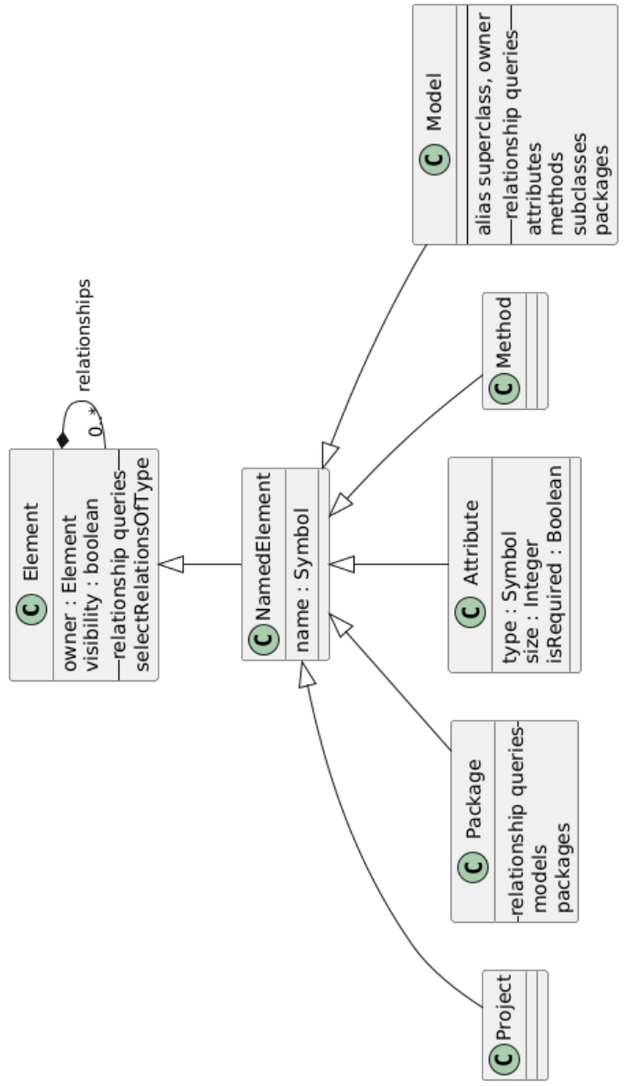
24	**Como** usuario del sistema de monitoreo de dispositivos Libera, **quiero** gestionar y supervisar dispositivos y sus componentes de forma centralizada, **para** poder leer, actualizar y controlar atributos y señales de cada dispositivo de manera confiable y eficiente.	As a user of the Libera device monitoring system, I want to manage and supervise devices and their components centrally, so that I can read, update, and control the attributes and signals of each device reliably and efficiently.	@startuml
			
			class LiberaBrillliancePlus {
			+init_device()
			+always_executed_hook()
			+read_attr()
			+write_attr()
			+command()
			-m_libera
			-m_signalDdc
			-m_signalDdcRaw
			-m_signalSA
			-m_signalADC
			-m_signalPM
			}
			
			"Tango::Device_4Impl" <|--
			LiberaBrillliancePlus
			
			class LiberaClient {
			+AddAttr()
			+AddSignal()
			+Connect()
			+Disconnect()
			+UpdateAttr()
			-m_attr
			-m_signals
			-m_thread
			}
			
			LiberaBrillliancePlus "1" *-- "1" LiberaClient
			
			class LiberaAttr {
			{abstract} +Read()
			-m_client
			}
			
			class LiberaScalarAttr {
			+Read()
			+Write()
				

			-m_attr
			-m_reader
			-m_writer
			-m_path
			}
			
			LiberaAttr <|-- LiberaScalarAttr
			
			class LiberaSignal {
			+Realloc()
			+Update()
			{abstract} #GetData()
			{abstract} -UpdateSignal()
			-m_thread
			-m_path
			}
			
			class LiberaSignalAttr {
			+GetData()
			-m_columns
			-m_stream
			-m_dod
			-m_buf
			}
			
			LiberaSignal <|-- LiberaSignalAttr
			
			class LiberaSignalSAHistory {
			-Realloc()
			-GetData()
			-UpdateSignal()
			-Refill()
			-Statistics()
			}
			
			LiberaSignalAttr <|-- LiberaSignalSAHistory
			
			LiberaBrilliancePlus o-- "*" LiberaSignal
			
			LiberaClient *-- "*" LiberaAttr	

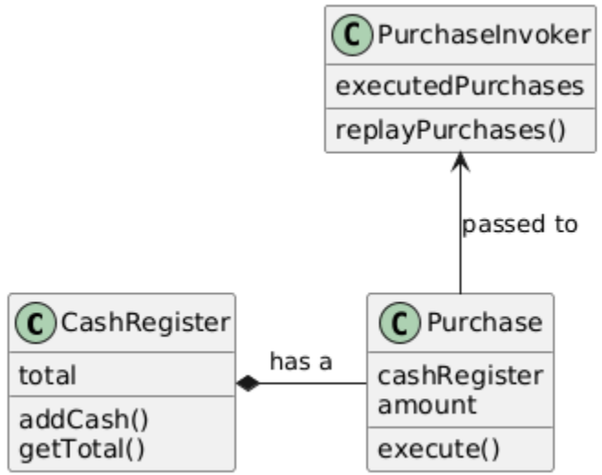
			LiberaClient --o LiberaAttr
			
			LiberaClient *-- "*" LiberaSignal
			
			class RootNode {
			  +Connect()
			  +Disconnect()
			}
			
			class Node {
			  +Get()
			  +Set()
			  +Notify()
			}
			
			class Stream {
			  +Open()
			  +Read()
			  +Close()
			}
			
			class DataOnDemand {
			  +Open()
			  +Read()
			  +Close()
			}
			
			LiberaClient o-- "2" RootNode
			LiberaScalarAttr o-- "*" Node
			LiberaSignalSAHistory "1" o-- "1" Stream
			LiberaSignalAttr o-- DataOnDemand
			
			@enduml	

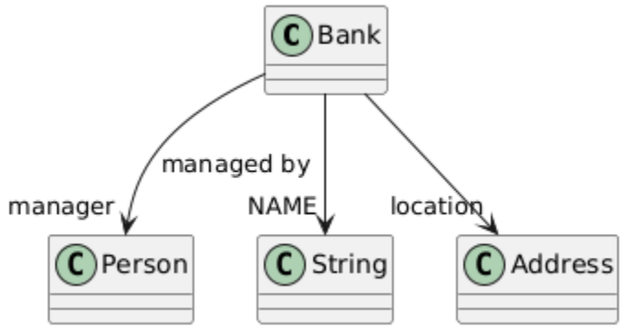
25	**Como** usuario del sistema de procesamiento de datos, **quiero** configurar y ejecutar trabajos que transformen y procesen grandes volúmenes de información, **para** obtener resultados procesados de manera eficiente y controlada.	As a user of the data processing system, I want to configure and execute jobs that transform and process large volumes of information, so that I can obtain processed results efficiently and in a controlled manner.	@startuml
			class Job {
			+setMapperClass(cls)
			+setCombinerClass(cls)
			+setReducerClass(cls)
			..
			+waitForCompletion(verbose): boolean
			}
			
			class Mapper {
			# setup(context)
			# cleanup(context)
			..
			# map(key, value, context)
			--
			+ run(context)
			}
			
			class MapContext {
			
			}
			
			class Reducer {
			# setup(context)
			# cleanup(context)
			..
			# reduce(key, values, context)
			--
			+ run(context)
			
			}
			
			class TaskInputOutputContext {
			+ write(key, value)
			}
			
			TaskInputOutputContext <|-- MapContext
			TaskInputOutputContext <|--
			ReduceContext
			JobContext <|-- Job
				

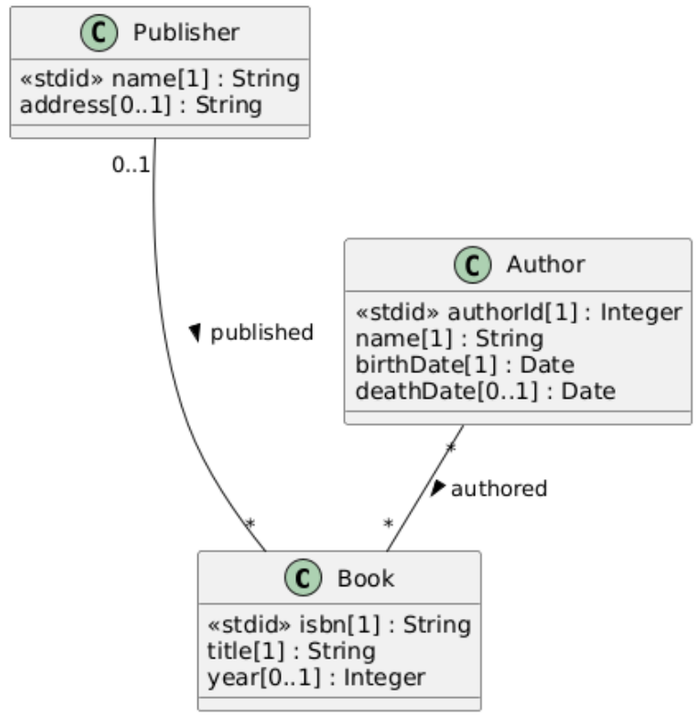
			@enduml	

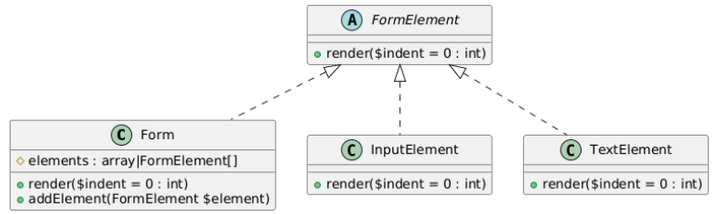
26	**Como** usuario del sistema, **quiero** gestionar proyectos, paquetes, atributos y métodos, **para** organizar y estructurar los elementos de un modelo de manera clara.	As a system user, I want to manage projects, packages, attributes, and methods, so that I can organize and structure the elements of a model clearly.	@startuml
			
			Element <|-- NamedElement
			NamedElement <|-- Project
			NamedElement <|-- Package
			NamedElement <|-- Attribute
			NamedElement <|-- Method
			NamedElement <|-- Model
			
			Element *- "0..*" Element : relationships
			
			class Element {
			  owner : Element
			  visibility : boolean
			  -- relationship queries --
			  selectRelationsOfType
			}
			class NamedElement {
			  name : Symbol
			}
			
			class Project
			
			class Package {
			  -- relationship queries --
			  models
			  packages
			}
			
			class Attribute {
			  type : Symbol
			  size : Integer
			  isRequired : Boolean
			}
			
			class Model {
			  --
				

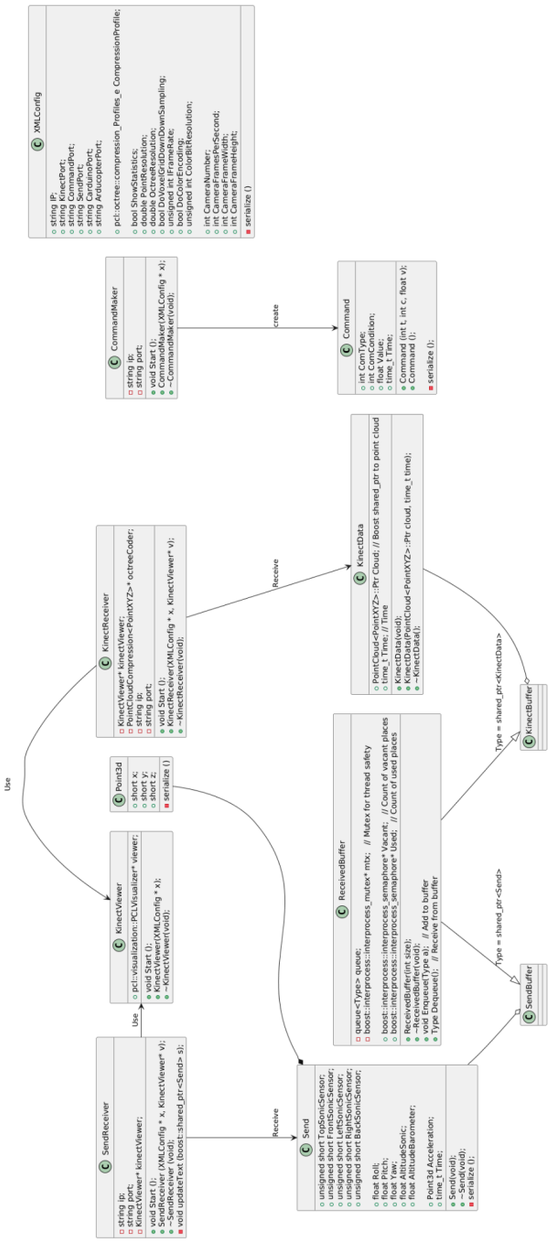
			alias superclass, owner -- relationship queries -- attributes methods subclasses packages } @enduml	

27	**Como** cajero del sistema, **quiero** registrar compras y actualizar automáticamente el total en la caja, **para** mantener un control preciso de todas las transacciones y poder revisar compras previas cuando sea necesario.	As a cashier of the system, I want to record purchases and automatically update the total in the register, so that I can maintain precise control of all transactions and review previous purchases when necessary.	@startuml
			class CashRegister {
			total
			addCash()
			getTotal()
			}
			
			class Purchase {
			cashRegister
			amount
			execute()
			}
			
			class PurchaseInvoker {
			executedPurchases
			replayPurchases()
			}
			
			Purchase -left-* CashRegister : has a
			PurchaseInvoker <-down- Purchase :
			passed to
			@enduml
				

28	**Como** administrador del sistema bancario, **quiero** registrar la información de cada banco con su nombre, ubicación y el gerente responsable, **para** asegurar una correcta identificación y gestión de las sucursales.	As an administrator of the banking system, I want to record the information of each bank, including its name, location, and responsible manager, so that I can ensure proper identification and management of the branches.	@startuml Bank --> "manager" Person : "managed by" Bank --> "NAME" String Bank --> "location" Address @enduml	

29	**Como** usuario del sistema bibliotecario, **quiero** registrar libros con sus datos (ISBN, título y año), junto con sus autores y la editorial que los publica, **para** mantener un catálogo completo y organizado que facilite la búsqueda y gestión de la colección.	As a library system user, I want to register books with their data (ISBN, title, and year), along with their authors and the publishing house, so that I can maintain a complete and organized catalog that facilitates searching and managing the collection.	@startuml
			
			class Book {
			«stdid» isbn[1] : String
			title[1] : String
			year[0..1] : Integer
			}
			
			class Publisher {
			«stdid» name[1] : String
			address[0..1] : String
			}
			
			class Author {
			«stdid» authorId[1] : Integer
			name[1] : String
			birthDate[1] : Date
			deathDate[0..1] : Date
			}
			
			Publisher "0..1" --- "*" Book : published >
			
			Author "*" -- "*" Book : authored >
			
			@enduml
				

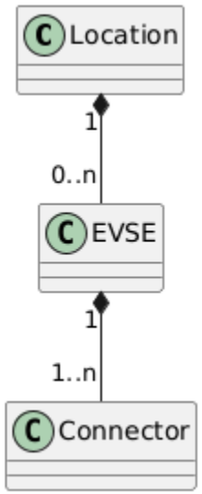
30	**Como** usuario del sistema, **quiero** poder crear formularios compuestos por distintos elementos (entradas de texto y campos de entrada), **para** mostrar de manera organizada la información y permitir que los formularios se puedan renderizar dinámicamente según los elementos agregados.	As a system user, I want to create forms composed of different elements (text entries and input fields), so that I can display information in an organized manner and allow the forms to be dynamically rendered according to the added elements.	@startuml
			class Form {
			#elements : array|FormElement[]
			+render($indent = 0 : int)
			+addElement(FormElement $element)
			}
			
			abstract class FormElement {
			+render($indent = 0 : int)
			}
			
			class InputElement {
			+render($indent = 0 : int)
			}
			
			class TextElement {
			+render($indent = 0 : int)
			}
			
			FormElement <|.. TextElement
			FormElement <|.. InputElement
			FormElement <|.. Form
			@enduml
				

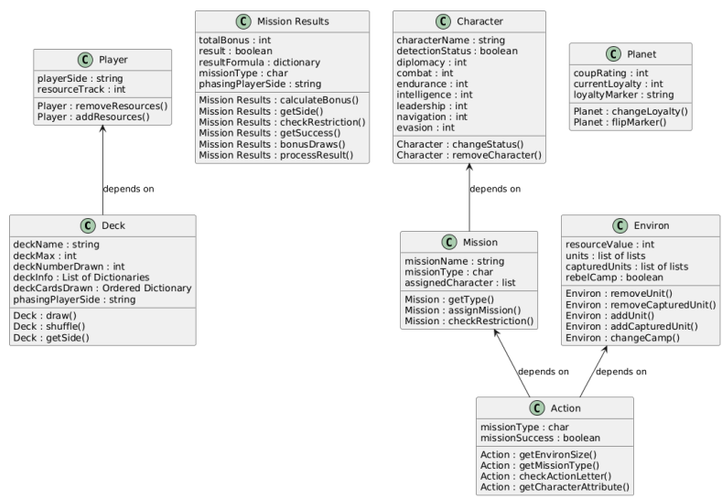
31	**Como** operador de un sistema de captura y transmisión de datos, **quiero** que los módulos de recepción de comandos, sensores y datos Kinect gestionen la adquisición, el procesamiento y la visualización de la información en tiempo real, **para** poder controlar el dispositivo de manera remota, supervisar su estado mediante los datos de sensores y obtener una representación visual confiable del entorno capturado por la cámara Kinect.	As an operator of a data capture and transmission system, I want the command reception, sensor, and Kinect data modules to manage the acquisition, processing, and visualization of information in real time, so that I can remotely control the device, monitor its status through sensor data, and obtain a reliable visual representation of the environment captured by the Kinect camera.	@startuml
			
			class Command {
			+int ComType;
			+int ComCondition;
			+float Value;
			+time_t Time;
			+Command (int t, int c, float v);
			+Command ();
			
			-serialize ();
			}
			
			class Point3d {
			+short x;
			+short y;
			+short z;
			
			-serialize ()
			}
			
			class Send {
			+unsigned short TopSonicSensor;
			+unsigned short FrontSonicSensor;
			+unsigned short LeftSonicSensor;
			+unsigned short RightSonicSensor;
			+unsigned short BackSonicSensor;
			
			+float Roll;
			+float Pitch;
			+float Yaw;
			+float AltitudeSonic;
			+float AltitudeBarometer;
			
			+Point3d Acceleration;
			+time_t Time;
			
			+Send(void);
			+~Send(void);
			-serialize ();
				

			}
			
			Point3d -down-* Send
			
			class KinectData {
			+PointCloud<PointXYZ>::Ptr Cloud; //
			Boost shared_ptr to point cloud
			+time_t Time; // Time
			+KinectData(void);
			+KinectData(PointCloud<PointXYZ>::Ptr
			cloud, time_t time);
			+~KinectData();
			}
			
			class ReceivedBuffer {
			-queue<Type> queue;
			-boost::interprocess::interprocess_mutex*
			mtx; // Mutex for thread safety
			
			+boost::interprocess::interprocess_semaph
			ore* Vacant; // Count of vacant places
			
			+boost::interprocess::interprocess_semaph
			ore* Used; // Count of used places
			
			+ReceivedBuffer(int size);
			+~ReceivedBuffer(void);
			+void Enqueue(Type a); // Add to buffer
			+Type Dequeue(); // Receive from buffer
			}
			
			ReceivedBuffer --|> KinectBuffer : Type =
			shared_ptr<KinectData>
			ReceivedBuffer --|> SendBuffer : Type =
			shared_ptr<Send>
			
			Send --o SendBuffer
			KinectData --o KinectBuffer	

			class CommandMaker {
			+void Start ();
			+CommandMaker(XMLConfig * x);
			+~CommandMaker(void);
			
			-string ip;
			-string port;
			}
			CommandMaker --> Command : create
			
			class KinectReceiver {
			+void Start ();
			+KinectReceiver(XMLConfig * x,
			KinectViewer* v);
			+~KinectReceiver(void);
			
			-KinectViewer* kinectViewer;
			-PointCloudCompression<PointXYZ>*
			octreeCoder;
			-string ip;
			-string port;
			}
			KinectReceiver --> KinectData : Receive
			
			class KinectViewer {
			+void Start ();
			+KinectViewer(XMLConfig * x);
			+~KinectViewer(void);
			+pcl::visualization::PCLVisualizer* viewer;
			}
			
			class SendReceiver {
			+void Start ();
			+SendReceiver (XMLConfig * x,
			KinectViewer* v);
			+~SendReceiver (void);
			
			-string ip;
			-string port;
			-KinectViewer* kinectViewer;	

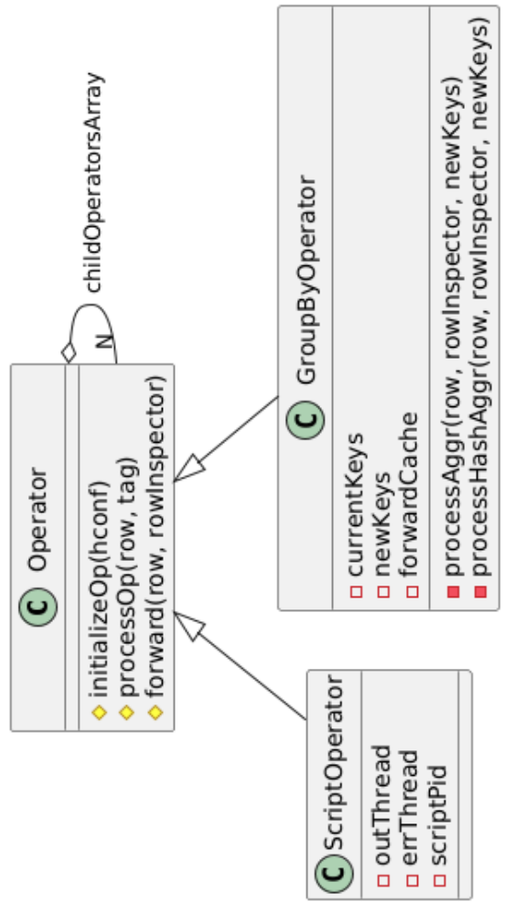
			-void updateText
			(boost::shared_ptr<Send> s);
			}
			SendReceiver --> Send : Receive
			
			SendReceiver -right-> KinectViewer : Use
			KinectReceiver -left-> KinectViewer : Use
			
			class XMLConfig {
			+string IP;
			+string KinectPort;
			+string CommandPort;
			+string SendPort;
			+string CarduinoPort;
			+string ArducopterPort;
			
			+pcl::octree::compression_Profiles_e
			CompressionProfile;
			
			+bool ShowStatistics;
			+double PointResolution;
			+double OctreeResolution;
			+bool DoVoxelGridDownDownSampling;
			+unsigned int IFrameRate;
			+bool DoColorEncoding;
			+unsigned int ColorBitResolution;
			
			+int CameraNumber;
			+int CameraFramesPerSecond;
			+int CameraFrameWidth;
			+int CameraFrameHeight;
			
			-serialize ()
			}
			
			@enduml	

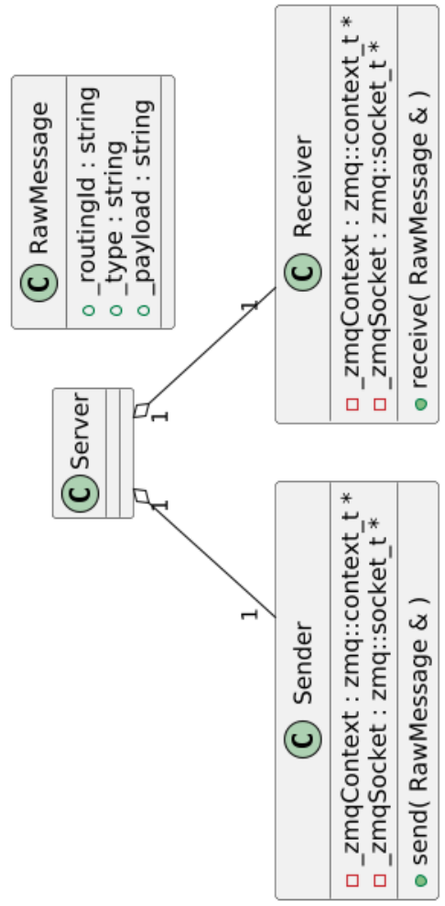
32	**Como** administrador de estaciones de carga, **quiero** que cada ubicación (Location) pueda gestionar múltiples puntos de suministro eléctrico (EVSE), y que cada punto de suministro disponga de uno o varios conectores, **para** ofrecer a los usuarios la posibilidad de cargar sus vehículos eléctricos de forma organizada y flexible según el tipo de conector disponible.	As an administrator of charging stations, I want each location to manage multiple electric supply points (EVSE), and for each supply point to have one or more connectors, so that users can charge their electric vehicles in an organized and flexible manner according to the type of connector available.	@startuml Location "1" *-- "0..n" EVSE EVSE "1" *-- "1..n" Connector @enduml	

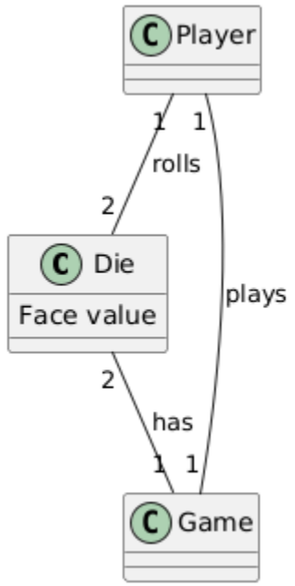
33	**Como** jugador **quiero** gestionar mis recursos, personajes y asignar misiones con ayuda de cartas y acciones **para** influir en el entorno, los planetas y obtener resultados que cambien el rumbo de la partida.	As a player, I want to manage my resources, characters, and assign missions using cards and actions, so that I can influence the environment, planets, and achieve results that change the course of the game.	@startuml
			
			class Deck {
			deckName : string
			deckMax : int
			deckNumberDrawn : int
			deckInfo : List of Dictionaries
			deckCardsDrawn : Ordered Dictionary
			phasingPlayerSide : string
			Deck : draw()
			Deck : shuffle()
			Deck : getSide()
			}
			
			class Player {
			playerSide : string
			resourceTrack : int
			Player : removeResources()
			Player : addResources()
			}
				

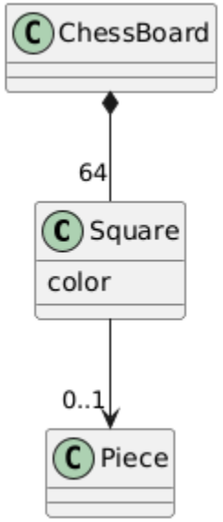
			class "Mission Results" {
			totalBonus : int
			result : boolean
			resultFormula : dictionary
			missionType : char
			phasingPlayerSide : string
			Mission Results : calculateBonus()
			Mission Results : getSide()
			Mission Results : checkRestriction()
			Mission Results : getSuccess()
			Mission Results : bonusDraws()
			Mission Results : processResult()
			}
			
			class Character {
			characterName : string
			detectionStatus : boolean
			diplomacy : int
			combat : int
			endurance : int
			intelligence : int
			leadership : int
			navigation : int
			evasion : int
			Character : changeStatus()
			Character : removeCharacter()
			}
			
			class Environ {
			resourceValue : int
			units : list of lists
			capturedUnits : list of lists
			rebelCamp : boolean
			Environ : removeUnit()
			Environ : removeCapturedUnit()
			Environ : addUnit()
			Environ : addCapturedUnit()
			Environ : changeCamp()
			}	

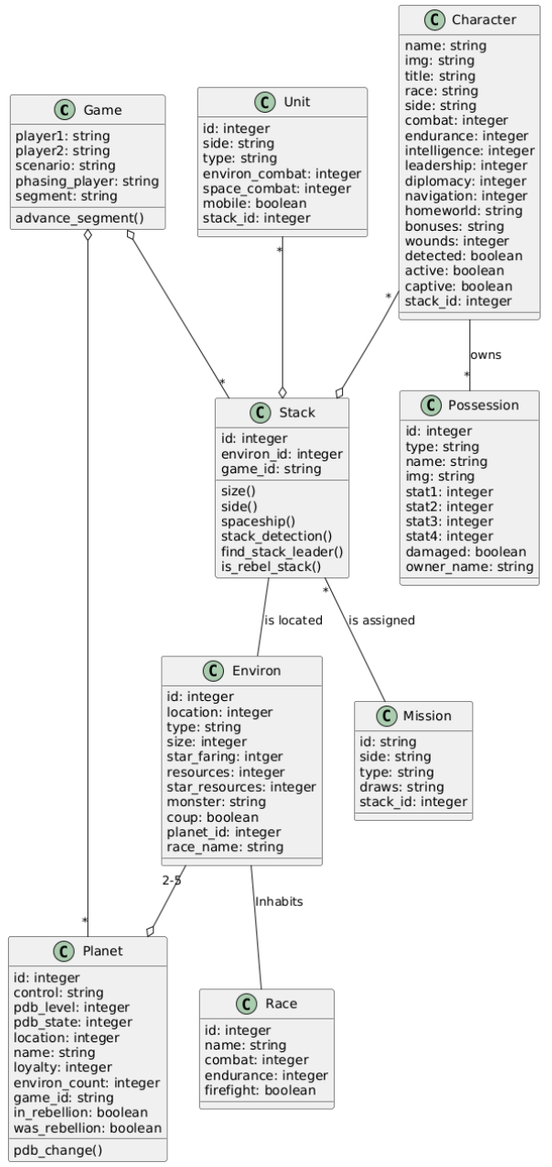
			class Planet {
			coupRating : int
			currentLoyalty : int
			loyaltyMarker : string
			Planet : changeLoyalty()
			Planet : flipMarker()
			}
			
			class Action {
			missionType : char
			missionSuccess : boolean
			Action : getEnvironSize()
			Action : getMissionType()
			Action : checkActionLetter()
			Action : getCharacterAttribute()
			}
			
			class Mission {
			missionName : string
			missionType : char
			assignedCharacter : list
			Mission : getType()
			Mission : assignMission()
			Mission : checkRestriction()
			}
			
			Mission <-- Action : depends on
			Environ <-- Action : depends on
			Character <-- Mission : depends on
			Player <-- Deck : depends on
			@enduml	

34	**Como** usuario del sistema de procesamiento de datos **quiero** utilizar operadores que puedan ejecutar scripts y agrupar datos automáticamente **para** transformar, agregar y dirigir los datos procesados a otros operadores de manera eficiente.	As a user of the data processing system, I want to use operators that can execute scripts and automatically group data to transform, aggregate, and direct processed data to other operators efficiently.	@startuml
			class Operator {
			  # initializeOp(hconf)
			  # processOp(row, tag)
			  # forward(row, rowInspector)
			
			}
			
			class ScriptOperator {
			  - outThread
			  - errThread
			  - scriptPid
			}
			
			class GroupByOperator {
			  - currentKeys
			  - newKeys
			  - forwardCache
			  - processAggr(row, rowInspector, newKeys)
			  - processHashAggr(row, rowInspector, newKeys)
			}
			
			Operator <|-- ScriptOperator
			Operator <|-- GroupByOperator
			Operator o-- "N" Operator: childOperatorsArray
			@enduml
				

35	**Como** usuario del sistema, **quiero** que el servidor gestione el envío y la recepción de mensajes en bruto mediante componentes dedicados, **para** asegurar una comunicación eficiente y organizada entre los diferentes módulos del sistema.	As a system user, I want the server to handle sending and receiving raw messages through dedicated components, so that communication between different system modules is efficient and well-organized.	@startuml
			
			class Server {
			}
			
			class Sender {
			+ send(RawMessage &)
			- _zmqContext : zmq::context_t *
			- _zmqSocket : zmq::socket_t *
			}
			
			class Receiver {
			+ receive(RawMessage &)
			- _zmqContext : zmq::context_t *
			- _zmqSocket : zmq::socket_t *
			}
			
			class RawMessage {
			+ _routingId : string
			+ _type : string
			+ _payload : string
			}
			
			Server "1" o-- "1" Sender
			Server "1" o-- "1" Receiver
			
			@enduml
				

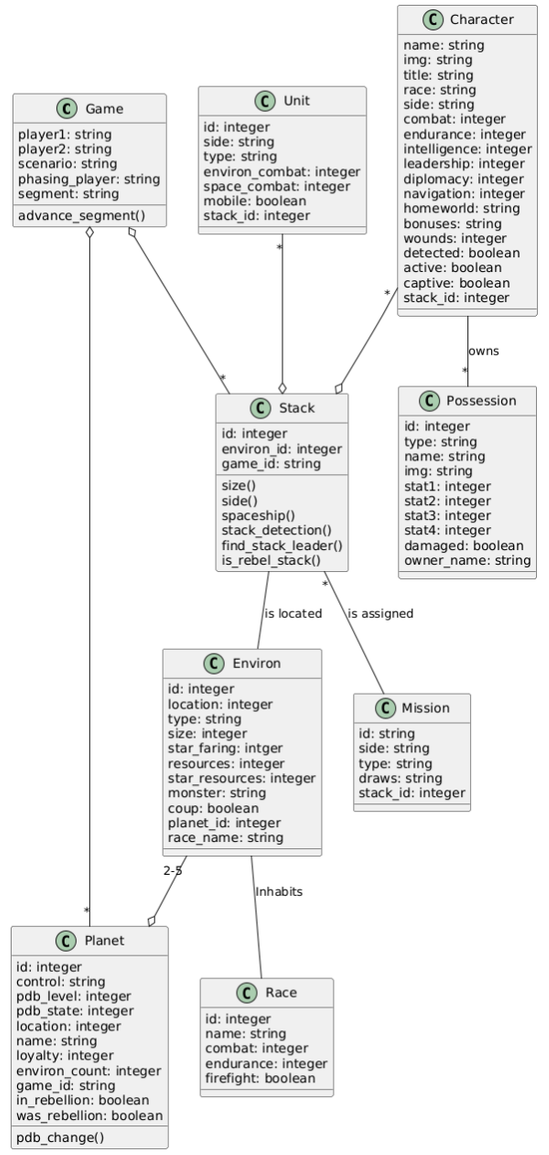
36	**Como** jugador **quiero** poder lanzar dados (Die) durante una partida (Game) **para** que el resultado de los dados determine mi progreso o acciones dentro del juego.	As a player, I want to roll dice (Die) during a match (Game), so that the dice results determine my progress or actions within the game.	@startuml
			class Die{
			Face value
			}
			
			Player "1" -- "2" Die : rolls
			Player "1" -- "1" Game : plays
			Die "2" -- "1" Game : has
			@enduml
				

37	**Como** jugador de ajedrez **quiero** que cada casilla del tablero pueda contener una pieza **para** poder ubicar y mover las piezas correctamente durante la partida.	As a chess player, I want each square on the chessboard to be able to contain a piece, so that I can place and move pieces correctly during the game.	@startuml
			
			class Square {
			color
			}
			
			ChessBoard *-- "64" Square
			
			Square --> "0..1" Piece
			
			@enduml
				

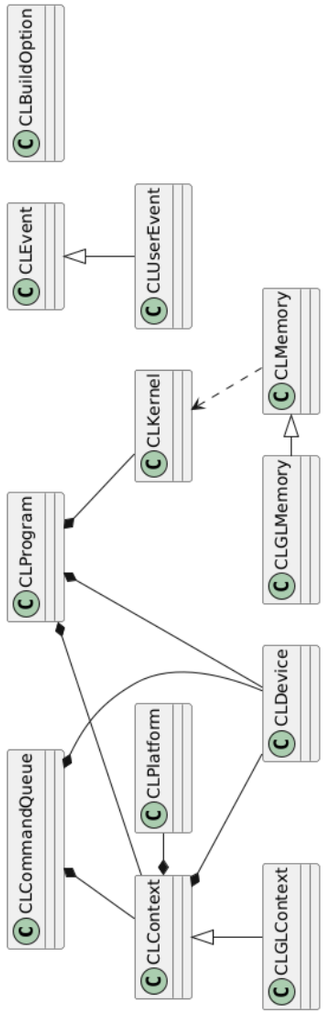
38	**Como** jugador, **quiero** controlar personajes, planetas, misiones y ejércitos dentro del juego, **para** vivir una experiencia estratégica en la que pueda tomar decisiones, conquistar territorios y enfrentarme a otros jugadores según el escenario que se desarrolle.	As a player, I want to control characters, planets, missions, and armies within the game, so that I can have a strategic experience where I make decisions, conquer territories, and compete against other players depending on the evolving scenario.	@startuml
			
			class Game{
			player1: string
			player2: string
			scenario: string
			phasing_player: string
			segment: string
			
			advance_segment()
			}
			class Character{
			name: string
			img: string
			title: string
			race: string
			side: string
			combat: integer
			endurance: integer
			intelligence: integer
			leadership: integer
			diplomacy: integer
			navigation: integer
			homeworld: string
			bonuses: string
			wounds: integer
			detected: boolean
			active: boolean
			captive: boolean
			stack_id: integer
			}
			
			class Environ{
			id: integer
			location: integer
			type: string
			size: integer
			star_faring: intger
			resources: integer
			star_resources: integer
				

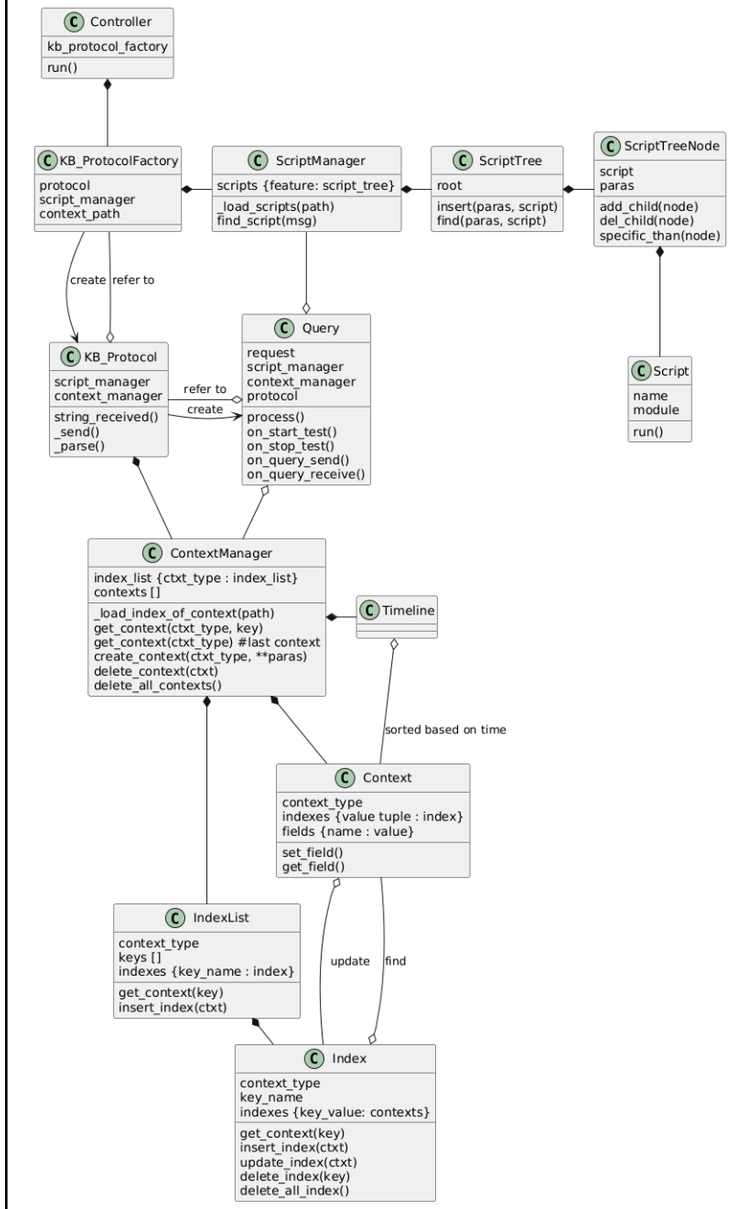
			monster: string
			coup: boolean
			planet_id: integer
			race_name: string
			}
			class Unit{
			id: integer
			side: string
			type: string
			environ_combat: integer
			space_combat: integer
			mobile: boolean
			stack_id: integer
			}
			
			class Mission{
			id: string
			side: string
			type: string
			draws: string
			stack_id: integer
			}
			class Planet{
			id: integer
			control: string
			pdb_level: integer
			pdb_state: integer
			location: integer
			name: string
			loyalty: integer
			environ_count: integer
			game_id: string
			in_rebellion: boolean
			was_rebellion: boolean
			
			pdb_change()
			}
			
			class Possession{
			id: integer	

			type: string
			name: string
			img: string
			stat1: integer
			stat2: integer
			stat3: integer
			stat4: integer
			damaged: boolean
			owner_name: string
			}
			class Race{
			id: integer
			name: string
			combat: integer
			endurance: integer
			firefight: boolean
			}
			class Stack{
			id: integer
			environ_id: integer
			game_id: string
			
			size()
			side()
			spaceship()
			stack_detection()
			find_stack_leader()
			is_rebel_stack()
			}
			Game o-- "*" Stack
			Game o-- "*" Planet
			Character "*" --o Stack
			Character -- "*" Possession: owns
			Environ "2-5" --o Planet
			Environ -- Race: Inhabits
			Unit "*" --o Stack
			Stack "*" -- Mission: is assigned
			Stack -- Environ: is located
			@enduml	

39	**Como** jugador del juego estratégico, **quiero** gestionar mis personajes, unidades, misiones, posesiones y pilas en distintos planetas y entornos, **para** poder planificar mis acciones, avanzar en el juego, controlar recursos y lograr objetivos estratégicos basados en la interacción de razas, planetas y entornos.	As a strategic game player, I want to manage my characters, units, missions, possessions, and stacks on different planets and environments, so that I can plan actions, advance in the game, control resources, and achieve strategic objectives based on the interaction of races, planets, and environments.	@startuml
			
			class Server
			
			class Game {
			void start()
			void stop()
			void addRobot(string iName)
			void addPlayer(string iName)
			Robot getAvailableRobot()
			void fillGameStateMessage (orwell::messages::GameState & iMessage)
			
			map<string routingId, Robot &> _robots
			map<string routingId, Player &> _players
			map<string, Team &> _teams
			bool _isGameStarted
			}
			
			note right of Game {
			routingId = Robot name
			}
			
			note "Do we need explicit link between robot and player ?" as N1
			N1 -- Game
			
			class Player {
			string name
			Team & _team
			}
			class Robot{
			void fillRobotStateMessage (orwell::messages::RobotState & iMessage)
				

			string name
			}
			
			class Team {
			void addPlayer(Player iPlayer)
			void removePlayer(Player iPlayer)
			list<Player &> _players
			}
			
			Game "1" *-- "0-*" Player
			Game "1" *-- "0-*" Robot
			Game "1" *-- "0-*" Team
			Team "1" o-- "0-*" Player
			Server "1" -- "1" Game
			Robot "1" -- "1" Player
			
			@enduml	

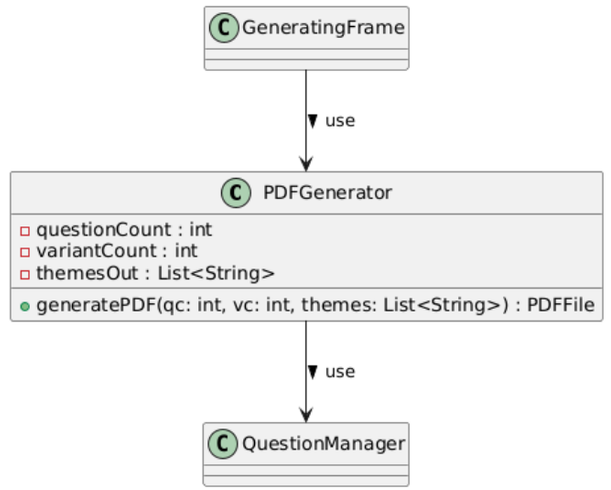
40	**Como** desarrollador de aplicaciones que usan OpenCL, **quiero** tener un sistema de clases que gestione colas de comandos, contextos, dispositivos y memoria, **para** poder organizar y ejecutar tareas de cómputo de manera eficiente y reutilizable.	As a developer of applications using OpenCL, I want to have a class system that manages command queues, contexts, devices, and memory, so that I can organize and execute computing tasks efficiently and in a reusable manner.	@startuml class CLCommandQueue class CLContext class CLDevice class CLEvent class CLUserEvent class CLKernel class CLMemory class CLPlatform class CLProgram class CLBuildOption class CLGLContext class CLGLMemory ' Relaciones principales CLCommandQueue *-- CLContext CLCommandQueue *- CLDevice CLContext *- CLPlatform CLContext *-- CLDevice CLEvent <|-- CLUserEvent CLKernel <.. CLMemory CLProgram *-- CLKernel CLProgram *-- CLContext CLProgram *-- CLDevice CLMemory <|-left- CLGLMemory CLContext <|-- CLGLContext @enduml	

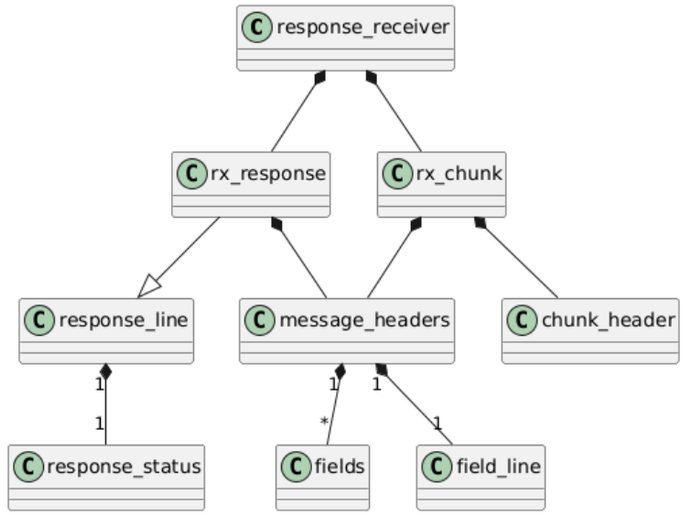
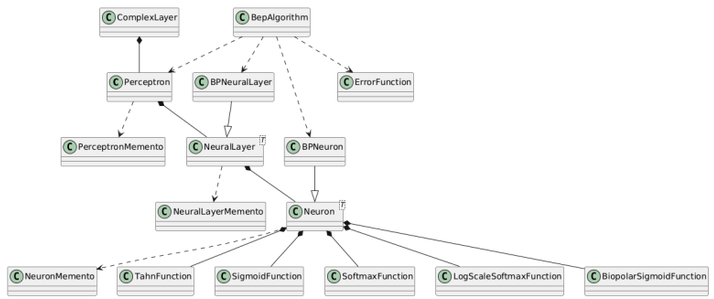
41	**Como** desarrollador del sistema de gestión de protocolos, **quiero** que el Controller gestione fábricas de protocolos (KB_ProtocolFactory) y coordine la ejecución, **para** poder inicializar y ejecutar la comunicación entre dispositivos de manera centralizada.	As a developer of the protocol management system, I want the Controller to manage protocol factories (KB_ProtocolFactory) and coordinate execution, so that I can initialize and run communication between devices in a centralized way.	@startuml
			class Controller {
			kb_protocol_factory
			
			run()
			}
			
			class KB_ProtocolFactory {
			protocol
			script_manager
			context_path
			}
			
			class KB_Protocol {
			script_manager
			context_manager
			string_received()
			_send()
			_parse()
			}
			
			class Query {
			request
			script_manager
			context_manager
			protocol
			process()
			on_start_test()
			on_stop_test()
			on_query_send()
			on_query_receive()
			}
			
			class ScriptManager {
			scripts {feature: script_tree}
			
			_load_scripts(path)
			find_script(msg)
			}
				

			class ScriptTree{
			root
			
			insert(paras, script)
			find(paras, script)
			}
			
			class ScriptTreeNode{
			script
			paras
			
			add_child(node)
			del_child(node)
			specific_than(node)
			}
			
			class Script {
			name
			module
			run()
			}
			
			class ContextManager {
			index_list {ctxt_type : index_list}
			contexts []
			
			_load_index_of_context(path)
			get_context(ctxt_type, key)
			get_context(ctxt_type) #last context
			create_context(ctxt_type, **paras)
			delete_context(ctxt)
			delete_all_contexts()
			}
			
			class IndexList {
			context_type
			keys []
			indexes {key_name : index}
			
			get_context(key)	

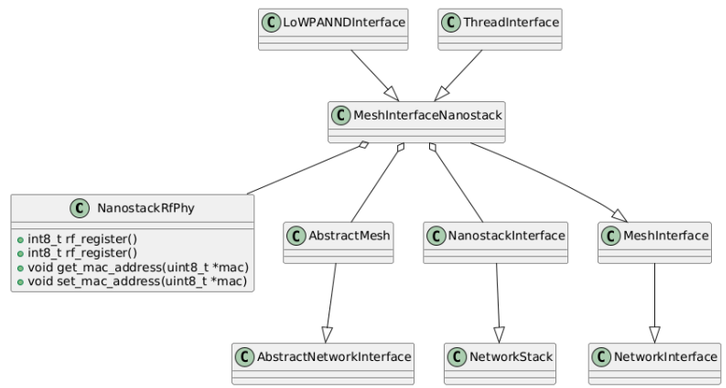
			insert_index(ctxt)
			}
			
			class Index {
			context_type
			key_name
			indexes {key_value: contexts}
			
			get_context(key)
			insert_index(ctxt)
			update_index(ctxt)
			delete_index(key)
			delete_all_index()
			}
			
			class Context{
			context_type
			indexes {value tuple : index}
			fields {name : value}
			
			set_field()
			get_field()
			}
			
			Controller *-- KB_ProtocolFactory
			KB_ProtocolFactory --> KB_Protocol: create
			KB_ProtocolFactory *- ScriptManager
			KB_Protocol o-up- KB_ProtocolFactory:refer to
			KB_Protocol -> Query: create
			KB_Protocol *-- ContextManager
			Query o-- ContextManager
			Query o-up- ScriptManager
			Query o- KB_Protocol: refer to
			ScriptManager *- ScriptTree
			ScriptTree *- ScriptTreeNode
			ScriptTreeNode *-- Script
			ContextManager *-- IndexList
			ContextManager *- Context	

			ContextManager *- Timeline Timeline o-- Context: sorted based on time IndexList *- Index Index o-up- Context: find Context o-- Index: update @enduml	

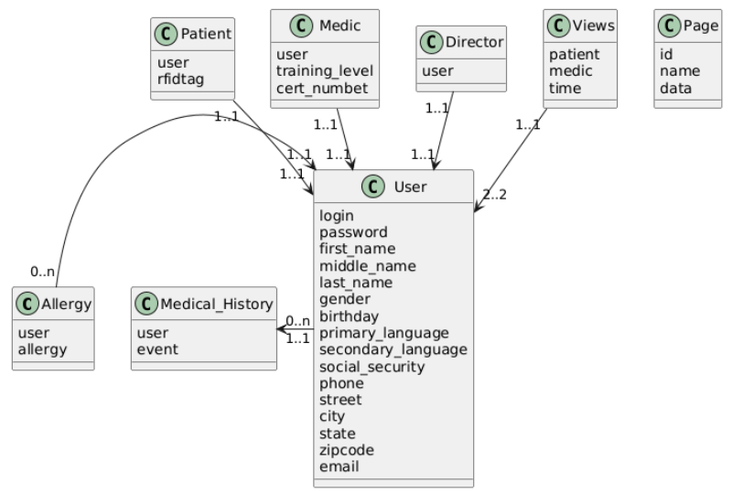
42	**Como** usuario del sistema, **quiero** generar un archivo PDF con un número específico de preguntas y variantes de diferentes temas, **para** que el sistema utilice PDFGenerator y QuestionManager para automatizar la creación de exámenes de manera organizada y reproducible.	As a system user, I want to generate a PDF file with a specific number of questions and variants on different topics, so that the system uses PDFGenerator and QuestionManager to automate exam creation in an organized and reproducible way.	@startuml
			class PDFGenerator {
			-questionCount : int
			-variantCount : int
			-themesOut : List<String>
			+generatePDF(qc: int, vc: int, themes: List<String>) : PDFFile
			}
			
			GeneratingFrame --> PDFGenerator : use
			>
			PDFGenerator --> QuestionManager : use
			>
			@enduml
				

43	**Como** desarrollador del sistema, **quiero** que el sistema reciba y procese respuestas de red estructuradas en líneas y fragmentos, **para** que cada response_receiver pueda gestionar rx_response y rx_chunk, asociando correctamente los encabezados y campos de mensaje, permitiendo un manejo organizado de los datos recibidos.	As a system developer, I want the system to receive and process structured network responses in lines and chunks, so that each response_receiver can manage rx_response and rx_chunk, correctly associating message headers and fields, allowing organized handling of received data.	@startuml response_receiver *-- rx_response response_receiver *-- rx_chunk rx_response --|> response_line rx_response *-- message_headers response_line "1" *-- "1" response_status rx_chunk *-- chunk_header rx_chunk *-- message_headers message_headers "1" *-- "*" fields message_headers "1" *-- "1" field_line @enduml	
44	**Como** investigador en inteligencia artificial, **quiero** entrenar y configurar una red neuronal que aprenda patrones de datos, **para** obtener predicciones y resultados confiables que me ayuden a resolver problemas complejos como clasificación, reconocimiento o predicción.	As an artificial intelligence researcher, I want to train and configure a neural network that learns data patterns, so that I can obtain reliable predictions and results that help solve complex problems such as classification, recognition, or forecasting.	@startuml class Perceptron class PerceptronMemento class NeuralLayer<T> class NeuralLayerMemento class Neuron<T> class NeuronMemento class ComplexLayer class BPNeuralLayer class BPNeuron class TahnFunction class SigmoidFunction class SoftmaxFunction class LogScaleSoftmaxFunction class BiopolarSigmoidFunction ' Relaciones principales Neuron *-- TahnFunction Neuron *-- SigmoidFunction	

			Neuron *-- SoftmaxFunction Neuron *-- LogScaleSoftmaxFunction Neuron *-- BiopolarSigmoidFunction Neuron ..> NeuronMemento NeuralLayer ..> NeuralLayerMemento Perceptron *-- NeuralLayer Perceptron ..> PerceptronMemento NeuralLayer *-- Neuron ComplexLayer *-- Perceptron BPNeuralLayer --|> NeuralLayer BPNeuron --|> Neuron BepAlgorithm ..> Perceptron BepAlgorithm ..> BPNeuralLayer BepAlgorithm ..> BPNeuron BepAlgorithm ..> ErrorFunction @enduml	

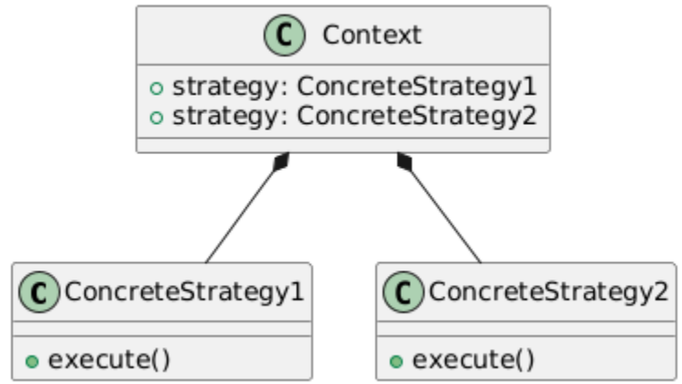
45	**Como** desarrollador de redes de malla, **quiero** que MeshInterfaceNanostack se comunique con la capa física NanostackRfPhy y con interfaces abstractas de red, **para** gestionar la comunicación inalámbrica y permitir la interoperabilidad con diferentes protocolos de red.	As a developer of mesh networks, I want MeshInterfaceNanostack to communicate with the NanostackRfPhy physical layer and abstract network interfaces, so that I can manage wireless communication and enable interoperability with different network protocols.	@startuml
			
			class NanostackRfPhy {
			+int8_t rf_register()
			+int8_t rf_register()
			+void get_mac_address(uint8_t *mac)
			+void set_mac_address(uint8_t *mac)
			}
			
			class MeshInterfaceNanostack {
			}
			MeshInterfaceNanostack o--
			NanostackRfPhy
			MeshInterfaceNanostack o-- AbstractMesh
			MeshInterfaceNanostack o--
			NanostackInterface
			
			class MeshInterface {
			}
				

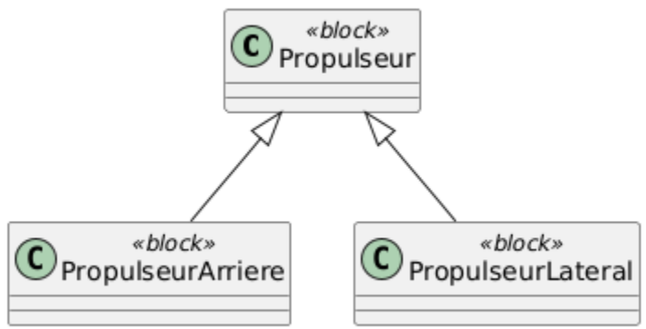
			NanostackInterface --|> NetworkStack MeshInterfaceNanostack --|> MeshInterface MeshInterface --|> NetworkInterface LoWPANNDInterface --|> MeshInterfaceNanostack ThreadInterface --|> MeshInterfaceNanostack AbstractMesh --|> AbstractNetworkInterface class AbstractNetworkInterface @enduml	

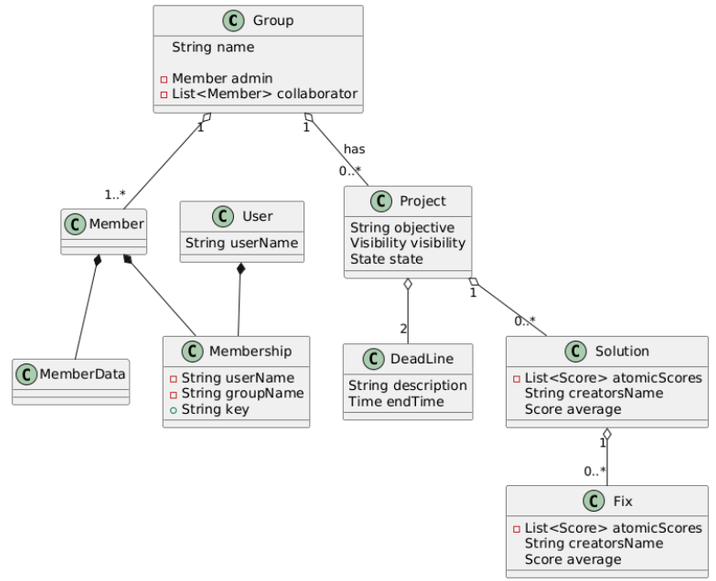
46	**Como** administrador del sistema de salud, **quiero** que los usuarios puedan asociarse con roles específicos como paciente, médico o director, **para** gestionar de manera correcta los permisos y funcionalidades de cada tipo de usuario.	As a health system administrator, I want users to be associated with specific roles such as patient, doctor, or director, so that the permissions and functionalities for each type of user can be managed correctly.	@startuml
			
			Allergy "0..n"->"1..1" User
			Medical_History "0..n" <- "1..1" User
			Patient "1..1"-->"1..1" User
			Medic "1..1"-->"1..1" User
			Director "1..1"-->"1..1" User
			Views "1..1"-->"2..2" User
			
			class Allergy{
			user
			allergy
			}
			
			class Director{
			user
			}
			
			class Medic{
			user
			training_level
			cert_numbet
			}
				

			class Medical_History{
			user
			event
			}
			
			class Page{
			id
			name
			data
			}
			
			class Patient{
			user
			rfidtag
			}
			
			class User{
			login
			password
			first_name
			middle_name
			last_name
			gender
			birthday
			primary_language
			secondary_language
			social_security
			phone
			street
			city
			state
			zipcode
			email
			}
			
			class Views{
			patient
			medic
			time
			}	

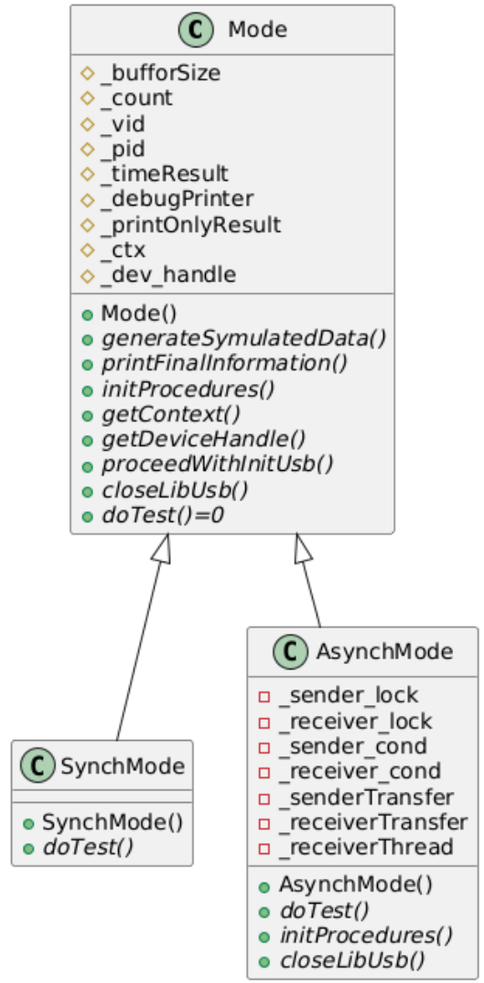
			@enduml	

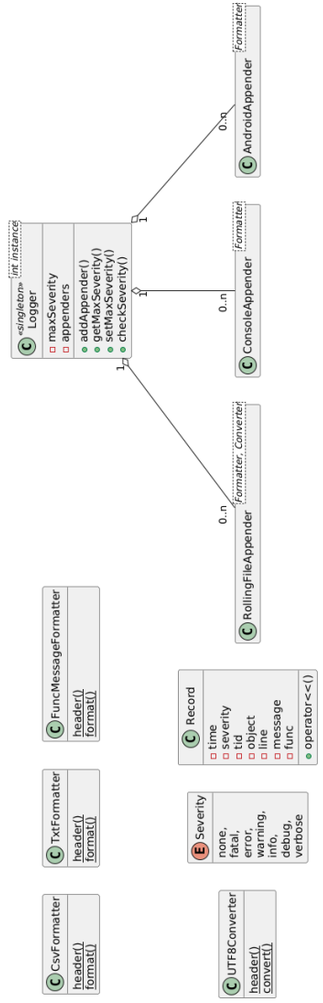
47	**Como** desarrollador del sistema, **quiero** que la clase Context pueda trabajar con distintas estrategias (ConcreteStrategy1 y ConcreteStrategy2) a través de un método execute(), **para** que el comportamiento del sistema pueda cambiar dinámicamente según la estrategia seleccionada sin modificar el código del Context.	As a system developer, I want the Context class to work with different strategies (ConcreteStrategy1 and ConcreteStrategy2) through an execute() method, so that the system behavior can change dynamically according to the selected strategy without modifying the Context code.	@startuml
			class ConcreteStrategy1 {
			+execute()
			}
			
			class ConcreteStrategy2 {
			+execute()
			}
			
			class Context {
			+strategy: ConcreteStrategy1
			+strategy: ConcreteStrategy2
			}
			
			Context *-- ConcreteStrategy1
			Context *-- ConcreteStrategy2
			@enduml
				

48	**Como** desarrollador de la simulación de naves espaciales, **quiero** modelar un sistema de propulsores que incluya propulsores traseros y laterales, **para** gestionar de manera organizada y extensible los diferentes tipos de propulsión en la nave, aprovechando la herencia para reutilizar la funcionalidad común de los propulsores.	As a developer of the spaceship simulation, I want to model a propulsion system that includes rear and side thrusters, so that I can organize and extend the different types of propulsion on the ship, leveraging inheritance to reuse common thruster functionality.	@startuml class Propulseur <<block>> class PropulseurArriere <<block>> class PropulseurLateral <<block>> Propulseur <|-- PropulseurArriere Propulseur <|-- PropulseurLateral @enduml	

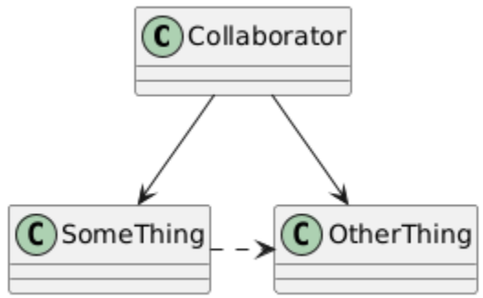
49	**Como** administrador de proyectos colaborativos, **quiero** gestionar grupos con miembros y sus datos, asignar miembros a proyectos y registrar soluciones y correcciones con sus puntajes, **para** organizar de manera eficiente la colaboración, seguimiento de tareas y evaluación de resultados dentro de cada grupo.	As an administrator of collaborative projects, I want to manage groups with members and their data, assign members to projects, and record solutions and corrections with their scores, so that I can efficiently organize collaboration, task tracking, and result evaluation within each group.	@startuml
			
			Group "1" o-- "1..*" Member
			Member *-- MemberData
			Member *-- Membership
			User *-- Membership
			Group "1" o-- "0..*" Project : has
			Project "1" o-- "0..*" Solution
			Solution "1" o-- "0..*" Fix
			Project o-- "2" DeadLine
			
			class User {
			  String userName
			}
			
			class Membership {
			  -String userName
			  -String groupName
			  +String key
			}
			
			class Solution {
			  -List<Score> atomicScores
			  String creatorsName
			  Score average
			}
			
			class Project {
			  String objective
			  Visibility visibility
			  State state
			}
			
			class Fix {
			  -List<Score> atomicScores
			  String creatorsName
			  Score average
			}
			
			class Group {
				

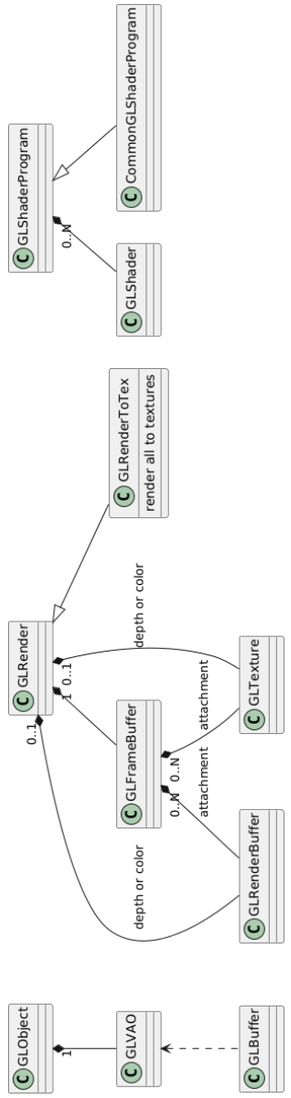
			String name
			
			-Member admin
			-List<Member> collaborator
			}
			
			class DeadLine {
			String description
			Time endTime
			}
			@enduml	

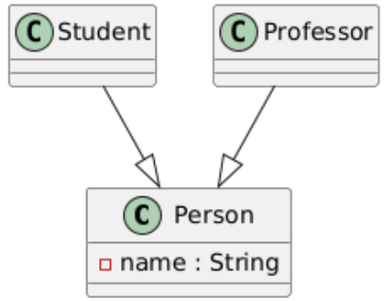
50	**Como** desarrollador de pruebas de dispositivos USB, **quiero** definir modos de operación síncrono y asíncrono con procedimientos de inicialización, manejo de contexto y ejecución de pruebas, **para** simular datos, ejecutar pruebas y obtener resultados de manera controlada y flexible según el modo de operación seleccionado.	As a developer of USB device testing, I want to define synchronous and asynchronous operation modes with initialization procedures, context handling, and test execution, so that I can simulate data, run tests, and obtain results in a controlled and flexible manner according to the selected operation mode.	@startuml
			Mode <|-- SynchMode
			Mode <|-- AsynchMode
			Class Mode {
			+ Mode()
			+{abstract} generateSymulatedData()
			+{abstract} printFinalInformation()
			+{abstract} initProcedures()
			+{abstract} getContext()
			+{abstract} getDeviceHandle()
			+{abstract} proceedWithInitUsb()
			+{abstract} closeLibUsb()
			+{abstract} doTest()=0
			#_bufforSize
			#_count
			#_vid
			#_pid
			#_timeResult
			#_debugPrinter
			#_printOnlyResult
			#_ctx
			#_dev_handle
			}
			class SynchMode {
			+SynchMode()
			+{abstract} doTest()
			}
			class AsynchMode {
			+AsynchMode()
			+{abstract} doTest()
			+{abstract} initProcedures()
			+{abstract} closeLibUsb()
			-_sender_lock
			-_receiver_lock
			-_sender_cond
			-_receiver_cond
			-_senderTransfer
			-_receiverTransfer
			-_receiverThread
			}
			@enduml
				

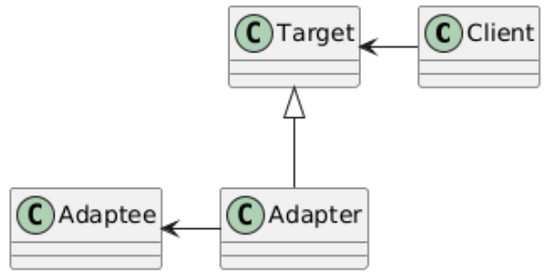
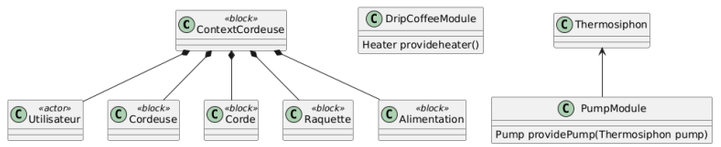
51	**Como** desarrollador del sistema, **quiero** que el logueo del sistema funcione como un singleton capaz de gestionar múltiples appenders (RollingFileAppender, ConsoleAppender, AndroidAppender) y distintos niveles de severidad (Severity), **para** registrar mensajes del sistema de forma centralizada y consistente, aplicando diferentes formatos (CsvFormatter, TxtFormatter, FuncMessageFormatter) y conversores (UTF8Converter) según sea necesario.	As a system developer, I want the system logging to function as a singleton capable of managing multiple appenders (RollingFileAppender, ConsoleAppender, AndroidAppender) and different severity levels (Severity), so that system messages can be recorded centrally and consistently, applying different formats (CsvFormatter, TxtFormatter, FuncMessageFormatter) and converters (UTF8Converter) as needed.	@startuml
			class Logger<int instance> <<singleton>> {
			+addAppender()
			+getMaxSeverity()
			+setMaxSeverity()
			+checkSeverity()
			-maxSeverity
			-appenders
			}
			
			class RollingFileAppender<Formatter, Converter>
			class ConsoleAppender<Formatter>
			class AndroidAppender<Formatter>
			
			Logger "1" o-- "0..n" RollingFileAppender
			Logger "1" o-- "0..n" ConsoleAppender
			Logger "1" o-- "0..n" AndroidAppender
			
			class CsvFormatter {
			{static} header()
			{static} format()
			}
			
			class TxtFormatter {
			{static} header()
			{static} format()
			}
			
			class FuncMessageFormatter {
			{static} header()
			{static} format()
			}
			
			class UTF8Converter {
			{static} header()
			{static} convert()
			}
			
			enum Severity {
				

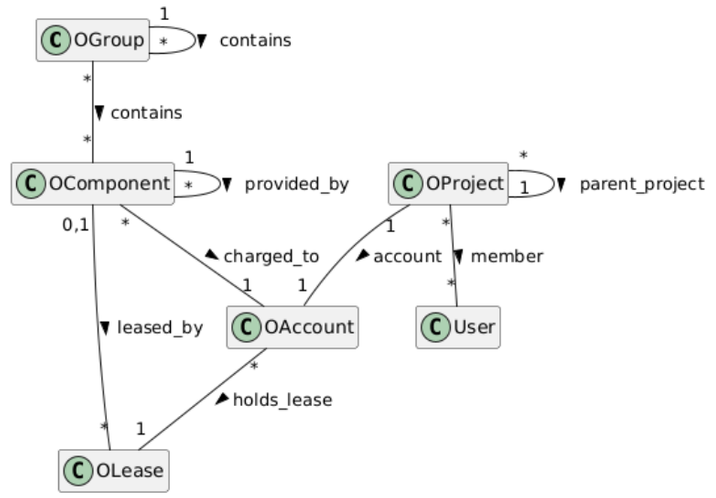
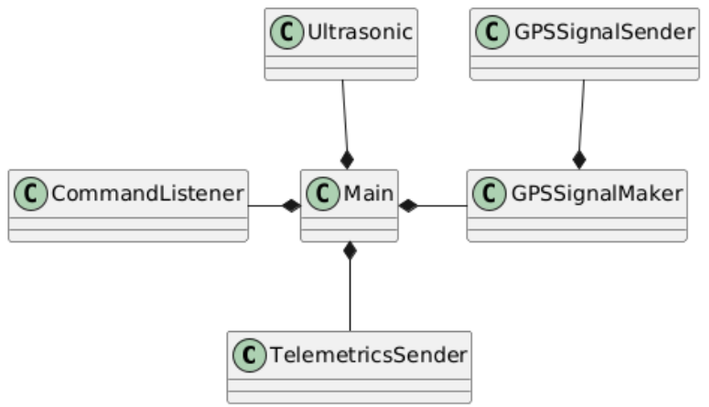
			none,
			fatal,
			error,
			warning,
			info,
			debug,
			verbose
			}
			
			class Record {
			+operator<<()
			-time
			-severity
			-tid
			-object
			-line
			-message
			-func
			}
			
			@enduml	

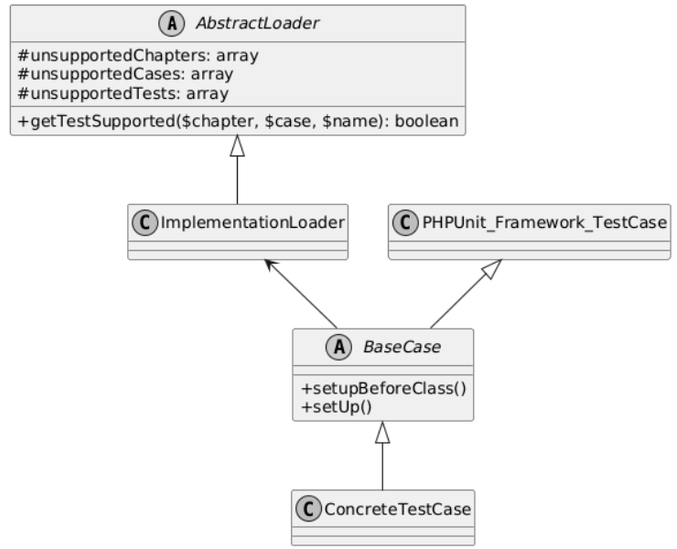
52	**Como** usuario del sistema, **quiero** que los colaboradores puedan interactuar con diferentes elementos (SomeThing y OtherThing) y que estos elementos puedan relacionarse entre sí, **para** facilitar la gestión y coordinación de recursos dentro del sistema.	As a system user, I want collaborators to interact with different elements (SomeThing and OtherThing) and for these elements to relate to each other, so that resource management and coordination within the system are facilitated.	@startuml
			
			class Collaborator {
			}
			
			Collaborator -down-> SomeThing
			Collaborator -down-> OtherThing
			SomeThing .right.> OtherThing
			
			@enduml
				

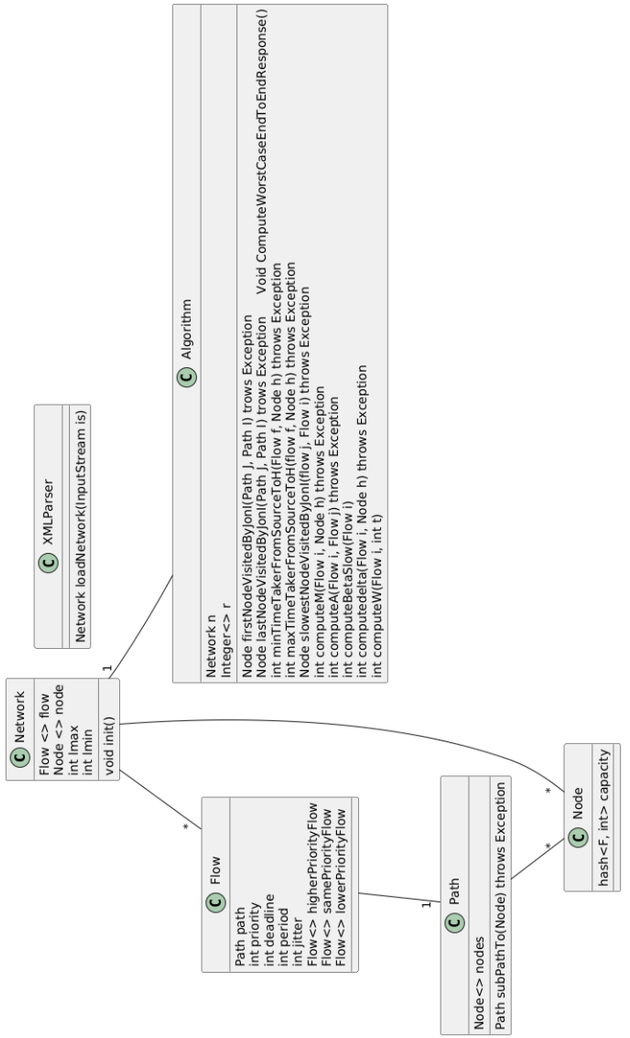
53	**Como** desarrollador gráfico del sistema, **quiero** que las clases GLRender, GLFrameBuffer, GLTexture, GLRenderBuffer y GLShaderProgram interactúen entre sí, **para** gestionar la renderización de escenas gráficas, adjuntar buffers y texturas, y aplicar programas de shaders de manera flexible para generar salidas visuales en pantalla o en texturas.	As a graphics developer of the system, I want the classes GLRender, GLFrameBuffer, GLTexture, GLRenderBuffer, and GLShaderProgram to interact with each other, so that I can manage the rendering of graphical scenes, attach buffers and textures, and apply shader programs flexibly to generate visual outputs on screen or in textures.	@startuml
			
			class GLBuffer
			class GLVAO
			class GLObject
			GLObject "1" *-- GLVAO
			class GLRenderBuffer
			class GLTexture
			class GLFrameBuffer
			GLFrameBuffer "0..N" *-- GLRenderBuffer : attachment
			GLFrameBuffer "0..N" *-- GLTexture : attachment
			
			class GLRender
			GLRender "1" *-- GLFrameBuffer
			GLRender "0..1" *-- GLTexture : depth or color
			GLRender "0..1" *-- GLRenderBuffer : depth or color
			
			class GLRenderToTex {
			render all to textures
			}
			
			GLRender <|-- GLRenderToTex
			GLVAO <.. GLBuffer
			
			class GLShader
			class GLShaderProgram
			
			GLShaderProgram "0..N" *-- GLShader
			
			class CommonGLShaderProgram
			GLShaderProgram <|-- CommonGLShaderProgram
			
			@enduml
				

54	**Como** administrador del sistema educativo, **quiero** que tanto estudiantes como profesores hereden información básica de la clase Persona, como el nombre, **para** mantener un modelo unificado de los individuos y facilitar la gestión de datos.	As an educational system administrator, I want both students and professors to inherit basic information from the Person class, such as the name, so that there is a unified model of individuals and data management is simplified.	@startuml
			class Person {
			- name : String
			}
			Student --|> Person
			Professor --|> Person
			@enduml
				

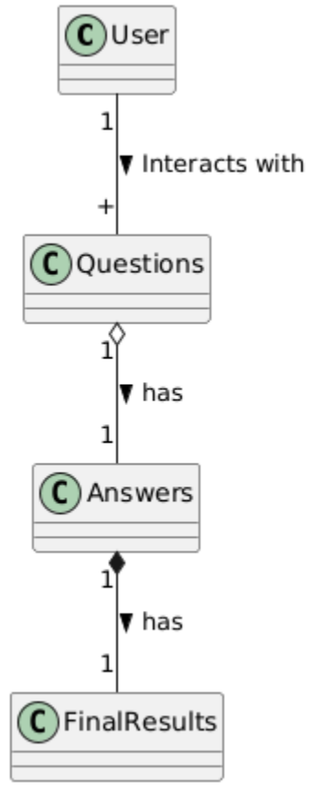
55	**Como** usuario del sistema, **quiero** que el adaptador permita que un cliente interactúe con una clase existente (Adaptee) a través de una interfaz común (Target), **para** que pueda utilizar funcionalidades de clases incompatibles sin modificar su código.	As a system user, I want the adapter to allow a client to interact with an existing class (Adaptee) through a common interface (Target), so that I can use functionalities of incompatible classes without modifying their code.	@startuml class Client class Target class Adapter class Adaptee Target <--R Client Target <|-- Adapter Adaptee <--R Adapter @enduml	
56	**Como** usuario del sistema, **quiero** interactuar con el ContextCordeuse que integra los elementos Cordeuse, Corde, Raquette y Alimentation, **para** que pueda gestionar de manera centralizada el proceso de cordaje y sus componentes asociados.	As a system user, I want to interact with the ContextCordeuse, which integrates the elements Cordeuse, Corde, Raquette, and Alimentation, so that I can manage the stringing process and its associated components in a centralized way.	@startuml class ContextCordeuse <<block>> class Utilisateur <<actor>> class Cordeuse <<block>> class Corde <<block>> class Raquette <<block>> class Alimentation <<block>> ContextCordeuse *-- Utilisateur ContextCordeuse *-- Cordeuse ContextCordeuse *-- Corde ContextCordeuse *-- Raquette ContextCordeuse *-- Alimentation @enduml	

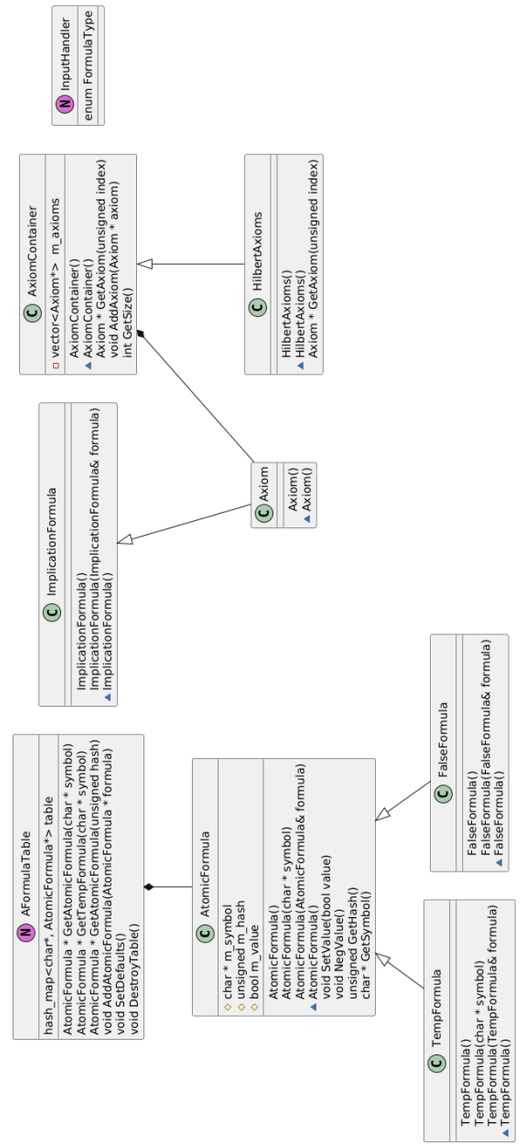
57	**Como** administrador de proyectos, **quiero** organizar proyectos en grupos y asignar componentes a cuentas, **para** llevar un control claro de los recursos, los costos y los usuarios que participan en cada proyecto.	As a project administrator, I want to organize projects into groups and assign components to accounts, so that I can maintain clear control of resources, costs, and users participating in each project.	@startuml hide members OGroup "*" -- "*" OComponent : contains > OGroup "1" -- "*" OGroup : contains > OComponent "*" -- "1" OAccount : charged_to > OComponent "1" -- "*" OComponent : provided_by > OLease "*" -- "0,1" OComponent: < leased_by OProject "1" -- "1" OAccount : account > OProject "*" -- "*" User: member > OAccount "*" -- "1" OLease : holds_lease > OProject "*" -- "1" OProject: parent_project > @enduml	
58	**Como** operador del sistema, **quiero** que el módulo principal gestione el envío de telemetría, la recepción de comandos, las señales GPS y la información del sensor ultrasónico, **para** poder supervisar y controlar de manera eficiente el sistema en tiempo real.	As a system operator, I want the main module to manage telemetry sending, command listening, GPS signals, and ultrasonic sensor data, so that I can monitor and control the system efficiently in real time.	@startuml Class TelemetricsSender Class CommandListener Class GPSSignalMaker Class GPSSignalSender Class Ultrasonic Class Main Main *-down- TelemetricsSender Main *-left- CommandListener Main *-up- Ultrasonic Main *-right- GPSSignalMaker GPSSignalMaker *-up- GPSSignalSender @enduml	

59	**Como** desarrollador del sistema, **quiero** que el cargador administre e implemente casos de prueba concretos a partir de clases base abstractas y del framework PHPUnit, **para** ejecutar pruebas de manera organizada, identificar cuáles son compatibles y garantizar la correcta validación del software.	As a system developer, I want the loader to manage and implement concrete test cases based on abstract base classes and the PHPUnit framework, so that tests can be executed in an organized manner, supported tests can be identified, and proper software validation is ensured.	@startuml
			
			abstract class AbstractLoader {
			#unsupportedChapters: array
			#unsupportedCases: array
			#unsupportedTests: array
			+getTestSupported($chapter, $case, $name): boolean
			}
			abstract class BaseCase {
			+setupBeforeClass()
			+setUp()
			}
			class ConcreteTestCase
			
			class ImplementationLoader
			class PHPUnit_Framework_TestCase
			
			AbstractLoader <|-- ImplementationLoader
			ImplementationLoader <-- BaseCase
			PHPUnit_Framework_TestCase <|--
			BaseCase
			BaseCase <|-- ConcreteTestCase
			
			@enduml
				

60	**Como** desarrollador de sistemas de comunicaciones, **quiero** modelar una red con nodos, flujos y rutas que puedan ser analizados por un algoritmo y cargados desde un archivo XML, **para** que pueda calcular métricas de rendimiento, tiempos de respuesta y planificar la gestión de flujos de manera eficiente dentro de la red.	As a communications system developer, I want to model a network with nodes, flows, and routes that can be analyzed by an algorithm and loaded from an XML file, so that I can calculate performance metrics, response times, and efficiently plan flow management within the network.	@startuml
			class Network{
			Flow <> flow
			Node <> node
			int lmax
			int lmin
			void init()
			}
			
			class Flow{
			Path path
			int priority
			int deadline
			int period
			int jitter
			Flow<> higherPriorityFlow
			Flow<> samePriorityFlow
			Flow<> lowerPriorityFlow
			}
			
			class Path{
			Node<> nodes
			Path subPathTo(Node) throws Exception
			}
			
			class Node{
			hash<F, int> capacity
			}
			
			class Algorithm{
			Network n
			Integer<> r
			Node firstNodeVisitedByJonI(Path J, Path I) trows Exception
			Node lastNodeVisitedByJonI(Path J, Path I) trows Exception Void ComputeWorstCaseEndToEndResponse()
			int minTimeTakerFromSourceToH(Flow f, Node h) throws Exception
			int maxTimeTakerFromSourceToH(flow f,
				

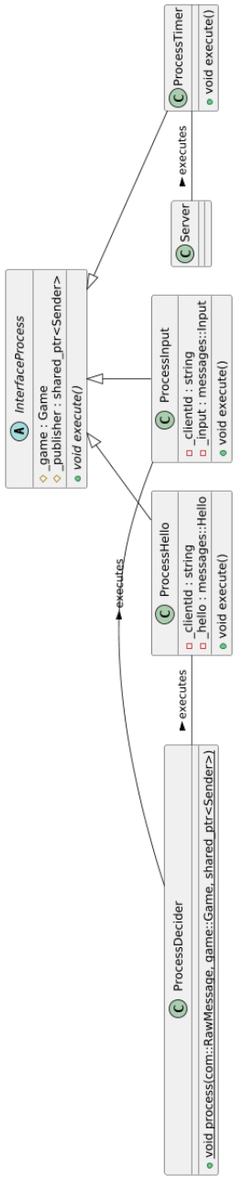
			Node h) throws Exception
			  Node slowestNodeVisitedByJonI(flow j, Flow i) throws Exception
			  int computeM(Flow i, Node h) throws Exception
			  int computeA(Flow i, Flow j) throws Exception
			  int computeBetaSlow(Flow i)
			  int computedelta(Flow i, Node h) throws Exception
			  int computeW(Flow i, int t)
			}
			
			class XMLParser{
			  Network loadNetwork(InputStream is)
			}
			
			Network "1" -- Algorithm
			Network -- "*" Node
			Network -- "*" Flow
			Flow -- "1" Path
			Path -- "*" Node
			
			@enduml	

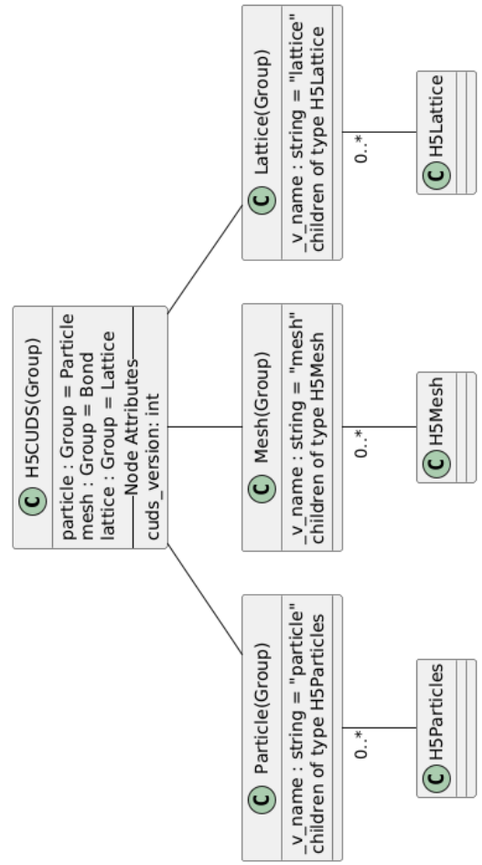
61	**Como** usuario de la aplicación, **quiero** responder preguntas y obtener resultados finales basados en mis respuestas, **para** conocer mi desempeño y recibir retroalimentación clara del sistema.	As an application user, I want to answer questions and obtain final results based on my responses, so that I can know my performance and receive clear feedback from the system.	@startuml Class User Class Questions Class Answers Class FinalResults User"1" -- "+"Questions : Interacts with > Questions"1" o-- "1"Answers : has > Answers"1" *-- "1"FinalResults : has > @enduml	

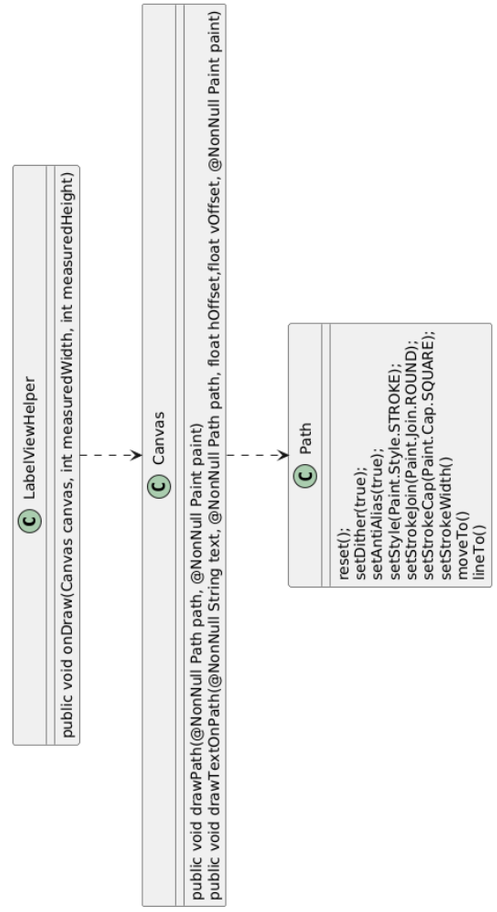
62	**Como** estudiante de lógica matemática, **quiero** representar y manipular diferentes tipos de fórmulas (atómicas, temporales, falsas, implicaciones y axiomas) organizadas en una estructura clara, **para** poder analizar, almacenar y recuperar axiomas de manera eficiente dentro de un sistema basado en las reglas de Hilbert.	As a student of mathematical logic, I want to represent and manipulate different types of formulas (atomic, temporal, false, implications, and axioms) organized in a clear structure, so that I can analyze, store, and retrieve axioms efficiently within a system based on Hilbert's rules.	@startuml
			
			class AtomicFormula {
			AtomicFormula()
			AtomicFormula(char * symbol)
			AtomicFormula(AtomicFormula& formula)
			~AtomicFormula()
			void SetValue(bool value)
			void NegValue()
			unsigned GetHash()
			char * GetSymbol()
			
			#char * m_symbol
			#unsigned m_hash
			#bool m_value
			}
			
			class TempFormula {
			TempFormula()
			TempFormula(char * symbol)
			TempFormula(TempFormula& formula)
			~TempFormula()
			}
			
			class Axiom {
			Axiom()
			~Axiom()
			}
			
			class FalseFormula{
			FalseFormula()
			FalseFormula(FalseFormula& formula)
			~FalseFormula()
			}
			
			class HilbertAxioms {
			HilbertAxioms()
				

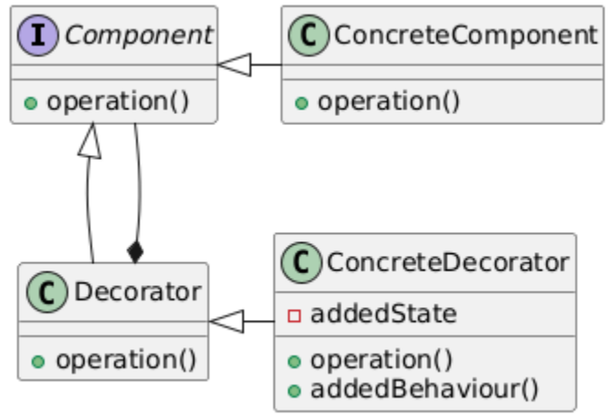
			~HilbertAxioms()
			Axiom * GetAxiom(unsigned index)
			}
			
			class AxiomContainer {
			AxiomContainer()
			~AxiomContainer()
			Axiom * GetAxiom(unsigned index)
			void AddAxiom(Axiom * axiom)
			int GetSize()
			
			-vector<Axiom*> m_axioms
			}
			
			class ImplicationFormula {
			ImplicationFormula()
			
			ImplicationFormula(ImplicationFormula&
			formula)
			~ImplicationFormula()
			}
			
			class AFormulaTable << (N, orchid) >> {
			AtomicFormula *
			GetAtomicFormula(char * symbol)
			AtomicFormula *
			GetTempFormula(char * symbol)
			AtomicFormula *
			GetAtomicFormula(unsigned hash)
			void
			AddAtomicFormula(AtomicFormula *
			formula)
			void SetDefaults()
			void DestroyTable()
			
			hash_map<char*, AtomicFormula*>
			table
			}
			
			class InputHandler << (N, orchid) >> {	

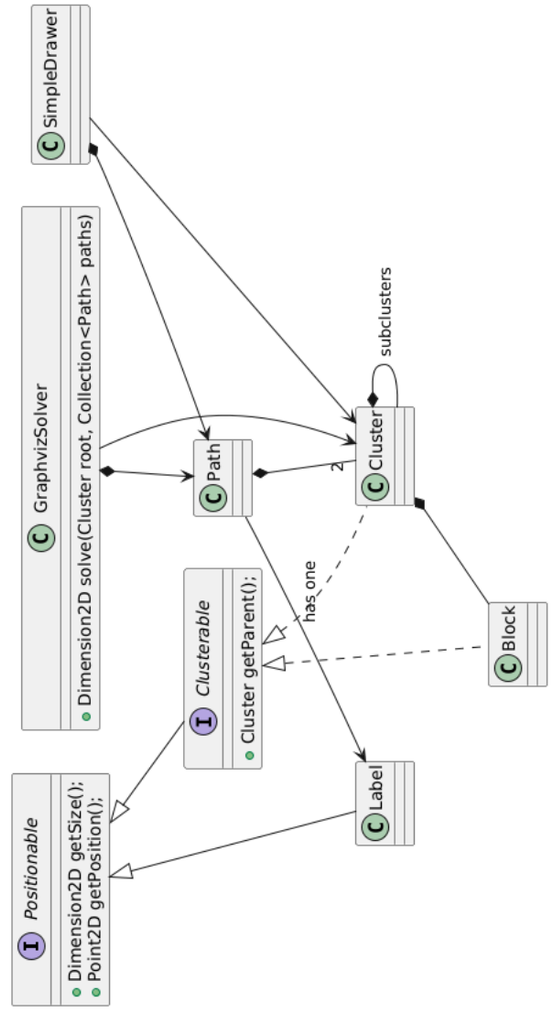
			enum FormulaType } AFormulaTable *-- AtomicFormula AtomicFormula <|-- FalseFormula AtomicFormula <|-- TempFormula ImplicationFormula <|-- Axiom AxiomContainer <|-- HilbertAxioms AxiomContainer *-- Axiom @enduml	

63	**Como** administrador de un sistema de juego en línea, **quiero** que el servidor procese diferentes tipos de mensajes de los clientes (como saludos, entradas de usuario y eventos de tiempo) mediante procesos especializados, **para** asegurar que cada interacción se gestione de forma correcta y eficiente dentro de la partida.	As an administrator of an online game system, I want the server to process different types of client messages (such as greetings, user inputs, and time events) through specialized processes, so that each interaction is handled correctly and efficiently within the game.	@startuml
			class Server
			class ProcessDecider {
			+{static} void process(com::RawMessage, game::Game, shared_ptr<Sender>)
			}
			abstract InterfaceProcess {
			+{abstract} void execute()
			
			# _game : Game
			# _publisher : shared_ptr<Sender>
			}
			class ProcessHello {
			+ void execute()
			
			- _clientId : string
			- _hello : messages::Hello
			}
			class ProcessInput {
			+ void execute()
			
			- _clientId : string
			- _input : messages::Input
			}
			class ProcessTimer {
			+ void execute()
			
			}
			InterfaceProcess <|-- ProcessHello
			InterfaceProcess <|-- ProcessInput
			InterfaceProcess <|-- ProcessTimer
			
			ProcessDecider - ProcessHello : executes
			>
			ProcessDecider - ProcessInput : executes
			>
			Server - ProcessTimer : executes >
			@enduml
				

64	**Como** investigador que trabaja con simulaciones científicas, **quiero** organizar y almacenar partículas, mallas y retículas en un sistema estructurado, **para** gestionar de manera clara los datos de mis experimentos y facilitar su análisis posterior.	As a researcher working with scientific simulations, I want to organize and store particles, meshes, and grids in a structured system, so that I can clearly manage the data from my experiments and facilitate subsequent analysis.	@startuml
			
			class H5CUDS as "H5CUDS(Group)" {
			particle : Group = Particle
			mesh : Group = Bond
			lattice : Group = Lattice
			-- Node Attributes --
			cuds_version: int
			}
			
			class Particle as "Particle(Group)" {
			_v_name : string = "particle"
			children of type H5Particles
			}
			
			class Mesh as "Mesh(Group)" {
			_v_name : string = "mesh"
			children of type H5Mesh
			}
			
			class Lattice as "Lattice(Group)" {
			_v_name : string = "lattice"
			children of type H5Lattice
			}
			
			H5CUDS -- Lattice
			H5CUDS -- Particle
			H5CUDS -- Mesh
			
			Lattice "0..*" -- H5Lattice
			Mesh "0..*" -- H5Mesh
			Particle "0..*" -- H5Particles
			
			@enduml
				

65	**Como** usuario de una aplicación móvil, **quiero** que las etiquetas de texto se dibujen de manera clara y adaptadas a trayectorias personalizadas, **para** visualizar la información de forma atractiva y legible dentro de la interfaz.	As a user of a mobile application, I want text labels to be drawn clearly and adapted to custom paths, so that I can display information in an attractive and readable way within the interface.	@startuml
			
			class LabelViewHelper {
			public void onDraw(Canvas canvas, int measuredWidth, int measuredHeight)
			
			}
			
			class Canvas {
			public void drawPath(@NonNull Path path, @NonNull Paint paint)
			public void drawTextOnPath(@NonNull String text, @NonNull Path path, float hOffset,float vOffset, @NonNull Paint paint)
			}
			
			class Path {
			reset();
			setDither(true);
			setAntiAlias(true);
			setStyle(Paint.Style.STROKE);
			setStrokeJoin(Paint.Join.ROUND);
			setStrokeCap(Paint.Cap.SQUARE);
			setStrokeWidth()
			moveTo()
			lineTo()
			}
			
			LabelViewHelper ..>Canvas
			Canvas ..>Path
			@enduml
				

66	**Como** usuario de un sistema, **quiero** poder extender las funciones de un componente sin modificar su estructura original, **para** añadir nuevas capacidades de manera flexible y mantener la simplicidad del diseño.	As a system user, I want to extend the functions of a component without modifying its original structure, so that I can add new capabilities flexibly while maintaining the simplicity of the design.	@startuml
			interface Component {
			+operation()
			}
			
			class ConcreteComponent {
			+operation()
			}
			
			class Decorator {
			+operation()
			}
			
			class ConcreteDecorator {
			-addedState
			+operation()
			+addedBehaviour()
			}
			Component <|-right- ConcreteComponent
			Component <|-- Decorator
			Component --* Decorator
			ConcreteDecorator -left-|> Decorator
			
			@enduml
				

67	**Como** usuario del sistema de visualización de gráficos, **quiero** organizar y representar jerárquicamente elementos como clusters, bloques y paths, **para** poder visualizar de manera clara la estructura de datos y sus relaciones, calcular posiciones y tamaños de los elementos, y dibujar los gráficos correctamente usando un motor de renderizado.	As a user of the graphics visualization system, I want to organize and represent elements hierarchically, such as clusters, blocks, and paths, so that I can clearly visualize the data structure and its relationships, calculate element positions and sizes, and correctly render the graphics using a rendering engine.	@startuml
			interface Positionable {
			+ Dimension2D getSize();
			+ Point2D getPosition();
			}
			
			interface Clusterable {
			+Cluster getParent();
			}
			
			Positionable <|-- Clusterable
			
			class Cluster
			
			Cluster *-- Cluster : subclusters
			Clusterable <|.. Cluster
			Cluster *-- Block
			Clusterable <|.. Block
			
			Path *-- "2" Cluster
			Path --> Label : has one
			Positionable <|-- Label
			
			SimpleDrawer --> Cluster
			SimpleDrawer *--> Path
			
			class GraphvizSolver {
			+ Dimension2D solve(Cluster root,
			Collection<Path> paths)
			}
			GraphvizSolver --> Cluster
			GraphvizSolver *--> Path
			'Clusterable --> Cluster : Parent
			
			@enduml
				

68	**Como** desarrollador del sistema, **quiero** que las entidades puedan ser gestionadas mediante operaciones de acceso a datos (buscar, guardar, actualizar y eliminar), **para** almacenar la información de manera consistente usando diferentes implementaciones de almacenamiento, como JPA o XML.	As a system developer, I want entities to be managed through data access operations (search, save, update, and delete), so that information can be stored consistently using different storage implementations, such as JPA or XML.	@startuml
			class DAO{
			find(id)
			findByName(name)
			update(Entity)
			save(Entity)
			delete(Entity)
			}
			
			Entity ..> DAO : "persist via"
			JpaDAO --|> DAO
			XmlDAO --|> DAO
			@enduml