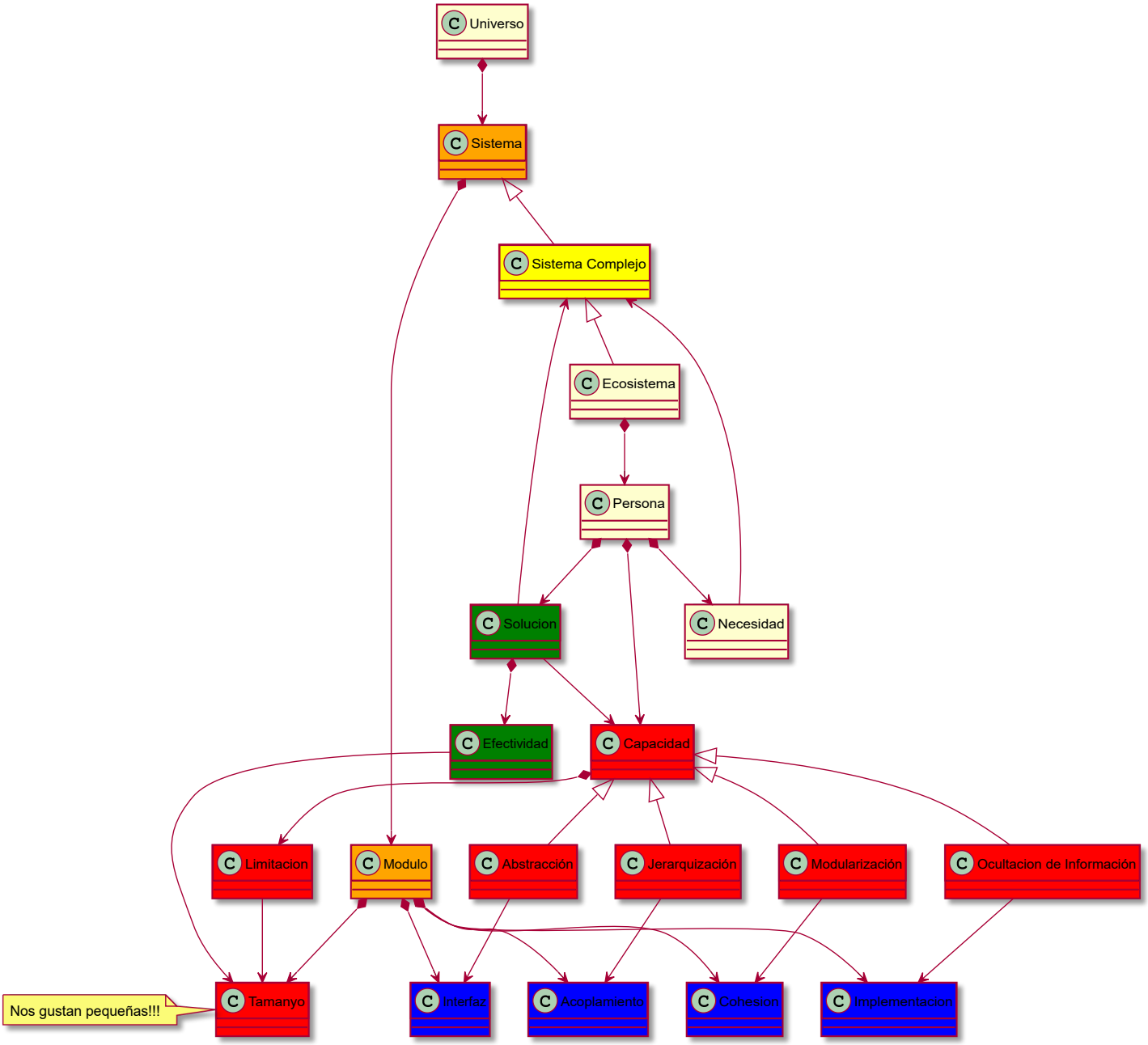
Sistemas Complejos

Santa Tecla
parqueNaturalSantaTecla@gmail.com
Version 0.0.1

Índice


Abstracción
Eficiencia
Jerarquización
Modularización
Sistema
Eficacia
Capacidad-Cualitativas
Sistema-Complejo
Efectividad
Capacidad-Cuantitativas
Necesidad
Encapsulación
Numero-Magíco-de-Miller


Universo
Sistema
Sistema Complejo
Ecosistema
Persona
Solucion
Necesidad
Efectividad
Capacidad
Limitacion
Modulo
Abstracción
Jerarquización
Modularización
Ocultacion de Información
Nos gustan pequeñas!!!
Tamanyo
Interfaz
Acoplamiento
Cohesion
Implementacion

Justificación: ¿Por qué?

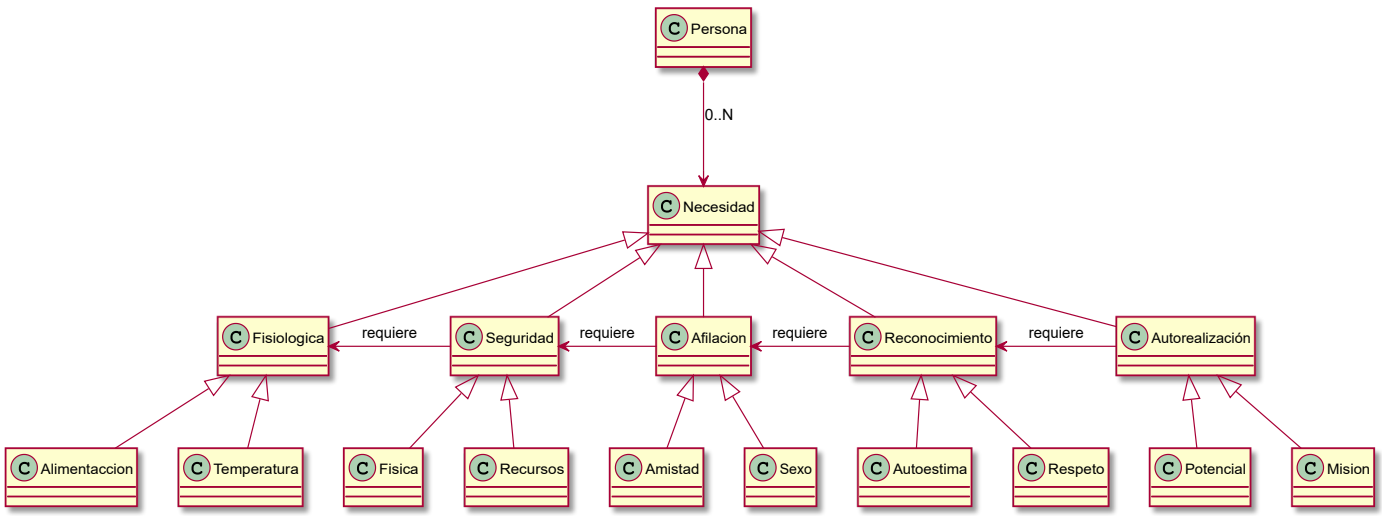
Necesidades

> *El hombre es la medida de todas las cosas*
>
> — *Protágoras*
> *Grecia Clásica*

> *EL hombre como centro del universo*
>
> — *Humanismo*
> *Renacimiento*

Pirámide de Maslow

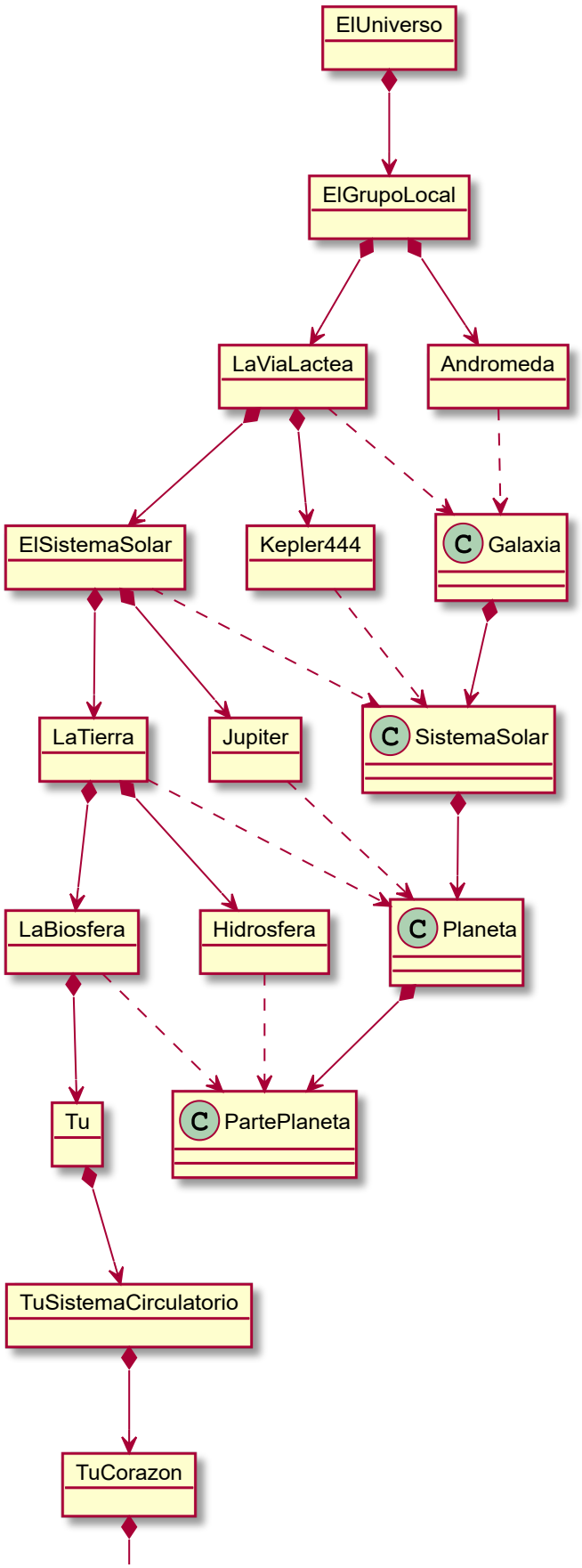
- Para evitar la ambigüedad
  - **UML** de **RUP**: Lenguaje Unificado de Modelado (*Unified Model Language*) de la metodología del Proceso Unificado de Rational (*Rational Unified Process*)
    - Cada símbolo (léxico) relacionado (sintaxis) en un diagrama tiene un significado estándar (semática). Así:
      - varias personas entienden lo mismo del mismo diagrama, no como con los gráficos de transparencias, ...
      - tú entiendes hoy el diagrama que dibujaste hace tiempo, no como una servilleta con un garabato, ...

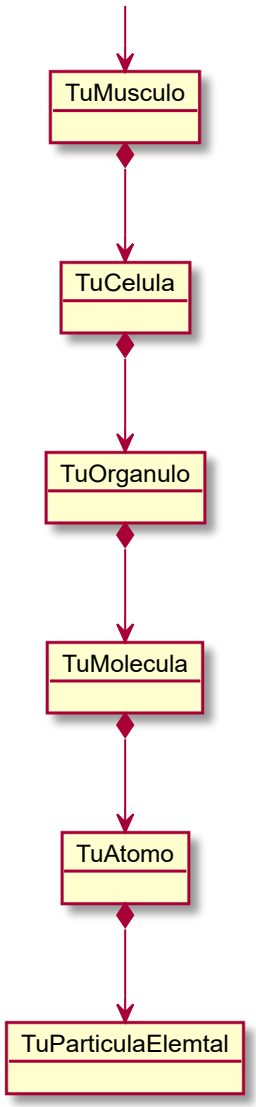


PlantUML	**Asciidoc**
Generación de diagramas de UML a partir de texto con sintaxis específica	Generación de documentación en HTML, PDF, ... con PlantUML incorporado
www.planttext.com/	asciidoctor.org/docs/asciidoc-writers-guide/

Universo

Todas las necesidades se satisfacen con los recursos del universo



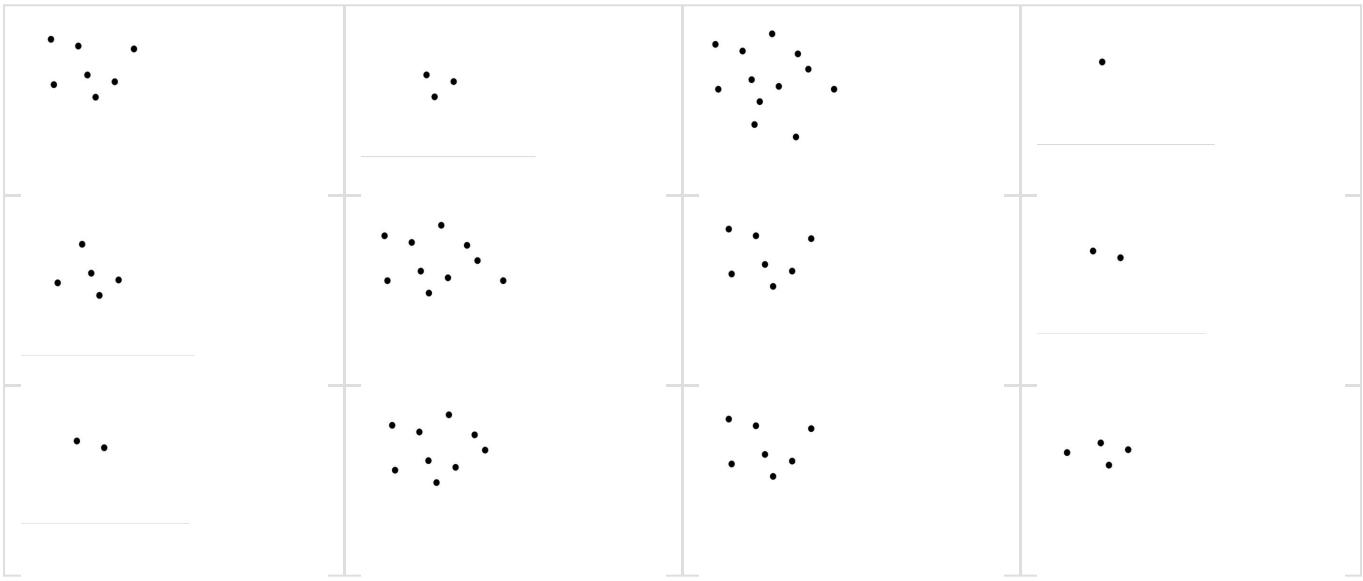


y para explotarlo hay que estudiarlo, conocerlo, ...

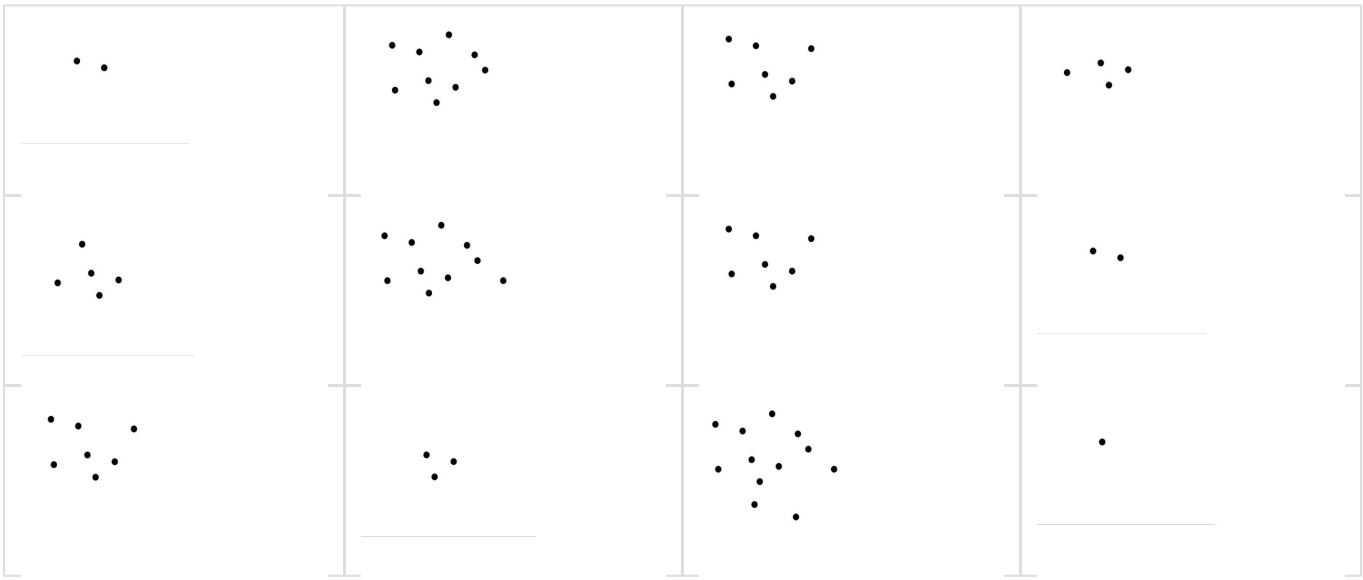
Escenarios

¿Qué tal manejas estos sencillos sistemas? Escenario con Puntos

- Dime cúantos elementos hay en cada imagen



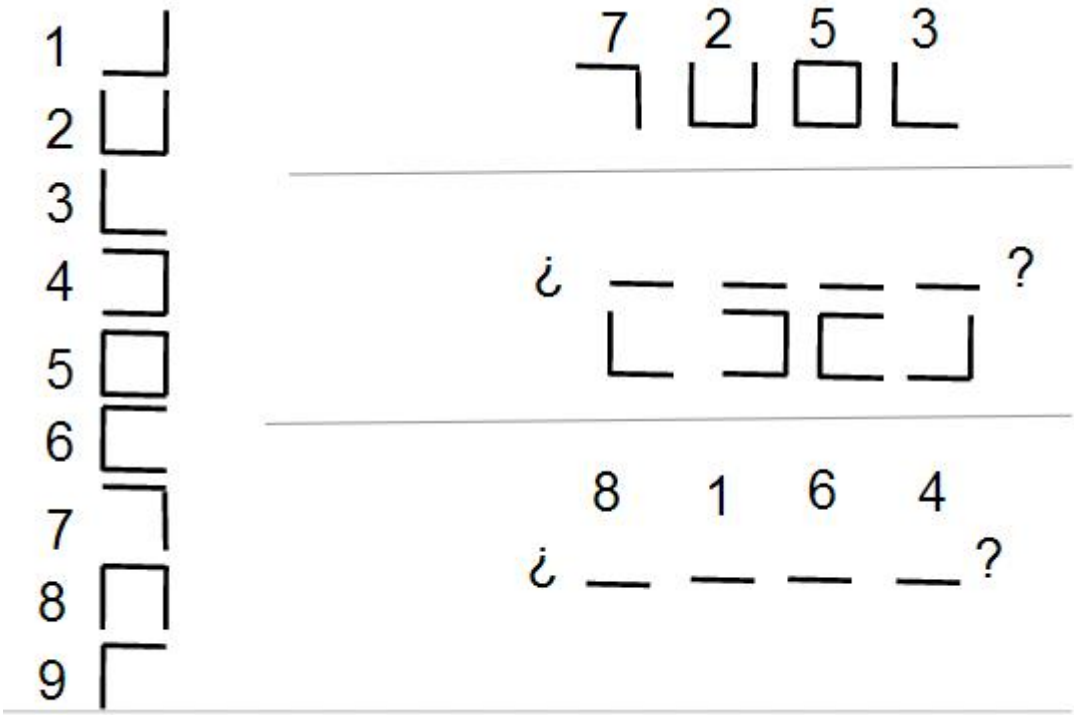
- Dime cúantos elementos hay en cada imagen y sé consciente cuándo los cuentas (siguiendo uno por uno con "golpes" de vista) o cuándo los ves (sin seguimiento)



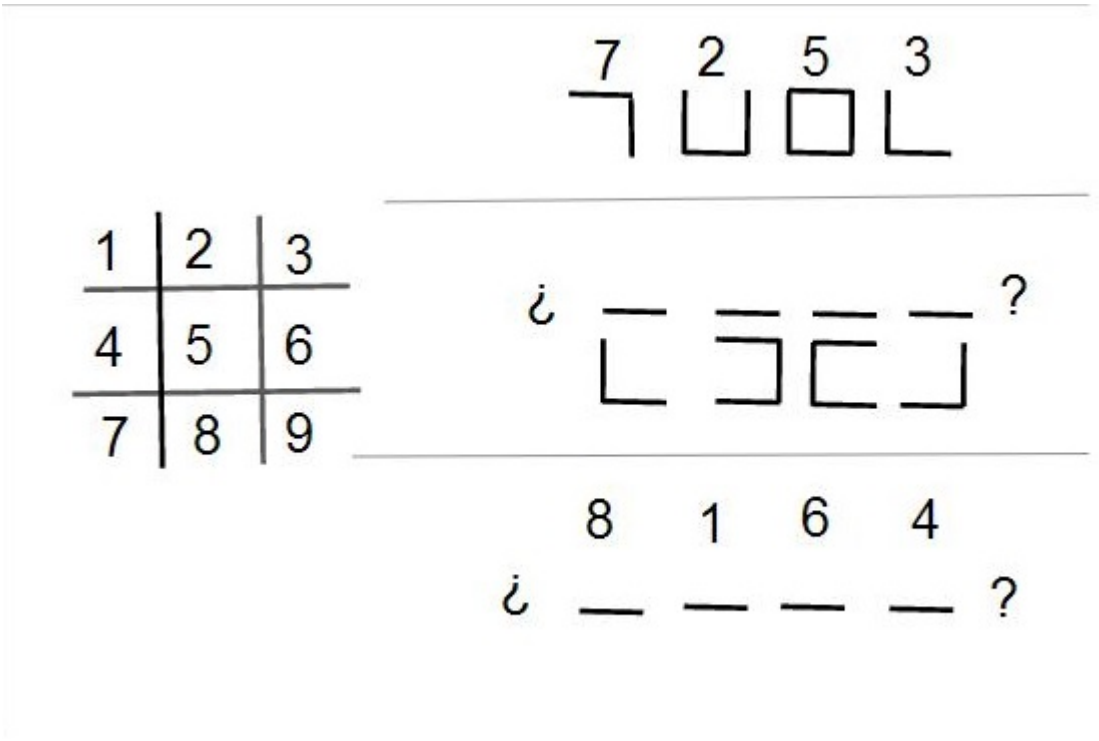
- La media en las personas es que hasta 4 elementos se ven y a partir de 5 se cuentan

¿Qué tal manejas estos sencillos sistemas? Escenario con Códigos

- Te voy a transmitir una clave con estos códigos, aprendetelos de memoria!

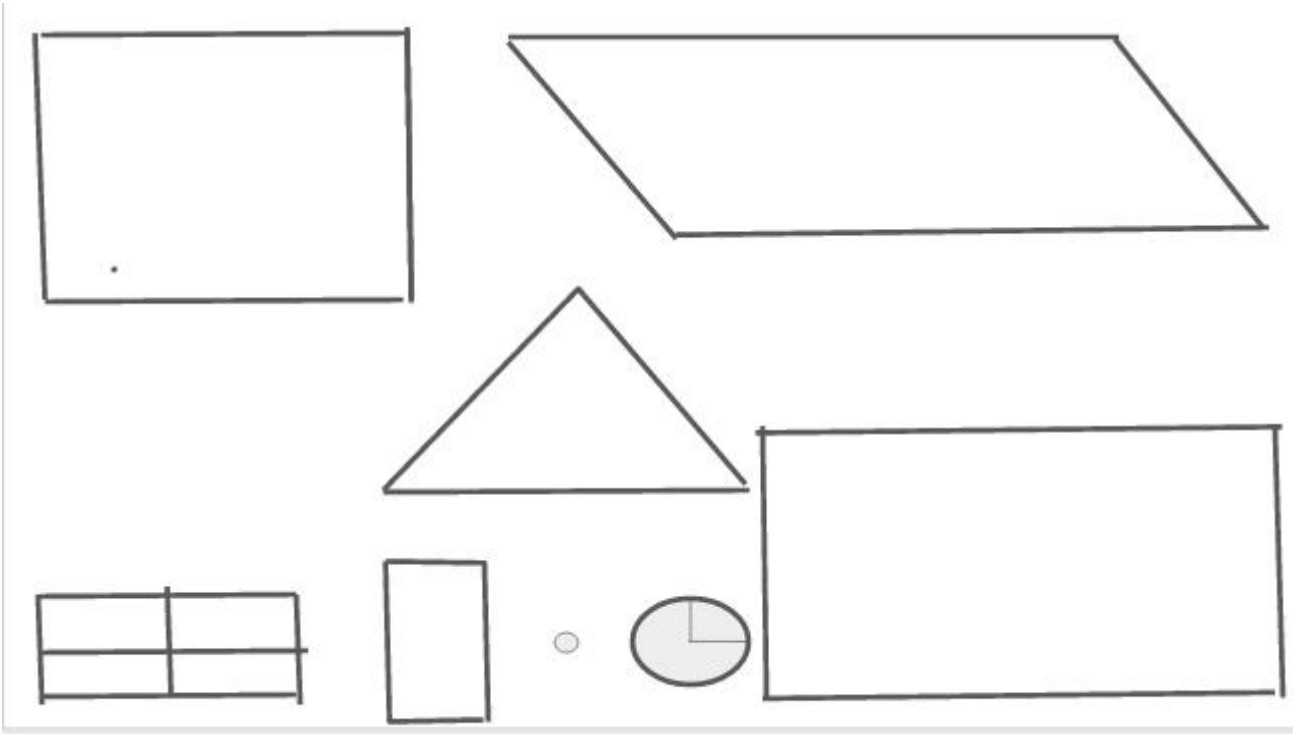


- Te voy a transmitir una clave con estos códigos: cada número con su contorno, aprendetelos de memoria!

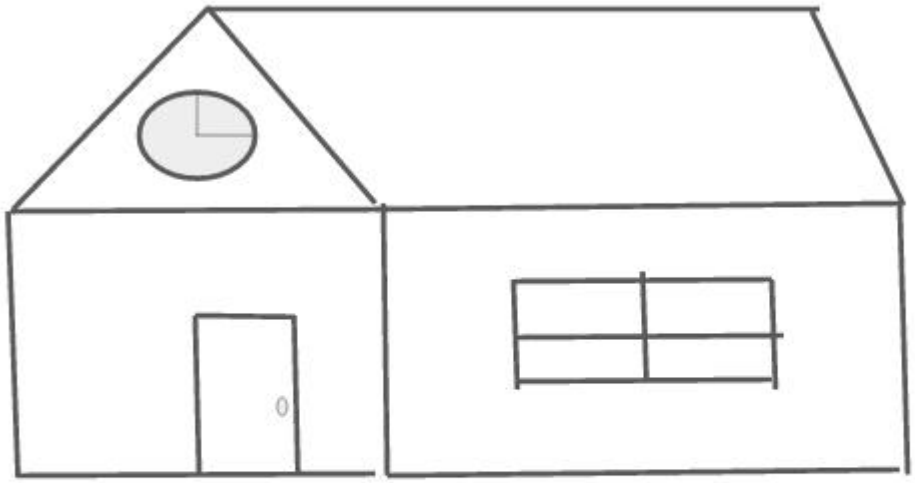


¿Qué tal manejas estos sencillos sistemas? Escenario con Figuras

- Mira el dibujo durante 5 segundos para memorizarlo y repetirlo con lápiz y papel

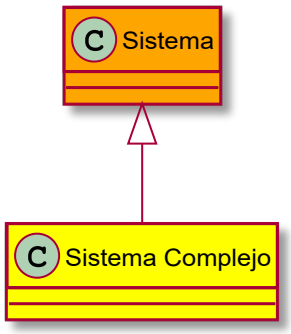


- Mira el dibujo durante 5 segundos para memorizarlo y repetirlo con lápiz y papel



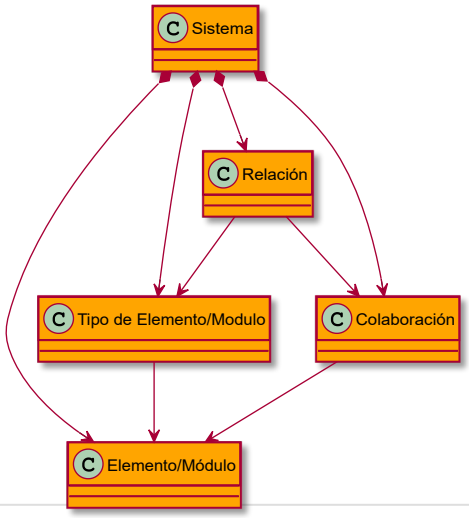
- Asociamos **muchos elementos** en **un todo**

Definición: ¿Qué?



Sistema

> *Un Sistema es un conjunto de componentes interactuando o interdependientes formando un todo integrado. Cada sistema está delimitado por sus límites espacio/temporales e influenciado por su entorno, descrito por su estructura y propósito y expresado en su funcionamiento*
>
> — *Wiki?*



Sistema	***Sistema respiratorio***	***Película de amor***	***Numeros Romanos***	***Semaforo***
conjunto de componentes interactuando o interdependientes formando un todo integrado.	*nariz, laringe, faringe, traquea, pulmones, alveolos, ...*	*personajes*	*I, V, X, L, C, D y M*	*Rojo, verde y amarillo*
Cada sistema está delimitado por sus límites espacio/temporales	*fechas y lugares de la vida del ser vivo que contiene el sistema respiratorio*	*fechas y lugares de los personajes*	*fechas y lugar donde estén escritos*	*fechas y lugar de la instalación del semáforo*
e influenciado por su entorno,	*lo que respira: aire limpio vs contaminado, ...*	*la sociedad, las familias, una ex-pareja, ...*	*en desuso en favor de sistemas de numeración poscionales (indo-arábigo, maya, chino, ...) mucho más efectivos*	*fuente de energía, climatología, vándalos, artistas, ...*
descrito por su estructura	*la nariz se conecta con la laringe, la laringe con la traquea, ...*	*argumento que relaciona los personajes de la historia*	*grupos de máximo 3 elementos consecutivos, grupo con elemento y opcionalmente un elemento inferiro prefijo o sufijo, ...*	*de rojo a verde, de verda a amarillo y de amarillo a verdo y/o rojo, ...*

Sistema	***Sistema respiratorio***	***Película de amor***	***Numeros Romanos***	***Semaforo***
y propósito	*inyectar oxigeno al sistema circulatorio extrayendo monóxido de carbono, ...*	*transmitir emociones*	*regsitrar información cuantitativa*	*controlar el tráfico*
y expresado en su funcionamiento.	*inspiración y expiración*	*reproducción de la película*	*suma, resta, producto, división, ... son información cuantitativas registrada*	*luces con alimentación electrica*

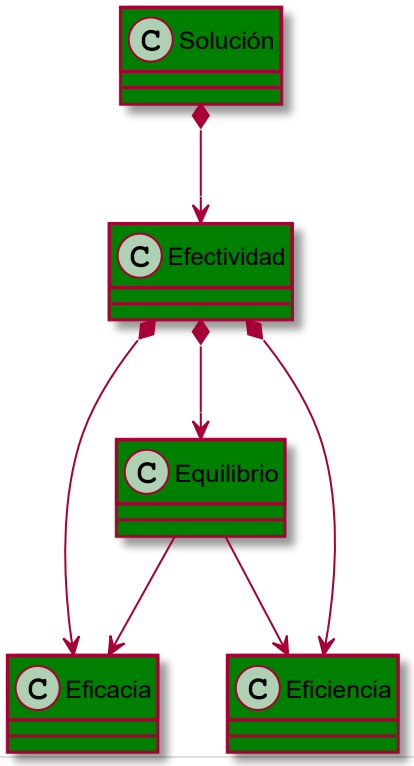
Sistema Complejo

> ***Un Sistema Complejo es aquel cuya complejida excede la capacidad intelectual humana***
>
> — *Booch*

Objetivos: ¿Para qué?

Efectividad

- **Efectividad** en una capacidad es un **equilibrio** entre:
  - **Eficacia**, alto cumplimiento de objetivos frente a incumplimientos por errores, desmotivación, cansancio, ...
  - **Eficiencia**, bajo consumo de recursos (tiempo, energía, espacio, ...) frente a consumos disparatados comparado con otras soluciones

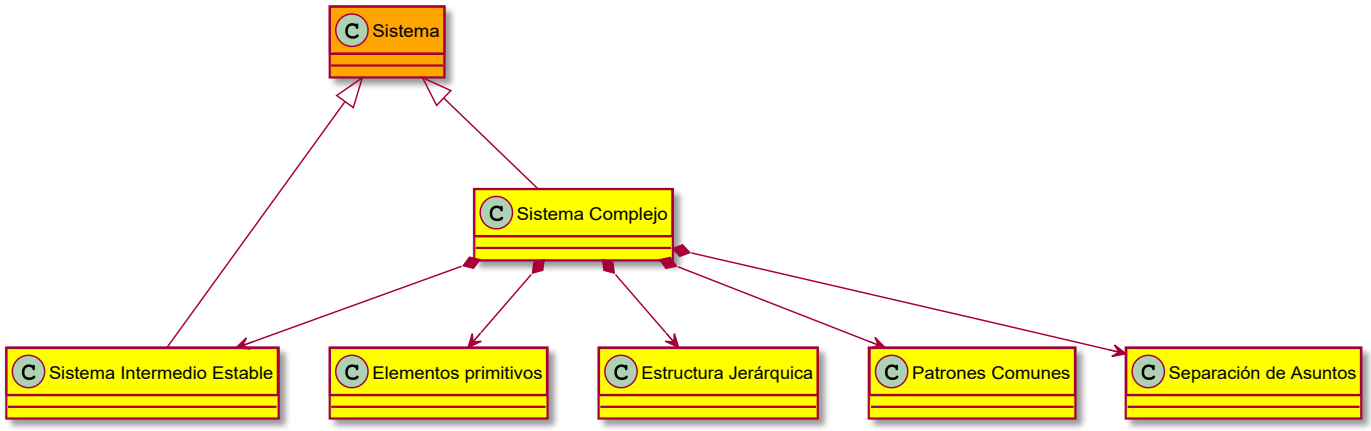


+	**(A) EFICAZ e INEFICIENTE** Haber alcanzado los retos sin cumplir con las pautas	**(B) EFICAZ y EFICIENTE** Haber alcanzado los retos con los recursos dispuestos
EFICACIA	**(C) INEFICAZ e INEFICIENTE** Haber fracasado en el cumplimiento de objetivos pese a extralimitar el uso de los medios	**(D) INEFICAZ y EFICIENTE** Haber utilizado bien los recursos disponibles sin alcanzar retos
-		
EFECTIVIDAD	- **EFICIENCIA** +	

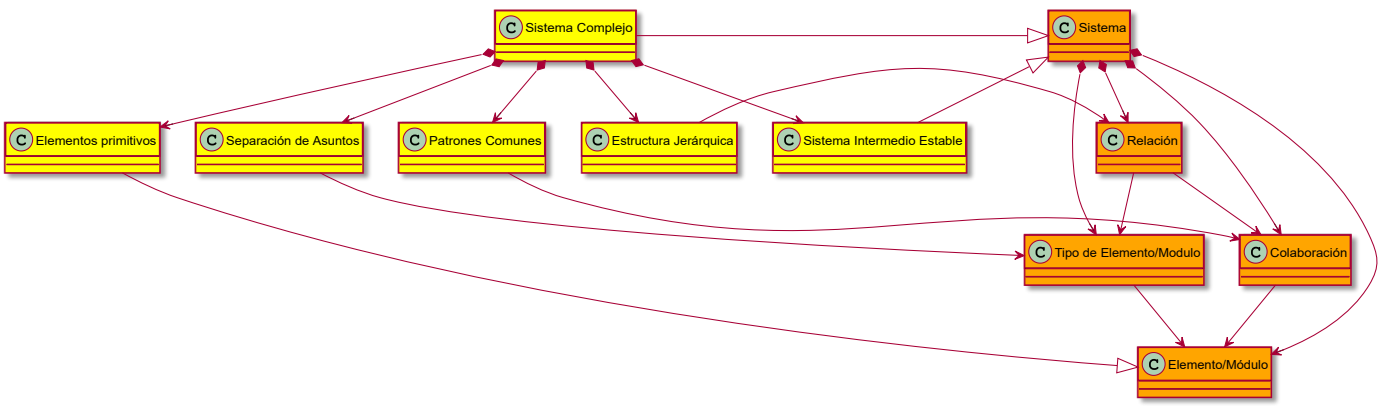
Efectividad *para Comprar el pan*	**Eficientemente**	**Ineficientemente**
Eficazmente	*Trae el pan en cinco minutos y devuelve la vuelta*	*Trae el pan en 20 minutos y no devuelve la vuelta*
Ineficazmente	*Trae el pan pero no la cantidad, tipos, ... oportunos en cinco minutos y devuelve la vuelta*	*Trae el pan pero no la cantidad, tipos, ... oportunos en 20 minutos y no devuelve la vuelta*

Descripción: ¿Cómo?

Características de Sistemas Complejos

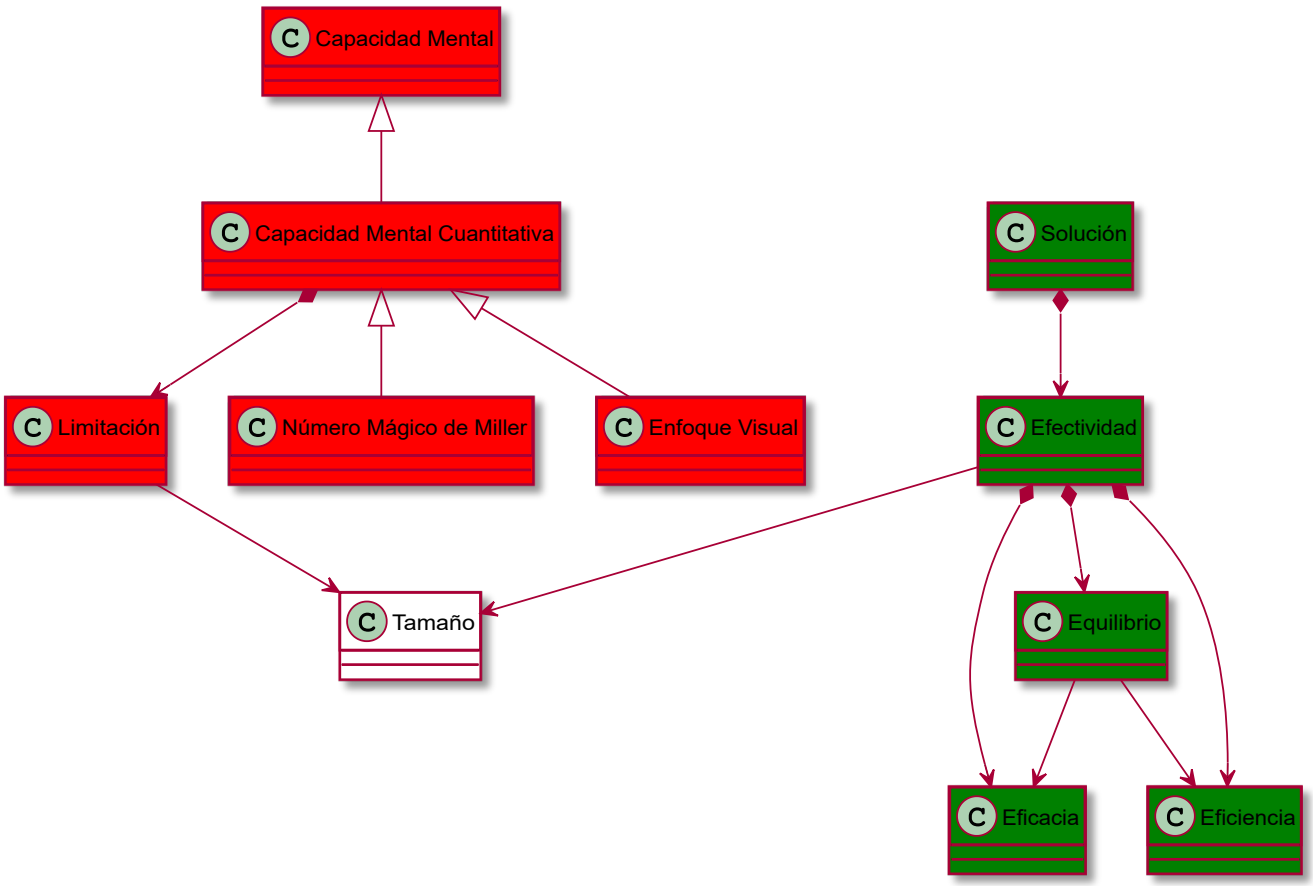


- **Estructura jerárquica**. Frecuentemente, la complejidad adquiere una forma jerárquica donde el sistema complejo está compuesto de subsistemas interrelacionados que a su vez tienen sus propios subsistemas y así hasta que se alcanza algún elemento del más bajo nivel. No solo son sistemas complejos jerárquicos sino que los niveles de su jerarquía representan los diferentes niveles de abstracción cada uno construido sobre otro y cada uno comprensible por sí mismo. En cada nivel de abstracción, encontramos una colección de elementos que colaboran para proveer servicios a niveles superiores
- **Elementos primitivos relativos**. La elección de qué componentes en un sistema son primitivos es relativamente arbitraria y mayormente está a discrección del observador del sistema
- **Separación de asuntos**. Las intra-conexiones de componentes son más fuertes que las inter-conexiones de componentes. Este hecho tiene el efecto de separar los componentes con dinámica de alta frecuencia (involucrando la interacción entre componentes) de los de dinámica de baja frecuencia. En términos sencillos,hay una clara separación de asuntos entre las partes de diferentes niveles de abstracción
- **Patrones comunes**. Los sistemas jerárquicos se componen generalmente de sólo unos pocos tipos diferentes de subsistemas en varias combinaciones y órdenes. Nos encontramos con una gran similitud en la forma de mecanismos compartidos unificando esta vasta jerarquía
- **Formas intermedias estables**. Un sistema complejo que funciona invariablemente se encuentra que ha evolucionado a partir de un sistema sencillo que funcionó. Un sistema complejo diseñado desde cero no funciona y no puede ser remendado para hacer que funcione. Usted tiene que comenzar de nuevo, a partir de un sistema sencillo de trabajo



Capacidades cuantitativas

- Con **Limitación para el Tamaño** de la carga del área de trabajo
- Inefectividad
  - **Ineficiencia**
    - **Ley de Hyks**: tiempo que tarda una persona en tomar una decisión respecto a la cantidad de posibles elecciones que tenga aumenta el tiempo de decisión logarítmicamente según aumenta el número de opciones
  - **Ineficacia**
    - **Número Mágico de Miller**: el número de objetos que una persona promedio puede tener en la memoria de trabajo es 7 ± 2. *Ejemplos*:
      - *Película de espías, 21 gramos, ... muchas historias paralelas poco relacionadas hasta el final*
      - *Código de números en cuadrícula, ... muchas frente a pocas reglas*
      - *Lenguaje natural, ... muchas palabras en muchas estructuras sintácticas*
    - **Enfoque Visual**: Ver hasta 4 elementos vs Contar a partir de 5 elementos
      - *Incluso no ves lo que ves (contar pases de pelota con ...)*



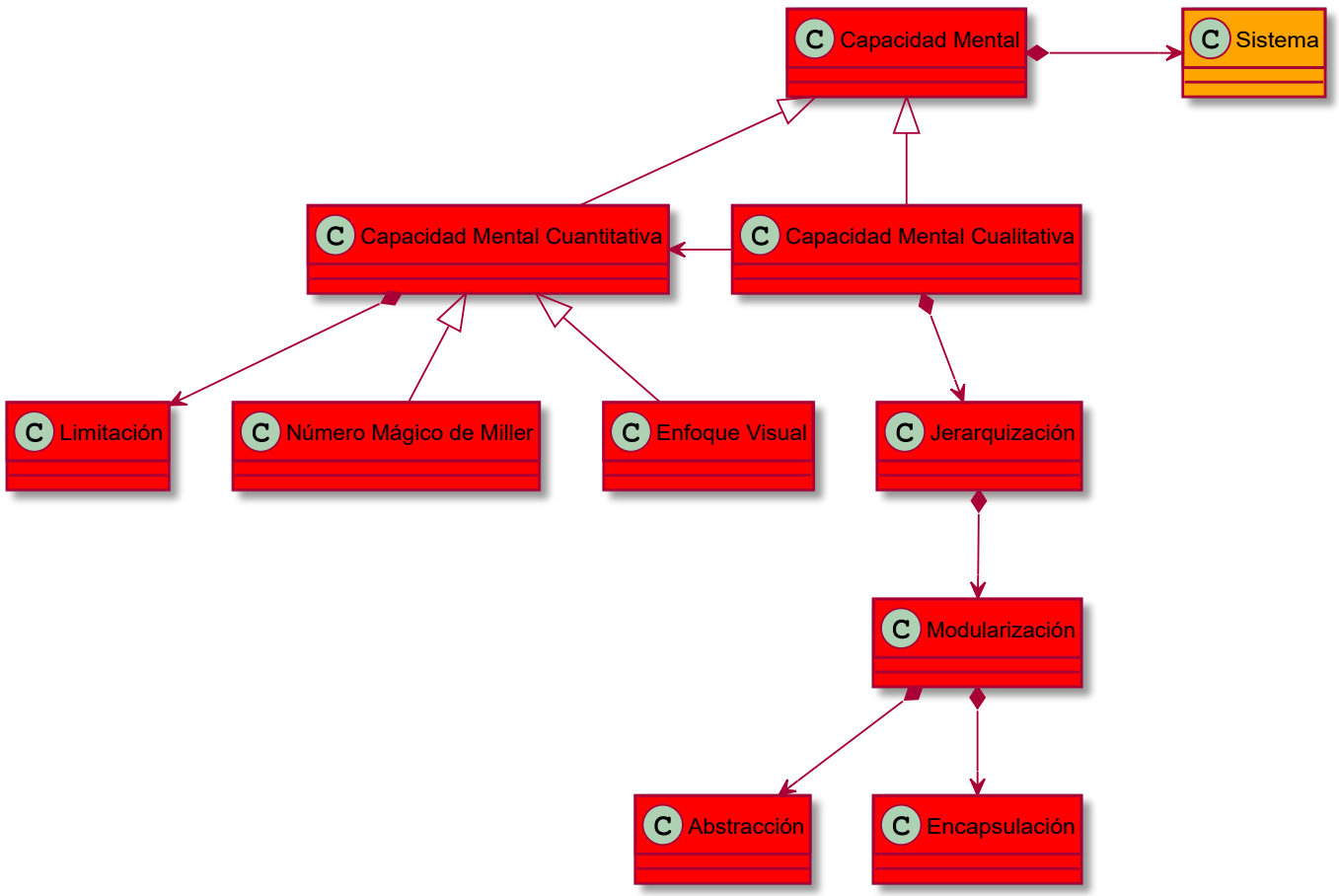
Ejemplo: Personas

- Las personas no son efectivas con tamaños elevados de elementos a gestionar por imposibilidad, agotamiento, desmotivación, distracción, ...)
  - No son eficaces por errores (¿nunca te has equivocado?, ...)
  - No son eficientes por
    - consumo de tiempo (calcula los primos del primer billón, a relación de un segundo por cálculo de si un número es primo, no hay segundos en la vida de una persona, ...)
    - consumo de espacio (calcula con lapiz y papel los posibles caminos entre todas las poblaciones de tu tierra natal, suponiendo que habrá miles, no hay hojas en el planeta, ...)

> *Divide et impera (divide y vencerás)*
>
> — *Imperio Romano*
> *Julio César*

Capacidades cualitativas

- La historia del ser humano disfruta de cuatro mecanismos mentales que facilitan enormemente nuestra comprensión de los sistemas complejos:
  - **Abstracción**
  - **Encapsulación**
  - **Modularización**
  - **Jerarquización**



Abstracción

> *La abstracción surge de un reconocimiento de similitudes entre ciertos objetos, situaciones o procesos en el mundo real, y la decisión de concentrarse en estas similitudes e ignorar por el momento, las diferencias*
>
> — *Dahl*
> *Dijkstra y Hoare*

> *La abstracción es "una descripción simplificada, o especificación, de un sistema que hace hincapié en algunos de los detalles o propiedades mientras que suprime otros del sistema. Una buena abstracción es la que hace hincapié en los detalles que son importantes para el lector o usuario y suprime detalles que son, al menos por ahora, de distracción*
>
> — *Shaw*

> *La abstracción es el proceso mental de extracción de las características esenciales de algo, ignorando los detalles superfluos*
>
> — *Booch*

- Implicaciones:
  - Una abstracción denota las **características esenciales** de un objeto que lo distinguen de todos los otros tipos de objetos y por lo tanto proporciona **límites conceptuales nítidamente definidos**, en relación con la perspectiva del espectador.
  - Una abstracción se centra en la **visión exterior** de un objeto y sirve para separar el comportamiento esencial de un objeto de su implantación
  - La abstracción es **eminentemente subjetiva**, dependiendo del interés del observador
  - Nos esforzamos para construir abstracciones de las **entidades porque son directamente paralelos al vocabulario del dominio de un determinado problema**.

Ejemplo: Varios

- Mundo real: un autobús de un pasajero o un mecánico, un ordenador, ...
- Software orientado a procesos: factorial, mostrar menú, ordenar, ...
- Software orientado a objetos: una fecha, un intervalo, un gestor de comunicaciones, un colección de datos, ...

Encapsulación

> *La encapsulación es proceso por el que se ocultan los detalles del soporte de las características esenciales de una abstracción*
>
> — *Booch*

- Hacer notar que en ninguno de los casos **no se trata de ocultar la información** en sí misma sino de ocultar los detalles del soporte de dicha información
  - La encapsulación se logra con mayor frecuencia a través de ocultación de información, que es el proceso de **ocultar todos los secretos de un objeto que no contribuyen a sus características esenciales**

- La encapsulación proporciona barreras explícitas entre las diferentes abstracciones y por lo tanto conduce a una clara **separación de asuntos**. El beneficio inmediato será la posibilidad de cambiar los soportes de las características de una abstracción sin afectar a todos los elementos que dependan de esas características porque ni los conocen, ni los mencionan
- Principio de Encapsulación: **todo aquello que no sea necesario dar a conocer, no se debe dar a conocer**

- Implicaciones:
  - Una vez realizada cierta abstracción hay que "trasladarlas" al lenguaje de programación. Esto conlleva decidir entre diversas estructuras de datos (estáticas o dinámicas, en memoria principal o secundaria, etc.) y/o diversos algoritmos (¿con variables auxiliares o no? ¿recursivo o iterativo?, etc.), siendo diversas las alternativas que recogen dichas características esenciales. Una vez que se selecciona una implantación, debe ser tratado como un secreto de la abstracción y oculta a la mayoría de los clientes. En la práctica, esto significa que cada clase debe tener dos partes:
    - La **interfaz** de una clase capta sólo su vista exterior, que abarca nuestra abstracción del comportamiento común a todas las instancias de la clase. La interfaz de una clase es el único lugar donde establecemos todas los suposiciones que un cliente puede hacer sobre cualquier instancia de la clase
    - La **implementación** de una clase comprende la representación de la abstracción, así como los mecanismos para conseguir el comportamiento deseado. La implementación encapsula detalles sobre los qué ningún cliente puede hacer suposiciones.
    - La **abstracción de un objeto debe preceder a las decisiones acerca de su implantación**.
    - **Ninguna parte de un sistema complejo debe depender de los detalles internos de cualquier otra parte**. Mientras que la abstracción ayuda a las personas a pensar en lo que están haciendo, la encapsulación permite hacer cambios fiables en el programa con un esfuerzo limitado.

Ejemplo: Varios

- Mundo real: un autobús, un ordenador, una universidad, ...
- Software orientado a procesos: factorial, mostrar menú, ordenar, ...
- Software orientado a objetos: una fecha, un intervalo, un gestor de comunicaciones, un colección de datos, ...

Modularización

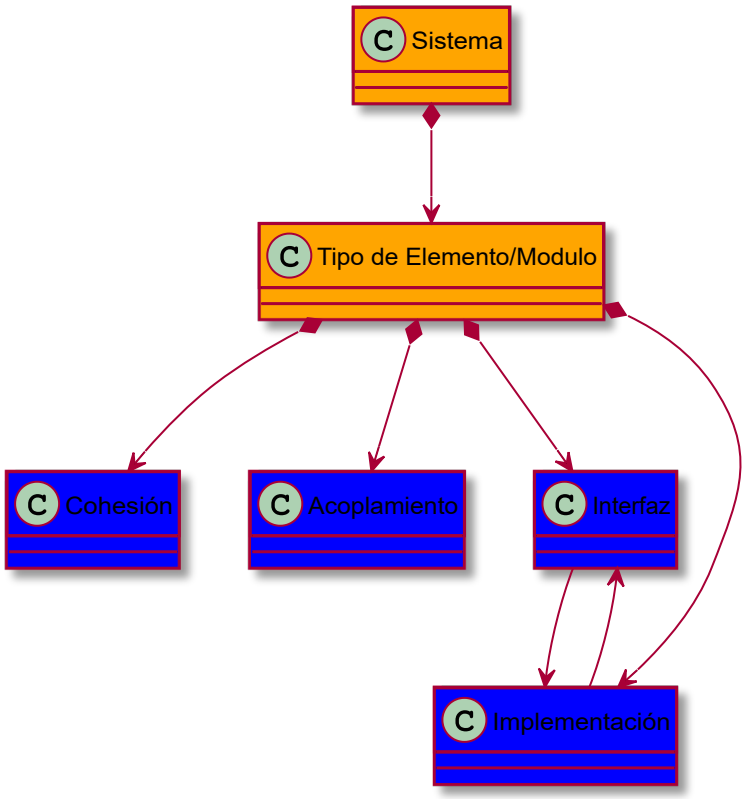
> *La **modularidad** es el proceso de descomposición de un sistema en un conjunto de piezas poco acoplados y cohesivos*
>
> — *Booch 96*

> *El **acoplamiento** "[...] es la medida de fuerza de la asociación establecida por una conexión ente un módulo -elemento- y otro. El acoplamiento fuerte complica un sistema porque los módulos son más difíciles de comprender, cambiar o corregir por sí mismos si están muy interrelacionados con otros módulos*
>
> — *Booch 96*

> *La **cohesión** mide el grado de conectividad entre los elementos de un solo módulo*
>
> — *Booch 96*

- Al diseñar un sistema de software complejo, es esencial para descomponer en partes más pequeñas y más pequeñas, cada una de las cuales podemos entonces refinar independientemente. De esta manera, satisfacemos la restricción muy real que existe sobre **la capacidad del canal de la cognición humana**: para entender cualquier nivel dado de un sistema, sólo necesitamos comprender algunas partes (en lugar de todas las partes) a la vez.
  - Para modularizar hay que **minimizar las dependencias entre módulos** (acoplamiento) que deben tener significado propio por sí mismo **agrupando abstracciones lógicamente relacionadas** (cohesión)
  - El **bajo acoplamiento de un modulo se basa en la abstracción** que limita su interfaz a lo esencial **y en la encapsulación** que oculta todos los detalles necesarios para su implantación pero innecesarios para otros módulos que se relacionen con éste
  - Dividir un programa en una serie de límites documentados bien definidos dentro del programa es de **gran valor en la comprensión del programa**
  - Debería ser posible cambiar la implementación de otros módulos sin el conocimiento de la aplicación de otros módulos y sin afectar el comportamiento de los otros módulos

> *La descomposición inteligente se dirige directamente a la complejidad inherente del software al obligar a una división del espacio de estados de un sistema*
>
> — *Parnas*



Ejemplo: Varios

- Mundo real: un autobús, un ordenador, una universidad, ...
- Software orientado a procesos: factorial, mostrar menú, ordenar, ...
- Software orientado a objetos: una fecha, un intervalo, un gestor de comunicaciones, un colección de datos, ...

Jearquización

> *Jerarquía es una clasificación u ordenamiento de las abstracciones*
>
> — *Booch*

> *La jerarquización es el proceso de estructuración por el que se produce una organización de un conjunto de elementos en grados o niveles de responsabilidad, de clasificación o de composición, ... entre otros*
>
> — *Fernando Arroyo*

- Implicaciones:
  - La abstracción es una buena cosa pero en todos los casos, excepto las aplicaciones más triviales, podemos encontrar muchas más abstracciones diferentes de lo que podemos comprender a la vez. La encapsulación ayuda a gestionar esta complejidad al ocultar el interior de la vista de nuestras abstracciones. La modularidad ayuda también, por que nos da una manera de agrupar abstracciones relacionados lógicamente. Aún así, esto no es suficiente. Un conjunto de abstracciones a menudo forma una jerarquía, y **mediante la identificación de estas jerarquías en nuestro diseño se simplifica enormemente nuestra comprensión del problema**.

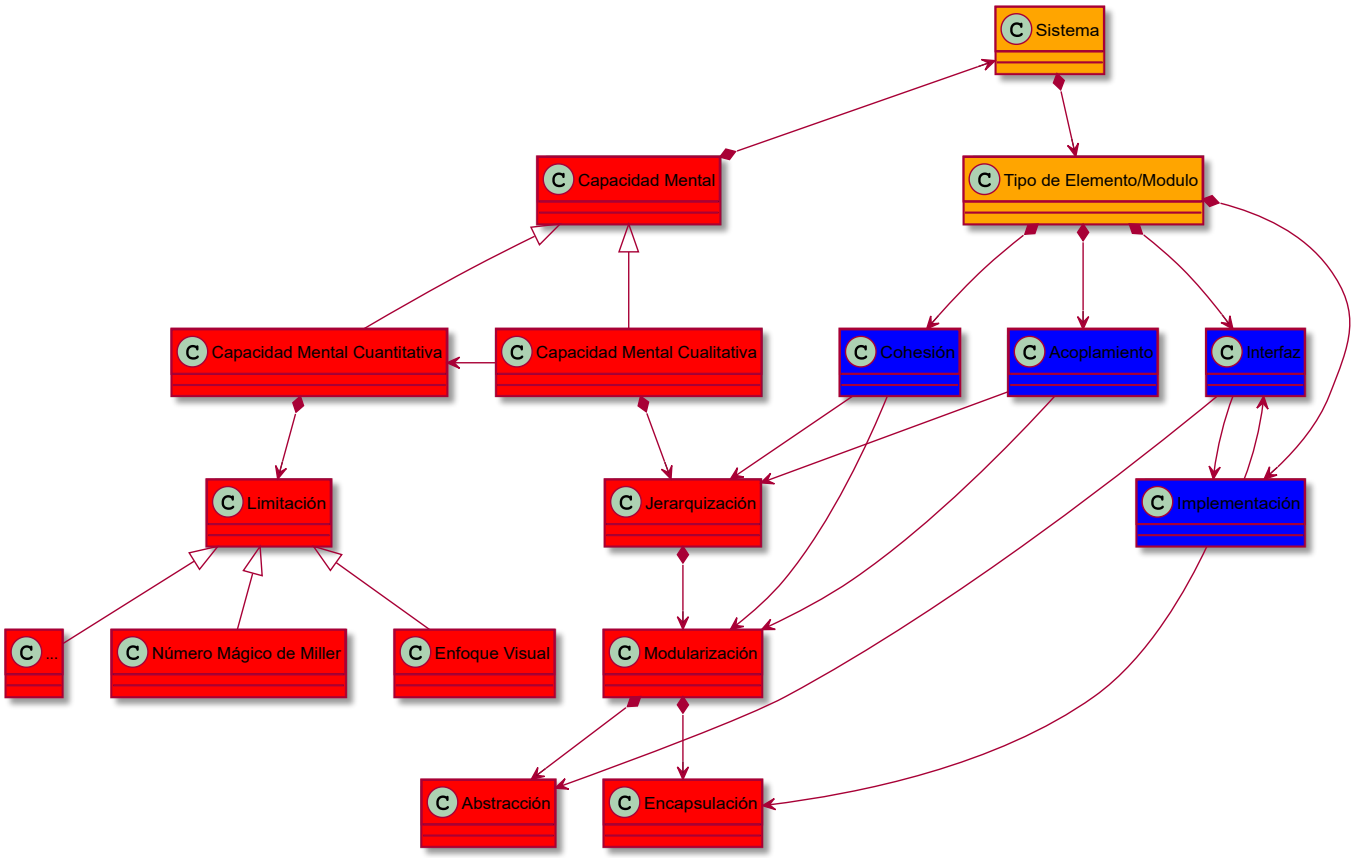
- La identificación de las jerarquías dentro de un sistema de software complejo a menudo **no es fácil**. Una vez que se exponen esas jerarquías, la estructura de un sistema complejo se vuelve muy simple y obtenemos la comprensión de la misma.
- Si no revelamos la estructura de clases de un sistema, tendríamos que multiplicar nuestro conocimiento sobre las propiedades de cada parte individual. Con la inclusión de la estructura de clases, captamos estas **propiedades comunes en un solo lugar**.
- Existen **dos jerarquías ortogonales del sistema: la estructura de clases y la estructura de objetos**. Cada jerarquía está en capas, con clases más abstractas y objetos construidos sobre otros más primitivos. La clase u objeto que se elija como primitivo está en relación con el problema en cuestión. Mirando dentro de cualquier nivel dado revela otro nivel de complejidad. Especialmente entre las partes de la estructura del objeto, existe una estrecha colaboración entre los objetos de ese mismo nivel de abstracción.
- **La estructura de clases y la estructura de objetos no son completamente independientes**; más bien, cada objeto en la estructura de objetos representa una instancia específica de una clase. Por lo general hay muchos más objetos que clases de objetos dentro de un sistema complejo. Al mostrar la "parte de", así como la jerarquía "es un", exponemos de forma explícita la redundancia del sistema considerado.
- La mayoría de los sistemas interesantes **no incorporan una única jerarquía**; en cambio, nos encontramos con que muchas jerarquías diferentes suelen estar presentes dentro del mismo sistema complejo. En nuestra experiencia, hemos encontrado que es esencial para ver un sistema desde ambas perspectivas, estudiando su jerarquía "es un" (clasificación), así como su jerarquía "parte de“ (composición)

> *Nuestra experiencia es que los sistemas de software complejos más exitosos son aquellos cuyos diseños incluyen explícitamente las estructuras de clases y objetos bien diseñadas y encarnan los cinco atributos de sistemas complejos descritos en la sección anterior. [...] Muy raramente nos encontramos con sistemas de software que se entregan a tiempo, que están dentro del presupuesto y que cumplen con sus requisitos, a menos que estén diseñados con estos factores en mente*
>
> — *Booch*

Ejemplo: Varios

- Mundo real: un autobús, un ordenador, una universidad, ... se componen de otros elementos y los hay de varias clases similares
- Software orientado a procesos: factorial, mostrar menú, ordenar, ... se componen de otros elementos y los hay de varias clases similares
- Software orientado a objetos: una fecha, un intervalo, un gestor de comunicaciones, un colección de datos, ... se componen de otros elementos y los hay de varias clases similares

Síntesis


Sistema
Capacidad Mental
Tipo de Elemento/Modulo
Capacidad Mental Cuantitativa
Capacidad Mental Cualitativa
Cohesión
Acoplamiento
Interfaz
Limitación
Jerarquización
Implementación
...
Número Mágico de Miller
Enfoque Visual
Modularización
Abstracción
Encapsulación