

Proyecto de Articulación Académica

IMPLICANCIAS EN MDS II



UAIOnline
Ultra»»



PROYECTO DE ARTICULACIÓN

- Implicancias en MDS II
 - Se espera, al finalizar el curso, analizar y diseñar los siguientes ítems:
 - Login / Logout:
 - Perfiles de usuario
 - Múltiples Idiomas
 - Diagrama de clases inicial

PROYECTO DE ARTICULACIÓN

- Contexto
 - Esta actividad deberá desarrollarse al finalizar la unidad 2, en clase y de manera grupal.
- Saberes previos del proyecto de articulación horizontal.
 - Arquitecturas (LUG)
- Objetivos en MDS II
 - En esta actividad se revisarán e integrarán diferentes conceptos vistos en la unidad 2, entre los que destacamos
 - Casos de uso
 - Clases y Objetos
 - Herencia
 - Agregación
 - Clases asociativas
 - Visibilidad de atributos y operaciones
 - Diagramas de secuencia

ACTIVIDADES A DESARROLLAR



UAIOnline
Ultra»»



LOG IN / LOG OUT - ALCANCE

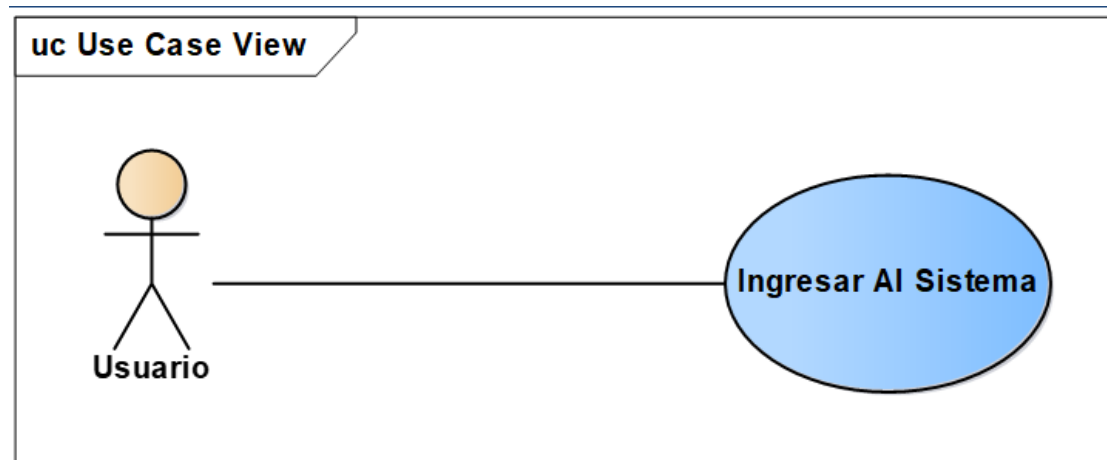
- Permite verificar y validar la identidad del usuario a través del ingreso de su nombre de usuario y su clave, asignándole el perfil que tenga en el sistema.
- Se debe describir cómo será la política de 'log-in' / 'log-out'. También deberán diferenciarse y documentarse los procesos que se correrán en el arranque del sistema (startup), el log in y el apagado de sistema (down), el log-out (permisos, auditoria, control de integridad, etc.). Se espera el uso del patrón singleton.
- Casos de uso asociados
 - Login
 - Logout

LOG IN / LOG OUT

- Objetivo
 - Permite verificar y validar la identidad del usuario a través del ingreso de su nombre de usuario y su clave, asignándole el perfil que tenga en el sistema.
- Descripción detallada de cómo funciona
 - El usuario podrá ingresar al sistema utilizando una pantalla que le solicite usuario y contraseña. Al ingresar, se creará una sesión que podrá ser consultada desde cualquier punto del sistema, respetando el concepto de identidad de objetos (un único objeto creado para mantener la sesión). Además, tendrá la opción de cerrar sesión (log-out) con la consecuente eliminación de dicho objeto

LOG IN / LOG OUT

- Casos de uso asociados
 - Login

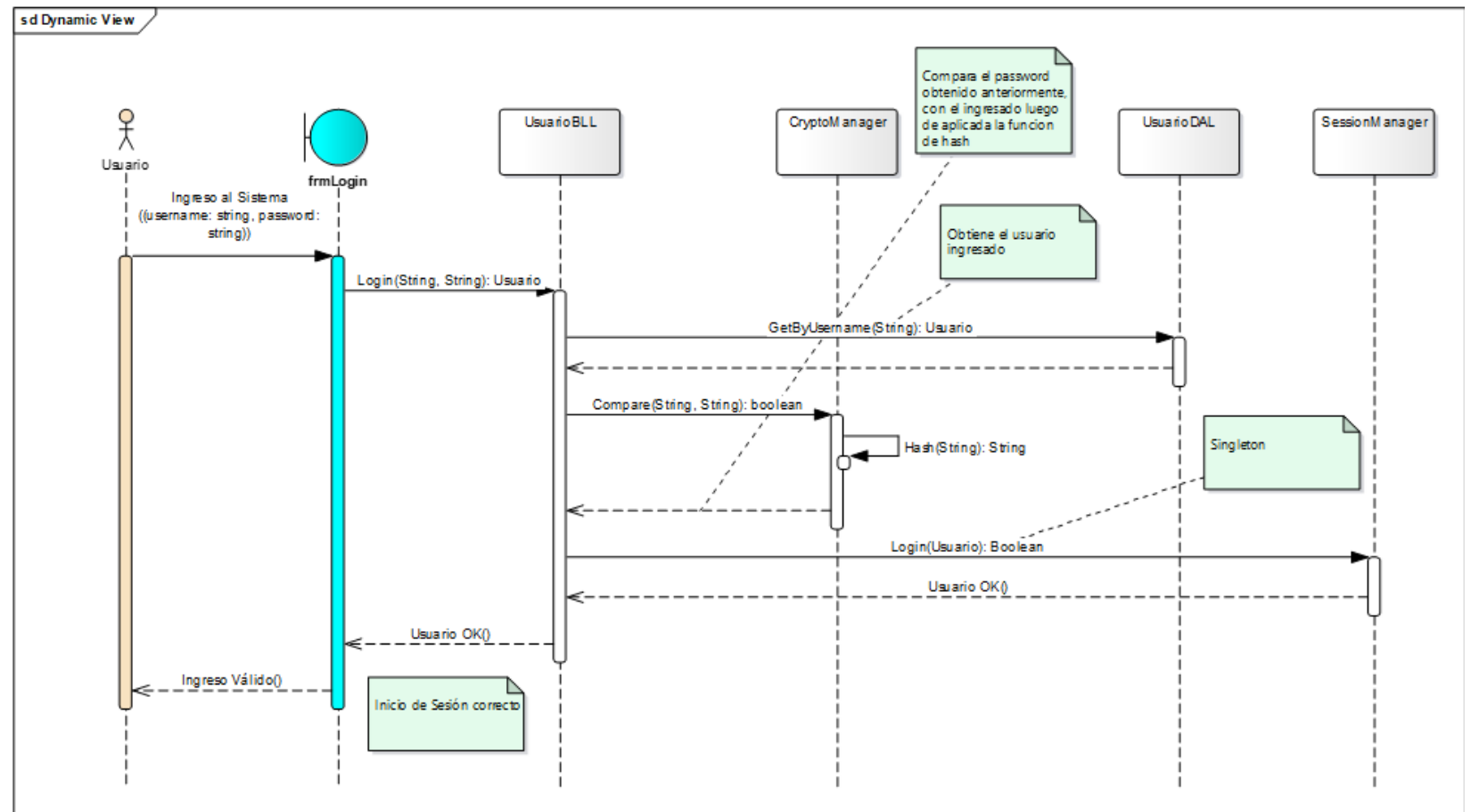


LOG IN / LOG OUT

- Especificación del caso de uso (debe ser consistente luego, con el diagrama de secuencia)
 - Nombre: Log in (iniciar sesión)
 - Actor: Usuario del sistema
 - Precondición: Que el usuario no haya iniciado sesión
 - Escenario principal
 - El actor ingresa su nombre de usuario y contraseña, luego presiona el botón ingresar
 - El sistema le da la bienvenida
 - Escenarios alternativos
 - El sistema le informa que sus credenciales son incorrectas
 - Postcondición: Una nueva sesión creada en el sistema

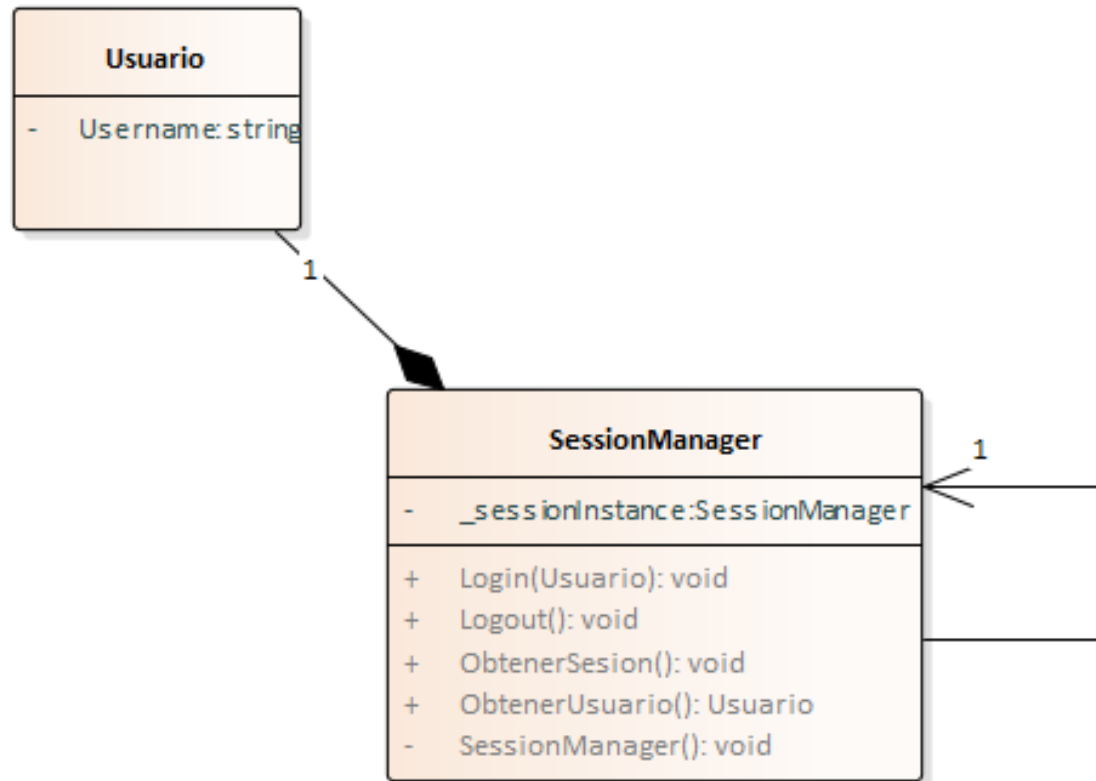
LOG IN / LOG OUT

- Diagrama de secuencia
- Login



LOG IN / LOG OUT

- Diagrama de clases
 - Log in



- La operación `ObtenerUsuario` debe ser estática para poder crear la sesión si no existe y devolverla como retorno del mensaje.
- Con este esquema creamos y mantenemos una única instancia del objeto `SesionManager` que será accesible desde todo el sistema para poder consultarlo

PERFILES DE USUARIO - ALCANCE

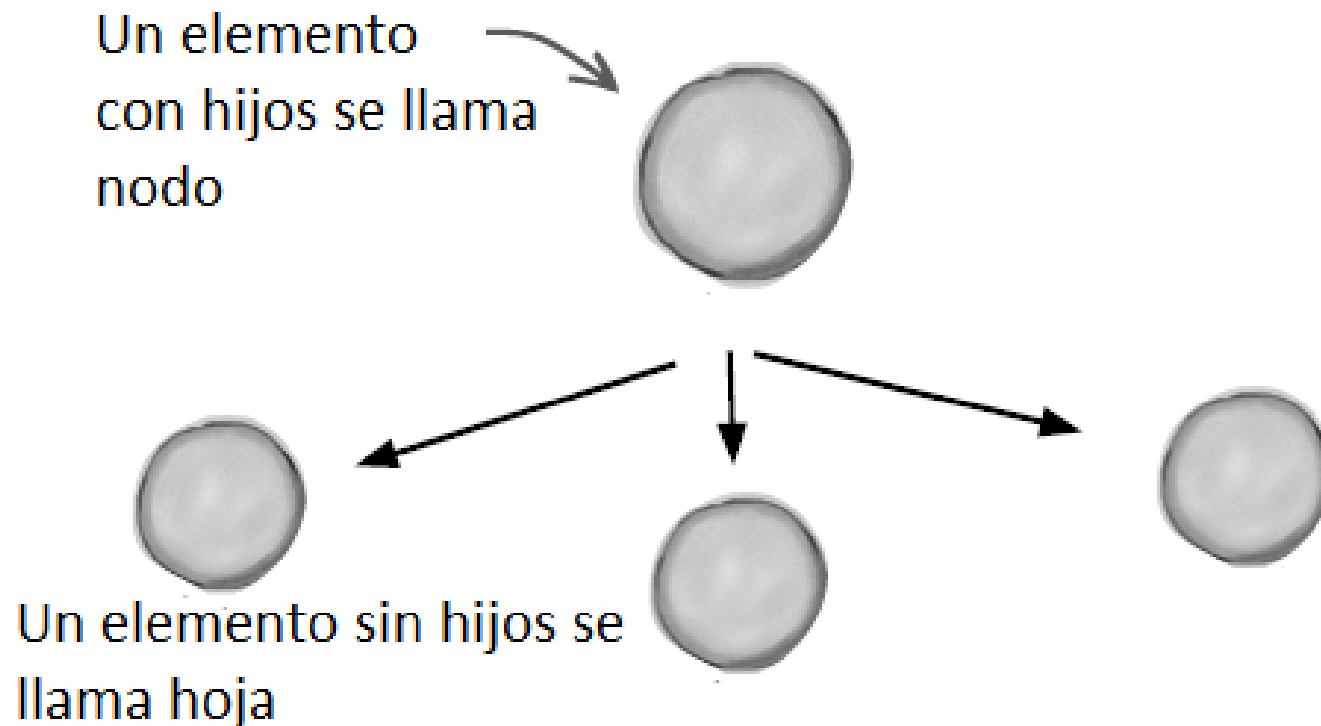
- Objetivo
 - El método seleccionado debe permitir la asignación rápida de los permisos y la definición de perfiles desde el sistema. Un permiso podrá ser atómico cuando represente a una funcionalidad o compuesto cuando agrupe a un conjunto de permisos
- Casos de uso asociados
 - Crear un permiso compuesto
 - Asignar un permiso (simple o compuesto) a un usuario

PERFILES DE USUARIO

- Descripción detallada de cómo funciona
 - Desarrollar un esquema de seguridad basados en patentes y familias. El sistema permitirá el acceso a sus diferentes funcionalidades en base a las “Patentes”, que estarán ordenados lógicamente en “Familias”.
 - Patente 1: Puede emitir factura
 - Patente 2: Puede realizar cobranza
 - Patente 3: Puede consultar status financiero
 - Familia 1: Vendedor
 - Patente 1, Patente 2
 - Familia 2: Gerente de Ventas
 - Familia 1
 - Patente 3.

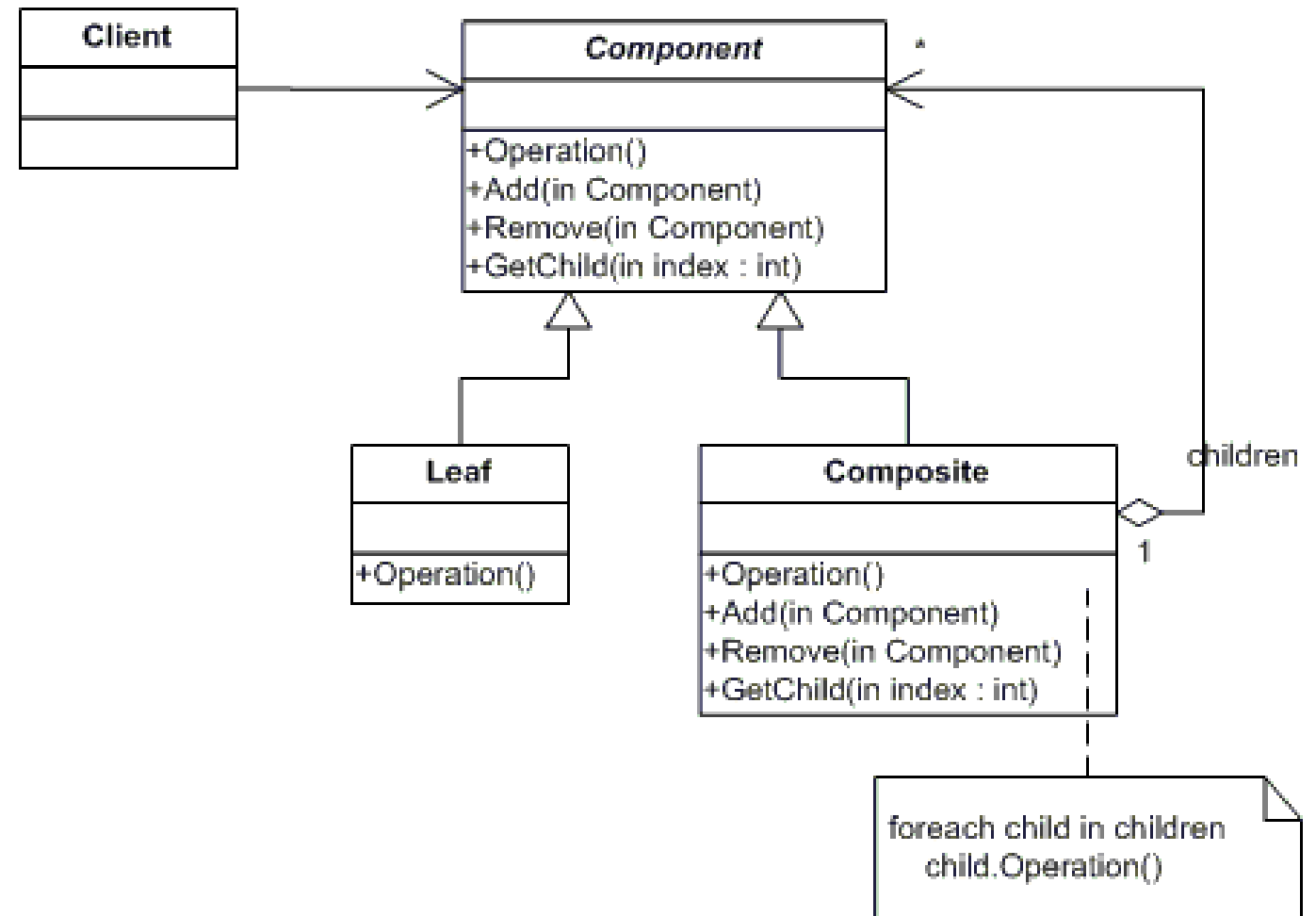
PERFILES DE USUARIO - SOLUCIÓN

- Componer objetos en estructuras de árbol para representar jerarquías de parte-todo.
- Define jerarquías de clases formadas por objetos primitivos y compuestos



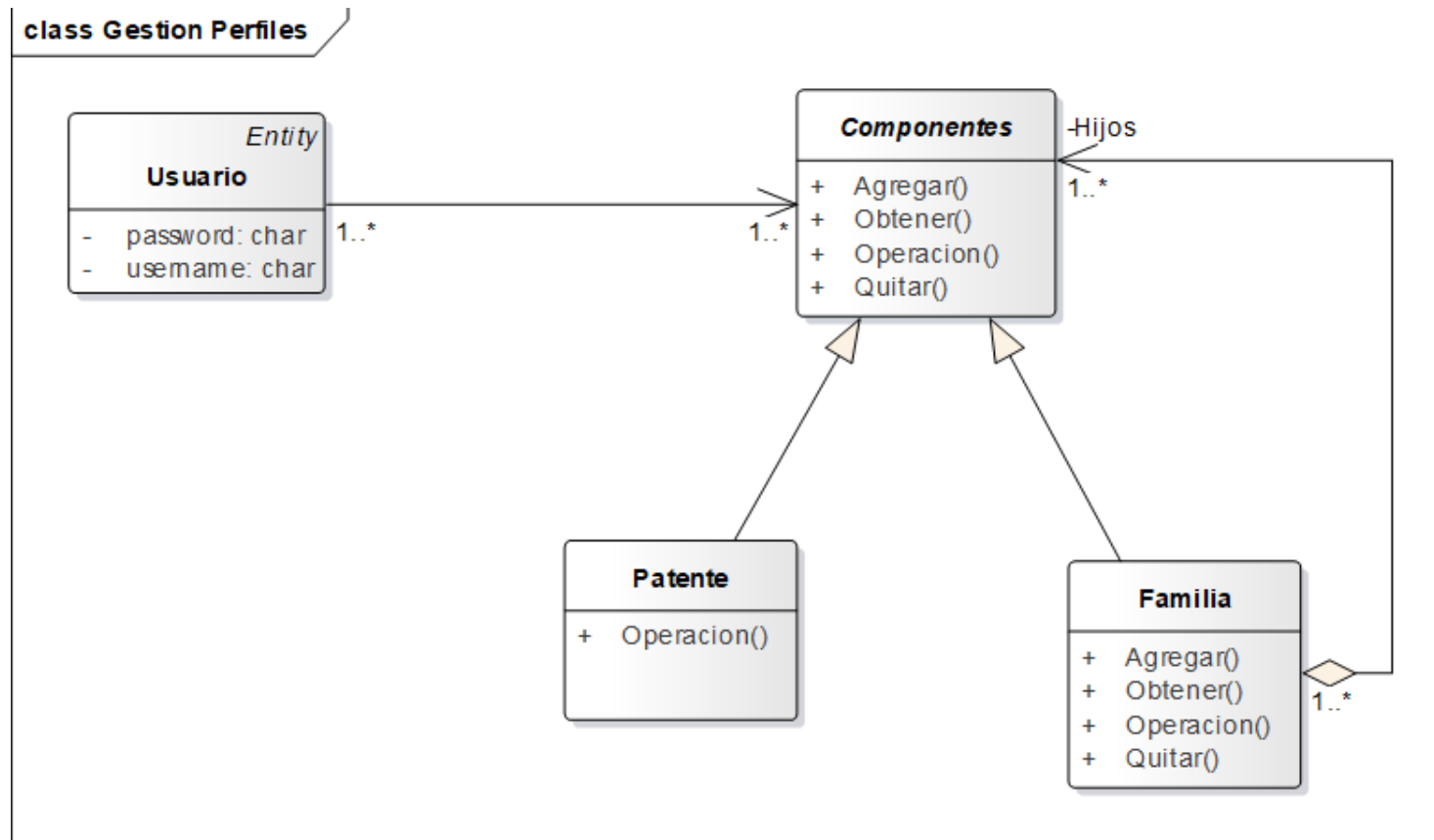
PERFILES DE USUARIO: PATRÓN COMPOSITE

- En este patrón se observa:
 - Herencia
 - Agregación
 - Relación “A muchos” (puede ser 1...* o *...* según la necesidad)



PERFILES DE USUARIO: PATRÓN COMPOSITE - SOLUCIÓN

■ Diagrama de clases



Una familia podrá tener dentro muchos "Componentes" que pueden ser "Patentes" o "Familias"

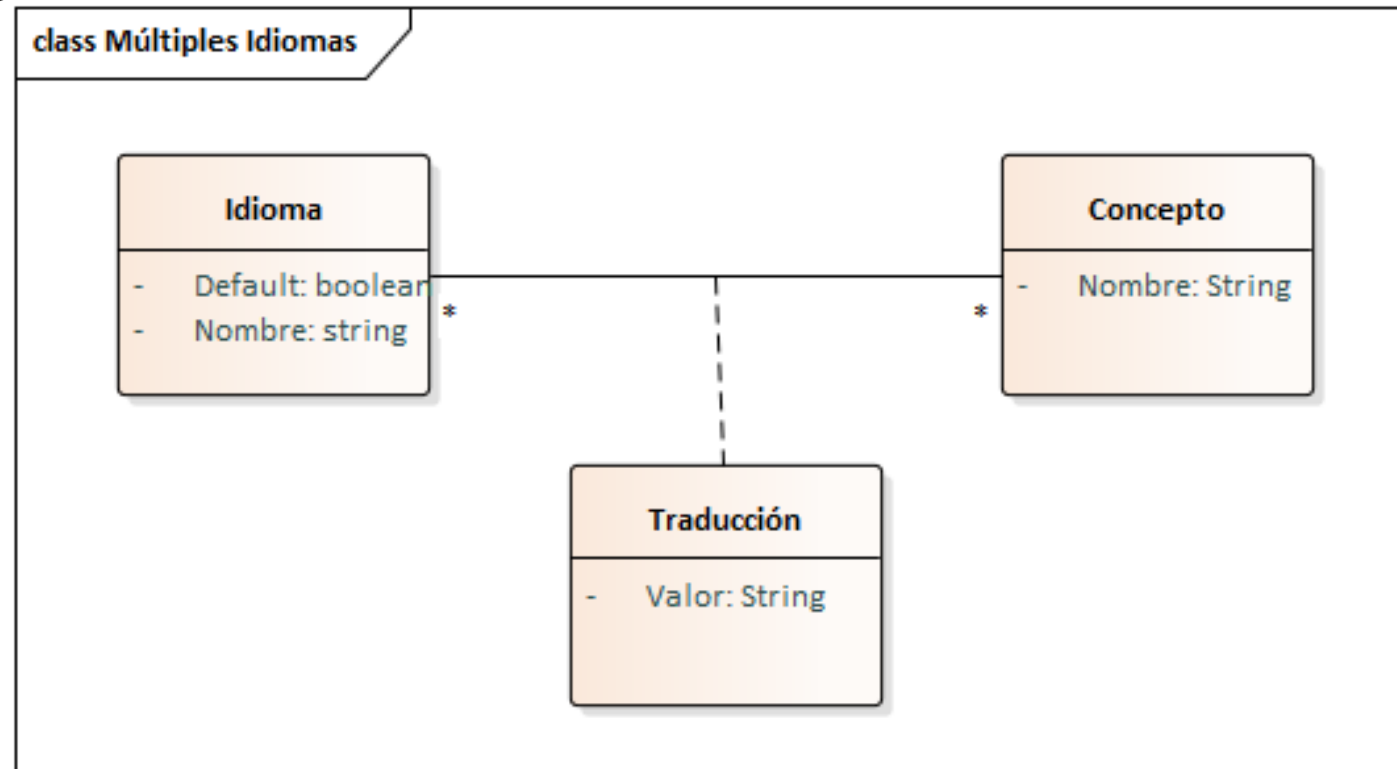
Para poder consultar todo el árbol es necesario recurrir a la recursividad. Cada familia podrá tener N componentes que a la vez pueden ser familias o patentes.

MÚLTIPLES IDIOMAS- ALCANCE

- Objetivo
 - Debe permitir el cambio de idioma de todas las leyendas y títulos que se lean en las interfaces de usuario
 - Se debe crear un modelo de objetos relacionado que permita crear y mantener múltiples idiomas de manera dinámica.
- Descripción detallada de como funciona
 - Los usuarios que dispongan de los permisos necesarios, podrán crear nuevos idiomas así también como traducir palabras de idiomas existentes. Se espera que al crear un nuevo idioma, se copien las traducciones del idioma por default que exista en el sistema en ese momento, para que luego el usuario pueda traducir manualmente.
- Casos de uso asociados
 - Crear Idioma
 - Traducir Palabras

MÚLTIPLES IDIOMAS- ALCANCE

■ Diagrama de clases





Fin de la clase



UAI

**Universidad Abierta
Interamericana**