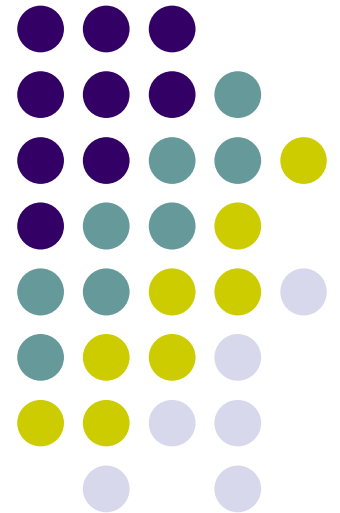
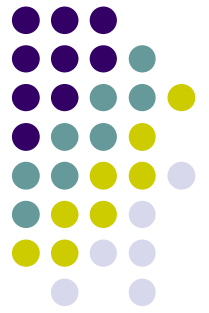


Metodologías de Diseño y Programación

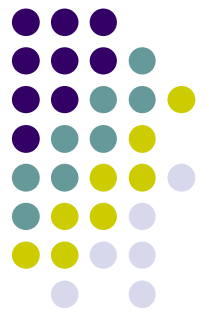
Diseño
Diagramas de Colaboración





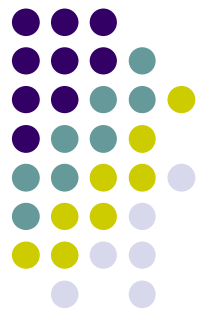
Contenido

- Diagramas de Interacción
- Notación



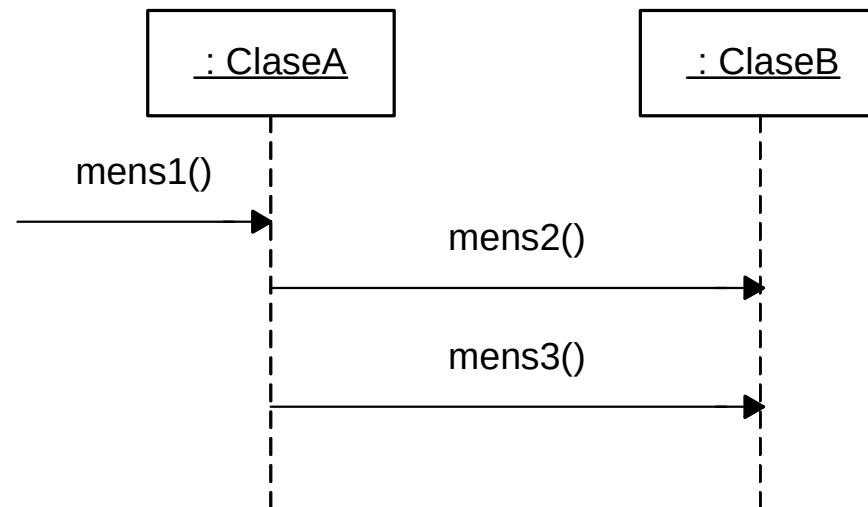
Diagramas de Interacción

- UML incluye los **diagramas de interacción** que sirven para mostrar ejemplos de cómo ciertos objetos interactúan a través de mensajes para la realización de tareas
- Existen dos tipos de diagramas de interacción que son semánticamente equivalentes entre sí
 - Diagramas de secuencia
 - Diagramas de colaboración

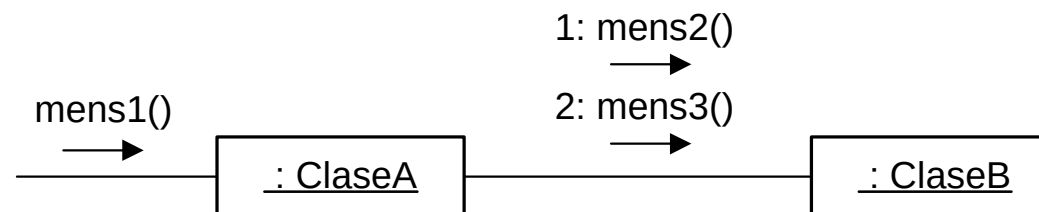


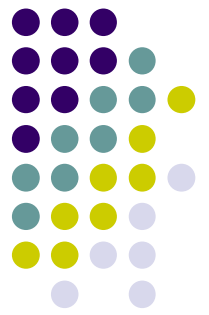
Diagramas de Interacción (2)

- Un Diagrama de Secuencia



- Su Diagrama de Colaboración equivalente





Notación Instancias

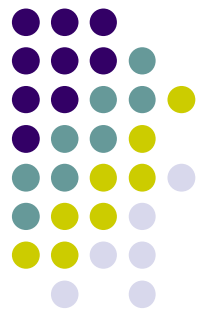
- Las instancias se representan igual que en los diagramas de instancias
- Corresponden a una instancia “cualquiera” de una cierta clase (no a una instancia real)

: Persona

Sin nombre

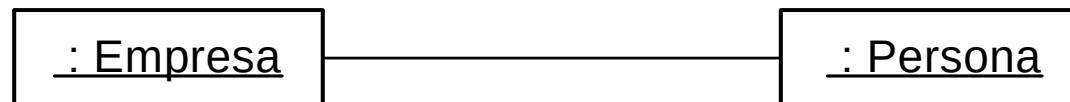
p : Persona

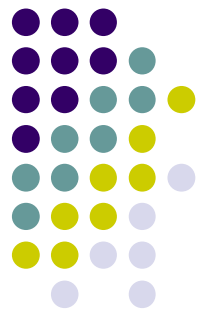
Con nombre



Notación Links

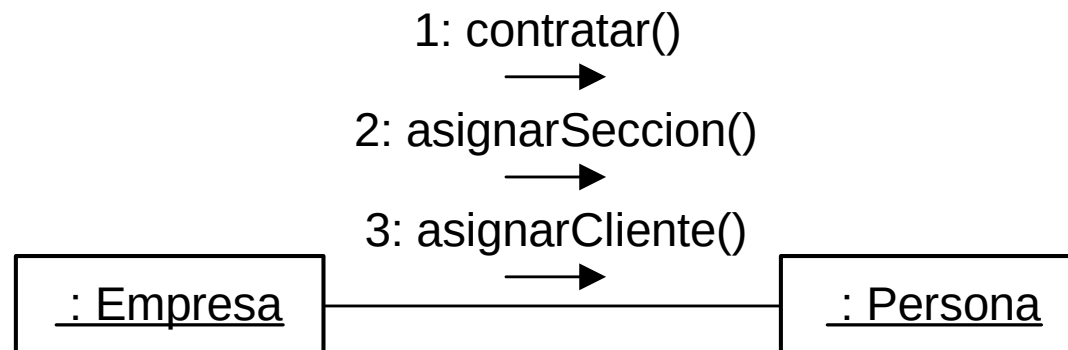
- Representa una conexión entre instancias que indica navegabilidad y visibilidad entre ellas
- Establece una relación de cliente/servidor entre las instancias

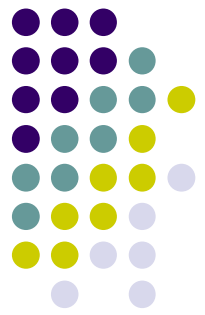




Notación Mensajes

- Los mensajes son representados mediante una flecha etiquetada
- Un mensaje está asociado a un link y tiene asignado un número de secuencia que determina el orden de ocurrencia

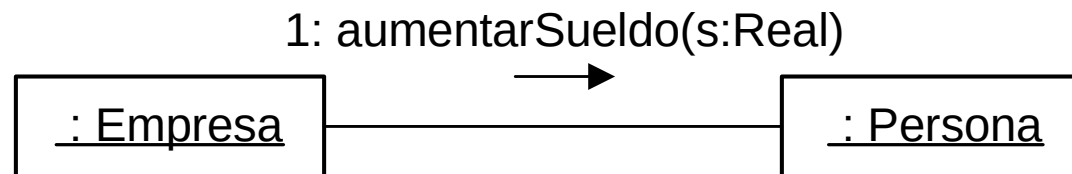


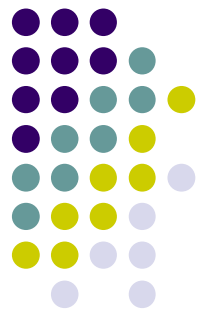


Notación

Parámetros

- Los parámetros se muestran entre paréntesis a la derecha del nombre del mensaje
- Opcionalmente se puede mostrar además su tipo

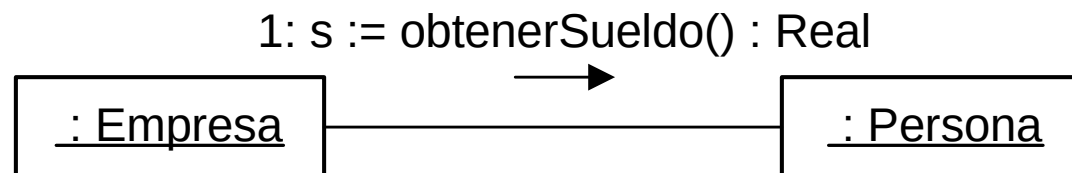


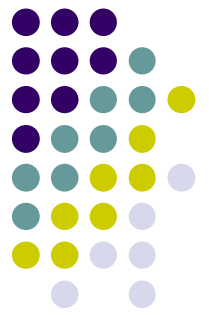


Notación

Tipo de Retorno

- El valor de retorno puede ser mostrado a la izquierda del mensaje, con un ($:=$) en medio
- Opcionalmente se puede mostrar el tipo del valor de retorno





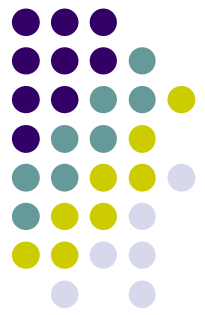
Notación

Sintaxis de Mensajes

- La sintaxis de los mensajes es la siguiente:

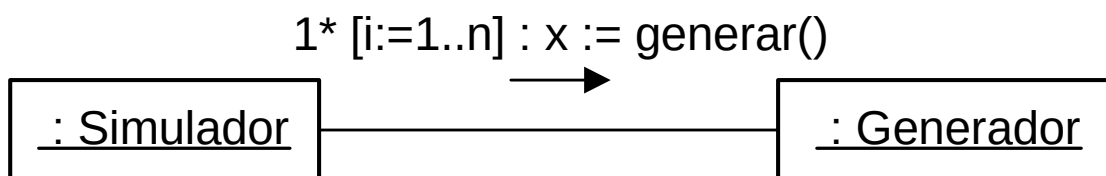
`[ret :=] mensaje([param [: TipoParam]]) [: TipoRet]`

- Donde:
 - `ret` almacena el resultado de la operación (opcional)
 - `mensaje` es el nombre del mensaje enviado (y de la operación invocada)
 - `param` son argumentos usados en el envío
 - `TipoParam` es el tipo de cada parámetro (opcional)
 - `TipoRet` es el tipo del recorrido de la operación (opcional)



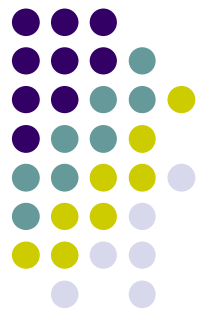
Notación Iteración

- Las iteraciones se indican mediante un asterisco (*) a continuación del numero de secuencia del mensaje
- Esto expresa que el mensaje es enviado en forma repetida (en un loop) al receptor



```
class Simulador {
    Generador gen;

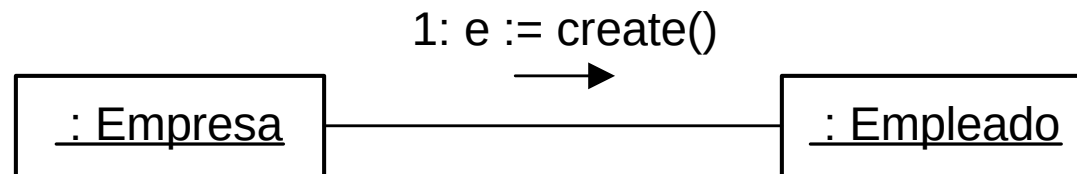
    void unaOper() {
        for (i from 1 to n) {
            x = gen.generar();
        }
    }
}
```

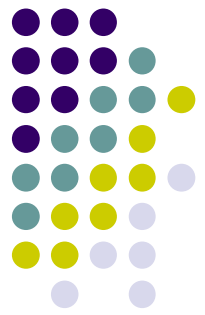


Notación

Creación de Instancias

- La forma de ilustrar la creación de una instancia es enviando el mensaje **create**
- Este mensaje puede incluir parámetros
- Lo usual es especificar un nombre para la instancia para poder utilizarla después

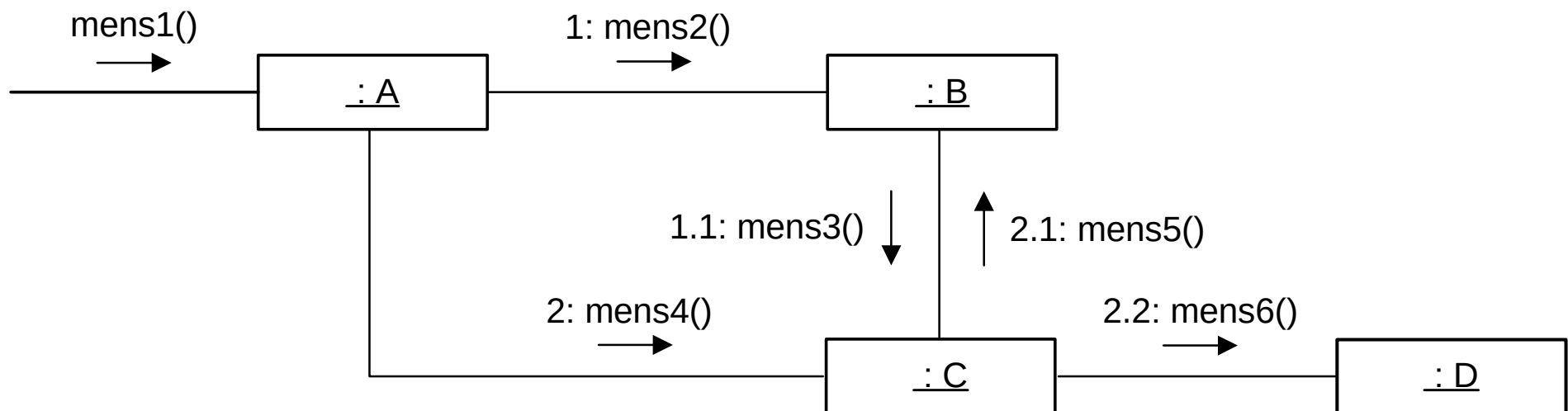


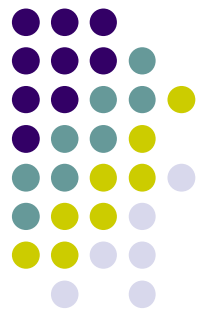


Notación

Números de Secuencia

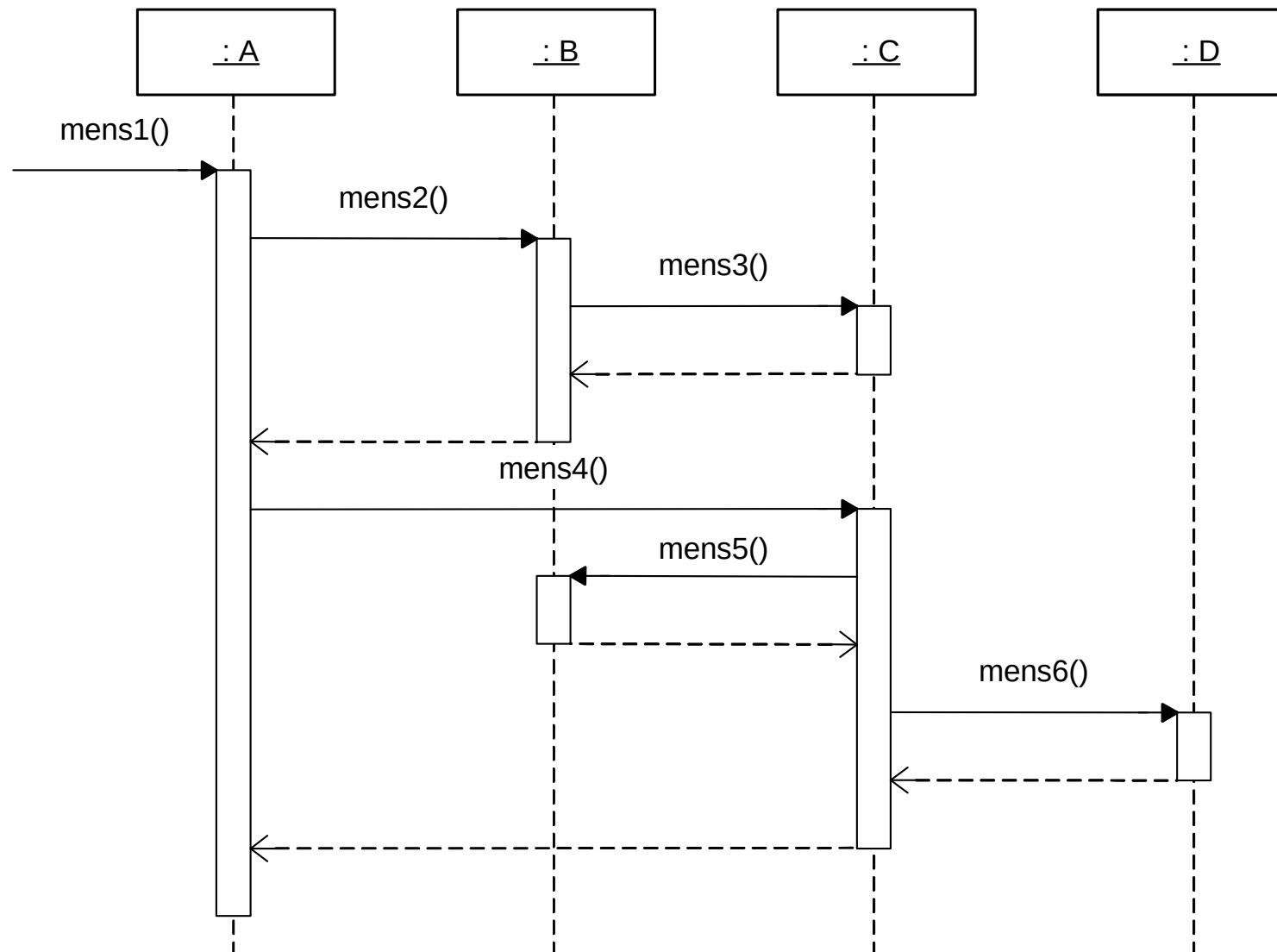
- El orden de ocurrencia de los mensajes viene dado por los números de secuencia
- El mensaje que inicia la interacción generalmente no es numerado

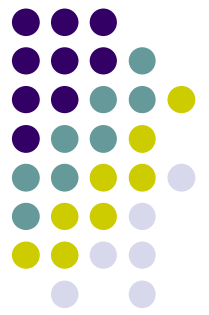




Notación

Números de Secuencia (2)

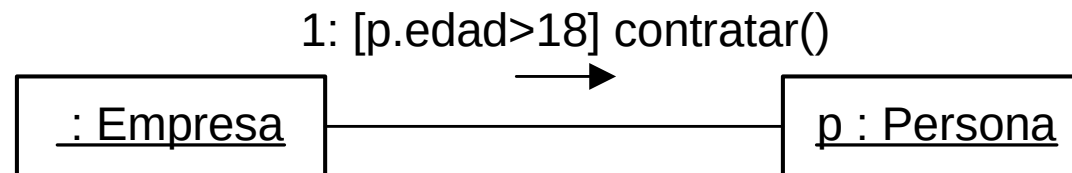




Notación

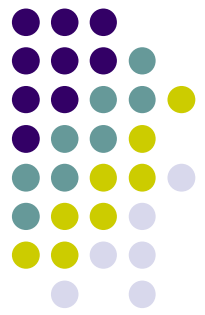
Mensajes Condicionales

- Un mensaje es enviado únicamente si su guarda es satisfecha
- La guarda se muestra entre paréntesis rectos ([]) a la izquierda del mensaje



Notación

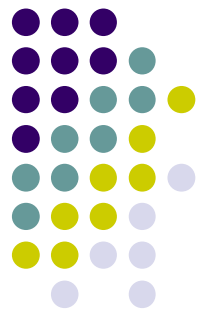
Colecciones



- Los multiobjetos de los diagramas de interacción representan una colección de objetos de una cierta clase



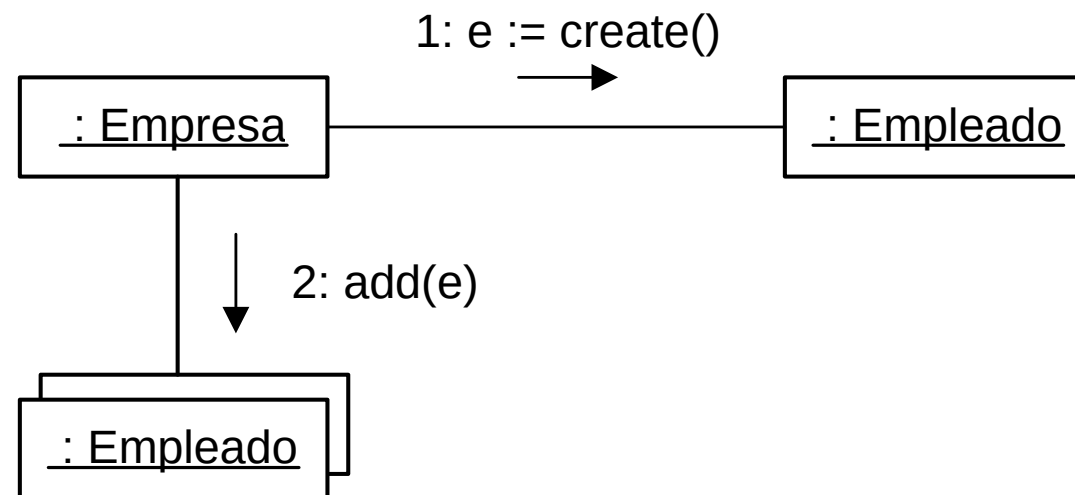
Colección de instancias
de la clase Empleado

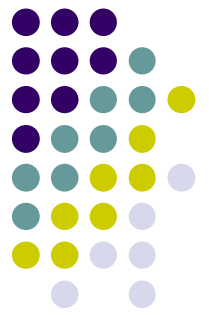


Notación

Mensajes a Colecciones

- Un mensaje a una colección representa un mensaje al objeto colección mismo
- No un broadcast a todos los elementos contenidos en él

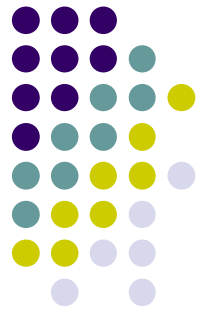




Notación

Responsabilidad de Colecciones

- Las colecciones serán tratadas como meros contenedores de objetos
- Por lo que no podrán tener otra responsabilidad más que esa
- Proveerán solamente operaciones que permitan administrar los objetos contenidos
- En general las interfaces de Diccionario (add, remove, find, member, etc.) e Iterador (next, etc.) son suficientes para las colecciones



Notación

Resp. de Colecciones - Ejemplo

