

Proyecto: Base de datos

BackEnd

Definiciones en Proyecto

```
package basededatos;
import clases.*;

Contenedor personas;
Contenedor instituciones;
Contenedor ciudades;
Contenedor paises;
Contenedor relacionesP_I;
Contenedor relacionesP_C;
Contenedor relacionesI_C;
Contenedor relacionesC_P;
```

Contenedor Relación:	id1	id2	Contenedor1	Contenedor2
relacionesP_I	Persona	Institucion	personas	instituciones
relacionesP_C	Persona	Ciudad	personas	ciudades
relacionesI_C	Institucion	Ciudad	instituciones	ciudades
realcionesC_P	Ciudad	Pais	ciudades	paises

Lectura de Objetos (Agregar a contenedor)

Ejemplos:

```
package basededatos;
import clases.*;
import utiles.Leer;
Leer leer_pais(paises); //Devuelve objeto Pais

paises.agregarObjeto(nuevoObjeto); // Se agrega a contenedor

/**Funciona para personas, instituciones, ciudades, paises.**/
```

Se envían los contenedores con el objetivo de verificar si no existe redundancia de ID al leer los datos del objeto.

Lectura de Objetos (Modificar en contenedor)

Ejemplos:

```
package basededatos;
import clases.*;
import utiles.Leer;

// Obtiene el ID de la persona que se va a modificar, i es la cubeta de posición en contenedor
// Se hace mientras leerObjeto no devuelva null
int ID = ((Persona)personas.leerObjeto(i)).get_id();

//Devuelve objeto Persona, con nuevos datos, ID no cambia
Leer.leer_persona(personas, false, ID);

// Modificar el objeto en contenedor
personas.modificarObjeto(i, nuevoObjeto);
```

```
/**Funciona para personas, instituciones, ciudades, países.**/
```

Se envían los contenedores con el objetivo de verificar si no existe redundancia de ID al leer los datos del objeto.

Eliminar un objeto de contenedor

Ejemplos:

```
// [i,j,k] es la cubeta de posición en contenedor
personas.eliminarObjeto(i);
relacionesI_C.eliminarObjeto(j);
países.eliminarObjeto(k);
```

Lectura de Relaciones

Ejemplos:

```
package basededatos;
import clases.*;
import utiles.LeerRel;

// Devuelve una Relacion entre persona e institución
LeerRel.leer_relacion(personas, instituciones);

// Devuelve una Relacion entre persona y ciudad
LeerRel.leer_relacion(personas, ciudades);

// Devuelve una Relacion entre institución y ciudad
LeerRel.leer_relacion(instituciones, ciudades);

// Devuelve una Relacion entre ciudad y país // Se añade al contenedor
LeerRel.leer_relacion(ciudades, países);
relacionesC_P.agregarObjeto(nuevoObjeto);
```

Se envían los contenedores con el objetivo de verificar si existen los ID's que el usuario ingresa

Otros métodos de la clase Contenedor

```
// Devuelve true solo si todos los objetos en el son null
personas.esVacio();

//Devuelve el tamaño (int) de array interno
personas.getLength();

//Imprime todos los objetos del contenedor que no son null
relacionesC_P.listarObjetos();
```

Métodos de la clase General

De esta clase Heredan las clases:

- Persona
- Institucion
- Ciudad
- País

Propiedades comunes a las clases: id (int) , nombre (String)

Métodos usados con relaciones:

```
General.estaEnRelaciones(Contenedor c, idABuscar int)
```

Este método se usa para comprobar la existencia del objeto (this) en alguna relación con otro(s), se utiliza al momento de eliminar un objeto. Esto con el objetivo de cuidar la integridad de los datos.

El idABuscar corresponde al id propio del objeto ubicado en el tipo de relación que almacena el Contenedor c. (Ver Tabla)

Ejemplos:

```
//El método devuelve true si encuentra en el contenedor al menos un objeto con las propiedades (this).
```

```
// Para personas en contenedor relacionesP_I se usa el id1  
persona.estaEnRelaciones(relacionesP_I, 1)
```

```
// Para ciudades en contenedor relacionesI_C se usa el id2  
ciudad.estaEnRelaciones(relacionesI_C, 2)
```

General.borrarOcurenciasEnRelaciones(Contenedor c, idABuscar int)

Su funcionamiento es igual al anterior, solo que en vez de comprobar la existencia de la relación. Elimina todas y cada una de las ocurrencias de relación donde aparecen las propiedades del objeto (this).

El idABuscar corresponde al id propio del objeto ubicado en el tipo de relación que almacena el Contenedor c. (Ver Tabla)

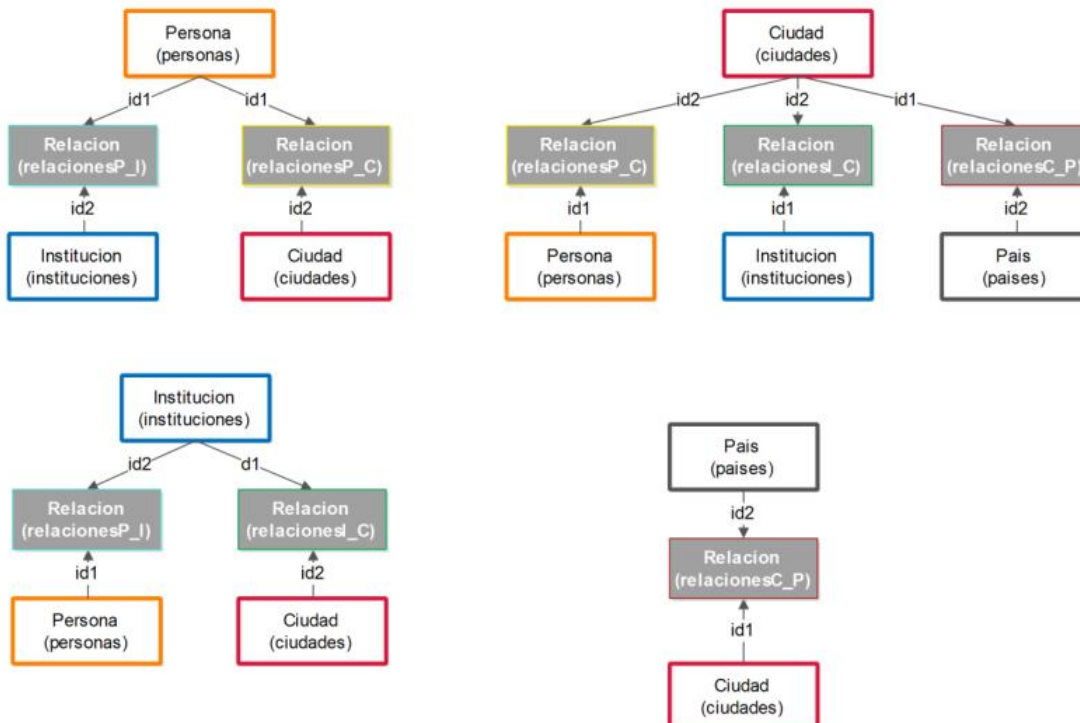
Ejemplos:

```
//El método elimina todas las relaciones donde esta objeto con las propiedades (this).
```

```
// Para personas en contenedor relacionesP_I se usa el id1  
persona.borrarOcurenciasEnRelaciones(relacionesP_I, 1)
```

```
// Para ciudades en contenedor relacionesI_C se usa el id2  
ciudad.borrarOcurenciasEnRelaciones(relacionesI_C, 2)
```

Para resumir, la siguiente imagen muestra las comprobaciones que se deberían hacer al eliminar un objeto con sus relaciones.



Métodos de existencia (Usado en Leer.leer):

General .exi steEn(Contenedor c)

Este método comprueba si existe el objeto con las mismas propiedades en el Contenedor c, devuelve **true** si lo encuentra.

Ejemplos:

```
persona.exi ste(personas)
ciudad.exi ste(ciudades)
```

FrontEnd:

Uso de Menús:

```
package basededatos;
import utiles.Menu;

// Todos estos metodos retornan un entero...
Menu.menu("Pri nci pal ");

Menu.menu("Personas");
Menu.menu("Insti tuciones");
Menu.menu("Ci udades");
Menu.menu("Pai ses");

Menu.menu("Rel aci ones");

Menu.menu("Personas-Insti tuciones");
Menu.menu("Personas-Ci udades");
Menu.menu("Insti tuciones-Ci udades");
Menu.menu("Ci udades-Pai ses");

// Cada uno de estos menús lee un fichero donde se guardan las opciones e imprime cada línea
// Para luego leer un entero con la opción del usuario.
```