

Trabajo Práctico Nro. 10

Herramientas para el Modelado Estructural Básico: Diagramas de Clases y Diagramas de Objetos

Lista de Conceptos Tratados:

Clase; Atributo; Método; Operación; Relaciones entre Clases: Generalización/Especialización, Asociación, Agregación, Composición, Dependencia, Realización, Instanciación; Roles y Multiplicidad en una relación; Instancia de una Clase (Objeto); Instancia de una relación (Link).

Ejercicio 10.1

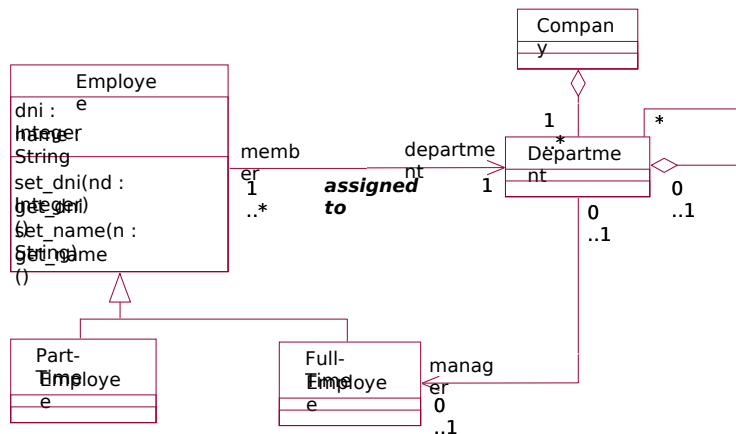
Para cada una de las siguientes afirmaciones discuta e indique si esta es Verdadera o Falsa.

	Verdadera	Falsa
Una clase es un descriptor de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.	<input type="checkbox"/>	<input type="checkbox"/>
Una clase representa un concepto físico del dominio de aplicación de un sistema que se está modelando.	<input type="checkbox"/>	<input type="checkbox"/>
Un diagrama de clases enfoca una perspectiva comportamental al modelar clases, sus atributos, sus operaciones, y relaciones entre clases.	<input type="checkbox"/>	<input type="checkbox"/>

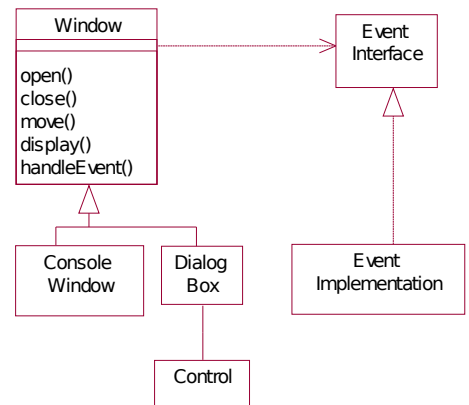
Ejercicio 10.2

- Nombre cada uno de los elementos de notación o sintaxis que están presentes en los siguientes diagramas de clases.
- Describa brevemente qué interpreta en cada uno de dichos diagramas.

a)



b)



Ejercicio 10.3

Considere las siguientes listas de términos. La de la izquierda, lista nombres de tipos de relaciones que se pueden dar entre clases. Mientras que la del centro, lista términos utilizados frecuentemente cuando se leen relaciones entre clases a partir de un diagrama. A la derecha se ven las representaciones en los diagramas.

- Trace la respectiva correspondencia que existe entre los ítems de las listas.

Especialización / Generalización	"USA"	
Dependencia	"ES PARTE DE"	
Realización	"ES UNA CATEGORÍA DE" ... "ES UN"	

Agregación

"IMPLEMENTA"



Asociación

"CONOCE A"



Ejercicio 10.4

Considere la siguiente lista de declaraciones.

- a) Una figura puede estar compuesta por otras figuras básicas, como por ejemplo: triángulos, elipses y cuadriláteros. Se sabe además que un tipo particular de elipse es el círculo, mientras que un tipo particular de cuadrilátero es el rectángulo. A su vez un tipo particular de rectángulo es el cuadrado. Por otra parte, los cuadriláteros y triángulos son tipos de polígonos. Los polígonos tienen un número de lados, y dichos lados están definidos por 2 puntos. Un punto está definido por una coordenada en el eje de las x, y otra en el eje de las y.
- b) Un país tiene provincias, una capital, limita con otros países y se localiza en un continente. Las provincias de un país limitan con otras provincias del mismo país y a su vez pueden limitar con otros países. Las provincias tienen ciudades y una de ellas es su capital.
- c) Java, C++, Smalltalk y Eiffel son lenguajes de programación orientados a objetos. A su vez, Pascal y C son lenguajes de programación procedurales y Prolog es un lenguaje de programación en lógica. El programa "JFS" es una clase de simulador de vuelo y está escrito en el lenguaje de programación Java.
- d) Una aerolínea ofrece vuelos transportando pasajeros en aviones entre diferentes ciudades. Cada vuelo parte de un aeropuerto de inicio y arriba a un aeropuerto de destino final. Algunos vuelos pueden tener escalas intermedias. Los aeropuertos están localizados en ciudades, pudiendo tener una ciudad más de un aeropuerto. A cada vuelo se le asigna un avión y una tripulación, la cual consta de 2 pilotos, 1 operador de comunicaciones, 2 comisarios de abordaje y 4 azafatas. Cada pasajero de un vuelo tiene asignado un asiento.
- e) Se tiene un procesador el cual recibe tareas, las mismas poseen una prioridad, un uso de memoria, y un uso de CPU. El procesador a medida que las tareas le son suministradas las almacena en una lista de espera ordenada de acuerdo a la prioridad. Existen otros procesadores los cuales ordenan las tareas de acuerdo al uso de memoria o el uso de CPU.
- f) Para un sistema bancario, modelar las clases caja de ahorro, cliente y banco para que, al menos, se puedan crear cajas de ahorro, depositar y extraer sobre una cuenta particular, listar el saldo de una cuenta, listar las cuentas de un cliente y cerrar cuentas.
- g) Modelar un conjunto de clases y sus operaciones que resuelvan el problema siguiente. Un país tiene que controlar el gasto público de las ciudades con más de 100.000 habitantes. Para ello, tiene información del monto recaudado por cada ciudad a través de cinco diferentes tipos de impuestos (denominados, aquí, de imp1, imp2, imp3, imp4 e imp5) e información acerca de gastos realizados en mantenimiento de la ciudad. Este país necesita un sistema que le informe cuales son las ciudades que gastan más de lo que recaudan, y las provincias que tienen más de la mitad de las ciudades en condición de déficit.

- **Para cada caso, identifique clases y relaciones entre clases, clasificando estas relaciones en generalización/especialización, asociación, agregación o composición.**
- **Construya los diagramas de clases correspondientes. Incluya atributos y operaciones considerando un contexto general.**
- **Construya diagramas de objetos para ejemplificar configuraciones posibles en cada caso.**

Ejercicio 10.5 (Caso de Estudio: Restaurantes)

En una ciudad existen diferentes restaurantes, algunos de ellos con varias sucursales. Los restaurantes ofrecen diferentes platos (de comida), pero no más de 20, a las personas que concurren a ellos. En general, a las personas les gustan diferentes platos, frecuentan varios restaurantes y ocasionalmente pueden concurrir a otros aunque no sean de los que frecuenta habitualmente. En particular, a las personas no les gusta un plato por sí mismo, sino cómo lo sirven en determinados restaurantes, aunque puede que para un plato particular no le guste cómo lo sirven en ninguno de los restaurantes. A su vez, un plato servido en un restaurante puede no gustarle a ninguna persona.

- **Identifique clases, atributos y relaciones entre clases, en el dominio del problema. Construya el diagrama de clases correspondiente.**

Ejercicio 10.6

Dadas las clases A, B y C, se definen las asociaciones R1 entre objetos de las clases A y B, y R2 entre objetos de las clases B y C. Sean los objetos a, b y c instancias de clases A, B y C, respectivamente.

➤ **Dibuje el diagrama de clases**

➤ **Determine si es posible, en un diagrama conteniendo estos objetos, que exista:**

- a.) un link entre a y b?
- b.) un link entre b y c?
- c.) un link entre a y c?
- d.) dos links entre b y c

Ejercicio 10.7

➤ **Identifique las clases, sus atributos y operaciones y las relaciones entre ellas para el siguiente problema**

Una cooperativa de agricultores requiere de un sistema que le aconseje cual es el cereal que puede sembrar en un determinado lote. Los lotes son clasificados como especiales cuando contienen ciertos minerales de interés primario para la cooperativa y comunes cuando contienen sólo minerales secundarios en la composición de la tierra. Para poder sembrar un cereal en un lote, éste debe contener determinados minerales. Los cereales se clasifican en granos de cosecha gruesa (como girasol, maíz,...), granos de cosecha fina (como trigo, avena,..) y pasturas (como alfalfa, trébol subterráneo,..). Un lote satisface los requerimientos de un cereal si contiene todos de los minerales que requiere ese cereal. Además, para el caso de las pasturas no se puede haber sembrado previamente otra pastura.

Ejercicio 10.8 (Caso de Estudio)**Dado el siguiente fragmento:**

En un agujero en el suelo, vivía un hobbit. No un agujero húmedo, sucio, repugnante, con restos de gusanos y olor a fango, ni tampoco un agujero seco, desnudo y arenoso, sin nada en que sentarse o que comer: era un agujero-hobbit, y eso significa comodidad.

Tenía una puerta redonda, perfecta como un ojo de buey, pintada de verde, con una manilla de bronce dorada y brillante, justo en el medio. La puerta se abría a un vestíbulo cilíndrico, como un túnel: un túnel muy cómodo, sin humos, con paredes revestidas de madera y suelos enlosados y alfombrados, provisto de sillas barnizadas, y montones y montones de perchas para sombreros y abrigos; el hobbit era aficionado a las visitas. El túnel se extendía serpeando, y penetraba bastante, pero no directamente, en la ladera de la colina —La Colina, como la llamaba toda la gente de muchas millas alrededor—, y muchas puertecitas redondas se abrían en él, primero a un lado y luego al otro. Nada de subir escaleras para el hobbit: dormitorios, cuartos de baño, bodegas, despensas (muchas), armarios (habitaciones enteras dedicadas a ropa), cocinas. Comedores, se encontraban en la misma planta, y en verdad en el mismo pasillo. Las mejores habitaciones estaban todas a la izquierda de la puerta principal, pues eran las únicas que tenían ventanas, ventanas redondas, profundamente excavadas, que miraban al jardín y los prados de más allá, camino del río.

"El Hobbit"; J.R.R. Tolkien

➤ **Realizar un Diagrama de Clases que modele un " agujero-hobbit ".**

Ejercicio 10.9 (Caso de Estudio: Telecompras)

Considere como caso de estudio al sistema de procesamiento de órdenes de compra de la empresa *TeleShopping*, para el cual Ud. desarrollara el modelo de casos de uso y diferentes diagramas de actividades.

- **Identifique las clases, atributos y operaciones. Construya el diagrama de clases.**
- **Construya diagramas de objetos para ejemplificar configuraciones posibles antes y después de la ejecución de cada uno de los casos de uso “Ingresar orden de compra” y “Cancelar orden de compra”.**
- **Revise los diagramas de actividades que tenga contruidos, para completar flujos de objetos faltantes a partir del item anterior.**

Ejercicio 10.10 (Caso de Estudio: Biblioteca)

Considere el caso de estudio del sistema para la biblioteca *Libraccio*, para el cual Ud. desarrollara el modelo de casos de uso y diferentes diagramas de actividades.

- **Identifique clases, atributos y operaciones. Construya el diagrama de clases**
- **Construya diagramas de objetos para ejemplificar configuraciones posibles antes y después de la ejecución de los casos de uso del préstamo de un ejemplar de libro a un socio, y la devolución de ejemplar.**
- **Revise los diagramas de actividades que tenga contruidos, para completar flujos de objetos faltantes a partir del item anterior.**

Ejercicio 10.11 (Caso de Estudio: Cajero Automático)

Considere la narrativa describiendo la operatoria de un cajero automático, para el cual Ud. desarrollara el modelo de casos de uso y diferentes diagramas de actividades.

- **Identifique clases, atributos, operaciones. Construya el diagrama de clases.**
- **Construya diagramas de objetos para ejemplificar configuraciones posibles antes y después de la ejecución de los casos de uso siguientes: extracción de un monto de dinero de una cuenta; depósito de un monto de dinero en una cuenta; transferencia de un monto entre dos cuentas de un mismo cliente y un mismo banco.**
- **Revise los diagramas de actividades que tenga contruidos, para completar flujos de objetos faltantes a partir del item anterior.**

Ejercicio 10.12 (Ingeniería Reversa - C++)

Dado el siguiente código fuente escrito en C++:

- **Identifique las clases involucradas y construya el diagrama de clases con sus relaciones, atributos y operaciones.**
- **Si fuera posible, identifique paquetes e interfaces.**

```
#include <iostream.h>
class Object { public:
    static int nombre_objetos = 0;
    int numero;
    Object();
    ~Object();
};

class Puerta : public Object { public:
    Puerta ();
    ~ Puerta ();
};

class Edificio : public Object { public:
```

```
Puerta * puerta;
int superficie;
Edificio(int);
~Edificio();
};

class Torre : public Object { public:
    int altura;
    Muralla * muralla;
    Torre(int, Muralla *);
    ~Torre ();
};

class TorrePrincipal : public Torre { public:
```

```

Castillo * castillo;
TorrePrincipal (Castillo *, int);
~ TorrePrincipal ();
};

class Muralla : public Object { public:
int longitud;
Torre * torre1;
Torre * torre2;
Castillo * castillo;
Muralla(int, int, int, Castillo *);
~ Muralla ();
};

class Castillo : public Edificio { public:
TorrePrincipal * torrePrincipal;
Muralla * muralla;
Castillo (int, int, int, int, int);
~Castillo ();
};

int Object::nombre_objetos = 0 ;

Object::Object() {
cout << "++ Object inicio" << endl;
nombre_objetos++;
numero = nombre_objetos ;
cout << "++ Object fin" <<
nombre_objetos << endl;
}
Object::~~Object() {
cout << "-- Object inicio " << endl;
nombre_objetos--;
cout << "-- Object fin " << endl;
}

Puerta:: Puerta () {
cout << "++ Puerta inicio" << endl;
cout << "++ Puerta fin" << endl;
}

Puerta::~~ Puerta () {
cout << "-- Puerta inicio" << endl;
cout << "-- Puerta fin" << endl;
}

Edificio:: Edificio (int s) {
cout << "++ Edificio inicio " << endl;
puerta = new Puerta();
superficie = s;
cout << "++ Edificio fin" <<
superficie << endl;
}
Edificio::~~ Edificio () {
cout << "-- Edificio inicio " << endl;
delete puerta;
cout << "-- Edificio fin" << endl;
}

Torre:: Torre (int h, Muralla * m) {

```

```

cout << "++ Torre inicio " << endl;
altura = h;
muralla = m;
cout << "++ Torre fin " << hauteur
<< endl;
}

Torre::~~ Torre () {
cout << "-- Torre inicio " << endl;
cout << "-- Torre fin" << endl;
}

TorrePrincipal:: TorrePrincipal (Castillo * c, int h)
: Torre(h, NULL) {
cout << "++ TorrePrincipal inicio " <<
endl;
castillo = c;
cout << "++ TorrePrincipal fin" <<
endl;
}
TorrePrincipal::~~ TorrePrincipal () {
cout << "-- TorrePrincipal inicio " <<
endl;
cout << "-- TorrePrincipal fin" << endl;
}

Muralla::Muralla(int l, int h1, int h2, Castillo * c)
{
cout << "++ Muralla inicio " << endl;
longitud = l;
torre1 = new Torre(h1, this);
torre2 = new Torre(h2, this);
castillo = c;
cout << "++ Muralla fin " << longitud
<< endl;
}
Muralla::~~Muralla() {
cout << "-- Muralla inicio " << endl;
delete torre1;
delete torre2;
cout << "--Muralla fin" << endl;
}

Castillo:: Castillo (int l, int h1, int h2, int hd, int
s) : Edificio
cout << "++ Castillo inicio " << endl;
muralla= new Muralla(l, h1, h2, this);
torrePrincipal(this, hd);
cout << "++ Castillo fin" << endl;
}
Castillo::~~Castillo () {
cout << "-- Castillo inicio" << endl;
delete muralla;
delete torrePrincipal;
cout << "-- Chateau fin" << endl;
}

main() {
Castillo castillo (400, 40, 60, 80,
10000);
}

```

Ejercicio 10.13 (Modelo de Modelos: Metamodelo)

Reflexione acerca de las definiciones de los conceptos de: Clase, Atributo, Operación, Método, Relación entre clases, y Roles en una relación.

- **Construya un diagrama de clases que muestre las relaciones entre estos conceptos. Incluya los atributos y operaciones que considere necesarios.**

Ejercicio 10.14 (Caso de Estudio: Comparación con herramientas utilizadas en el paradigma estructurado)

- **¿Cuál o cuáles de herramientas del paradigma estructurado serían comparables a los diagramas de clases, respecto de lo que modelan?**
- **Destaque similitudes y diferencias.**

Bibliografía de Apoyo Sugerida

[1] – The Unified Modeling Language – Reference Manual. J. Rumbaugh, I. Jacobson and G. Booch. Addison Wesley Longman, Inc. 1999. ISBN 0-201-30998-X. Código de consulta en Biblioteca Central UNICEN: 001.642 R936-1.

- Resumen de la notación de UML: Appendix B, páginas 519 a 530.
- Clases, Relaciones y Diagramas de Clases: Part 2 - Chapter 4 (Static View), páginas 41 a 59.
- Diagramas de Objetos: Part 2 - Chapter 4 (Static View), páginas 59 a 61.

[2] – The Unified Modeling Language – User Guide. G. Booch, J. Rumbaugh and I. Jacobson. Addison Wesley Longman, Inc. 1999. ISBN 0-201-57168-4. Código de consulta en Biblioteca Central UNICEN: 001.642 B724-3.

- Conceptos generales sobre diagramas de UML: Section 2 – Chapter 7, páginas 91 a 104.
- Resumen de la notación de UML: Appendix A, páginas 435 a 437.
- Clases: Chapter 4 y 9.
- Relaciones: Chapter 5 y 10.
- Diagramas de Clases: Chapter 8.
- Diagramas de Objetos: Chapter 14.
- Paquetes e Interfaces: Chapter 11 y 12.
- Instancias: Chapter 13.