

Programación Orientada a Objetos 2014-1 Relaciones entre Objetos

Profesora: M. Angélica Caro

(26-05-2014)

Enunciado

Considere el diagrama de clases UML de la Figura 1:

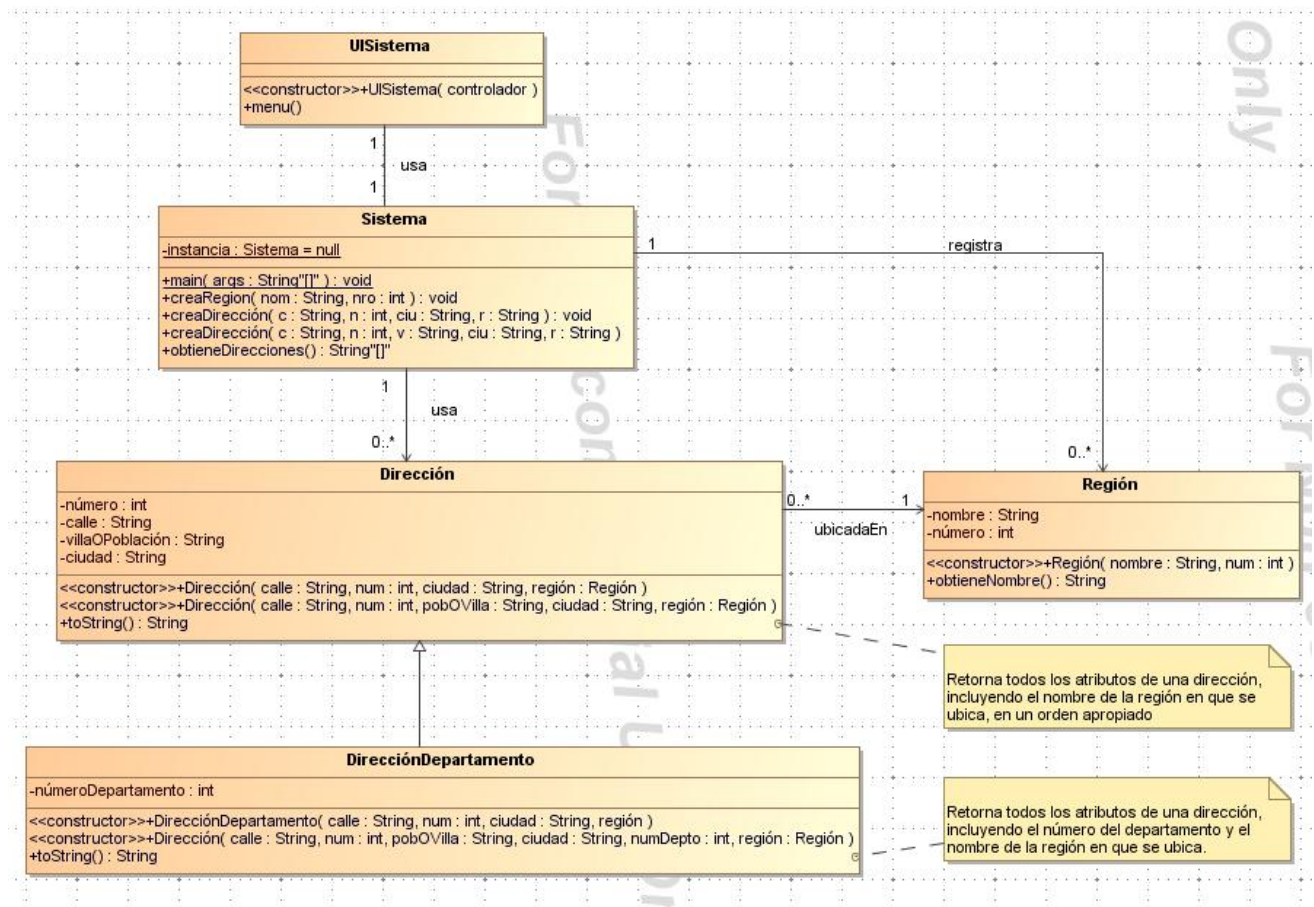


Figura 1. Diagrama de clases Sistema de Direcciones

Ejercicio 1:

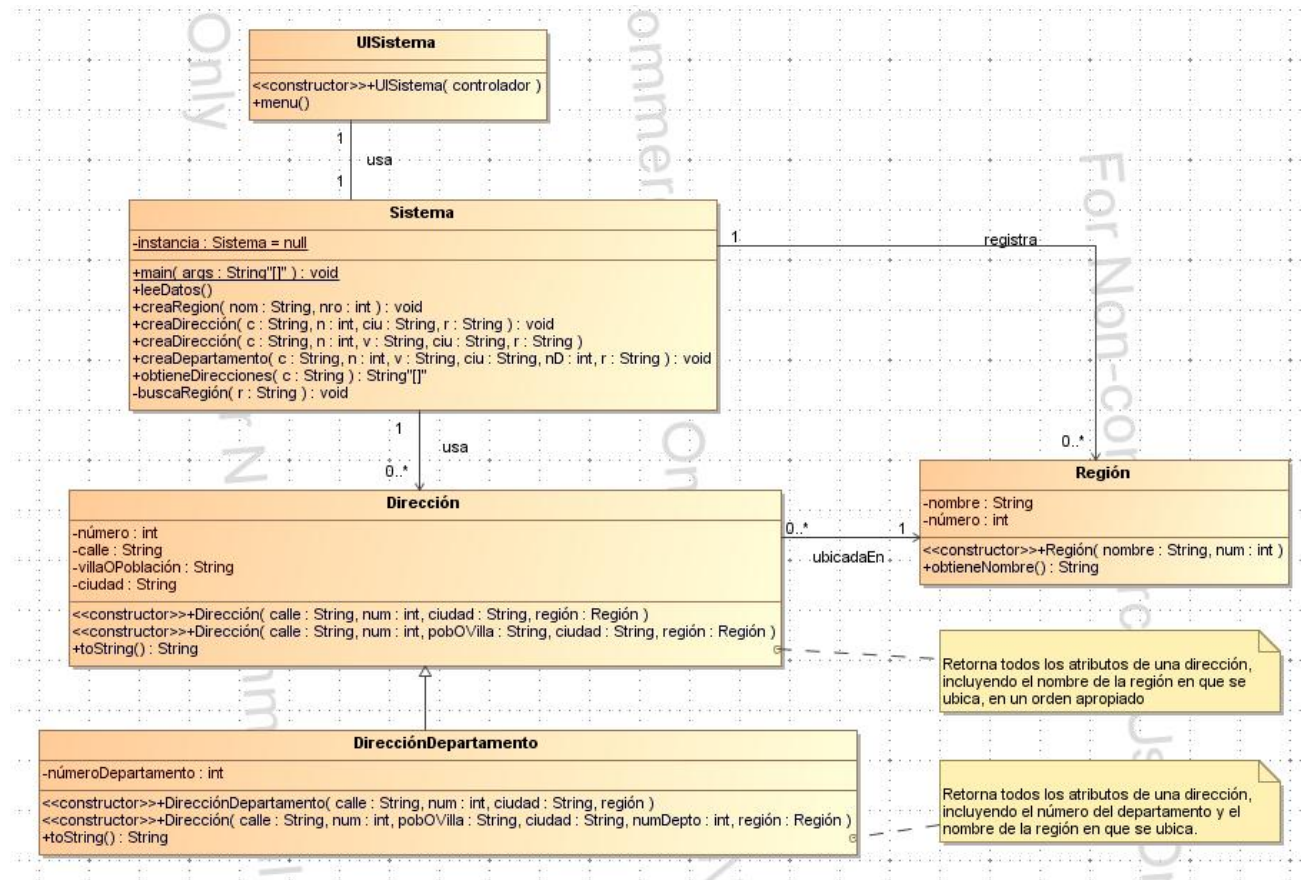
Habiendo implementado las clases entidades presentadas de la Figura 1, a continuación se pide implementar las clases Sistema e IUSistema, considerando lo siguiente:

- La clase Sistema, que representa el **control** de la aplicación. En su implementación debe considerar las relaciones que posee y los métodos definidos para ella:
 - El método main() es el que habilita la ejecución de la aplicación, en este caso a través de la instancia de la clase UISistema.
 - El método creaRegión(), crea una región a partir de los datos recibidos por parámetro. El método no debe crear regiones repetidas.
 - El método creaDirección(), en sus dos versiones, crea una dirección a partir de los datos recibidos por parámetro.



- El método `obtieneDirecciones()`, retorna un arreglo con todas las direcciones creadas en el sistema.
- b) La clase `IUSistema`, que representa la **vista** de la aplicación. En su implementación debe considerar las relaciones que posee y los métodos definidos para ella:
 - El método `menú()` es el que habilita la ejecución de la aplicación, en este caso a través de la instancia de la clase `UISistema`. El menú presentará al usuario las siguientes opciones:
 1. Crea Región
 2. Crea Dirección
 3. Lista Direcciones
 4. Fin
 - En las opciones 1 y 2, se leerán desde teclado los datos para crear una Región y una Dirección respectivamente.
 - En la opción 3 se mostrarán por pantalla los datos de todas las direcciones creadas en el sistema.

Ejercicio 2:



Considere una nueva versión para la aplicación del Sistema de Direcciones, de acuerdo con la figura 2.



Habiendo implementado las clases entidades presentadas de la Figura 1, a continuación se pide implementar las clases Sistema e IUSistema, considerando lo siguiente:

- c) La clase Sistema, que representa el **control** de la aplicación. En su implementación debe considerar las relaciones que posee y los métodos definidos para ella:
- El método main() es el que habilita la ejecución de la aplicación, en este caso a través de la instancia de la clase UISistema.
 - El método creaRegión(), crea una región a partir de los datos recibidos por parámetro. El método no debe crear regiones repetidas.
 - El método creaDirección(), en sus dos versiones, crea una dirección a partir de los datos recibidos por parámetro.
 - El método obtieneDirecciones(), retorna un arreglo con todas las direcciones creadas en el sistema en la ciudad especificada por parámetro.
 - El método leeDatos, lee desde los archivos Región y Direcciones, los datos de las regiones y direcciones a crear en el sistema. El formato de los archivos es el siguiente:

Regiones :
nombre, numero

Direcciones:
tipo,calle,numero,numerodepto,población,ciudad, región

donde:
tipo=2 es departamento
tipo = 1 no departamento

cuando la dirección no tiene población o villa viene un "-"

Un posible ejemplo del archivo:

```
Arica,15  
Iquique,1  
Maule,7  
BioBio,8
```

Un posible ejemplo del archivo:

```
2,Av. Ecuador ,980,105,Centenario,Chillan,8  
2,Sargento Aldea,567,204,-,Chillan,8  
1,Los Copihues,123,-,Chillan,8  
1,Las Amapolas,123,J.Ñuble,Chillan,8  
2,Freire,567,804,-,Concepcion,8
```

- d) La clase IUSistema, que representa la **vista** de la aplicación. En su implementación debe considerar las relaciones que posee y los métodos definidos para ella:
- El método menú() es el que habilita la ejecución de la aplicación, en este caso a través de la instancia de la clase UISistema. El menú presentará al usuario las siguientes opciones:
 1. Lee Direcciones
 2. Lista Direcciones de una ciudad
 3. Fin
 - En la opción 1 se leerán los datos de regiones y direcciones desde los archivos correspondientes.
 - En la opción 2 se mostrarán por pantalla los datos de todas las direcciones de una ciudad específica leída desde teclado.