

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

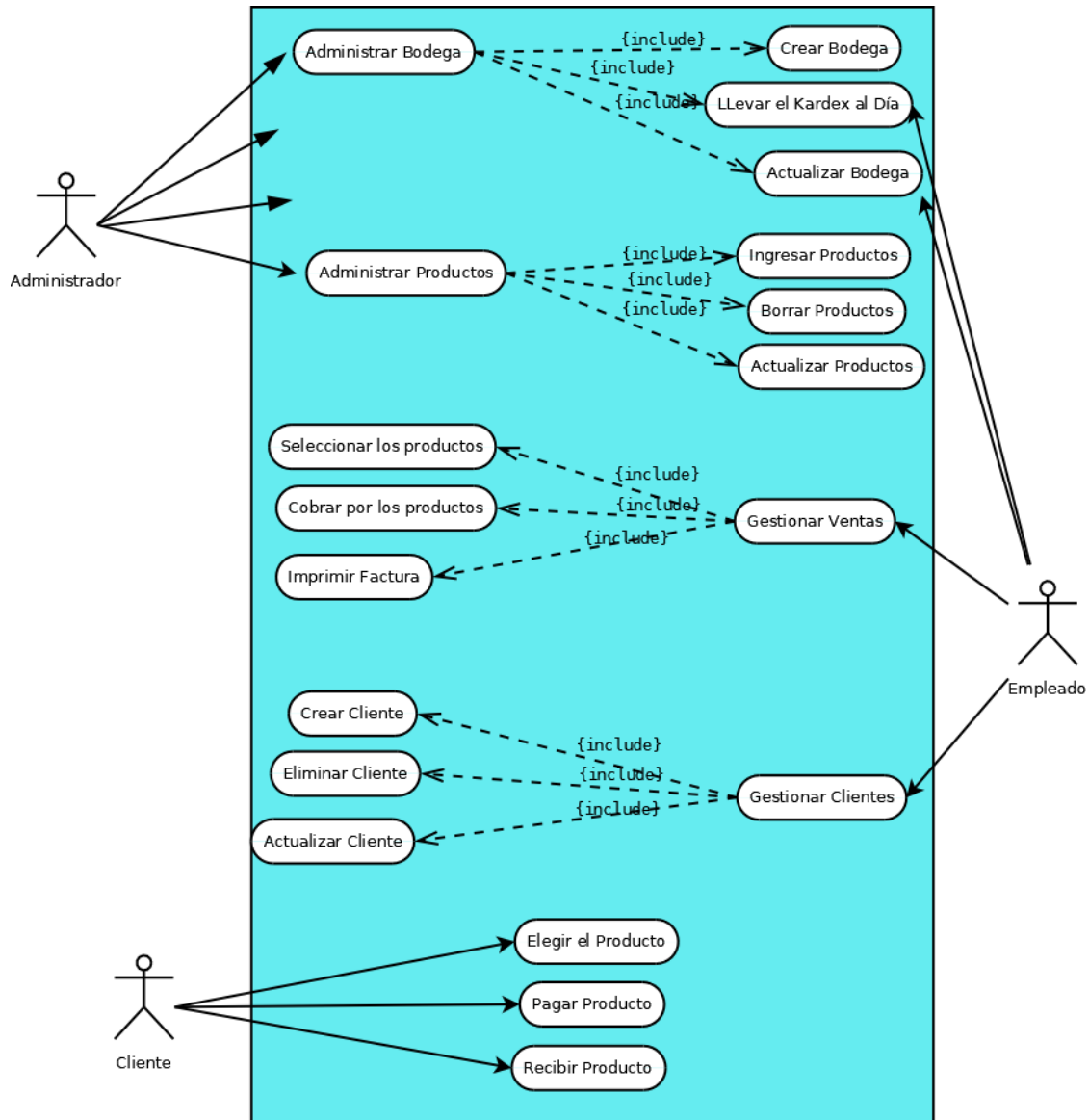
Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

Para el primer avance se debe presentar el análisis del proyecto y la estructura inicial del mismo:

- Diagrama de casos de uso
- Diagramas de clases
- Base de datos
- Proyecto EAR con WAR y JAR incluido
- Mapeo relacional de objetos hacia las entidades con JPA
- Patrón de diseño DAO

El análisis que se utilizó para elaborar este sistema se basa en los diagramas de uso General y en el diagrama de Clases, para luego llevar todo a la programación en java utilizando la IDE de NetBeans.

DIAGRAMA DE USO GENERAL PARA EL PROYECTO HELADERIA



INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

Según (Microsoft, 2021) Los diagramas de casos de uso muestran el comportamiento esperado del sistema. No muestran el orden en que se realizan los pasos. La definición de la límite de sistema determina qué se considera externa o interna para el sistema. Un actor representa una función que desempeña un objeto externo. Un objeto puede desempeñar varias funciones y, por tanto, está representado por varios actores. Una asociación muestra la participación del actor en el caso de uso.

Un caso de uso es un conjunto de eventos que se produce cuando un actor usa un sistema para completar un proceso. Normalmente, un caso de uso es un proceso relativamente grande, no un paso o una transacción individual.

Referee Certification System Diagram

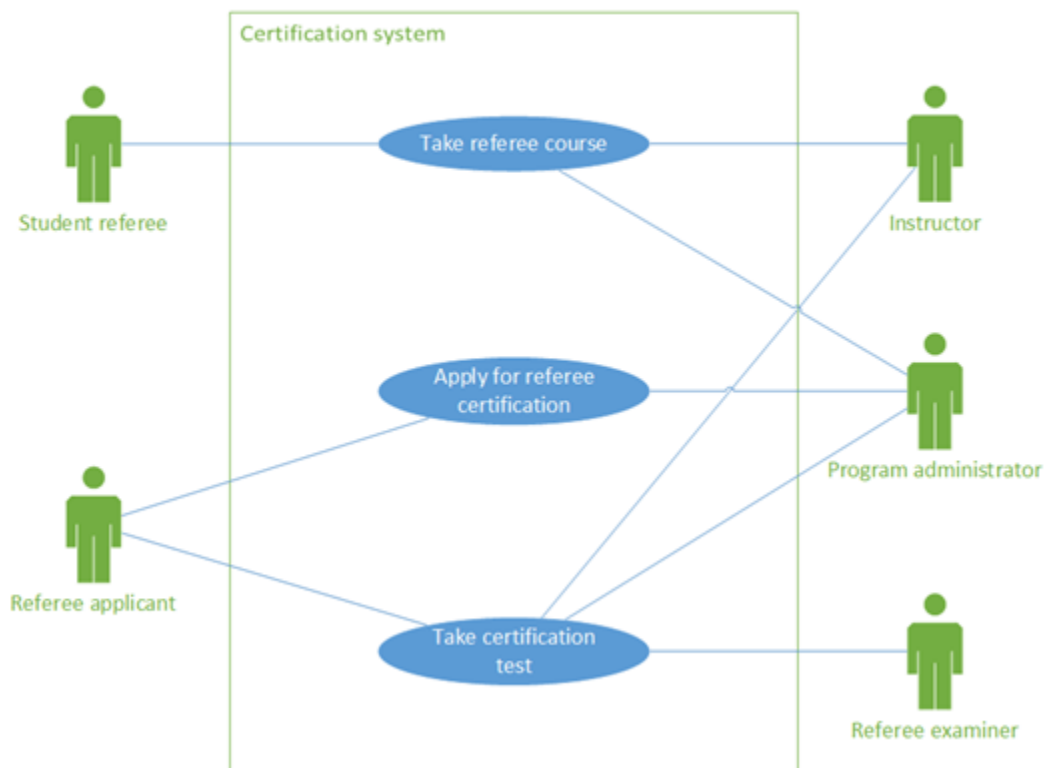


Figura 1: Según (Microsoft, 2021) este es un ejemplo de diagrama de casos de uso

Se pueden usar formas de conector para indicar relaciones entre las formas del diagrama. Hay cinco conectores disponibles:

INSTITUTO TECNOLÓGICO QUITO

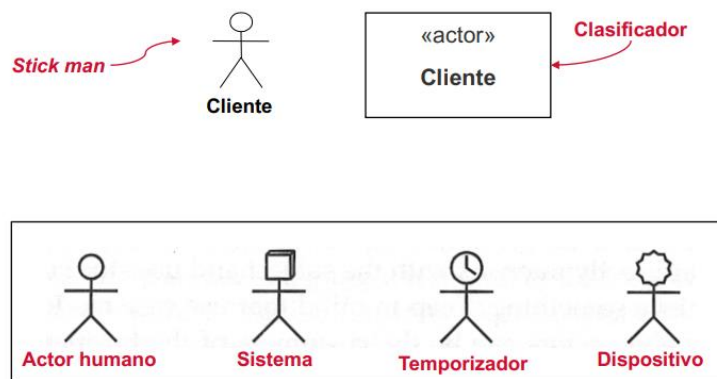
INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

Connector	Descripción
Asociadas 	Muestra la relación de un actor con un caso de uso.
Dependencia 	Indica que un caso de uso tiene una dependencia en otra.
Generalización 	Indica que un caso de uso es una forma específica de lograr los objetivos del caso de uso general.
Incluir 	Muestra cómo se divide un caso de uso en pasos más pequeños.
Extendido 	Muestra que un caso de uso agrega funcionalidad a otro.

Figura 2: Según (Microsoft, 2021) estas son las formas de conectores que existen para un diagrama

Según (Garcia & Garcia, 2018) estos son los tipos de actores que se pueden presentar:



Símbolos utilizados para representar tipos de actores

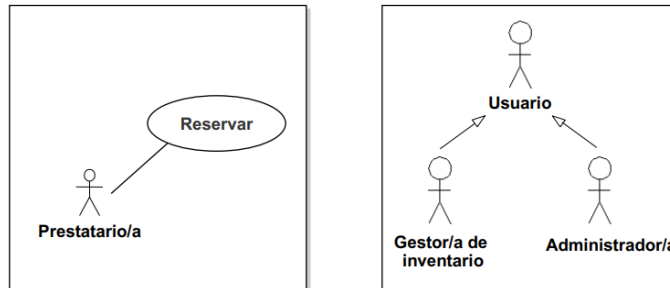
Figura 3: Según (Garcia & Garcia, 2018) estos son los tipos de actores

Estas son las relaciones entre actores:

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba



Asociación entre un actor y un caso de uso

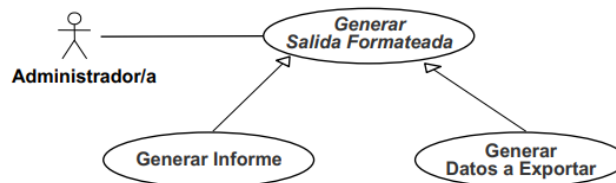
Relaciones de generalización entre actores

Figura 4: Según (Garcia & Garcia, 2018) estos son los tipos de relaciones entre actores

Relaciones de los casos de uso:

Generalización:

- Una relación de generalización relaciona un caso de uso especializado con otro caso de uso más general
- El hijo hereda las relaciones y comportamiento del padre y puede agregar atributos y operaciones propios
- El caso de uso hijo añade comportamiento al caso de uso padre insertando secuencias de acción adicionales en la secuencia del padre en puntos arbitrarios
- También puede modificar algunas operaciones y secuencias heredadas, pero debe hacerse de forma que la intención del padre se mantenga
- El caso de uso padre puede ser abstracto



Relaciones de generalización entre casos de uso

Figura 5: Según (Garcia & Garcia, 2018) esta es la relación de Generalización

Extensión:

- Dependencia entre dos casos de uso que especifica que el comportamiento de un caso de uso base (extendido) puede ser extendido con comportamiento adicional definido en otro caso de uso (extensor)
- El caso de uso extendido define un comportamiento que tiene significado con independencia del caso de uso extensor
- El comportamiento del caso de uso extensor incrementa el del caso de uso base sólo en determinadas condiciones
- Un caso de uso extensor puede extender varios casos de uso base y puede, a su vez, ser extendido por otro caso de uso
- La extensión tiene lugar en puntos de extensión pertenecen al caso de uso extendido
- Indican el lugar donde se insertan los fragmentos de comportamiento del caso de uso extensor

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

< nombre > [: <explicación>]

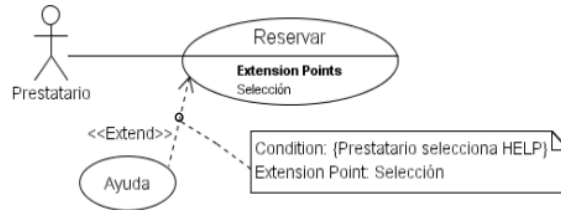


Figura 6: Según (Garcia & Garcia, 2018) esta es la relación de Extensión

Inclusión:

- Relación entre dos casos de uso que indica que el comportamiento de un caso de uso (incluido) se inserta en el comportamiento de otro caso de uso (base o inclusor) en la localización especificada en este último
- La inclusión no es condicional
- El propósito de la inclusión es la reutilización de porciones de comportamiento comunes a varios casos de uso
- Un caso de uso incluido puede insertarse en varios casos de uso base y puede, a su vez, incluir otros casos de uso
- Un caso de uso base puede tener relaciones de inclusión con varios casos de uso incluidos
- La ejecución es análoga a las llamadas a procedimientos
- Notación de la relación de inclusión: símbolo de dependencia con el estereotipo «include»

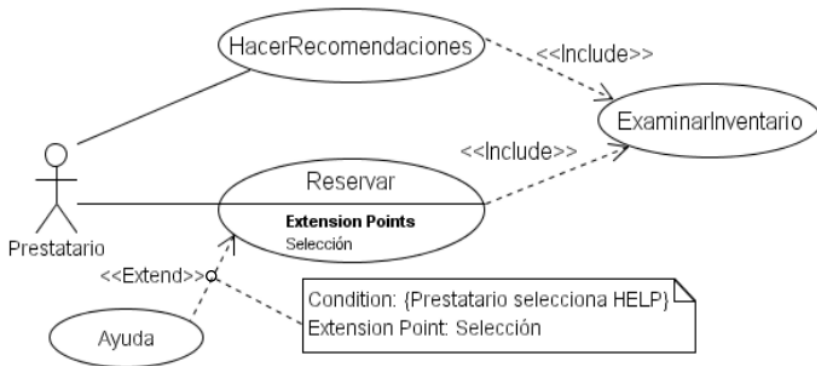


Figura 7: Según (Garcia & Garcia, 2018) esta es la relación de Inclusión

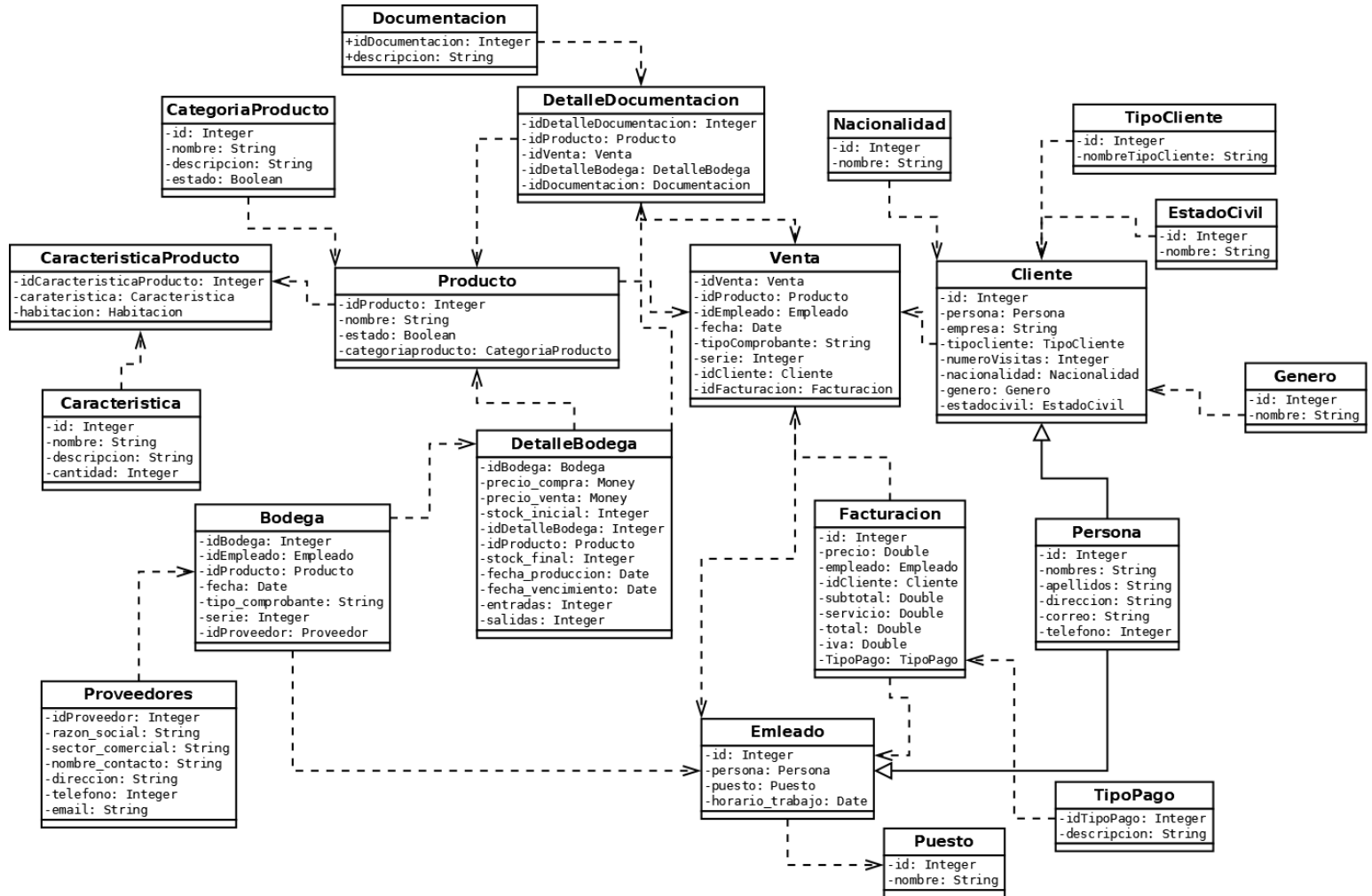
El diagrama de uso general que se realizó para este ejercicio se presenta a continuación:

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

DIAGRAMA DE CLASES PARA EL PROYECTO HELADERIA

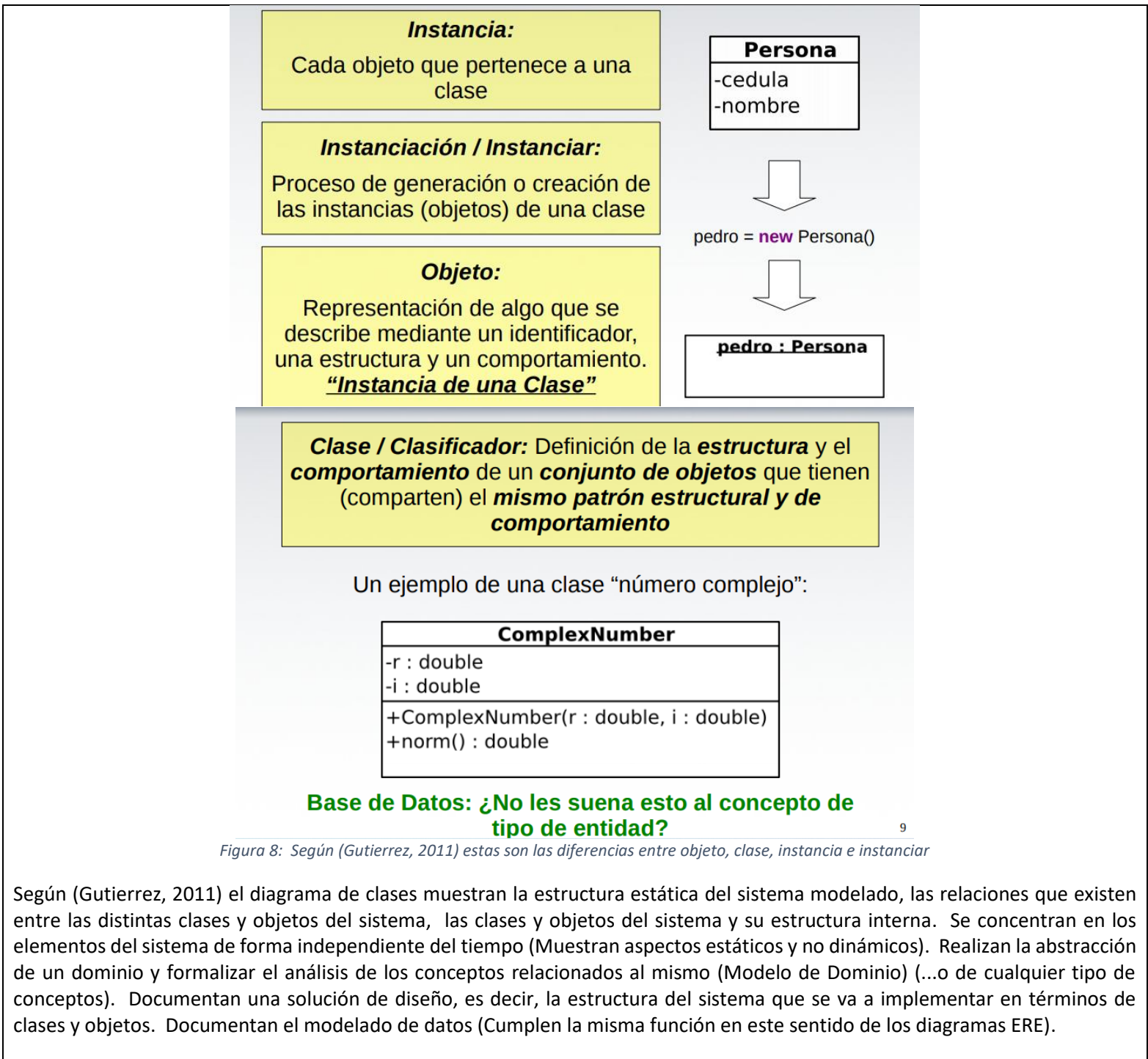


Según (Gutierrez, 2011) estos son las diferencias entre instancia, objeto, instanciar y clase

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba



INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

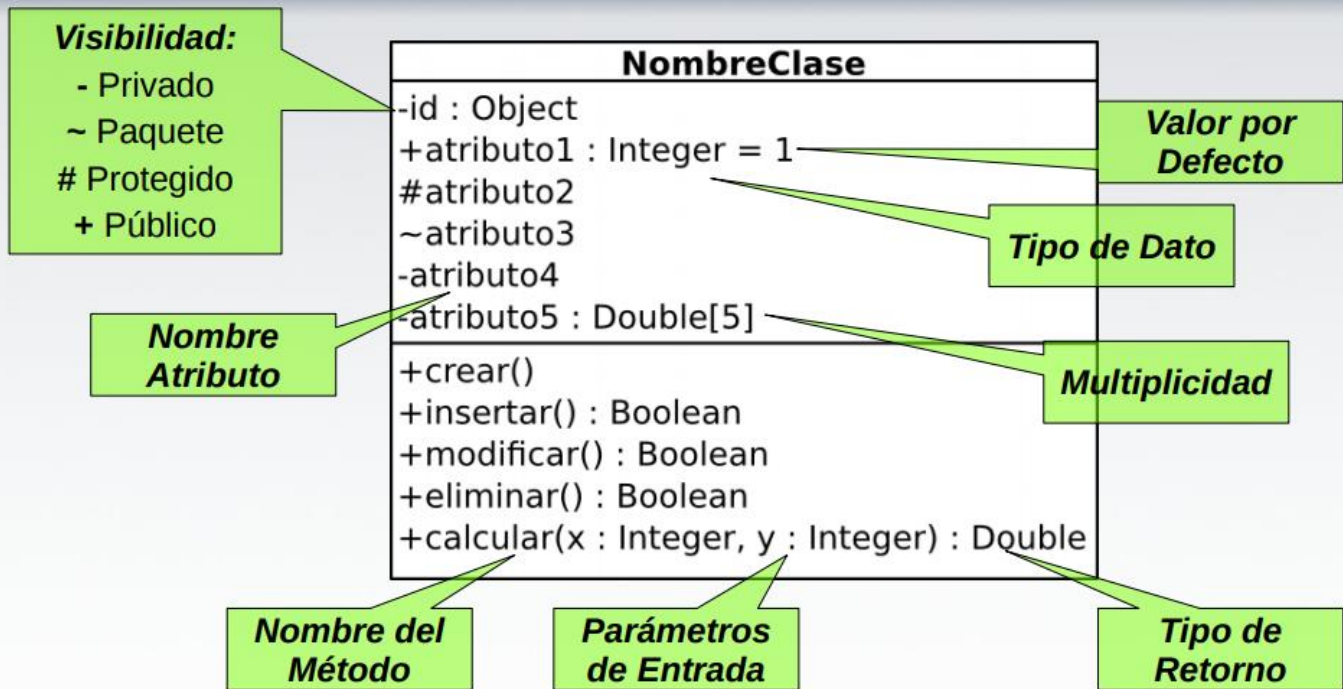


Figura 9: Según (Gutierrez, 2011) es la estructura de una clase en el Diagrama de clases

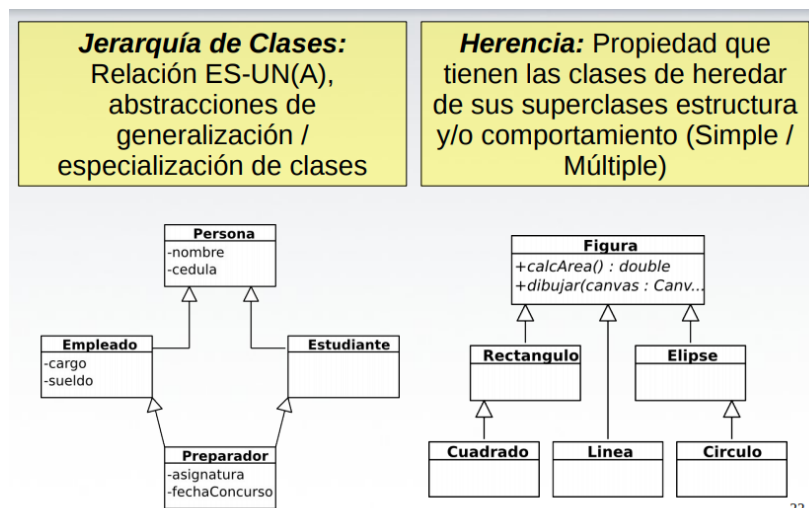


Figura 10: Según (Gutierrez, 2011) estas son un tipo de relaciones entre las clases

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

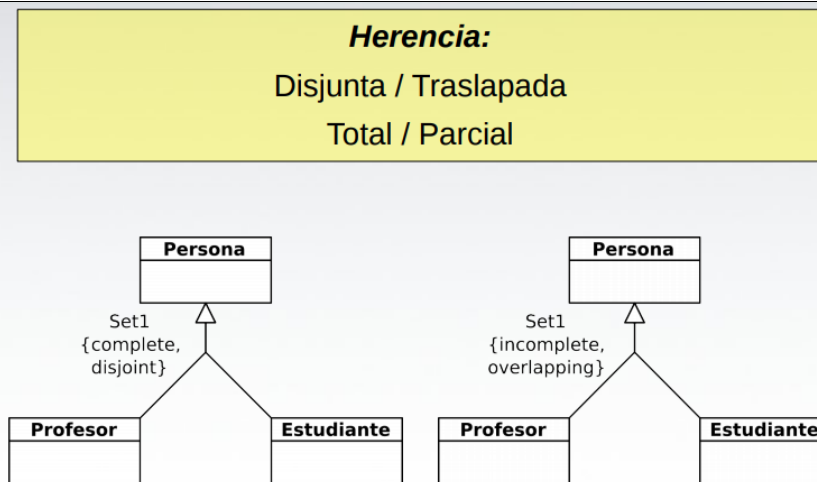


Figura 11: Según (Gutierrez, 2011) estas son un tipo de relaciones entre las clases

Vista lógica o Estructural

- Polimorfismo: se puede usar el mismo nombre para la definición de un método en varias clases sin importar la relación entre las mismas.
- Reescritura o sobrecarga: permite nombrar código diferente con el mismo nombre para más de una clase de objetos.
- Encadenamiento tardío: permite seleccionar el código adecuado al objeto definido en la invocación del método.

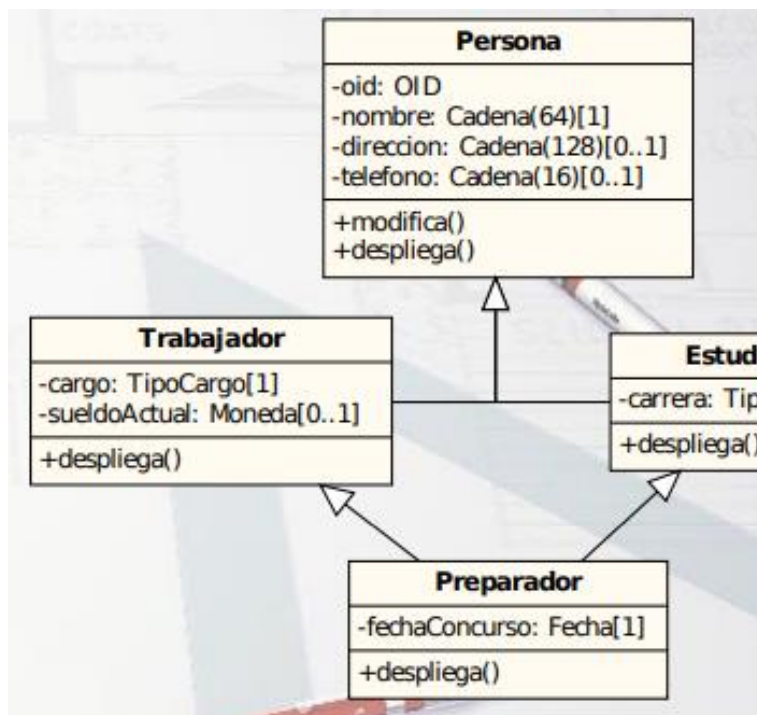


Figura 12: Según (Gutierrez, 2011) este diagrama es un ejemplo lógico-estructural

Asociaciones: Representan relaciones estructurales entre las clases (la forma en que están relacionadas entre si las clases)

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

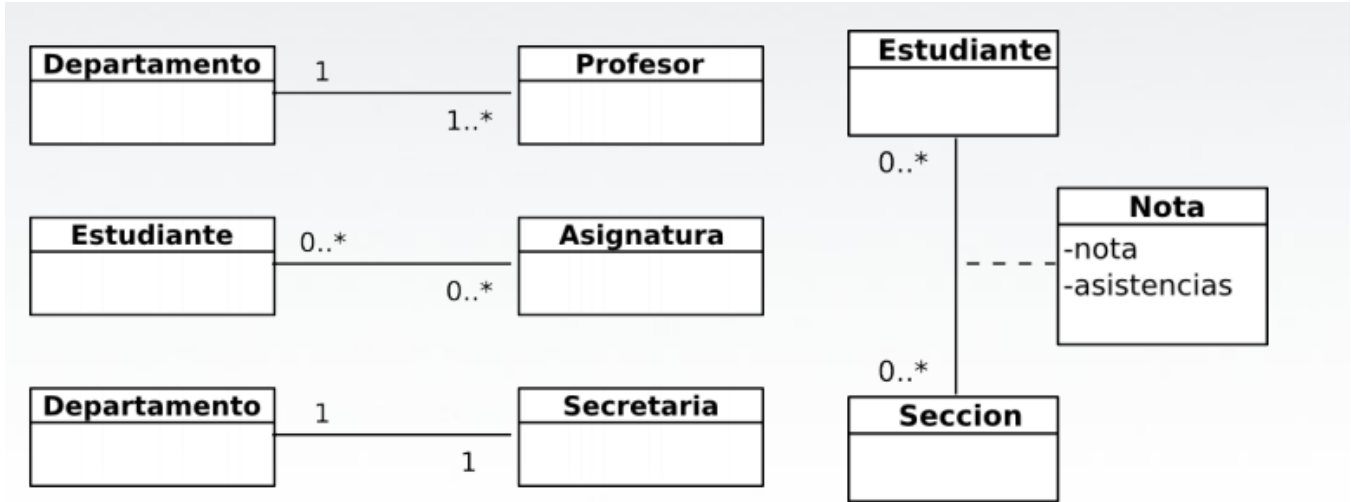
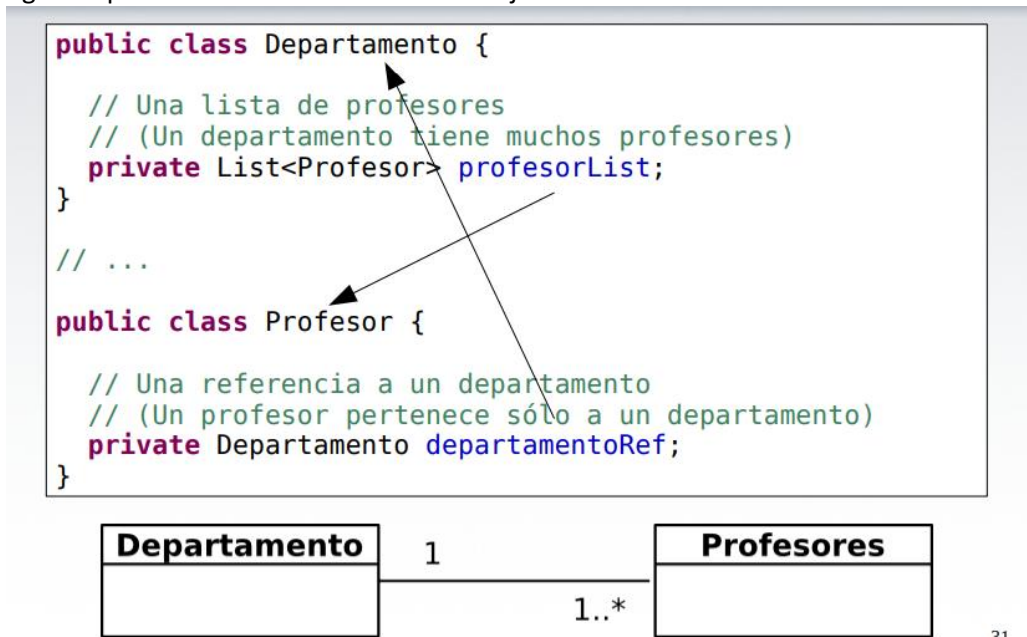


Figura 13: Según (Gutierrez, 2011) este diagrama es un ejemplo de las asociaciones que se pueden presentar entre las clases y como se deben representar

A continuación, esta figura representa como se debe escribir en java una asociación 1 a 1 :



31

Figura 14: Según (Gutierrez, 2011) este es un ejemplo de una relación 1:1 entre clases y su representación en java

A continuación, esta figura representa como se debe escribir en java una relación N:M

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

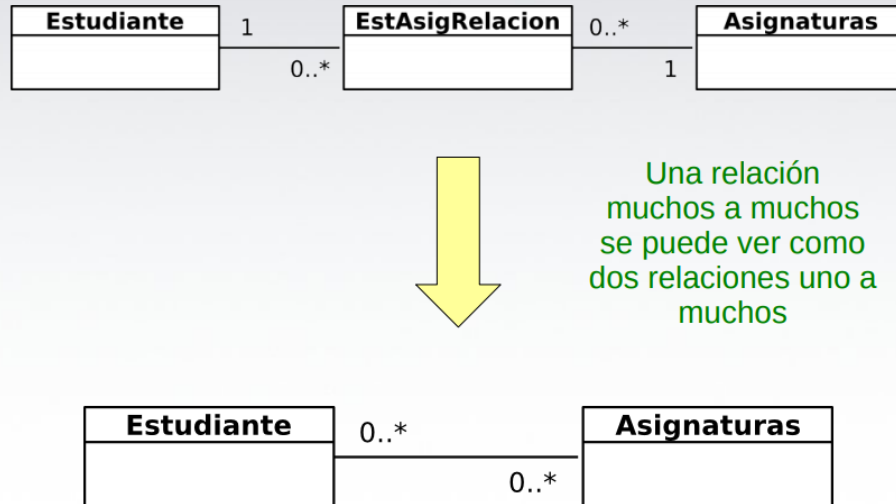


Figura 15: Según (Gutierrez, 2011) esta es la representación en el diagrama de clases

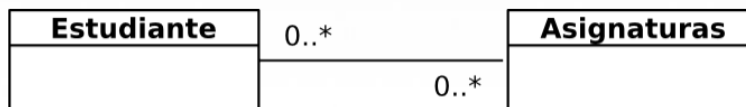
```

public class Estudiante {
    // Una lista de EstAsigRelacion (Una clase relación)
    private List<EstAsigRelacion> estAsigRelacionList;
}

public class EstAsigRelacion {
    // referencias cruzadas a las dos clases relacionadas
    private Estudiante estudianteRef;
    private Asignatura asignaturaRef;
}

public class Asignatura {
    // Una lista de EstAsigRelacion (Una clase relación)
    private List<EstAsigRelacion> estAsigRelacionList;
}

```



33

Figura 16: Según (Gutierrez, 2011) este es un ejemplo de una relación N:M entre clases y su representación en java

Navegabilidad: representan relaciones estructurales entre las clases (la forma en que están relacionadas entre si las clases)

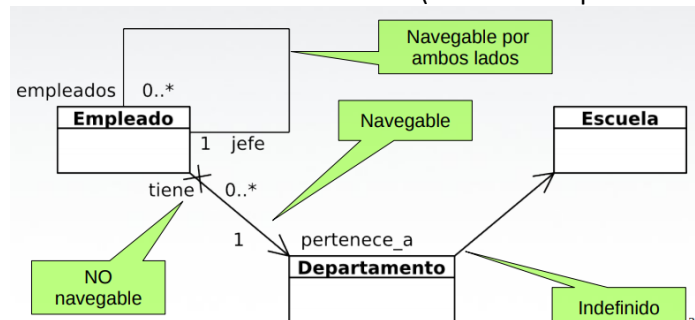


Figura 17: Según (Gutierrez, 2011) este es un ejemplo de Navegabilidad

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

Agregación: Es una relación en la que una de las clases representa un todo y la otra representa parte de ese todo

Composición: Es una forma más fuerte de la agregación, en la que el todo no puede existir sin sus partes

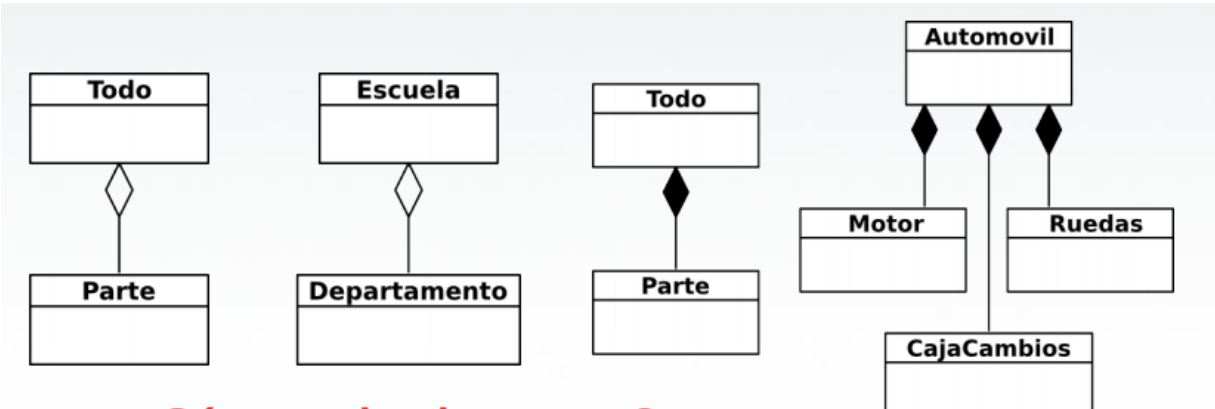


Figura 18: Según (Gutierrez, 2011) este es un ejemplo de Agregación y Composición

Composición: Las partes no pueden existir sin el todo En contradicción con el ejemplo anterior

Composición: El todo no puede existir sin las partes (Ejemplo Anterior)

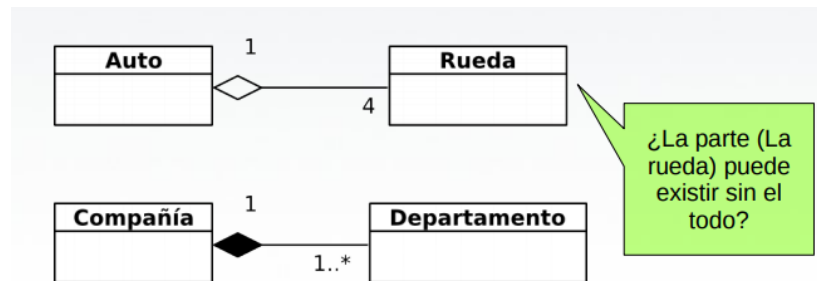


Figura 19: Según (Gutierrez, 2011) estos son ejemplos de composición

Agregación: ¡Las partes pueden ser compartidas por varios todos!

Composición: ¡Las partes NO pueden ser compartidas por varios todos!

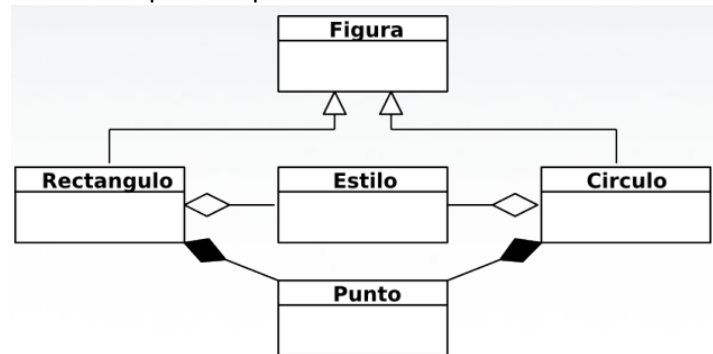


Figura 20: Según (Gutierrez, 2011) este es otro caso de Agregación y de Composición

Dependencia: Relación en la que una clase necesita (requiere) a otra para poder funcionar

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

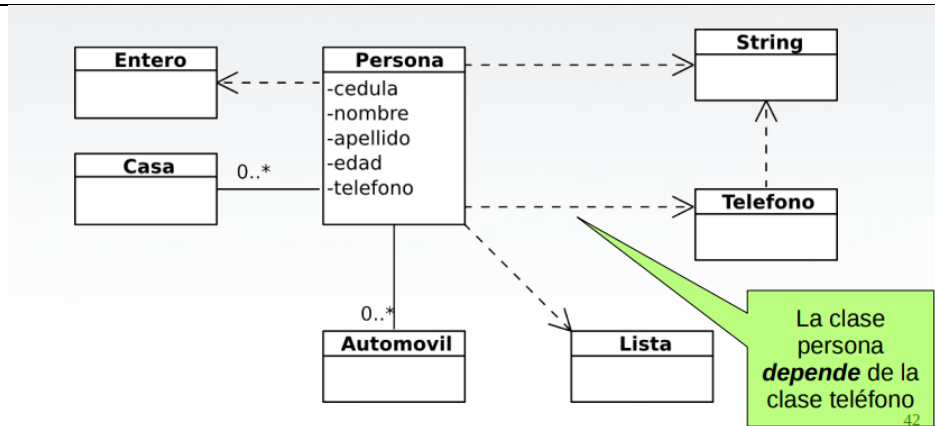


Figura 21: Según (Gutierrez, 2011) este es un ejemplo de dependencia

Interfaz: Clase asociada que describe su comportamiento visible. Conjunto de métodos que describen el comportamiento visible de una clase

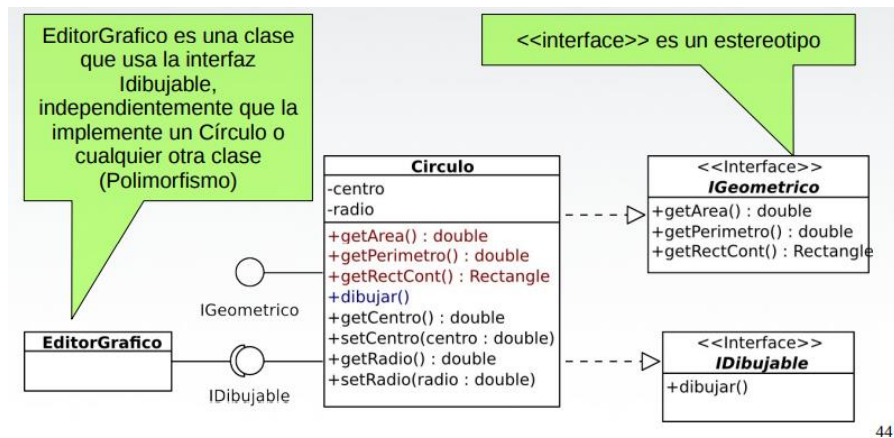


Figura 22: Según (Gutierrez, 2011) este es un ejemplo de Interfaz

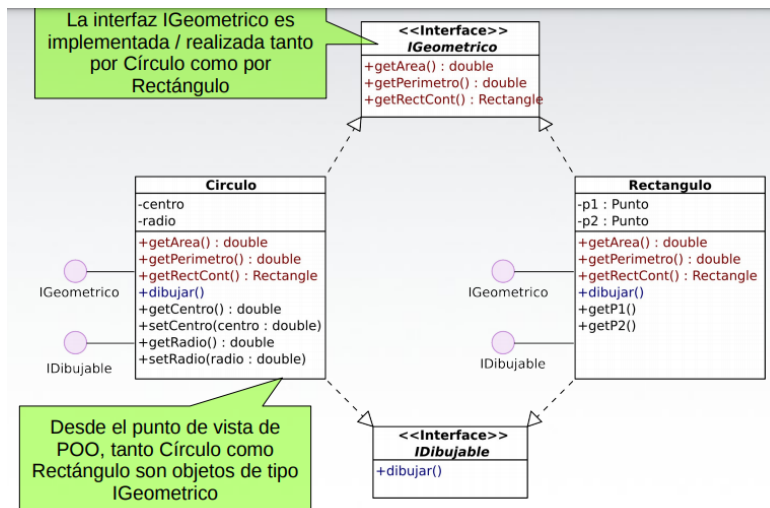


Figura 23: Según (Gutierrez, 2011) este es un ejemplo de Interfaz

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos

Fecha: 13 de marzo del 2021

Módulo: III Módulo

Semestre: Tercero

Modalidad: Nocturna

Materia: Lenguaje de Programación II

Profesor: Ing. Freddy Catucuamba

```
import java.awt.Point;
import java.awt.Rectangle;

public class Circulo implements IGeometrico, IDibujable {

    private double centro;
    private double radio;

    public double    getArea()        { /* de IGeometrico */ }
    public double    getPerimetro()   { /* de IGeometrico */ }
    public Rectangle getRectCont()    { /* de IGeometrico */ }

    public void dibujar()              { /* de IDibujable */ }

    public Point getCentro()           { /* de circulo */ }
    public void setCentro(...)         { /* de circulo */ }

    public double getRadio()           { /* de circulo */ }
    public void setRadio(...)          { /* de circulo */ }
}
```

16

```
import java.awt.Rectangle;

public interface IGeometrico {

    public double getArea();

    public double getPerimetro();

    public Rectangle getRectCont();
}
```

```
public interface IDibujable {

    public void dibujar();
}
```

```
List<IDibujable> elementosDibujar;

// ...

for (IDibujable dibujable :
    elementosDibujar) {
    // No importa si dibujable es
    // un círculo, rectángulo, etcétera
    // Los puedo manejar a todos igual
    // porque tienen una interfaz en común
    dibujable.dibujar();
}
```

Algunos de estos son círculos, otros son rectángulos, estrellas, líneas, etcétera... Pero todos implementan la interfaz IDibujable

¡El acto de magia de las interfaces y el polimorfismo!

48

Figura 24: Según (Gutierrez, 2011) este es un ejemplo de Interfaz en Java

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

CREACIÓN DE LA BASE DE DATOS

La creación de la base de datos se la realizó en el software POSTGRES SQL en donde se utilizó la versión 12 con el puerto 533. A continuación se muestra una captura de todas las tablas que se crearon en el proyecto y un extracto del código SQL del cual se ejecutó.

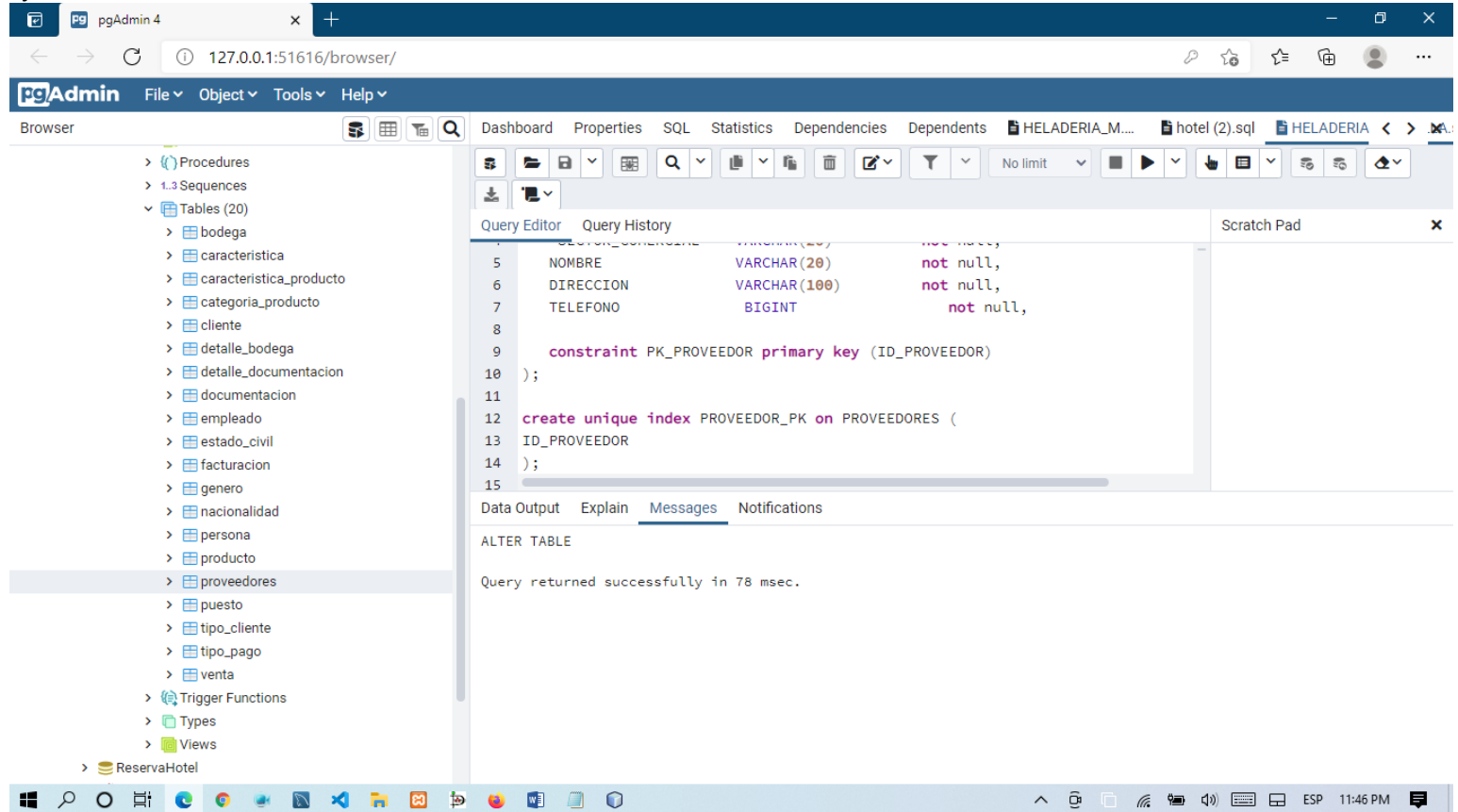


Figura 25: Este es un extracto del código sql, además de las tablas creadas en POSTGRES SQL.

CREACIÓN DE LOS ARCHIVOS EAR CON WAR Y JAR, MAPEO RELACIONAL Y CREACIÓN DE ARCHIVOS DAO

A continuación se presenta los pasos que se realizaron para crear un archivo ear con war y jar.

1.- Se creó un archivo JEE.

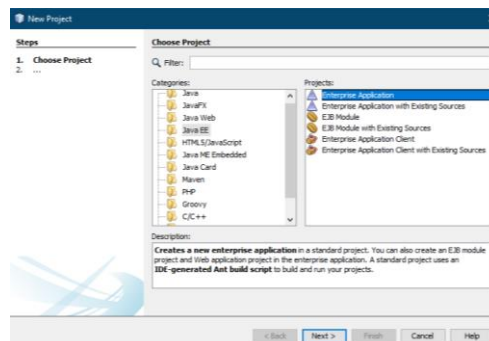


Figura 26: Esta es la captura de creación del archivo.

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos

Fecha: 13 de marzo del 2021

Módulo: III Módulo

Semestre: Tercero

Modalidad: Nocturna

Materia: Lenguaje de Programación II

Profesor: Ing. Freddy Catucuamba

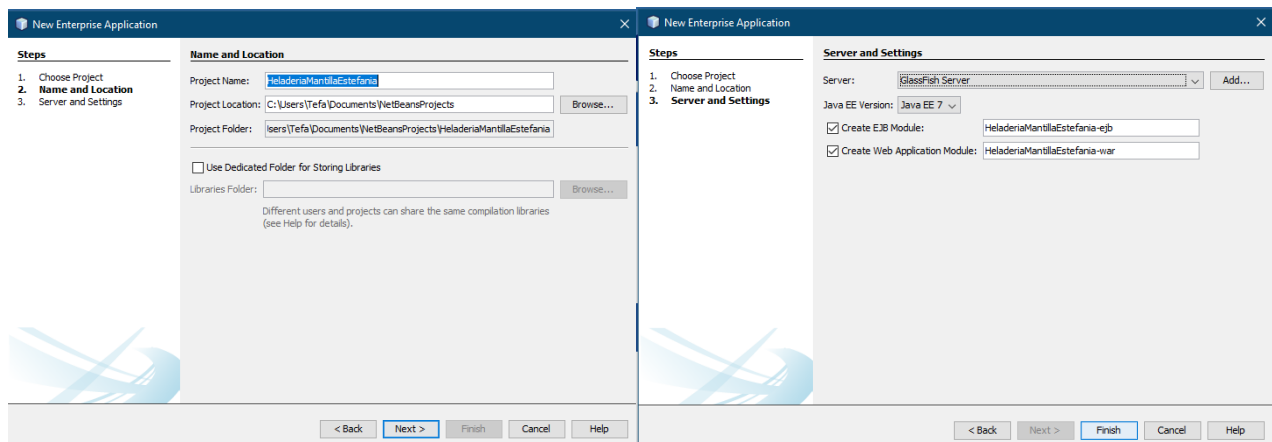


Figura 27: Esta es la captura de los detalles de creación.

2.- Creación de los paquetes donde se alojaron los modelos y los demás archivos

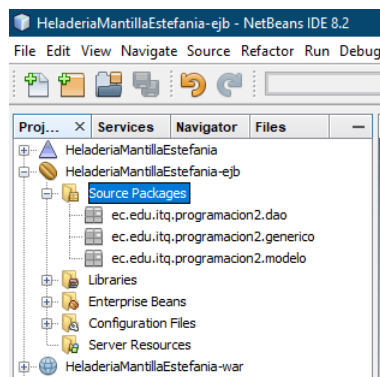


Figura 28: Esta es la captura de los paquetes en .EJB

3.- Creación de la conexión con la base de datos:

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos		Fecha: 13 de marzo del 2021
Módulo: III Módulo	Semestre: Tercero	Modalidad: Nocturna
Materia: Lenguaje de Programación II		Profesor: Ing. Freddy Catucuamba

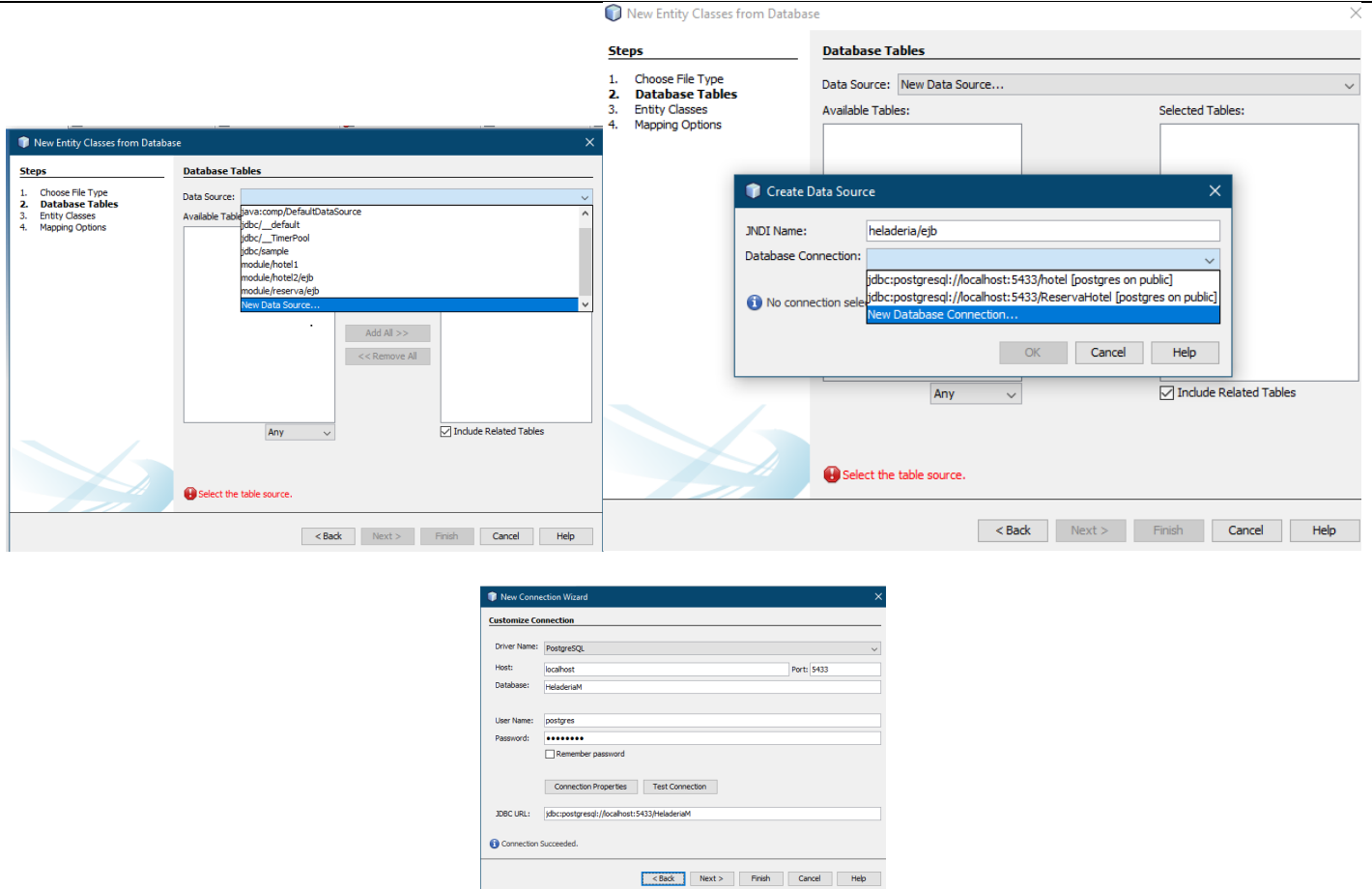
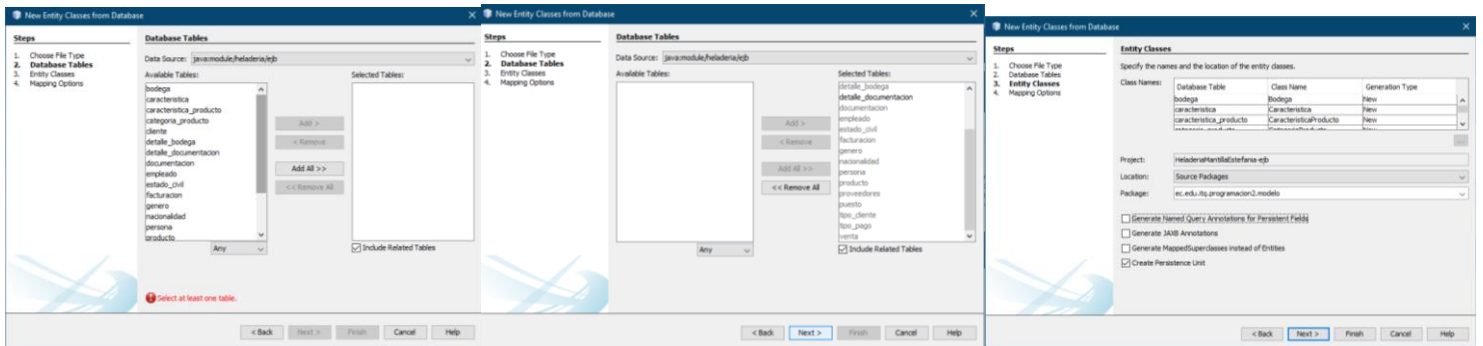


Figura 29: Conexión con la base de datos exitosa

3.- Cargar los modelos en los paquetes



INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos

Fecha: 13 de marzo del 2021

Módulo: III Módulo

Semestre: Tercero

Modalidad: Nocturna

Materia: Lenguaje de Programación II

Profesor: Ing. Freddy Catucuamba

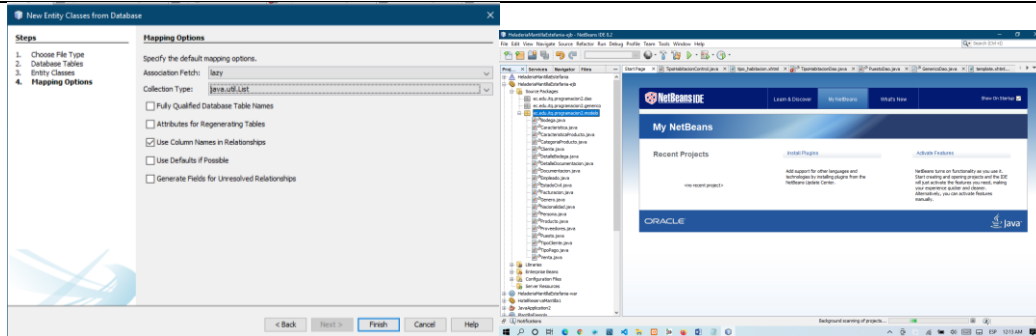


Figura 30: Creación de Modelos

4.- Creación de Dao y generación del archivo Genérico

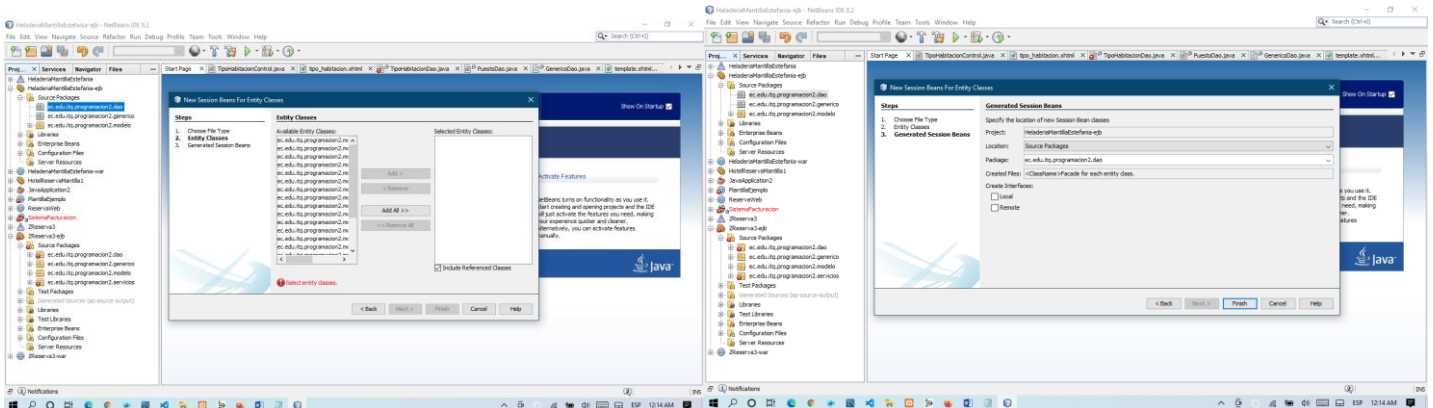


Figura 31: Creación de Dao

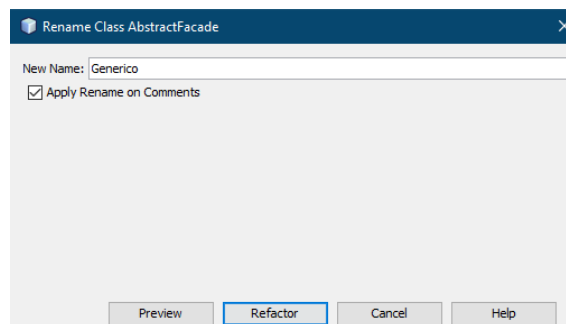


Figura 32: Generación de archivo genérico

5.- Renombre de los archivos a Dao

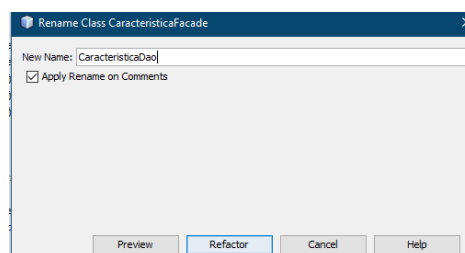


Figura 33: Creación de los archivos Dao

INSTITUTO TECNOLÓGICO QUITO

INFORME DE PROYECTO FINAL

Nombre: Rita Estefanía Mantilla Gallegos

Fecha: 13 de marzo del 2021

Módulo: III Módulo

Semestre: Tercero

Modalidad: Nocturna

Materia: Lenguaje de Programación II

Profesor: Ing. Freddy Catucuamba

6.- Eliminación del contenido de los archivos Dao

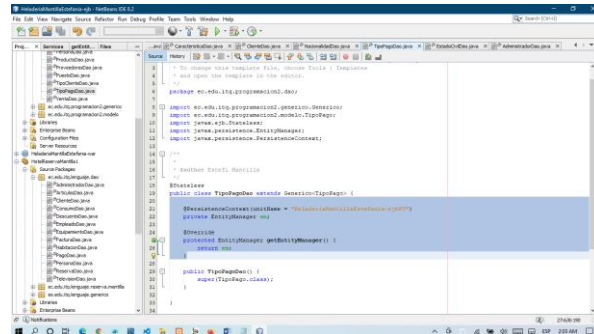


Figura 34: Eliminación del contenido de los archivos Dao.

Conclusiones:

- El diagrama de clases constituye un elemento de importancia para el modelado del sistema y la representación de distintos conceptos como el Modelo de Dominio el Dominio de Aplicación.
- El diagrama de usos general es una herramienta que pretende modelar el funcionamiento del sistema.

Recomendaciones:

- Se recomienda identificar todos los tipos de relaciones que existen para los dos diagramas así como también identificar correctamente las clases, objetos, instancias y como instanciar.

Link del deber en git hub:

<https://github.com/estefania906/DeberRamas.git>

Bibliografía:

Garcia, F., & Garcia, A. (2018). *FUNDAMENTOS DE LA VISTA DE CASOS DE USO*. Retrieved from <https://repositorio.grial.eu/bitstream/grial/1155/1/UML%20-%20Casos%20de%20uso.pdf>

Gutierrez, D. (2011). *UML Diagramas de Clases*. Retrieved from https://www.codecompiling.net/files/slides/UML_clase_04_UML_clases.pdf

Microsoft. (2021). *Crear un diagrama de casos de uso UML*. Retrieved from <https://support.microsoft.com/es-es/office/crear-un-diagrama-de-casos-de-uso-uml-92cc948d-fc74-466c-9457-e82d62ee1298>