

The background of the slide is a photograph of a city skyline, likely Rio de Janeiro, viewed from a high vantage point. A person is sitting on a rooftop in the foreground, looking out over the city. The image has a blue and green color grade. Overlaid on the image are several white, glowing, circular lines that spiral around various buildings, suggesting a digital or data-driven theme. In the top left corner, there is a green-to-white gradient bar.

BECA Java **Verano 2017**

Enrique Mingorance Cano

enrique.Mingorance.cano@everis.com





02

Empezando
a
Desarrollar

Estructura de un proyecto java

Conceptos Básicos

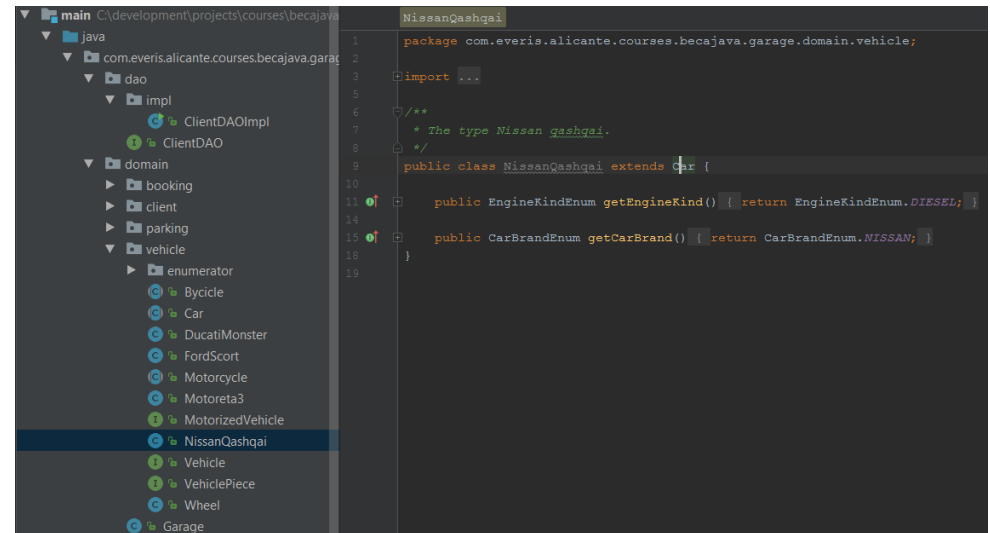
- Las clases se guardan en ficheros .java que se organizan en carpetas
- Cada nivel de carpeta representa un nivel de paquete siendo el raíz la ausencia de paquete
- Los paquetes son determinantes desde el punto de vista de la visibilidad de las clases
 - Las clases en el mismo paquete son accesibles
 - Las clases en diferentes paquetes necesitan ser públicas
- Un conjunto de clases puede ser empaquetado en un contenedor denominado jar

```
C:\development\projects\courses\becajava07>jar cvf pepe.jar .
```



Estructura de un proyecto java

Paquete II



> Este equipo > OSDisk (C:) > development > projects > courses > becajava07 > garage > src > main > java > com > everis > alicante > courses > becajava > garage > domain > vehicle >

Nombre	Fecha de modifica...	Tipo	Tamaño
enumerator	03/07/2017 8:44	Carpeta de archivos	
Bicycle.java	29/06/2017 16:47	IntelliJ IDEA	1 KB
Car.java	29/06/2017 16:36	IntelliJ IDEA	1 KB
DucatiMonster.java	29/06/2017 16:37	IntelliJ IDEA	1 KB
FordScort.java	29/06/2017 16:36	IntelliJ IDEA	1 KB
Motorcycle.java	29/06/2017 16:37	IntelliJ IDEA	1 KB
Motoreta3.java	29/06/2017 16:47	IntelliJ IDEA	1 KB
MotorizedVehicle.java	29/06/2017 16:36	IntelliJ IDEA	1 KB
NissanQashqai.java	29/06/2017 16:36	IntelliJ IDEA	1 KB

Estructura de un proyecto java

Paquete I – Convención

- Se escriben siempre con cualquier carácter del código ASCII en minúsculas, separando las palabras por un punto y sin espacios
- Empezando siempre por com, edu, gov, mil, net, org o uno de los códigos de dos letras establecidos por ISO-3166 <https://www.iso.org/obp/ui/#search>
- En caso de tener que utilizar palabras reservadas de java por convención se añade _ al final del nombre. Por ejemplo com.int_.example
- Los paquetes propios de java empiezan por java o javax.

Exemplos:

- Correctos
 - com.sun.eng
 - com.apple.quicktime.v2
 - edu.cmu.cs.bovik.cheese
- Incorrectos
 - com.sun_eng (por la separación entre sun_eng)
 - com.long.example (uso de palabra reservada)



Java

Empezando a Desarrollar

Estructura de un proyecto java

Fichero .java

- Las clases se definen en ficheros con extensión .java

// Filename: NewApp.java

```
// PART 1: (OPTIONAL) package declaration
package com.company.project.fragilePackage;
```

```
// PART 2: (ZERO OR MORE) import declarations
import java.io.*;
import java.util.*;
```

```
// PART 3: (ZERO OR MORE) top-level class and interface declarations
public class NewApp { }
```

```
class AClass { }
```

```
interface IOne { }
```

```
class BClass { }
```

```
interface ITwo { }
```

```
// ...
```

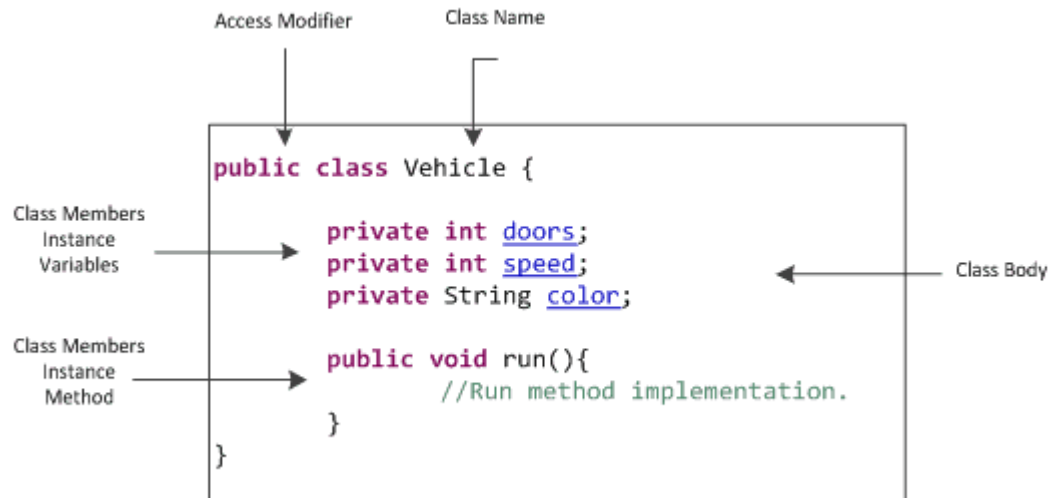
```
// end of file
```

- AST de un fichero java 8 <https://github.com/antlr/grammars-v4/blob/master/java8/Java8.g4>



Clase java

Body Básico



Método java

Método Básico

return type **method name** **value passed to the method**

```
int total( int aNumber) {  
    int a_Value = aNumber + 10;  
    return a_Value;  
}
```



Código básico java

```
public class Cube {  
    private int length;  
  
    public Cube(int length) { this.length = length; }  
  
    public double getVolume() { return (Math.pow(length, 3)); }  
}
```

```
public class CubeFactory {  
    public static Cube generateCube(int length) {  
        Cube cube = new Cube(length);  
        return cube;  
    }  
}
```

```
public class AplicacionFlipante {  
  
    public static void main(final String[] args) {  
        double volumeCubeOne = CubeFactory.generateCube( length: 5 ).getVolume();  
        System.out.println(volumeCubeOne);  
        double volumeCubeTwo = CubeFactory.generateCube( length: 2 ).getVolume();  
        System.out.println(volumeCubeTwo);  
    }  
}
```



Ejecutar una aplicación java

Clase

```
public class AplicacionFlipante {  
    public static void main(final String[] args){  
        System.out.println("Eres guapísimo!!");  
    }  
}
```

- Compilar

```
C:\development\projects\courses\becajava07\garage\src\main\java>javac AplicacionFlipante.java
```

- Ejecutar

```
C:\development\projects\courses\becajava07\garage\src\main\java>java AplicacionFlipante  
Eres guapísimo!!
```



Ejecutar una aplicación java

Jar

- Ejecutar

<https://stackoverflow.com/a/32439530>



A background image showing several pairs of hands clasped together on a wooden table, symbolizing teamwork and collaboration. The image has a green-to-blue color gradient.

03

Nuestro
Primer
Proyecto

Gestión de Taller Mecánico

Supuesto I

- Desde el taller “Mecanicos Ochenteros” nos piden que le desarrollemos una aplicación que les ayude a gestionar las citas con los clientes
- Todos los datos los vamos a manejar de tipo textual (menos los objetos)
- Quieren gestionar la entrada de coches, motos y bicicletas
 - Los vehículos constan de las siguientes partes:
 - Ruedas
 - Altura, anchura, pulgadas, marca
 - Volante (marca)
 - Motor (CV, CC, marca y consumo)
 - Quieren gestionar los siguientes tipos de coches
 - Nissan Qashqai
 - Ford Focus
 - Los siguientes tipos de motos
 - Ducati Monster
 - Vespino
 - Y las bicicletas:
 - Motoreta 3
 - BH Gacela

Gestión de Taller Mecánico

Supuesto II

- Los clientes constan de:
 - DNI
 - Name
 - Surname
 - Fecha de Nacimiento
- Las citas
 - Fecha de reserva
 - Tiempo estimado de reparación
 - Cliente
 - Vehiculo
 - Descripción de la cita o incidencia