

The background of the slide is a photograph of a person sitting on a rooftop, looking out over a city skyline. The image has a blue and green color grade. Overlaid on the image are several white, concentric circles and lines that suggest a digital or network theme. A large green circle is positioned on the left side of the image, partially overlapping the person and the city.

BECA Java

Enrique Mingorance Cano

enrique.Mingorance.cano@everis.com





Java 8

Conceptos
Básicos

Empezando
a Desarrollar

Nuestro
Primer
Proyecto

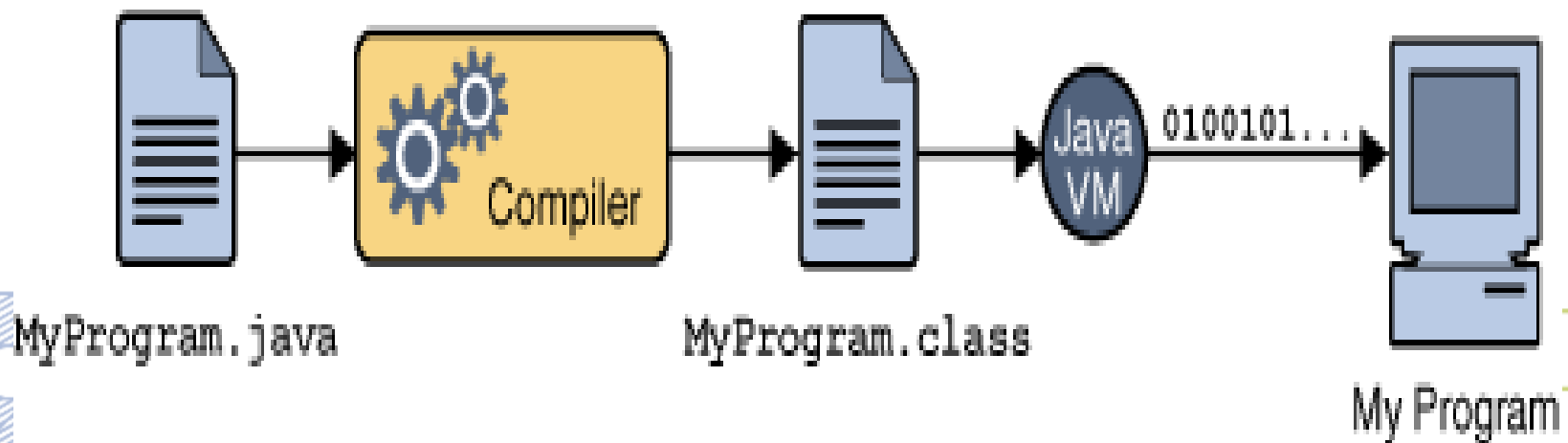


01

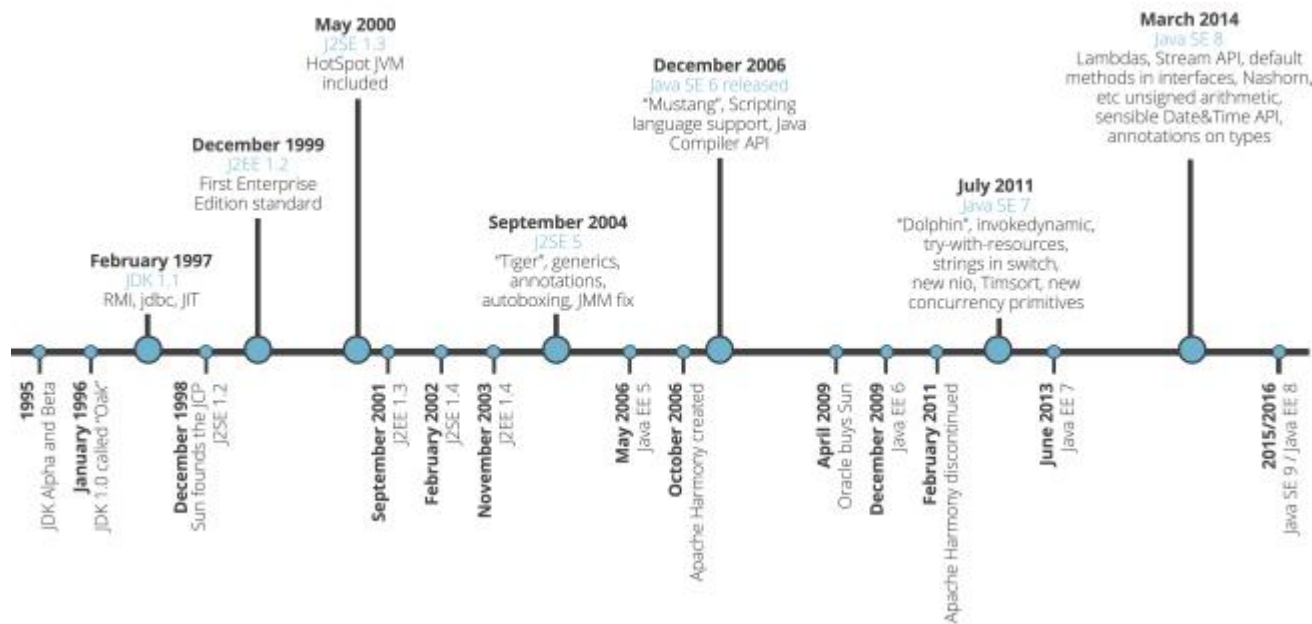
Conceptos Básicos

¿Qué es Java?

- Lenguaje de programación desarrollado por Sun Microsystems (posteriormente comprada por Oracle)
- Corre sobre una máquina virtual JVM (mismo código compatible en diferentes arquitecturas) que interpreta ficheros .class



Historia Java

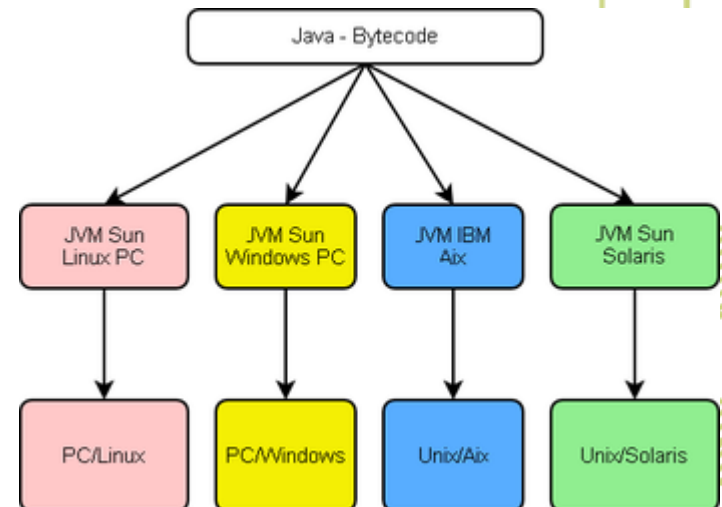


- https://en.wikipedia.org/wiki/Java_version_history

JVM I

Definición

- Es la máquina virtual o programa que provee el entorno de ejecución donde java se ejecuta
- Interpreta ficheros .class
- Diferentes máquinas virtuales para diferentes arquitecturas (multiplataforma)



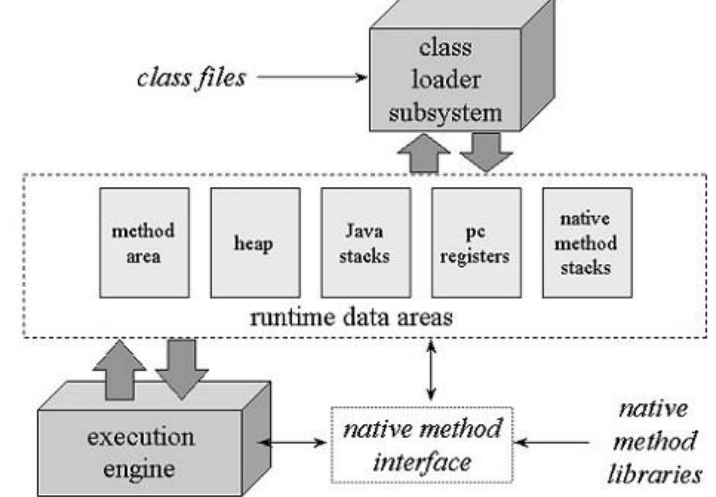
Java

Conceptos Básicos

JVM II

Arquitectura

- **Class Loader:** Responsable de encontrar y cargar los .class
- **Method Area:** Contiene la información de los tipos/clases cargados por el Class Loader
- **Heap/Stack:** Ahora lo veremos
- **Program Counter Registers (PC):** Guarda un registro de los hilos de ejecución
- **Native method stack:** Un programa de java puede llamar a métodos nativos de la arquitectura del sistema, por ejemplo aplicaciones escritas en C. Esta pieza contiene esos métodos nativos
- **Execution Engine:** Procesador/Intérprete/JIT Compiler que ejecuta las instrucciones.



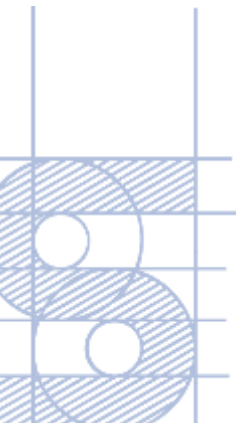
JVM Diagram

Figure by javawithease

JVM III

Heap / Space

- ... Antes veamos qué es una clase ...



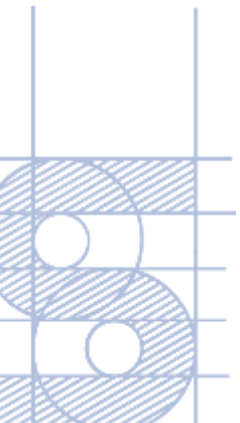
Objetos y Clases I

- Java es un lenguaje de programación orientado a objetos (POO) que soporta los siguientes conceptos
 - Polymorphism
 - Inheritance
 - Encapsulation
 - Abstraction
 - Classes
 - Objects
 - Instance
 - Method
 - Message Passing (o llamada a método, el nombre del método y sus parámetros, en términos generales, su signature)

Objetos y Clases II

Objeto

- Un objeto es una instancia de una clase
- La instancia se guarda en memoria (heap) y una referencia apunta a ella (stack). Lo veremos más adelante.
- Los objetos tiene estados y comportamientos (muy similar al mundo real)
 - Estados: nombre, color, tamaño
 - Comportamientos: ladrar, correr, saltar



Objetos y Clases III

Clase

- Es el cianotipo que usan los objetos para crearse (copia original, contrato, etc...)
- Las clases y java (como en unix) son sensibles a Mayúsculas y Minúsculas (no es lo mismo CarPart que carPart o carpart)
- Una clase está compuesta de la siguiente forma (AST) <https://github.com/antlr/grammars-v4/blob/master/java8/Java8.g4>

public
protected
private
abstract
static
final
strictfp

Empieza por
Letra
Ilimitada
secuencia de
Número y
Letras

classModifier* 'class' Identifier typeParameters? 'extends'? 'implements'? classBody

Objetos y Clases IV

Clase

```
1  public class Cube {  
2  
3      private int length;  
4      private int breadth;  
5      private int height;  
6  
7      public int getVolume() {  
8          return (length * breadth * height);  
9      }  
10 }
```

Objetos y Clases V

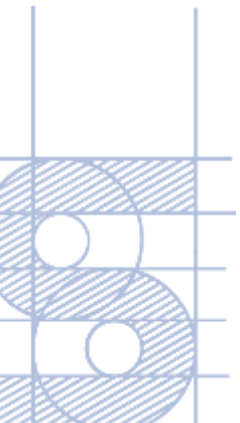
Instanciar una clase

```
public class CubeFactory {  
    Cube cube = new Cube();  
}
```

Java

Primer Programa

- Pero antes... qué necesitamos?...Librerías para ejecutar una clase Java



JRE / JDK

JRE

- Contiene la máquina virtual Java que interpreta los “bytecodes” de los programas.
- Contiene el conjunto de librerías y paquetes que todo programa Java utiliza y que vienen con la máquina virtual Java por defecto.
- Se utiliza en entornos de sólo ejecución, en entornos finales.
- <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

JRE / JDK

JDK

- Contiene JRE
- Contiene utilidad de compilación (*javac*).
- Otras utilidades, ejemplos y documentación.
- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Java

Conceptos Básicos

Java

Primer Programa (bis)

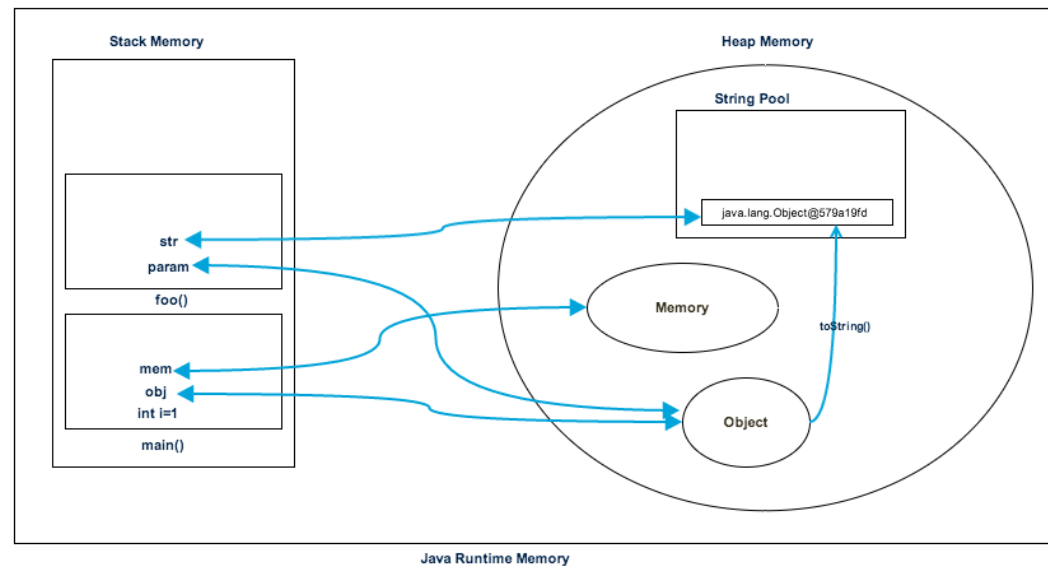
```
public class AplicacionFlipante {
    public static void main(final String[] args){
        System.out.println("Eres guapísimo!!");
    }
}
```

JVM III (bis)

Heap / Space

- ... Ahora sí! ...

```
class Memory {  
  
    public static void main(String[] args) { // Line 1  
        int i=1; // Line 2  
        Object obj = new Object(); // Line 3  
        Memory mem = new Memory(); // Line 4  
        mem.foo(obj); // Line 5  
    } // Line 9  
  
    private void foo(Object param) { // Line 6  
        String str = param.toString(); //// Line 7  
        System.out.println(str);  
    } // Line 8  
}
```

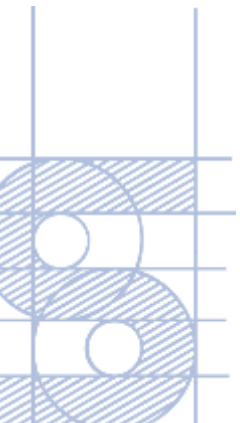


Java Runtime Memory

JVM IV

Garbage Collector

- Por simplificar se encarga de liberar la memoria del Heap, eliminando las referencias no referenciadas desde el stack
- más detalles en <http://www.journaldev.com/2856/java-jvm-memory-model-memory-management-in-java>

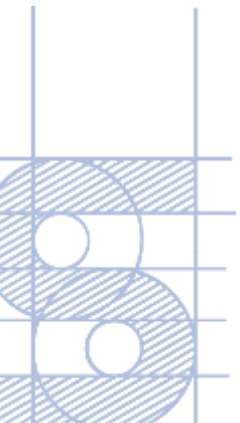


Palabras Reservadas

- La lista de palabras que no se deben utilizar bajo ninguna circunstancia como nombre de ninguna variable, método o clase.
- https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

Comentarios y documentación

- `/* */` Comentar bloques
- `//` Comentar líneas
- `/** */` Indica documentación



A low-angle, upward-looking photograph of several modern skyscrapers with glass facades, reaching towards a bright, hazy sky. The perspective creates a sense of height and architectural scale.

02

Empezando
a
Desarrollar

Estructura de un proyecto java

Conceptos Básicos

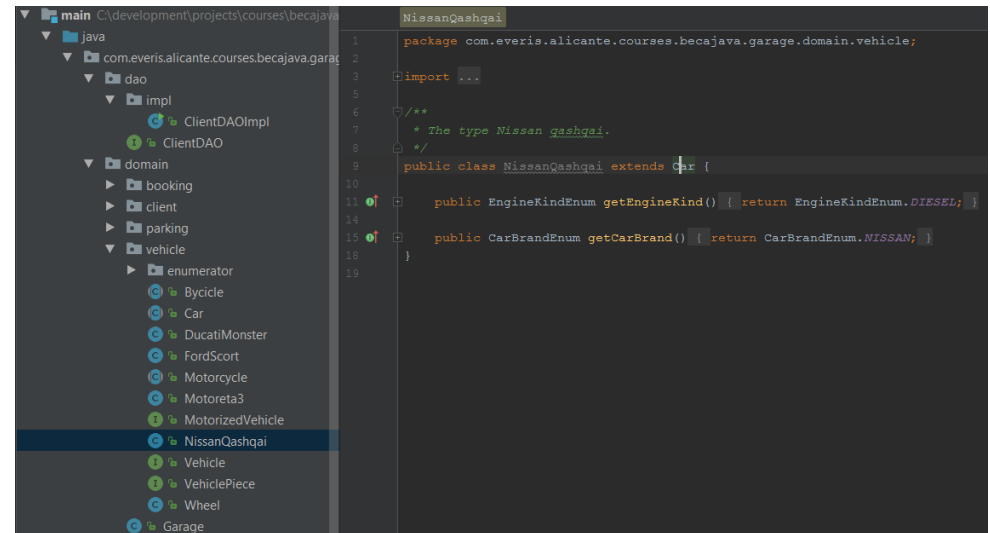
- Las clases se guardan en ficheros .java que se organizan en carpetas
- Cada nivel de carpeta representa un nivel de paquete siendo el raíz la ausencia de paquete
- Los paquetes son determinantes desde el punto de vista de la visibilidad de las clases
 - Las clases en el mismo paquete son accesibles
 - 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 { }
```

```
// ...
```

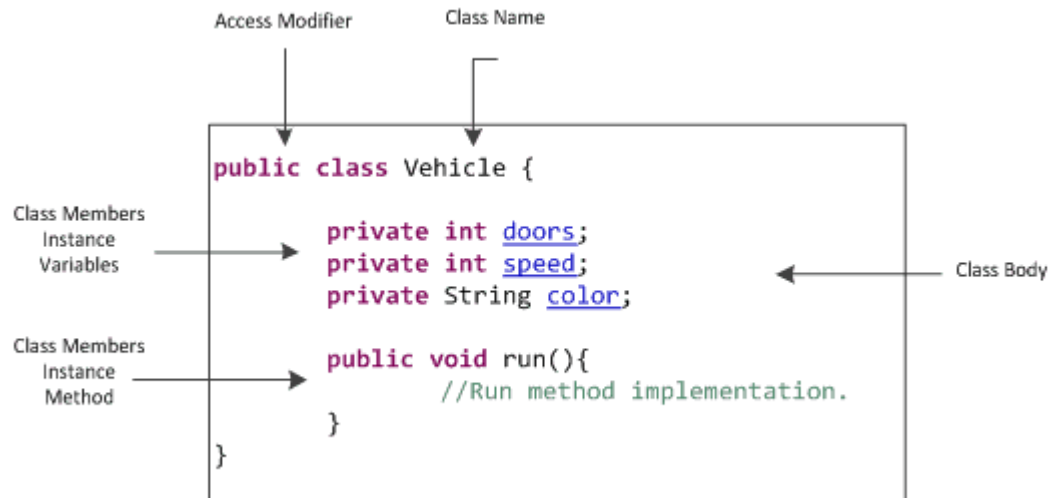
```
// end of file
```

- 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 is split vertically, with the left side having a green tint and the right side having a blue tint.

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