

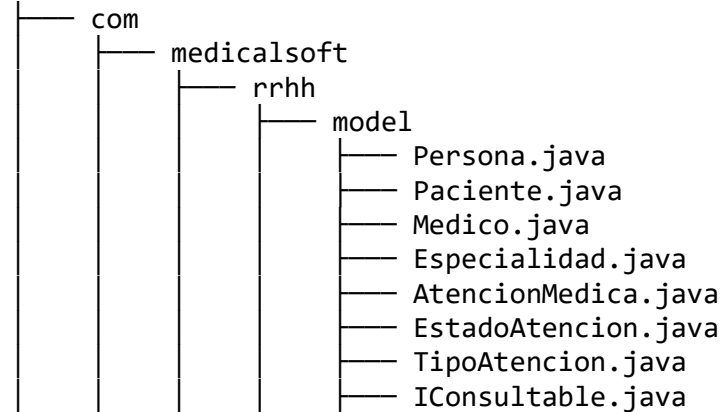
Contenido

Solución del Ejercicio 02	2
Solución del Ejercicio 03	6

Solución del Ejercicio 02

Elaboramos a nivel físico la siguiente estructura:

T:



Persona.java

```
package com.medicalsoft.rrhh.model;
public abstract class Persona implements IConsultable{
    ...
}
```

Paciente.java

```
package com.medicalsoft.rrhh.model;
import java.util.ArrayList;
public class Paciente extends Persona{
    ...
}
```

Medico.java

```
package com.medicalsoft.rrhh.model;
public class Medico extends Persona{
    ...
}
```

Especialidad.java

```
package com.medicalsoft.rrhh.model;
public class Especialidad{
    ...
}
```

AtencionMedica.java

```
package com.medicalsoft.rrhh.model;
import java.util.Date;
import java.text.SimpleDateFormat;
public abstract class AtencionMedica{
    ...
}
```

EstadoAtencion.java

```
package com.medicalsoft.rrhh.model;
public enum EstadoAtencion{
    ...
}
```

TipoAtencion.java

```
package com.medicalsoft.rrhh.model;
public enum TipoAtencion{
...
}
```

IConsultable.java

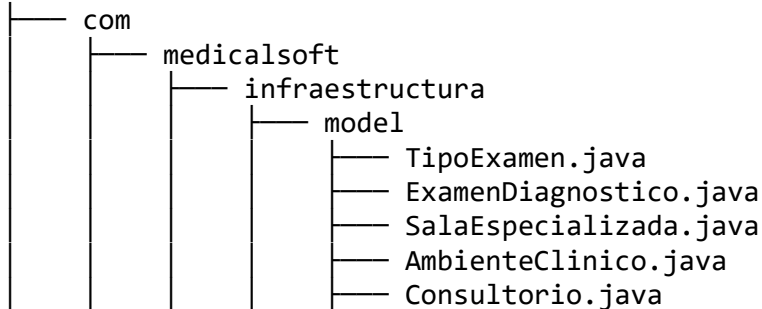
```
package com.medicalsoft.rrhh.model;
public interface IConsultable{
...
}
```

Corroboramos desde el directorio raíz que todo compila correctamente:

```
javac com/medicalsoft/rrhh/model/*.java
```

Elaboramos a nivel físico la siguiente estructura:

T:



TipoExamen.java

```
package com.medicalsoft.infraestructura.model;
public enum TipoExamen{
...
}
```

ExamenDiagnostico.java

```
package com.medicalsoft.infraestructura.model;
import java.util.Date;
import com.medicalsoft.rrhh.model.AtencionMedica;
import com.medicalsoft.rrhh.model.Paciente;
import com.medicalsoft.rrhh.model.Medico;
import com.medicalsoft.rrhh.model.TipoAtencion;
public class ExamenDiagnostico extends AtencionMedica{
...
}
```

SalaEspecializada.java

```
package com.medicalsoft.infraestructura.model;
public class SalaEspecializada extends AmbienteClinico{
...
}
```

AmbienteClinico.java

```
package com.medicalsoft.infraestructura.model;
import com.medicalsoft.rrhh.model.IConsultable;
public abstract class AmbienteClinico implements IConsultable{
...
}
```

Consultorio.java

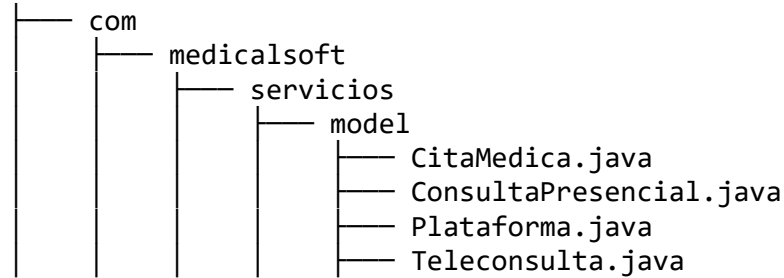
```
package com.medicalsoft.infraestructura.model;
public class Consultorio extends AmbienteClinico{
    ...
}
```

Corroboramos desde el directorio raíz que todo compila correctamente:

```
javac com/medicalsoft/rrhh/model/*.java com/medicalsoft/infraestructura/model/*.java
```

Elaboramos a nivel físico la siguiente estructura:

T:



CitaMedica.java

```
package com.medicalsoft.servicios.model;
import com.medicalsoft.rrhh.model.AtencionMedica;
import com.medicalsoft.rrhh.model.Paciente;
import com.medicalsoft.rrhh.model.Medico;
import com.medicalsoft.rrhh.model.TipoAtencion;
import java.util.Date;
public abstract class CitaMedica extends AtencionMedica{
    ...
}
```

ConsultaPresencial.java

```
package com.medicalsoft.servicios.model;
import com.medicalsoft.infraestructura.model.Consultorio;
import com.medicalsoft.rrhh.model.Paciente;
import com.medicalsoft.rrhh.model.Medico;
import java.util.Date;
public class ConsultaPresencial extends CitaMedica{
    ...
}
```

Plataforma.java

```
package com.medicalsoft.servicios.model;
public enum Plataforma{
    ...
}
```

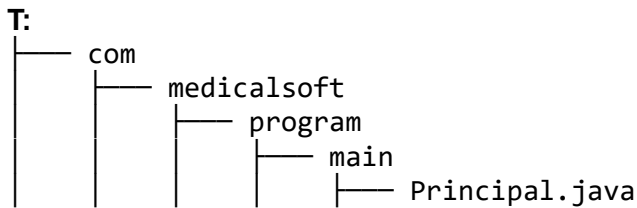
Teleconsulta.java

```
package com.medicalsoft.servicios.model;
import java.util.Date;
import com.medicalsoft.rrhh.model.Paciente;
import com.medicalsoft.rrhh.model.Medico;
public class Teleconsulta extends CitaMedica{
    ...
}
```

Corroboramos desde el directorio raíz que todo compila correctamente:

```
javac com/medicalsoft/rrhh/model/*.java com/medicalsoft/infraestructura/model/*.java  
com/medicalsoft/servicios/model/*.java
```

Elaboramos a nivel físico la siguiente estructura:



Principal.java

```
package com.medicalsoft.program.main;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import com.medicalsoft.rrhh.model.Especialidad;
import com.medicalsoft.rrhh.model.Medico;
import com.medicalsoft.rrhh.model.Paciente;
import com.medicalsoft.rrhh.model.AtencionMedica;
import com.medicalsoft.infraestructura.model.Consultorio;
import com.medicalsoft.infraestructura.model.SalaEspecializada;
import com.medicalsoft.infraestructura.model.ExamenDiagnostico;
import com.medicalsoft.infraestructura.model.TipoExamen;
import com.medicalsoft.servicios.model.ConsultaPresencial;
import com.medicalsoft.servicios.model.Teleconsulta;
import com.medicalsoft.servicios.model.Plataforma;
public class Principal{
    ...
}
```

Corroboramos desde el directorio raíz que todo compila y ejecuta correctamente:

```
javac com/medicalsoft/rrhh/model/*.java com/medicalsoft/infraestructura/model/*.java  
com/medicalsoft/servicios/model/*.java com/medicalsoft/program/main/*.java
```

```
java -cp  
com/medicalsoft/rrhh/model/*.class;com/medicalsoft/infraestructura/model/*.class;com  
/medicalsoft/servicios/model/*.class; com/medicalsoft/program/main/Principal
```

Solución del Ejercicio 03

1. Compile únicamente las clases contenidas en **com.medicalsoft.rrhh.model** desde el directorio raíz de trabajo.

```
javac com/medicalsoft/rrhh/model/*.java
```

2. Genere un componente .jar llamado "**medicalsoftRRHHModel.jar**" que contenga la estructura y clases en Bytecode contenidas en: **com.medicalsoft.rrhh.model**.

```
jar cf medicalsoftRRHHModel.jar com/medicalsoft/rrhh/model/*.class
```

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "**medicalsoftRRHHModel.jar**" (**com.medicalsoft.rrhh.model**). **Esto también puede realizarse de forma manual.**

```
rd com\medicalsoft\rrhh /s
```

Prosiga con las siguientes instrucciones.

3. Compile las clases contenidas en **com.medicalsoft.infraestructura.model** desde el directorio raíz de trabajo utilizando el .jar previamente creado: "**medicalsoftRRHHModel.jar**".

```
javac -cp medicalsoftRRHHModel.jar; com/medicalsoft/infraestructura/model/*.java
```

4. Genere un componente .jar llamado "**medicalsoftInfraModel.jar**" que contenga la estructura y clases en Bytecode contenidas en: **com.medicalsoft.infraestructura.model**.

```
jar cf medicalsoftInfraModel.jar com/medicalsoft/infraestructura/model/*.class
```

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "**medicalsoftInfraModel.jar**" (**com.medicalsoft.infraestructura.model**). Prosiga con las siguientes instrucciones. **Esto también puede realizarse de forma manual.**

```
rd com\medicalsoft\infraestructura /s
```

5. Compile las clases contenidas en **com.medicalsoft.servicios.model** desde el directorio raíz de trabajo utilizando los .jar previamente creados: "**medicalsoftRRHHModel.jar**" y "**medicalsoftInfraModel.jar**".

```
javac -cp medicalsoftRRHHModel.jar;medicalsoftInfraModel.jar; com/medicalsoft/servicios/model/*.java
```

6. Genere un componente .jar llamado "**medicalsoftServModel.jar**" que contenga la estructura y clases en Bytecode contenidas en: **com.medicalsoft.servicios.model**.

```
jar cf medicalsoftServModel.jar com/medicalsoft/servicios/model/*.class
```

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "**medicalsoftServModel.jar**" (**com.medicalsoft.servicios.model**). Prosiga con las siguientes instrucciones. **Esto también puede realizarse de forma manual.**

```
rd com\medicalsoft\servicios /s
```

7. Compile las clases contenidas en **com.medicalsoft.program.main** desde el directorio raíz de trabajo utilizando los .jar previamente creados: "**medicalsoftRRHHModel.jar**", "**medicalsoftInfraModel.jar**" y "**medicalsoftServModel.jar**".

```
javac -cp medicalsoftRRHHModel.jar;medicalsoftInfraModel.jar;medicalsoftServModel.jar; com/medicalsoft/program/main/*.java
```

8. Genere un componente .jar ejecutable llamado "**medicalsoftPrincipal.jar**" que contenga la estructura y clases en Bytecode contenidas en **com.medicalsoft.program.main**. Comprueba que funcione. Prosiga con la siguiente instrucción.

Como es un JAR ejecutable creamos un archivo MANIFEST.MF

```
Class-Path: medicalsoftRRHHModel.jar medicalsoftInfraModel.jar medicalsoftServModel.jar  
Main-Class: com/medicalsoft/program/main/Principal
```

```
jar cfm medicalsoftPrincipal.jar MANIFEST.MF com/medicalsoft/program/main/*.class
```

Comprobamos

```
java -jar medicalsoftPrincipal.jar
```

Respuesta final de la pregunta:

```
javac com/medicalsoft/rrhh/model/*.java
```

```
jar cf medicalsoftRRHHModel.jar com/medicalsoft/rrhh/model/*.class
```

```
javac -cp medicalsoftRRHHModel.jar; com/medicalsoft/infraestructura/model/*.java
```

```
jar cf medicalsoftInfraModel.jar com/medicalsoft/infraestructura/model/*.class
```

```
javac -cp medicalsoftRRHHModel.jar;medicalsoftInfraModel.jar; com/medicalsoft/servicios/model/*.java
```

```
jar cf medicalsoftServModel.jar com/medicalsoft/servicios/model/*.class
```

```
javac -cp  
medicalsoftRRHHModel.jar;medicalsoftInfraModel.jar;medicalsoftServModel.jar;  
com/medicalsoft/program/main/*.java
```

```
jar cfm medicalsoftPrincipal.jar MANIFEST.MF com/medicalsoft/program/main/*.class
```

Contenido del MANIFEST.MF:

```
Class-Path: medicalsoftRRHHModel.jar medicalsoftInfraModel.jar medicalsoftServModel.jar  
Main-Class: com/medicalsoft/program/main/Principal
```