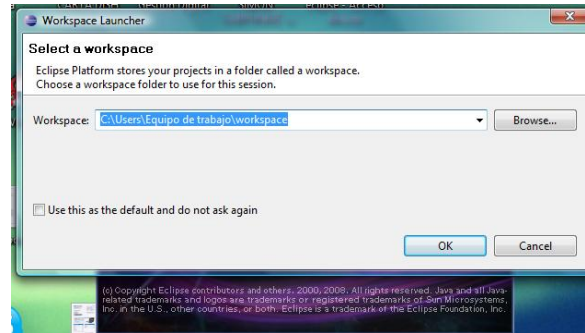


Conceptos básicos de Orientación a Objetos, manejo con Eclipse.

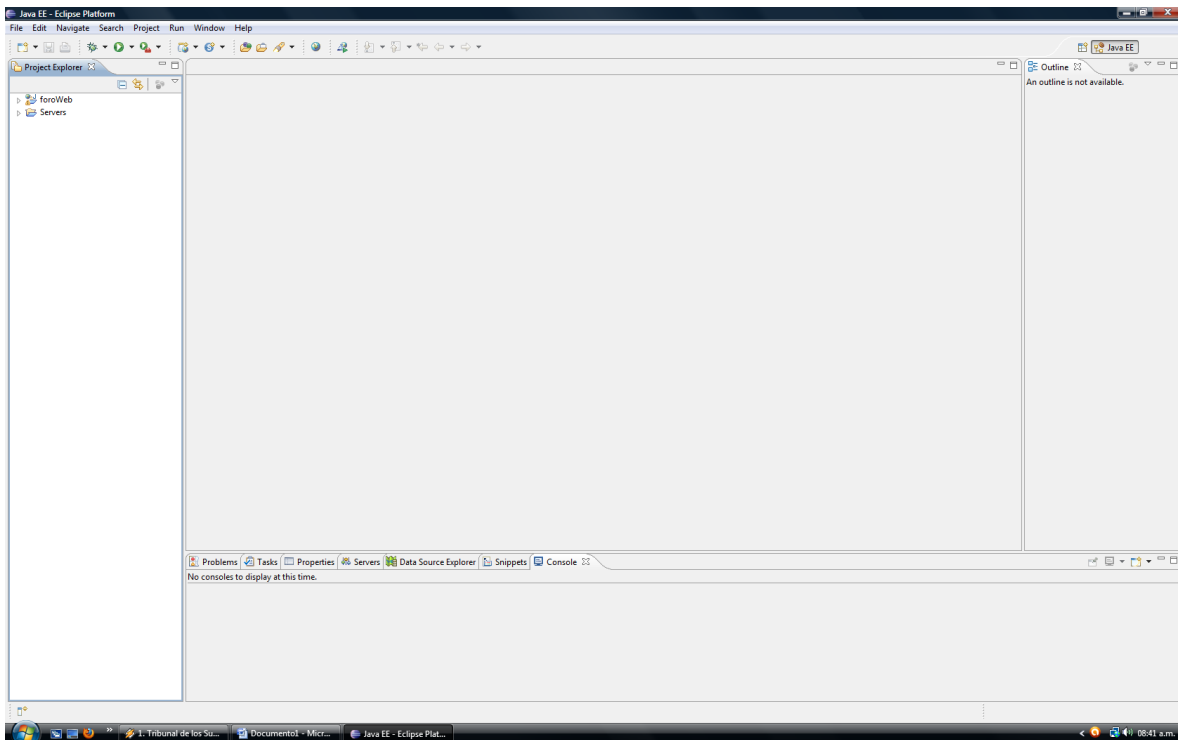
Arrancar Eclipse.

Después de descomprimir la aplicación, dar doble clic sobre el ícono ejecutable para arrancar la aplicación.



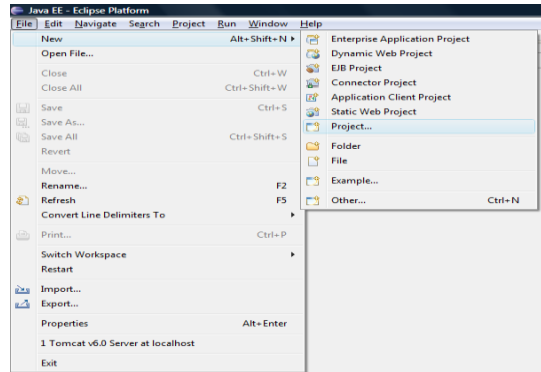
En caso de que se presente la pantalla de selección de workspace, seleccionar el que se desea utilizar y dar clic en OK.

Se desplegará la pantalla principal del entorno de desarrollo con los proyectos que se tengan en el workspace.

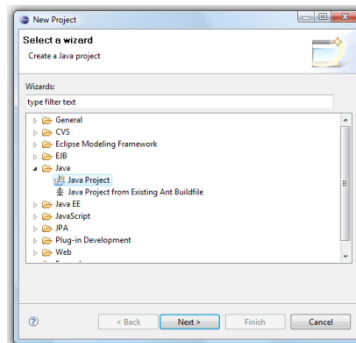


Crear un nuevo proyecto.

En la barra de menú principal, seleccionar File → New → Project

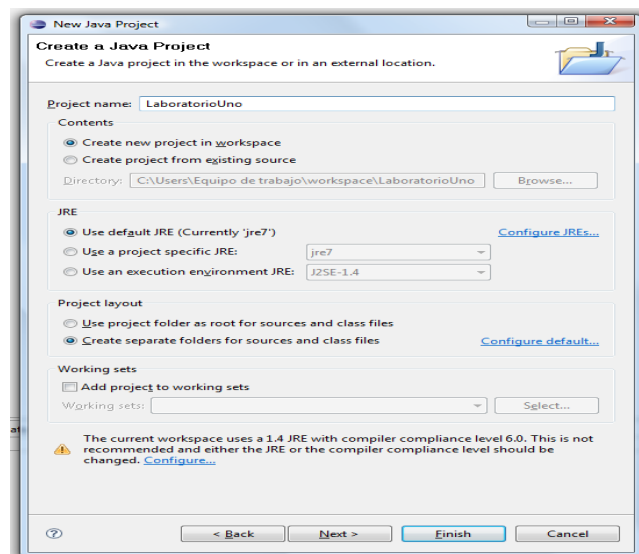


Seleccionar Java → Java Project

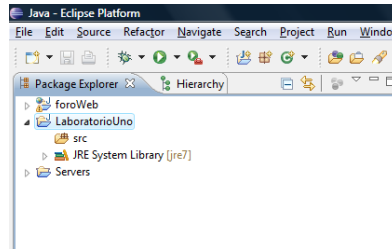


Clic Next >

En la siguiente pantalla, escribir el nombre del proyecto, en este caso LaboratorioUno, y dar clic en Next >



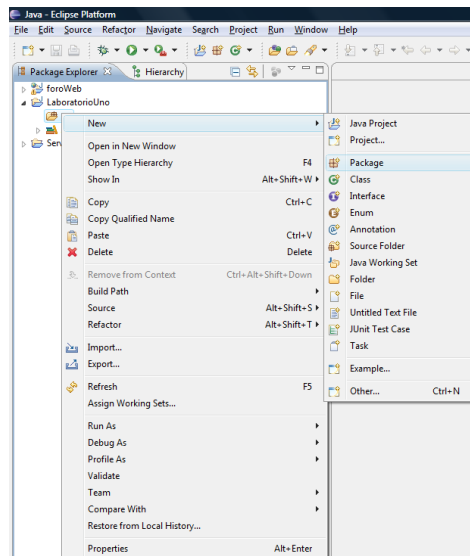
En la siguiente pantalla, dar clic en Finish, se habrá creado el proyecto.



Creación de paquetes.

Es conveniente crear paquetes para almacenar las clases que tengan funcionalidad en común, para crear un paquete, se deben seguir los siguientes pasos:

Clic derecho sobre el directorio *src* del proyecto en donde se esté trabajando, seleccionar New → Package.

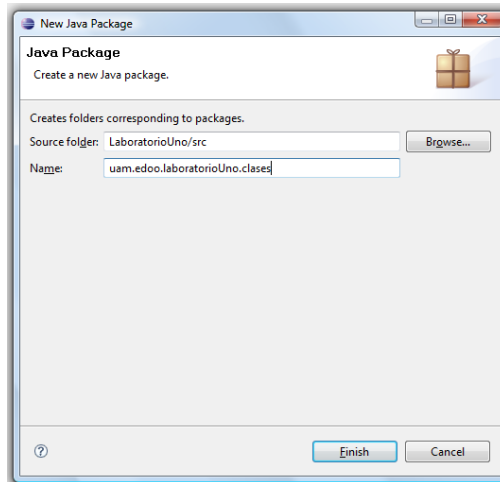


Escribir el nombre del paquete deseado, el nombre no puede contener espacios, puede contener puntos pero no al inicio ni al final del nombre, en este caso se utilizará como nombre de paquete base: `uam.edoo.laboratorioUno`, seguido de los identificadores del tipo de clases que se crearán en el paquete.

En este caso se crearán los paquetes:

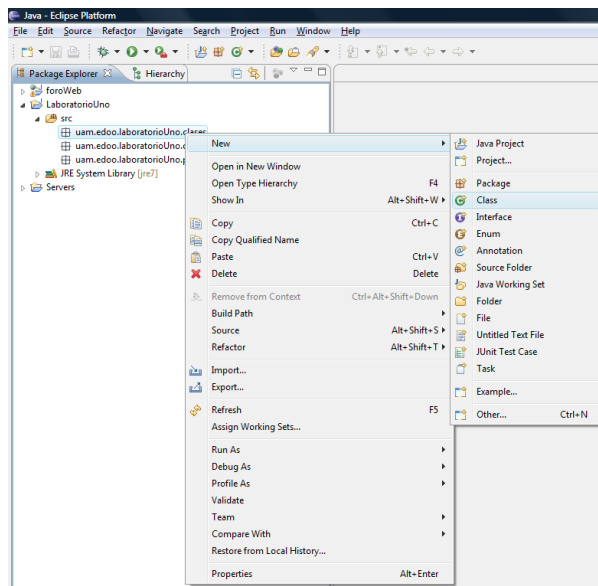
- `uam.edoo.laboratorioUno.clases`
- `uam.edoo.laboratorioUno.interfaces`
- `uam.edoo.laboratorioUno.operaciones`
- `uam.edoo.laboratorioUno.principal`

Escribir el nombre del paquete y dar clic en Finish, se habrá creado el paquete en el proyecto.

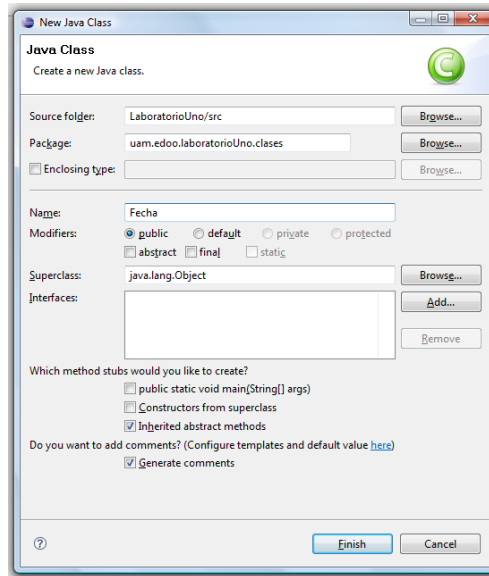


Creación de clases.

Para crear una clase, se debe seleccionar el paquete en donde se desea crear la clase, dar clic derecho → New → Class



En la siguiente pantalla, escribir el nombre de la clase, en este caso se creará una clase llamada Fecha.



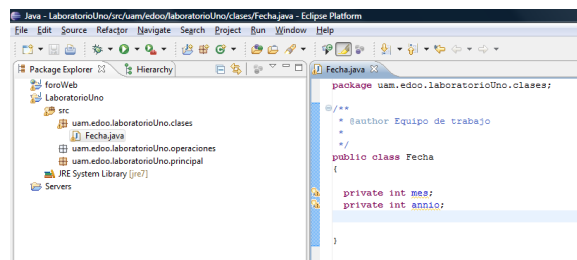
Clic en Finish, se habrá creado la clase en el paquete seleccionado.

Repetir los pasos anteriores para crear dentro del paquete de clases, las clases:

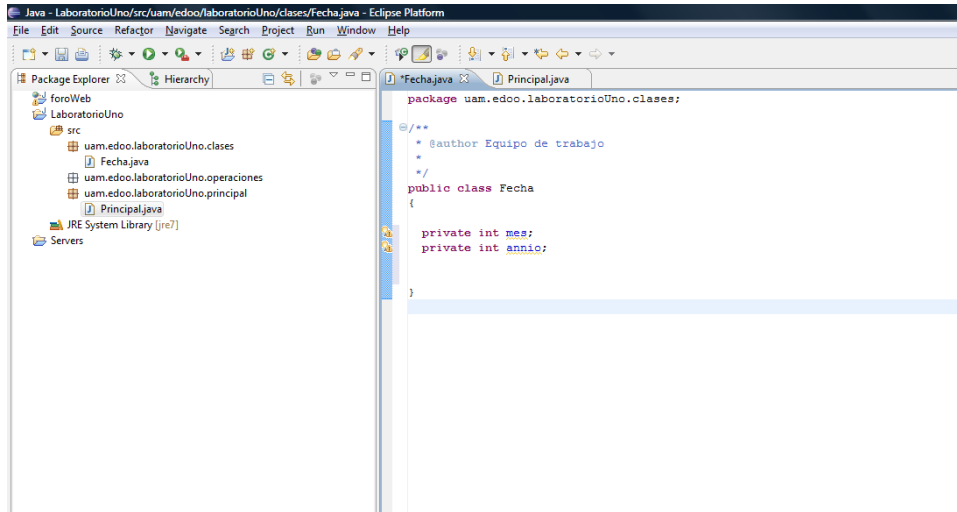
- Tarjeta
- TarjetaPelículas
- TarjetaVideoJuegos
- TarjetaCD

Creando atributos y métodos get y set para las clases.

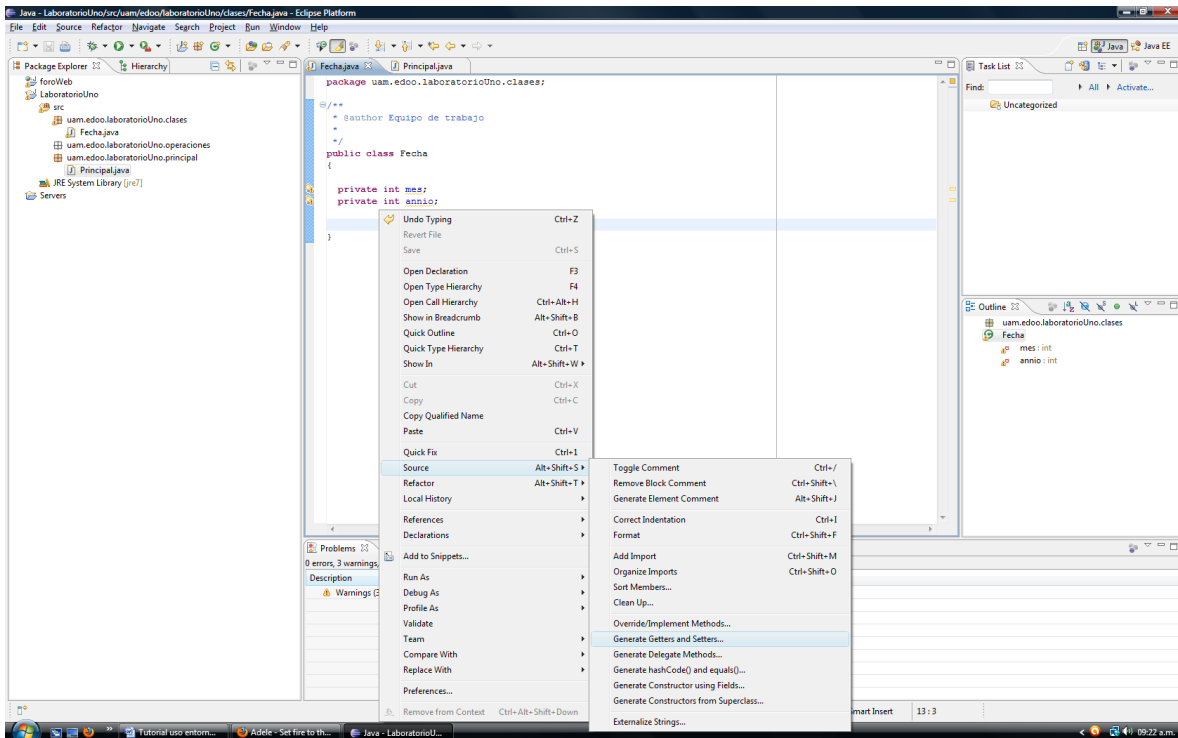
Crear dos atributos privados de tipo entero, uno llamado mes y otro llamado año en la clase Fecha.



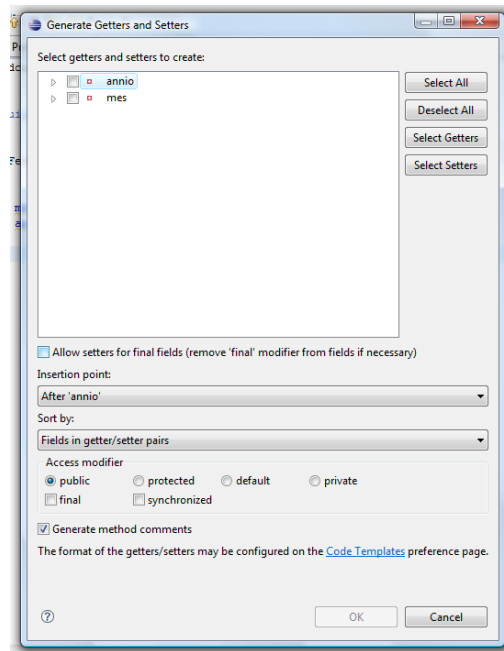
Es conveniente que los atributos de una clase sean accesibles a través de métodos get() y set(), ya que estos atributos han sido declarados como privados. A continuación se generarán atributos para la clase Fecha y se generarán sus métodos get() y set().



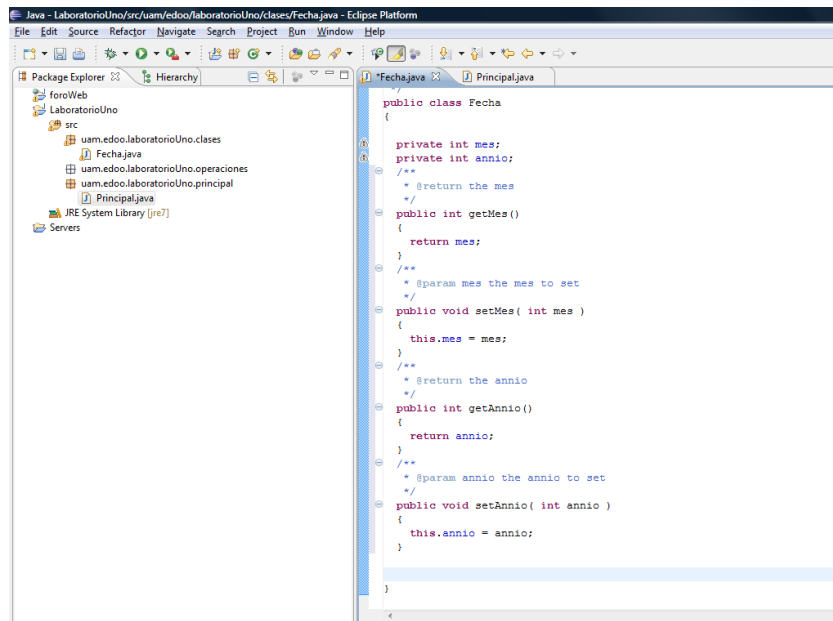
Para generar los métodos getters y setters, dar clic derecho sobre algún espacio en blanco que se encuentre dentro de la clase, seleccionar Source → Generate Getters and Setters.



Seleccionar aquellos atributos para los que se quieran generar los métodos get y set.



Se habrán generado los métodos get y set para los atributos seleccionados.



De esta forma, la clase Fecha queda de la siguiente manera:

Fecha

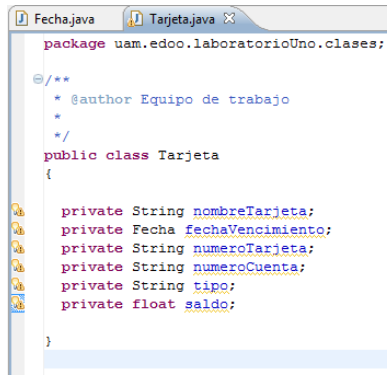
```
package uam.edoo.laboratorioUno.clases;

/**
 * @author Equipo de trabajo
 *
 */
public class Fecha
{

    private int mes;
    private int annio;
    /**
     * @return the mes
     */
    public int getMes()
    {
        return mes;
    }
    /**
     * @param mes the mes to set
     */
    public void setMes( int mes )
    {
        this.mes = mes;
    }
    /**
     * @return the annio
     */
    public int getAnnio()
    {
        return annio;
    }
    /**
     * @param annio the annio to set
     */
    public void setAnnio( int annio )
    {
        this.annio = annio;
    }

}
```

En la clase Tarjeta, crear los siguientes atributos y generar sus métodos get y set.



```
package uam.edoo.laboratorioUno.clases;

/**
 * @author Equipo de trabajo
 *
 */
public class Tarjeta
{
    private String nombreTarjeta;
    private Fecha fechaVencimiento;
    private String numeroTarjeta;
    private String numeroCuenta;
    private String tipo;
    private float saldo;
}
```

De esta forma, la clase Tarjeta quedará de la siguiente forma:

Tarjeta

```
package uam.edoo.laboratorioUno.clases;

/**
 * @author Equipo de trabajo
 *
 */
public class Tarjeta
{
    private String nombreTarjeta;
    private Fecha fechaVencimiento;
    private String numeroTarjeta;
    private String numeroCuenta;
    private String tipo;
    private float saldo;

    /**
     * @return the nombreTarjeta
     */
    public String getNombreTarjeta()
    {
        return nombreTarjeta;
    }

    /**
     * @param nombreTarjeta the nombreTarjeta to set
     */
    public void setNombreTarjeta( String nombreTarjeta )
    {
        this.nombreTarjeta = nombreTarjeta;
    }

    /**
     * @return the fechaVencimiento
     */
    public Fecha getFechaVencimiento()
    {
        return fechaVencimiento;
    }

    /**
     * @param fechaVencimiento the fechaVencimiento to set
     */
}
```

```

public void setFechaVencimiento( Fecha fechaVencimiento )
{
    this.fechaVencimiento = fechaVencimiento;
}
/**
 * @return the numeroTarjeta
 */
public String getNumeroTarjeta()
{
    return numeroTarjeta;
}
/**
 * @param numeroTarjeta the numeroTarjeta to set
 */
public void setNumeroTarjeta( String numeroTarjeta )
{
    this.numeroTarjeta = numeroTarjeta;
}
/**
 * @return the numeroCuenta
 */
public String getNumeroCuenta()
{
    return numeroCuenta;
}
/**
 * @param numeroCuenta the numeroCuenta to set
 */
public void setNumeroCuenta( String numeroCuenta )
{
    this.numeroCuenta = numeroCuenta;
}
/**
 * @return the tipo
 */
public String getTipo()
{
    return tipo;
}
/**
 * @param tipo the tipo to set
 */
public void setTipo( String tipo )
{
    this.tipo = tipo;
}
/**
 * @return the saldo
 */
public float getSaldo()
{
    return saldo;
}
/**
 * @param saldo the saldo to set
 */
public void setSaldo( float saldo )

```

```
{  
    this.saldo = saldo;  
}  
}
```

El método toString

El método toString se encuentra en la clase Object, de la cuál hereden todas las clases en Java de manera automática, este sirve para imprimir la información del objeto de manera más sencilla, a continuación se sobre escribirá el método toString de la clase Tarjeta, después de sobre escribir este método, la clase Tarjeta queda de la siguiente manera:

Tarjeta

```
package uam.edoo.laboratorioUno.clases;  
  
/**  
 * @author Equipo de trabajo  
 */  
public class Tarjeta  
{  
  
    private String nombreTarjeta;  
    private Fecha fechaVencimiento;  
    private String numeroTarjeta;  
    private String numeroCuenta;  
    private String tipo;  
    private float saldo;  
    /**  
     * @return the nombreTarjeta  
     */  
    public String getNombreTarjeta()  
    {  
        return nombreTarjeta;  
    }  
    /**  
     * @param nombreTarjeta the nombreTarjeta to set  
     */  
    public void setNombreTarjeta( String nombreTarjeta )  
    {  
        this.nombreTarjeta = nombreTarjeta;  
    }  
    /**  
     * @return the fechaVencimiento  
     */  
    public Fecha getFechaVencimiento()  
    {  
        return fechaVencimiento;  
    }  
    /**  
     * @param fechaVencimiento the fechaVencimiento to set
```

```

*/
public void setFechaVencimiento( Fecha fechaVencimiento )
{
    this.fechaVencimiento = fechaVencimiento;
}
/**
 * @return the numeroTarjeta
 */
public String getNumeroTarjeta()
{
    return numeroTarjeta;
}
/**
 * @param numeroTarjeta the numeroTarjeta to set
 */
public void setNumeroTarjeta( String numeroTarjeta )
{
    this.numeroTarjeta = numeroTarjeta;
}
/**
 * @return the numeroCuenta
 */
public String getNumeroCuenta()
{
    return numeroCuenta;
}
/**
 * @param numeroCuenta the numeroCuenta to set
 */
public void setNumeroCuenta( String numeroCuenta )
{
    this.numeroCuenta = numeroCuenta;
}
/**
 * @return the tipo
 */
public String getTipo()
{
    return tipo;
}
/**
 * @param tipo the tipo to set
 */
public void setTipo( String tipo )
{
    this.tipo = tipo;
}
/**
 * @return the saldo
 */
public float getSaldo()
{
    return saldo;
}
/**
 * @param saldo the saldo to set
 */

```

```

public void setSaldo( float saldo )
{
    this.saldo = saldo;
}

/**
 * Se sobre escribe el método toString para que regrese una cadena
 * (String) con la información que se desea se imprima en pantalla.
 * El símbolo + se utiliza para ir concatenando los valores a la cadena
 * a regresar, la fecha de vencimiento, al ser un objeto se imprime
 * a través del valor de sus atributos (mes y año)
 */
public String toString() {
    String mensaje = "";
    mensaje = "Titular: " + this.nombreTarjeta + "\n"+
        "Número de tarjeta: " + this.numeroTarjeta + "\n"+
        "Número de cuenta: " + this.numeroCuenta + "\n"+
        "Tipo: " + this.tipo + "\n"+
        "Fecha de vencimiento: " + fechaVencimiento.getMes() +
        "/" + fechaVencimiento.getAnio();
    return mensaje;
}
}

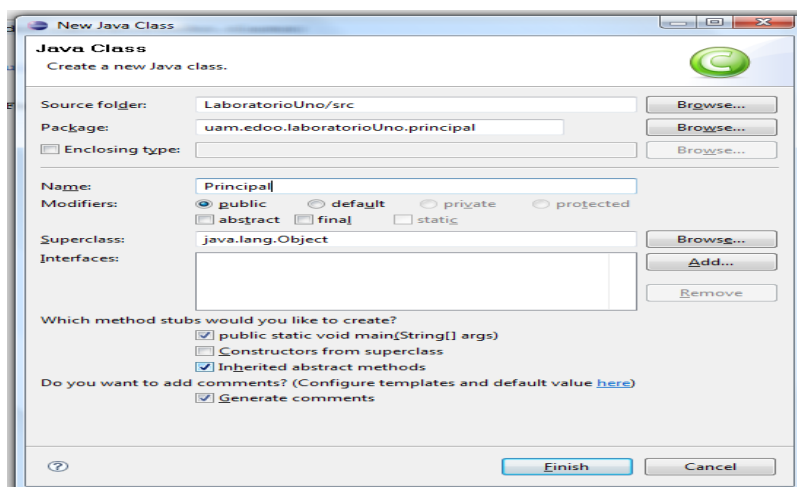
```

Creación de clases con método main.

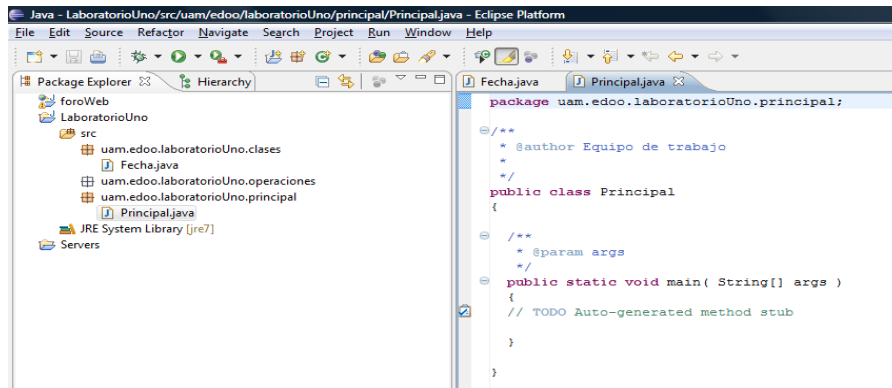
Para ejecutar la aplicación, se debe tener al menos una clase con un método principal (main), para crear esta clase se siguen los mismos pasos que para cualquier otra clase.

Clic derecho sobre el paquete → New → Class

Escribir el nombre de la clase, su utilizará Principal, y activar el checkbox que indica que se debe de crear el método *public static void main(String[] args)*.



Dar clic en Finish, se habrá creado la clase con el método main().



Imprimiendo en consola.

Para probar de manera inmediata la aplicación, se imprimirá un mensaje en la consola, en este caso “Ejecutando mi primera aplicación”, para imprimir en consola, se utiliza la instrucción

System.out.print() o System.out.println(). Colocando entre el paréntesis lo que se desea que se imprima, que puede ser un mensaje, una variable, atributos de un objeto o la unión de varios de estos que se pueden concatenar con el operador +.

Por el momento solo se imprimirá un mensaje, por lo tanto la clase Principal queda de la siguiente manera:

Principal

```
package uam.edoo.laboratorioUno.principal;

/**
 * @author Equipo de trabajo
 *
 */
public class Principal
{

    /**
     * @param args
     */
    public static void main( String[] args )
    {

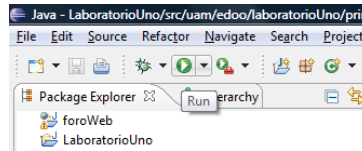
        System.out.println("Ejecutando mi primera aplicación");

    }

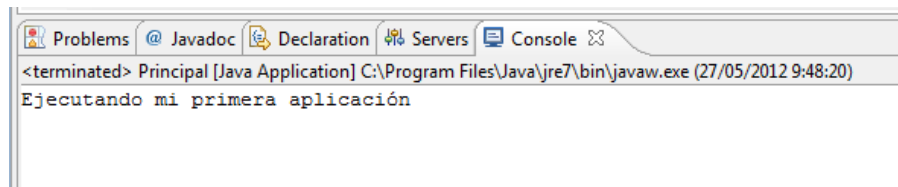
}
```

Ejecutando una aplicación en Eclipse.

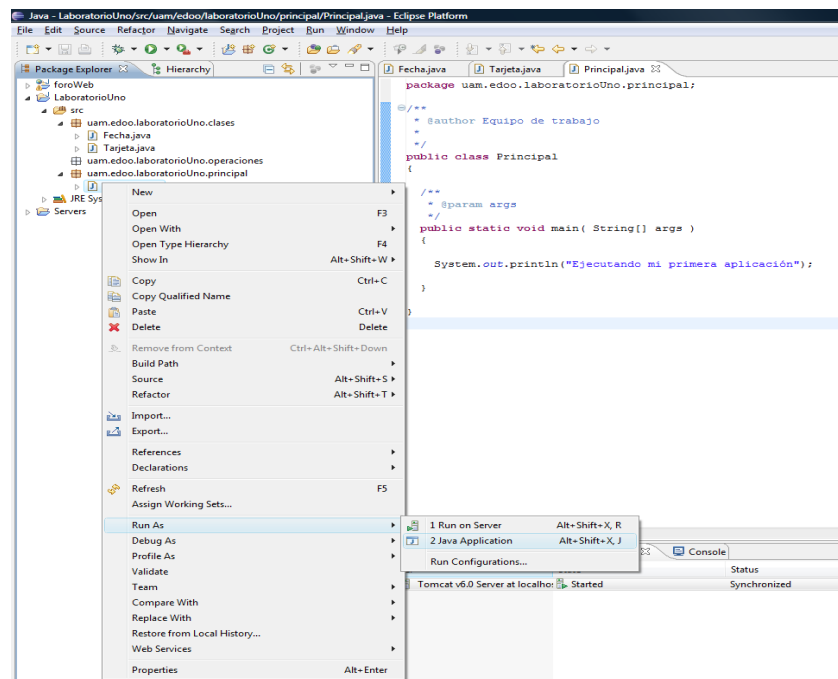
Para ejecutar una aplicación, se debe haber creado una clase que tenga el método main(), una forma de correr la aplicación es: dar clic sobre el nombre del proyecto y dar clic en el ícono de Run.



El resultado aparecerá en la pestaña Console.



La otra opción es dar clic derecho sobre la clase que contiene el método main(), seleccionar Run as y seleccionar Java Application.



Llenando un objeto.

En la clase Principal, se creará un método llamado llenarDatos() el cuál regresará un objeto de tipo Tarjeta con sus atributos llenos.

Para llenar los atributos de un objeto, se debe hacer uso de sus métodos set(), en caso de que el atributo sea un objeto de otra clase, se debe instanciar un objeto de ese tipo, llenarlo y asignarlo al atributo con el método set() correspondiente.

Una vez llenado, se imprimirán sus valores con ayuda del método toString, de esta forma la clase Principal queda de la siguiente manera:

Principal

```
package uam.edoo.laboratorioUno.principal;

import uam.edoo.laboratorioUno.clases.Fecha;
import uam.edoo.laboratorioUno.clases.Tarjeta;

/**
 * @author Equipo de trabajo
 *
 */
public class Principal
{
    /**
     * @param args
     */
    public static void main( String[] args )
    {
        //Se crea un objeto tipo Tarjeta que se llenará a partir del método
        //llenarDatos(), como se llena en otro método no es necesario
        //instanciarlo
        Tarjeta tarjeta;
        //Se manda a llamar el método que llena la tarjeta
        tarjeta = llenarDatos();
        //Se imprimen los datos usando con ayuda del método toString
        System.out.println(tarjeta.toString());
    }

    /**
     * Método que regresa un objeto tipo Tarjeta con
     * sus atributos llenos.
     * @return
     */
    public static Tarjeta llenarDatos() {
        Tarjeta tarjeta = new Tarjeta();
        Fecha fechaVencimiento = new Fecha();

        tarjeta.setNombreTarjeta( "Nombre del titular" );
    }
}
```



```

tarjeta.setNumeroCuenta( "123456" );
tarjeta.setNumeroTarjeta( "123 456 789" );
tarjeta.setTipo( "General" );
//Para indicar un tipo flotante, se coloca la letra F después del
valor
tarjeta.setSaldo( 100.50F );

//Se llena el atributo mes del objeto Fecha
fechaVencimiento.setMes( 11 );
//Se llena el atributo año del objeto Fecha
fechaVencimiento.setAño( 18 );

//Se asigna el objeto al atributo tipo Fecha de la tarjeta creada
tarjeta.setFechaVencimiento( fechaVencimiento );
return tarjeta;
}
}

```

Al ejecutar la aplicación se mostrará lo siguiente en consola.

```

<terminated> Principal [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (02/06/2012 13:13:41)
Titular: Nombre del titular
Número de tarjeta: 123 456 789
Número de cuenta: 123456
Tipo: General
Fecha de vencimiento: 11/18

```

Llenando datos desde teclado.

Se creará un método llamado desdeTeclado() el cuál solicitará datos al usuario y con ellos llenará los atributos necesarios.

Adicionalmente se creará un método llamado leerDatos() el cuál se encargará de leer desde teclado un dato como una cadena (String) y regresará el valor leído.

De esta forma, la clase Principal queda de la siguiente manera:

Principal

```

package uam.edoo.laboratorioUno.principal;

import java.io.*;

import uam.edoo.laboratorioUno.clases.Fecha;
import uam.edoo.laboratorioUno.clases.Tarjeta;

/**
 * @author Equipo de trabajo
 *
 */

```

```

public class Principal
{
    /**
     * @param args
     */
    public static void main( String[] args )
    {
        //Se crea un objeto tipo Tarjeta que se llenará a partir del método
        //llenarDatos(), como se llena en otro método no es necesario
        //instanciarlo
        Tarjeta tarjeta;
        //Se manda a llamar el método que llena la tarjeta
        //tarjeta = llenarDatos();

        //Se manda a llamar el método que llena la tarjeta
        //desde teclado
        tarjeta = desdeTeclado();

        //Se imprimen los datos usando con ayuda del método toString
        System.out.println(tarjeta.toString());
    }

    /**
     * Método que regresa un objeto tipo Tarjeta con
     * sus atributos llenos.
     * @return
     */
    public static Tarjeta llenarDatos() {
        Tarjeta tarjeta = new Tarjeta();
        Fecha fechaVencimiento = new Fecha();

        tarjeta.setNombreTarjeta( "Nombre del titular" );
        tarjeta.setNumeroCuenta( "123456" );
        tarjeta.setNumeroTarjeta( "123 456 789" );
        tarjeta.setTipo( "General" );
        //Para indicar un tipo flotante, se coloca la letra F después del
        valor
        tarjeta.setSaldo( 100.50F );

        //Se llena el atributo mes del objeto Fecha
        fechaVencimiento.setMes( 11 );
        //Se llena el atributo año del objeto Fecha
        fechaVencimiento.setAño( 18 );

        //Se asigna el objeto al atributo tipo Fecha de la tarjeta creada
        tarjeta.setFechaVencimiento( fechaVencimiento );
        return tarjeta;
    }

    /**
     * Método que crea y llena un objeto tipo
     * Tarjeta a través de valores leídos desde
     * teclado.

```

```

* @return
*/
public static Tarjeta desdeTeclado() {
    Tarjeta tarjeta = new Tarjeta();
    Fecha fechaVencimiento = new Fecha();

    //Variables que recibe el valor leído
    //desde teclado a partir del método
    //leerDato()
    String dato = "";
    String numeroTarjeta = "";

    System.out.println("Nombre del cliente:");
    dato = leerDato();
    tarjeta.setNombreTarjeta(dato);

    System.out.println("Numero de cuenta:");
    dato = leerDato();
    tarjeta.setNumeroCuenta(dato);

    //En este caso la lectura se hace usando
    //la variable numeroTarjeta
    System.out.println("Numero de tarjeta:");
    numeroTarjeta = leerDato();
    tarjeta.setNumeroTarjeta(numeroTarjeta);

    System.out.println("Tipo:");
    dato = leerDato();
    tarjeta.setTipo(dato);

    //Cuando se leen número es necesario
    //parsear (convertir) la cadena leída
    //a su equivalente numérico, en este caso
    //con la clase Float y su método parseFloat
    System.out.println("Saldo de la tarjeta:");
    dato = leerDato();
    tarjeta.setSaldo(Float.parseFloat(dato));

    //Para convertir cadenas a enteros, se utiliza
    //Integer.parseInt
    System.out.println("Mes de vencimiento:");
    dato = leerDato();
    fechaVencimiento.setMes(Integer.parseInt(dato));

    System.out.println("Año de vencimiento:");
    dato = leerDato();
    fechaVencimiento.setAño(Integer.parseInt(dato));

    tarjeta.setFechaVencimiento(fechaVencimiento);

    return tarjeta;
}
/**
 * Método que lee desde teclado y lo asigna a un String
 * para que sea regresado, se deben utilizar las clases
 * InputStreamReader y BufferedReader para realizar la
 * lectura

```

```

    * @return
    */
    public static String leerDato() {
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader entrada = new BufferedReader(input);
        String datoLeido="";

        try {
            datoLeido = entrada.readLine();
        } catch (IOException e) {
            System.out.println("ERROR DESDE TECLADO");
            e.printStackTrace();
        }

        return datoLeido;
    }
}

```

Al ejecutar la aplicación, se solicitarán los datos al usuario:

```

Principal [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (02/06/2012 13:37:31)
Nombre del cliente:
Nombre del cliente
Numero de cuenta:
Número de cuenta

```

Llenando los datos desde otra clase.

Lo más recomendable es no tener muchos métodos en la clase Principal, se creará una clase para llenar la información de una tarjeta, en este caso se creará la clase LlenarDatos en el paquete *uam.edoo.laboratorioUno.operaciones*.

Se copiarán ahí los métodos `llenarDatos()`, `desdeTeclado()`, `leerDato()`, como ya no serán llamados directamente desde el método `main()`, es decir estando en la misma clase, se les puede eliminar el atributo `static`.

De esta forma, la clase `LlenarDatos` quedará de la siguiente manera:

LlenarDatos

```

package uam.edoo.laboratorioUno.operaciones;

import uam.edoo.laboratorioUno.clases.Fecha;
import uam.edoo.laboratorioUno.clases.Tarjeta;

import java.io.*;

```

```

/**
 * @author Equipo de trabajo
 *
 */
public class LlenarDatos
{
    /**
     * Método que regresa un objeto tipo Tarjeta con
     * sus atributos llenos.
     * @return
     */
    public Tarjeta llenarDatos() {
        Tarjeta tarjeta = new Tarjeta();
        Fecha fechaVencimiento = new Fecha();

        tarjeta.setNombreTarjeta( "Nombre del titular" );
        tarjeta.setNumeroCuenta( "123456" );
        tarjeta.setNumeroTarjeta( "123 456 789" );
        tarjeta.setTipo( "General" );
        //Para indicar un tipo flotante, se coloca la letra F después del
valor
        tarjeta.setSaldo( 100.50F );

        //Se llena el atributo mes del objeto Fecha
        fechaVencimiento.setMes( 11 );
        //Se llena el atributo año del objeto Fecha
        fechaVencimiento.setAño( 18 );

        //Se asigna el objeto al atributo tipo Fecha de la tarjeta creada
        tarjeta.setFechaVencimiento( fechaVencimiento );
        return tarjeta;
    }
    /**
     * Método que crea y llena un objeto tipo
     * Tarjeta a través de valores leídos desde
     * teclado.
     * @return
     */
    public Tarjeta desdeTeclado() {
        Tarjeta tarjeta = new Tarjeta();
        Fecha fechaVencimiento = new Fecha();

        //Variables que recibe el valor leído
        //desde teclado a partir del método
        //leerDato()
        String dato = "";
        String numeroTarjeta = "";

        System.out.println("Nombre del cliente:");
        dato = leerDato();
        tarjeta.setNombreTarjeta(dato);

        System.out.println("Número de cuenta:");
        dato = leerDato();
    }
}

```

```

tarjeta.setNumeroCuenta(dato);

//En este caso la lectura se hace usando
//la variable numeroTarjeta
System.out.println("Numero de tarjeta:");
numeroTarjeta = leerDato();
tarjeta.setNumeroTarjeta(numeroTarjeta);

System.out.println("Tipo:");
dato = leerDato();
tarjeta.setTipo(dato);

//Cuando se leen número es necesario
//parsear (convertir) la cadena leida
//a su equivalente numérico, en este caso
//con la clase Float y su método parseFloat
System.out.println("Saldo de la tarjeta:");
dato = leerDato();
tarjeta.setSaldo(Float.parseFloat(dato));

//Para convertir cadenas a enteros, se utiliza
//Integer.parseInt
System.out.println("Mes de vencimiento:");
dato = leerDato();
fechaVencimiento.setMes(Integer.parseInt(dato));

System.out.println("Año de vencimiento:");
dato = leerDato();
fechaVencimiento.setAnio(Integer.parseInt(dato));

tarjeta.setFechaVencimiento(fechaVencimiento);

return tarjeta;
}
/**
 * Método que lee desde teclado y lo asigna a un String
 * para que sea regresado, se deben utilizar las clases
 * InputStreamReader y BufferedReader para realizar la
 * lectura
 * @return
 */
public String leerDato() {
    InputStreamReader input = new InputStreamReader(System.in);
    BufferedReader entrada = new BufferedReader(input);
    String datoLeido="";

    try {
        datoLeido = entrada.readLine();
    } catch (IOException e) {
        System.out.println("ERROR DESDE TECLADO");
        e.printStackTrace();
    }

    return datoLeido;
}
}

```

Ahora desde el método main() de la clase Principal, se debe invocar a los métodos correspondientes de la clase LlenarDatos de acuerdo a lo que se quiera realizar, para esto es necesario primero instanciar un objeto de tipo LlenarDatos, de esta forma la clase Principal ahora queda:

Principal

```
package uam.edoo.laboratorioUno.principal;

import java.io.*;

import uam.edoo.laboratorioUno.clases.Fecha;
import uam.edoo.laboratorioUno.clases.Tarjeta;
import uam.edoo.laboratorioUno.operaciones.LlenarDatos;

/**
 * @author Equipo de trabajo
 *
 */
public class Principal
{
    /**
     * @param args
     */
    public static void main( String[] args )
    {
        //Se crea un objeto tipo Tarjeta que se llenará a partir del método
        //llenarDatos(), como se llena en otro método no es necesario
        //instanciarlo
        Tarjeta tarjeta;
        //Se instancia un objeto del tipo LlenarDatos

        LlenarDatos llenarDatos = new LlenarDatos();
        //Se manda a llamar el método que llena la tarjeta que se
        //encuentra en la clase LlenarDatos
        tarjeta = llenarDatos.llenarDatos();
        System.out.println(tarjeta.toString());

        //Se manda a llamar el método que llena la tarjeta
        //desde teclado que se encuentra en la clase LlenarDatos
        tarjeta = llenarDatos.desdeTeclado();

        //Se imprimen los datos usando con ayuda del método toString
        System.out.println(tarjeta.toString());
    }
}
```