

# Escribiendo las clases de la aplicación

```
package edu.ehu.GB.MIF.nasdaqGame;
```

```
import java.io.*;
import java.net.*;
import java.util.*;
import java.util.regex.*;
```

```
/**
 *
 * @author German
 */
```

```
public final class Nasdaq {
    private static final String[] tickers={"IBM","AAPL","MSFT","ORCL","HPQ","ADBE","INTC","GOOG","EBAY","YHOO"};
    private static final int MESES=2;
    private static final Map<String, Map<String,Double>> cotizaciones=new TreeMap<>();
```

```
public Nasdaq(int meses) throws IOException {
    Map<String, Double> datos;
    for (String t:tickers) {
        cotizaciones.put(t, datos=new TreeMap<>());
        for (Stock s:findStoks(t,meses)) {
            datos.put(s.date+"_open", s.open);
            datos.put(s.date+"_close", s.close);
        }
    }
}
```

```
//Copiadas de Mikel
```

```
private static class Stock {...31 lines }
private static class WebPage {...23 lines }
private static String getText(InputStream in) throws IOException {...10 lines }
private static WebPage downloadWebPage(String address) throws MalformedURLException, IOException {...12 lines }
private static Stock[] findStoks(String ticker, int months) throws MalformedURLException, IOException {...23 lines }
```

M

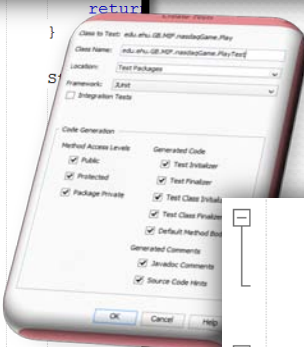
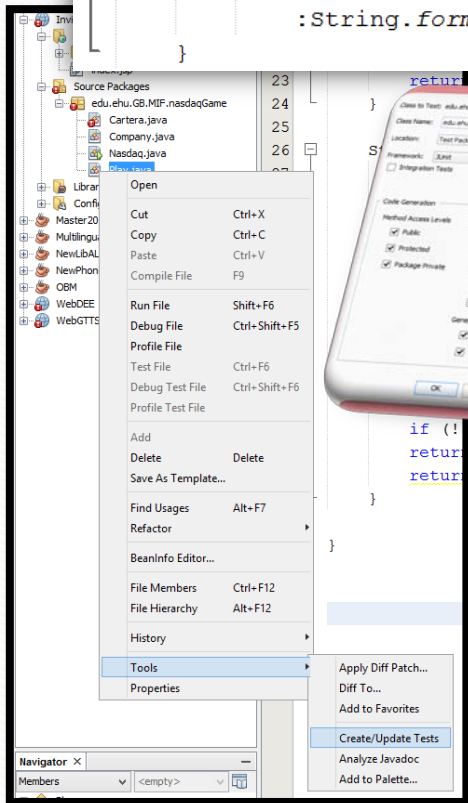
¿MVC?

VC

# Escribiendo las clases de la aplicación

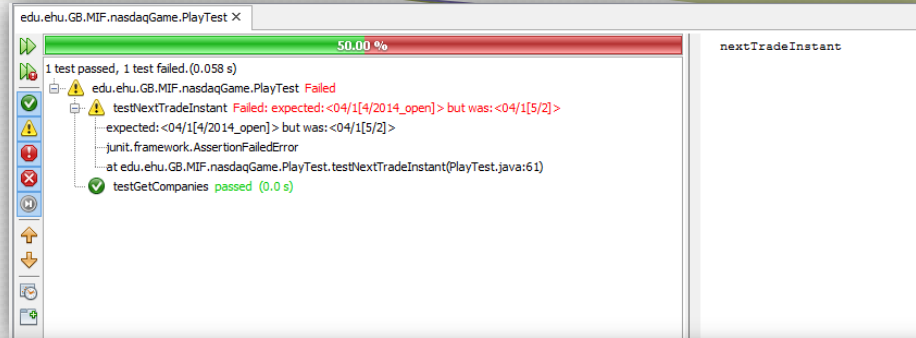
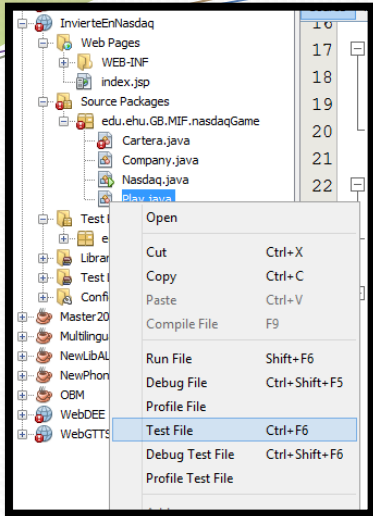
```
static String nextTradeInstant(String tradeInstant) {
    if (tradeInstant.endsWith("open")) return tradeInstant.replace("open", "close");
    tradeInstant=tradeInstant.replace("close", "open");
    int mes=Integer.parseInt(tradeInstant.substring(0,2));
    int dia=Integer.parseInt(tradeInstant.substring(3,5));
    int año=Integer.parseInt(tradeInstant.substring(6,10));
    int ultimoDia=0;
    switch (mes) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12: ultimoDia=31; break;
        case 2: ultimoDia=((año%4==0 && !(año%100==0)) || año%400==0)?29:28; break;
        case 4: case 6: case 9: case 11: ultimoDia=30;
    }
    return (mes!=12 || dia<31)
        ?String.format("%02d/%02d"+tradeInstant.substring(5),dia==ultimoDia?mes+1:mes,dia==ultimoDia?1:dia+1)
        :String.format("01/01/%02d"+tradeInstant.substring(10),año+1);
}
```

Ojo! no es robusta frente a entradas incorrectas



```
/**
 * Test of nextTradeInstant method, of class Play.
 */
@Test
public void testNextTradeInstant() {
    System.out.println("nextTradeInstant");
    String tradeInstant = "";
    String expectedResult = "";
    String result = Play.nextTradeInstant(tradeInstant);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    fail("The test case is a prototype.");
}
```

# Escribiendo las clases de la aplicación (tests)



```
/**
 * Test of nextTradeInstant method, of class Play.
 */
@Test
public void testNextTradeInstant() {
    System.out.println("nextTradeInstant");
    String tradeInstant = "04/13/2014_open";
    String expectedResult = "04/13/2014_close";
    String result = Play.nextTradeInstant(tradeInstant);
    assertEquals(expectedResult, result);
    assertEquals("04/15/2014_open", Play.nextTradeInstant("04/14/2014_close"));
    assertEquals("05/01/2014_open", Play.nextTradeInstant("04/30/2014_close"));
    assertEquals("02/29/2012_open", Play.nextTradeInstant("02/28/2012_close"));
    assertEquals("03/01/2015_open", Play.nextTradeInstant("02/28/2015_close"));
    assertEquals("01/01/2015_open", Play.nextTradeInstant("12/31/2014_close"));
}
```

