

Evidencia: Ocultamiento de datos

Objetivo: aplicar los conceptos de clases e instanciación de objetos con LDP Java, usando el ocultamiento de datos.

Modalidad: trabajo en grupo (2 ó 3 personas)

Nombre de los participantes:

- Sophia Torres
- Yasmin Hernández

Preguntas a responder:

Pgta 1. ¿Desde dónde es posible acceder a la variable time?

1. Cualquier otra clase
2. java.util package
3. myUtilities package
4. La clase TodaysDate
5. La clase TodaysDate y sus sub clase

Pgta 2. ¿Quiénes pueden acceder al atributo day?

-day es accesible desde cualquier otra clase en cualquier paquete.

Pgta 3. ¿Qué atributos de la clase tienen el método de acceso más restrictivo?

1. day
2. month
3. time
4. year

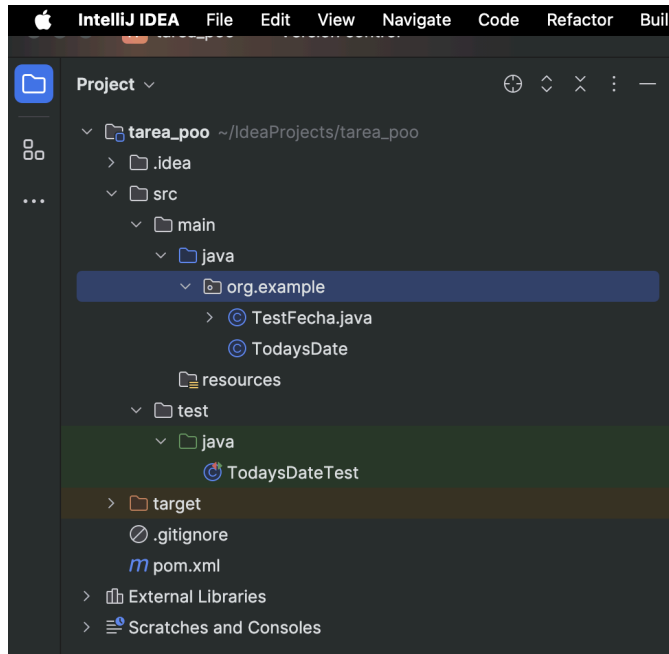
Pgta 4. ¿Desde dónde se puede acceder al atributo year?

1. Sólo dentro del package myUtilities.
2. Sub clases de TodaysDate en cualquier package
3. La clase TodaysDate
4. Sub clases de TodaysDate

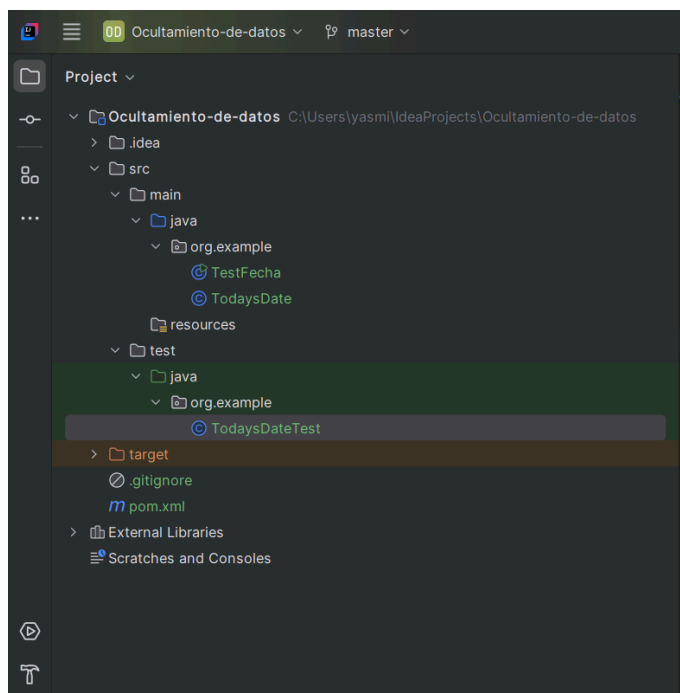
Evidencia fotográfica:

la estructura de su proyecto en su IDE y escritorio local.

Sophia:

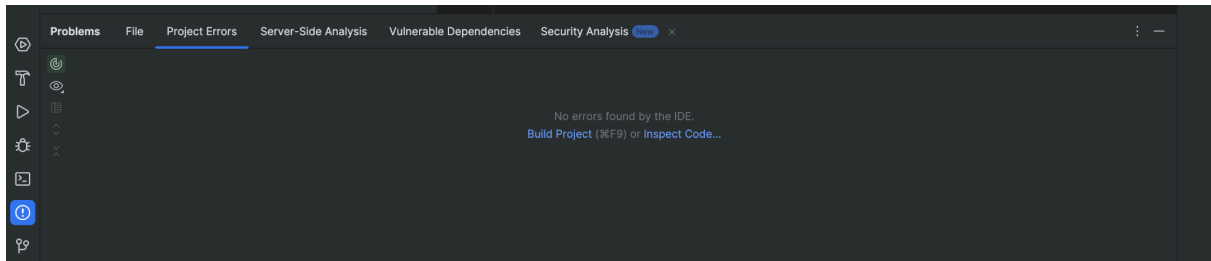


Yasmin:

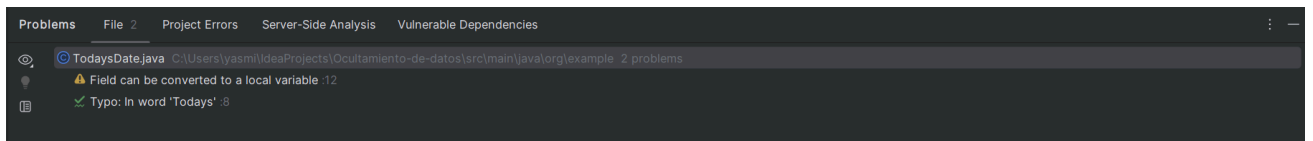


una compilación exitosa en su IDE.

Sophia:

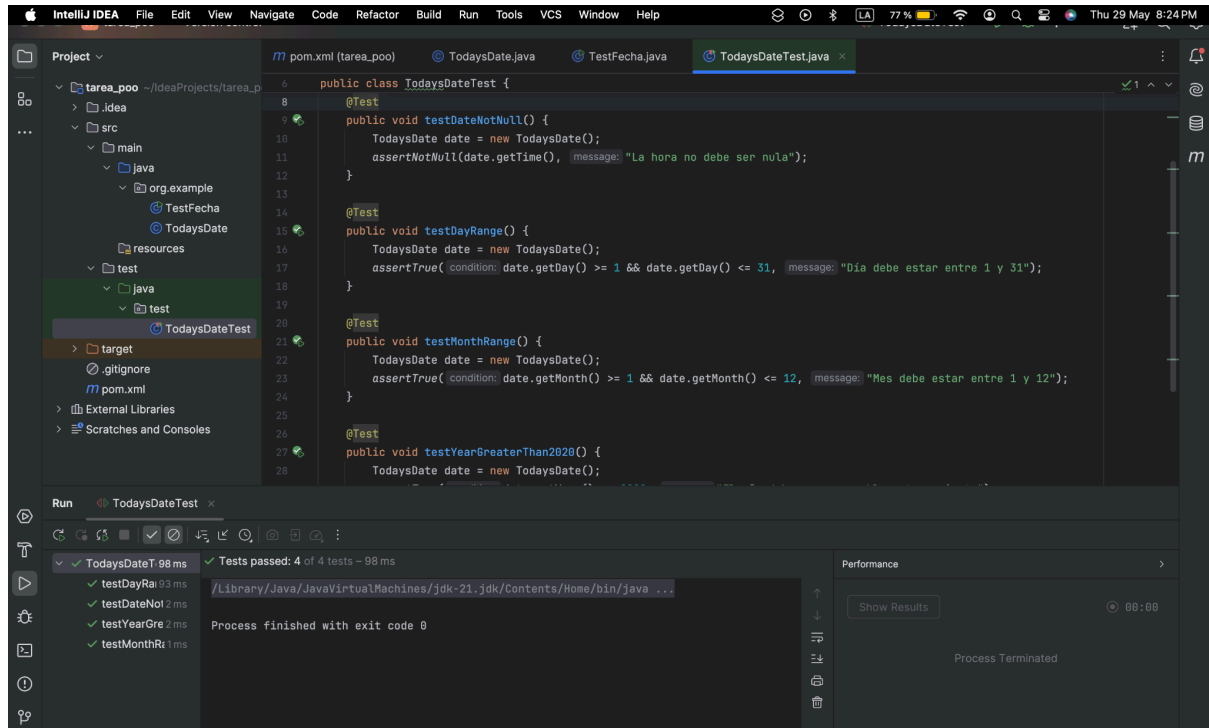


Yasmin:

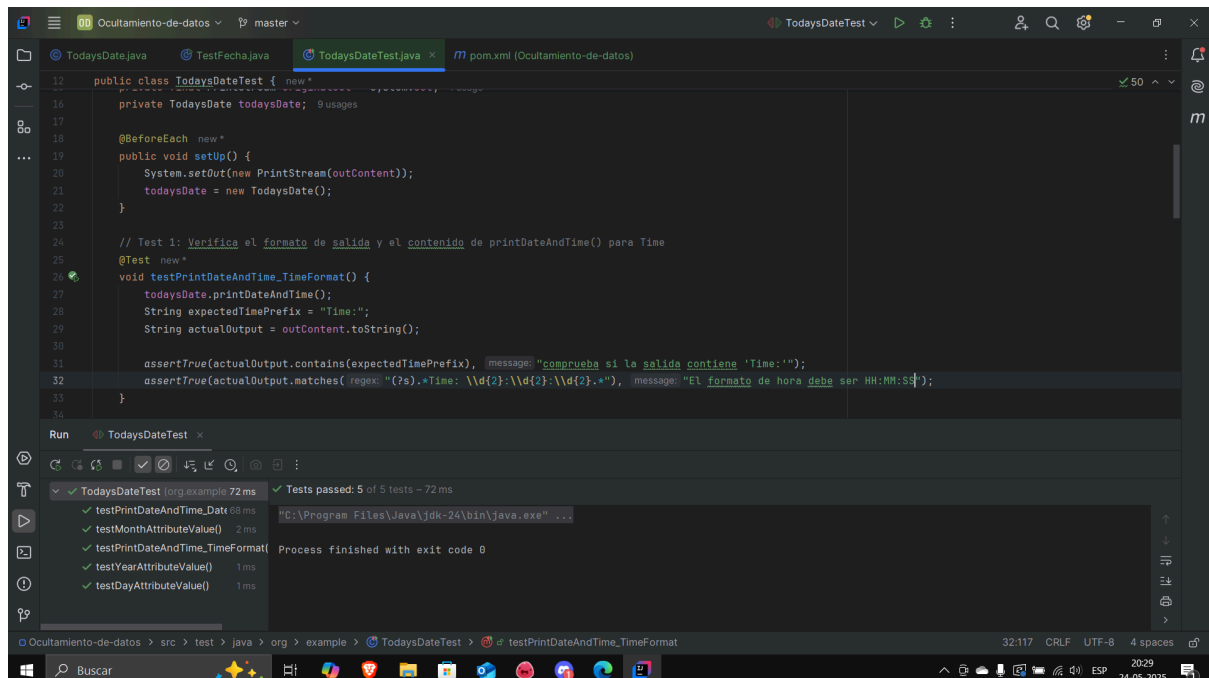


Una ejecución de pruebas unitarias en su IDE.

Sophia:

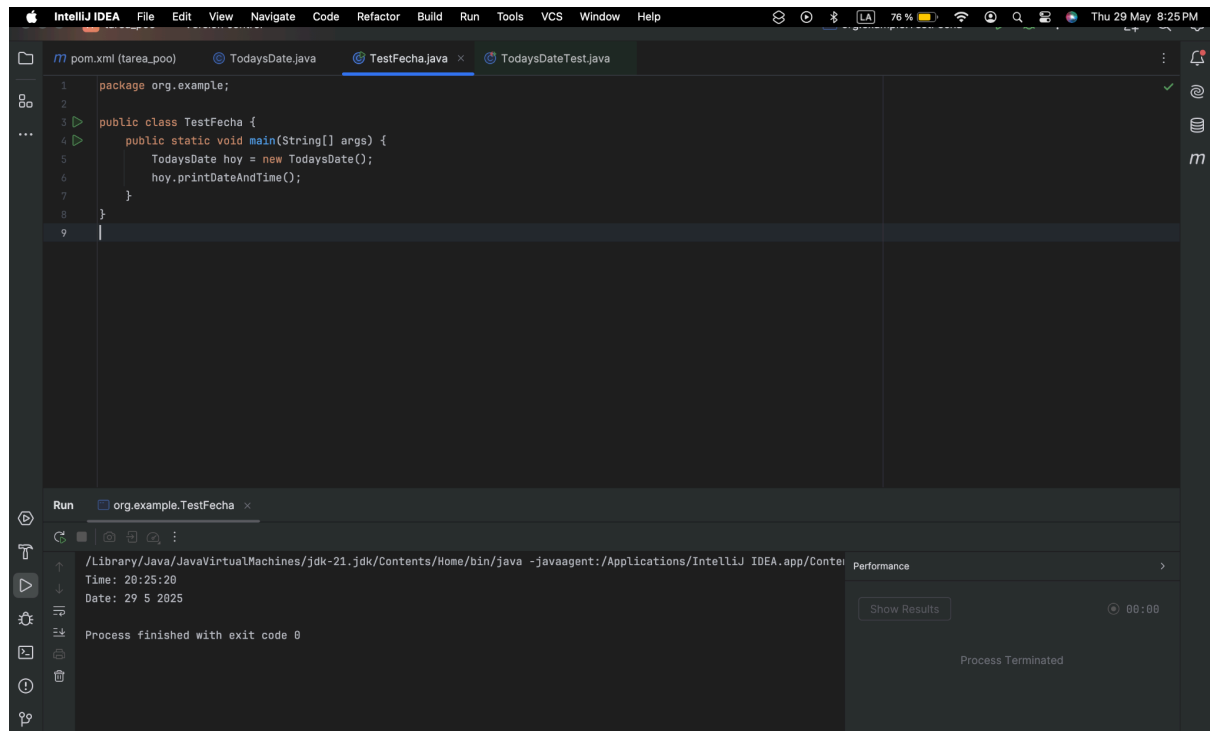


Yasmin:

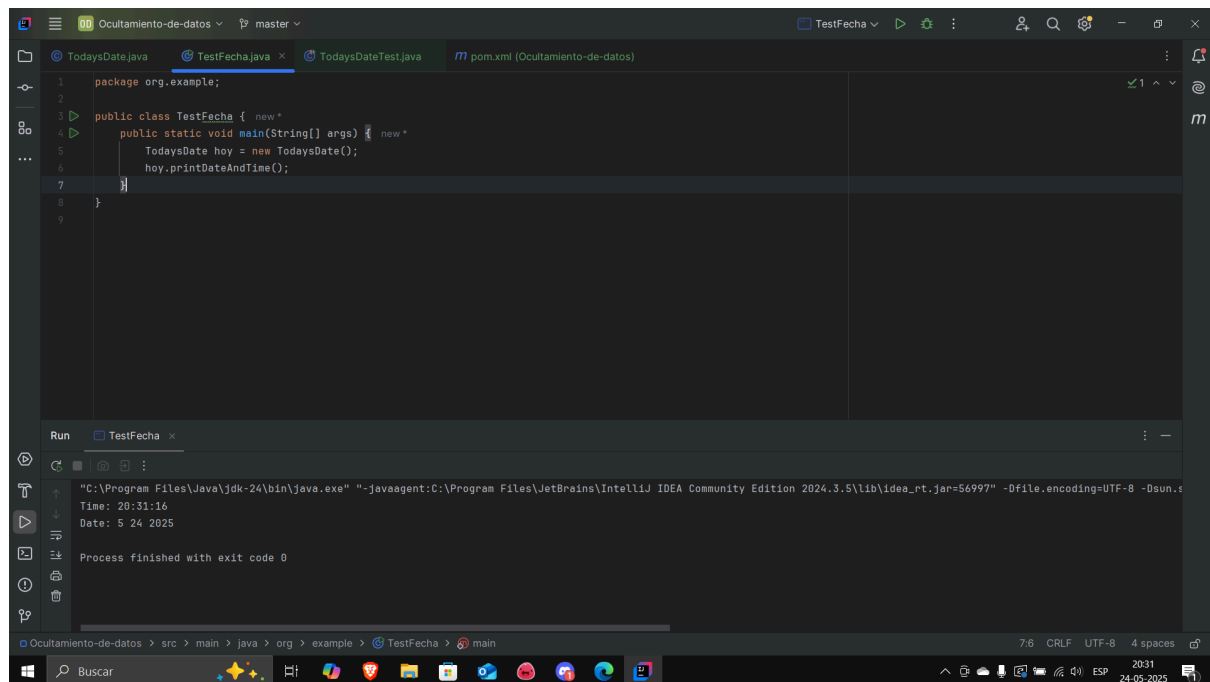


Una ejecución exitosa de su programa en su IDE.

Sophia:



Yasmin:



Cambios:

en time:

```
time= String.format("%02d:%02d:%02d",  
    calendar.get(Calendar.HOUR_OF_DAY),  
    calendar.get(Calendar.MINUTE),  
    calendar.get(Calendar.SECOND));
```

para ejecutar los tests

```
public int getMonth() {  
    return month;  
}  
  
public int getYear() {  
    return year;  
}  
  
public String getTime() {  
    return time;  
}
```

pruebas unitarias

```
import java.io.ByteArrayOutputStream;  
import java.io.PrintStream;  
import java.util.Calendar;  
import java.util.GregorianCalendar;  
  
public class TodaysDateTest {  
  
    private final ByteArrayOutputStream outContent = new  
ByteArrayOutputStream();  
    private final PrintStream originalOut = System.out;  
    private TodaysDate todaysDate;  
  
    @BeforeEach  
    public void setUp() {  
        System.setOut(new PrintStream(outContent));  
        todaysDate = new TodaysDate();  
    }  
  
    // Test 1: Verifica el formato de salida y el contenido de  
printDateAndTime() para Time  
    @Test
```

```

    void testPrintDateAndTime_TimeFormat() {
        todaysDate.printDateAndTime();
        String expectedTimePrefix = "Time:";
        String actualOutput = outContent.toString();

        assertTrue(actualOutput.contains(expectedTimePrefix),
"comprueba si la salida contiene 'Time:');
        assertTrue(actualOutput.matches("(?s).*Time:
\\d{2}:\\d{2}:\\d{2}.*"), "El formato de hora debe ser
HH:MM:SS");
    }

    // Test 2: Verifique el formato de salida y el contenido de
printDateAndTime() para Fecha
    @Test
    void testPrintDateAndTime_DateFormat() {
        todaysDate.printDateAndTime();
        String expectedDatePrefix = "Date:";
        String actualOutput = outContent.toString();

        assertTrue(actualOutput.contains(expectedDatePrefix),
"La salida debe contener 'Fecha:');
        assertTrue(actualOutput.matches("(?s).*Date: \\d{1,2}
\\d{1,2} \\d{4}.*"), "El formato de fecha debe ser M D AAAA");
    }

    // Test 3: Verifique la corrección del atributo 'día'
después de printDateAndTime()
    @Test
    void testDayAttributeValue() {
        GregorianCalendar calendar = new GregorianCalendar();
        int currentDay = calendar.get(Calendar.DATE);

        todaysDate.printDateAndTime();
        assertEquals(currentDay, todaysDate.day, "El atributo
'día' debe coincidir con el día actual.");
    }

    // Test 4: Verifique la exactitud del atributo 'mes' (a
través del getter) después de printDateAndTime()
    @Test
    void testMonthAttributeValue() {
        GregorianCalendar calendar = new GregorianCalendar();
        int currentMonth = calendar.get(Calendar.MONTH) + 1;

```

```

        todaysDate.printDateAndTime();
        assertEquals(currentMonth, todaysDate.getMonth(), "El
        atributo 'mes' debe coincidir con el mes actual.");
    }

    // Test 5: Verifique la exactitud del atributo 'año' (a
    través del getter) después de printDateAndTime()
    @Test
    void testYearAttributeValue() {
        GregorianCalendar calendar = new GregorianCalendar();
        int currentYear = calendar.get(Calendar.YEAR);

        todaysDate.printDateAndTime();
        assertEquals(currentYear, todaysDate.getYear(), "El
        atributo 'año' debe coincidir con el año actual.");
    }

    // Restaurar System.out después de cada prueba
    @org.junit.jupiter.api.AfterEach
    public void restoreStreams() {
        System.setOut(originalOut);
    }
}

```