

**Grado en Ciencia, Gestión e Ingeniería de Servicios**

# **Introducción a la Programación – Otros Materiales**

## **Índice**

- Preguntas Frecuentes - 2
- Trabajando con Codeboard.io - 7

©2022 Juan Manuel Vara Mesa, David Granada Mejía, Fco. Javier Pérez Blanco

Algunos derechos reservados. Este documento se distribuye bajo la licencia “Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en <https://creativecommons.org/licenses/by-sa/4.0/deed.es>

## Introducción a la Programación – FAQ

### DESARROLLO DE LA ASIGNATURA

#### **Semi presencialidad y pruebas de evaluación**

El hecho de que la asignatura se imparta en una titulación semipresencial no implica ningún cambio respecto a la realización de las pruebas de evaluación. Al tratarse de una titulación universitaria oficial, no podemos definir varias alternativas para realizar una prueba. En este sentido, el desarrollo de las pruebas debe ser igual al de la modalidad presencial, y la flexibilidad estriba fundamentalmente en la forma de impartir los contenidos. Tenéis la desventaja de no contar con el profesor in-situ en las clases magistrales (suplida en parte con las videoconferencias), pero la ventaja de poder visualizar y trabajar los contenidos a vuestro propio ritmo.

#### **Uso del foro**

Recordad que la mejor forma de compartir vuestras dudas, comentarios y sugerencias es el uso del FORO de la asignatura, no solo en el contexto de esta asignatura, si no en el de cualquiera de las que curséis a lo largo de vuestra vida académica, especialmente en aquellas, como esta, en las que lo hagáis en la modalidad semipresencial. Os invitamos a considerar algunas de las posibles ventajas derivadas del uso de los foros:

- Una vez publicada la posible solución, todos vuestros compañeros estarán al tanto de cómo resolver el problema que planteáis. Todos dispondréis de la misma información y, además, de un sitio donde volver a consultarla en el futuro, ya que el buscador que incluyen los foros, funciona razonablemente bien.
- Acorta los tiempos de respuesta, porque es probable que alguno de vuestros compañeros pueda ayudaros a responder a esa duda y lo haga más rápidamente que nosotros los profesores.
- Contribuye a la co-creación de conocimiento del que todos os beneficiaréis: es habitual que, a colación de una duda, un compañero plantee otra relacionada, con la consiguiente respuesta, etc.
- Aumenta vuestra visibilidad. En la mayoría de asignaturas valoramos de alguna forma la participación en clase. En el contexto en el que nos encontramos la mejor forma de evidenciar esta participación es estando presentes y/o interactuando en los foros, etc. En este sentido, poneos en el lugar de vuestros profesores e imaginad que estáis en la tesitura de un profesor que debe decidir cuál es la calificación de un alumno que ronda, pero no alcanza el aprobado. ¿Os parecería positivo que, a raíz de sus intervenciones en el foro, ese alumno le resultase familiar a ese profesor? Seguramente podéis intuir que la respuesta es positiva.
- Finalmente, algo que probablemente os interesa saber es que la única forma de que sepamos que nos habéis enviado un correo a la bandeja de la asignatura es entrando a Aula Virtual y pinchando en la asignatura y, como podréis intuir, hay días en los que no nos es posible conectarnos al aula virtual. En cambio, los mensajes del foro generan el envío de un correo a nuestra bandeja de correo personal, que como también intuiréis, consultamos más a menudo. Por lo tanto, es muy probable que respondamos más rápido a una duda planteada en el foro, que a una duda que nos llega por correo.

## Planteando dudas

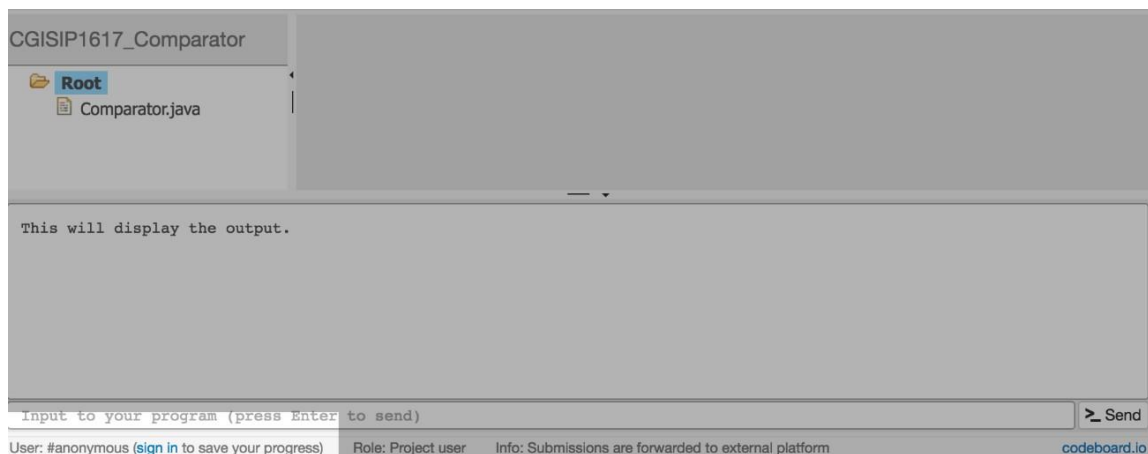
Es muy importante que planteéis dudas concretas que podamos responder y que probablemente generen más debate. La interacción que nos permite la plataforma resultará mucho más rica si las dudas o comentarios que nos hacéis llegar son lo suficientemente concretas. No tiene mucho sentido plantear una cuestión del tipo: "No entiendo el Tema X ...". Este tipo de cuestiones es prácticamente imposible de responder a través del foro y, sobre todo, no contribuyen a que avancéis, más bien al contrario.

Lo deseable es que os enfrentéis a los ejercicios que os vayamos planteando y, cuando lleguéis a lugares donde os sea complicado avanzar, recurráis a nuestra ayuda para revolver o abordar ese problema concreto. Ese avance resultará mucho más provechoso.

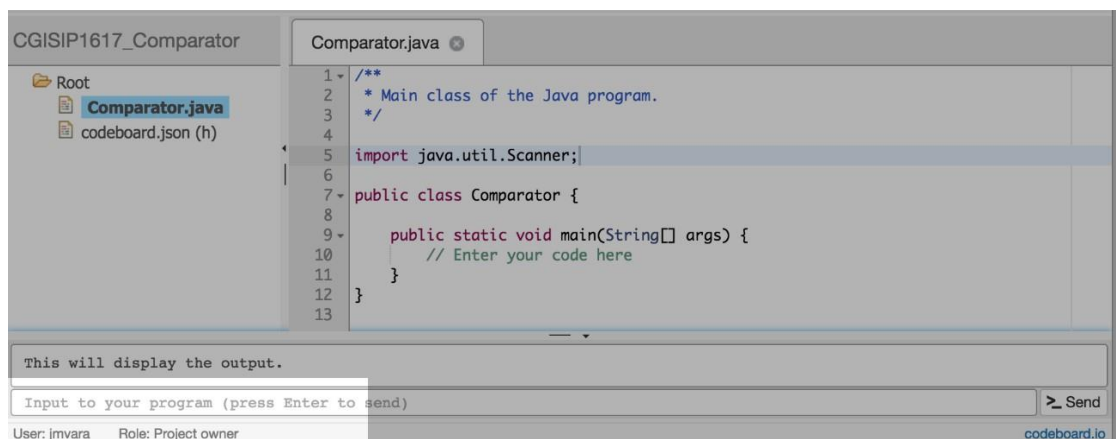
## CODEBOARD.IO

### Registrar envíos

A la hora de usar Codeboard para resolver los diferentes ejercicios, no olvidéis loguearos en la plataforma antes de comenzar a trabajar. De lo contrario, cuando pulséis el botón SUBMIT, el envío quedará registrado a nombre del usuario #anonymous, tal y como muestra la siguiente imagen.

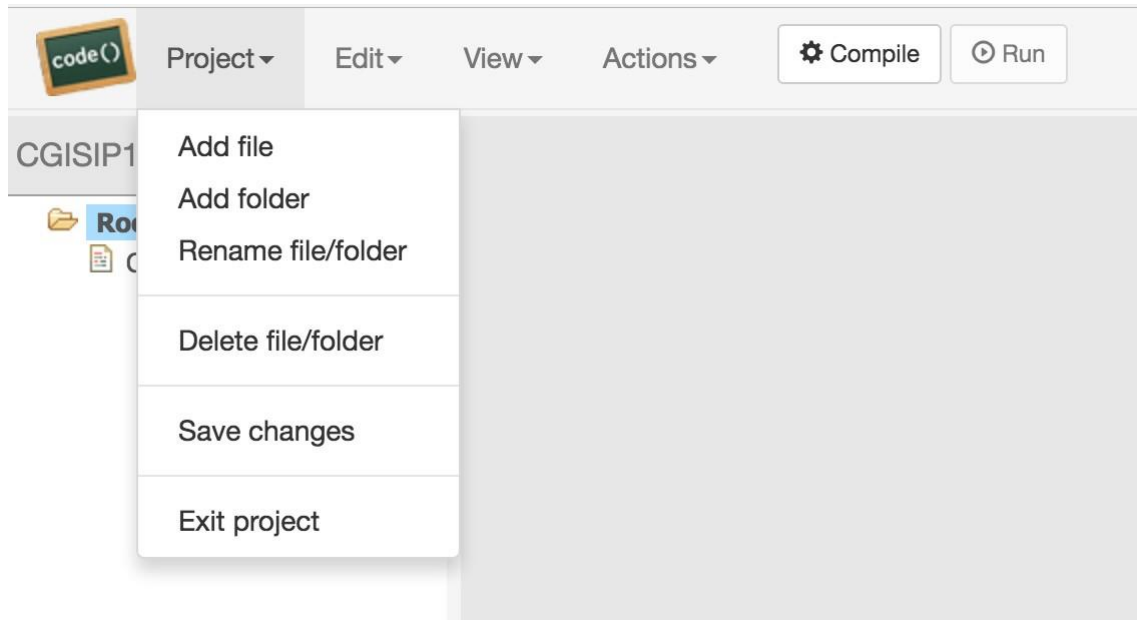


Por el contrario, si os logueais (podéis utilizar el enlace sign in que se muestra junto al nombre usuario en la esquina inferior izquierda de la pantalla), vuestro envío quedará registrado con vuestro nombre usuario, tal y como muestra la siguiente imagen.



## Guardar proyectos

Para poder consultar vuestros proyectos, y repasar las soluciones que habéis ido elaborando para los diferentes ejercicios, no olvidéis guardar vuestro trabajo antes de cerrar el proyecto. Para ello, utilizad el menú "Project", opción "Save Changes".



## Varios Envíos

El sistema registra automáticamente todos los envíos que realizamos y los ordena automáticamente por fecha de envío. Por tanto, si por error realizamos varios envíos para un mismo ejercicio, el profesor dispondrá de todos ellos y además visualizará automáticamente el último que realizamos, con lo que el hecho de realizar varios envíos no supondrá ningún problema.

## ERRORES TÍPICOS DE PROGRAMACIÓN

### Uso correcto de if

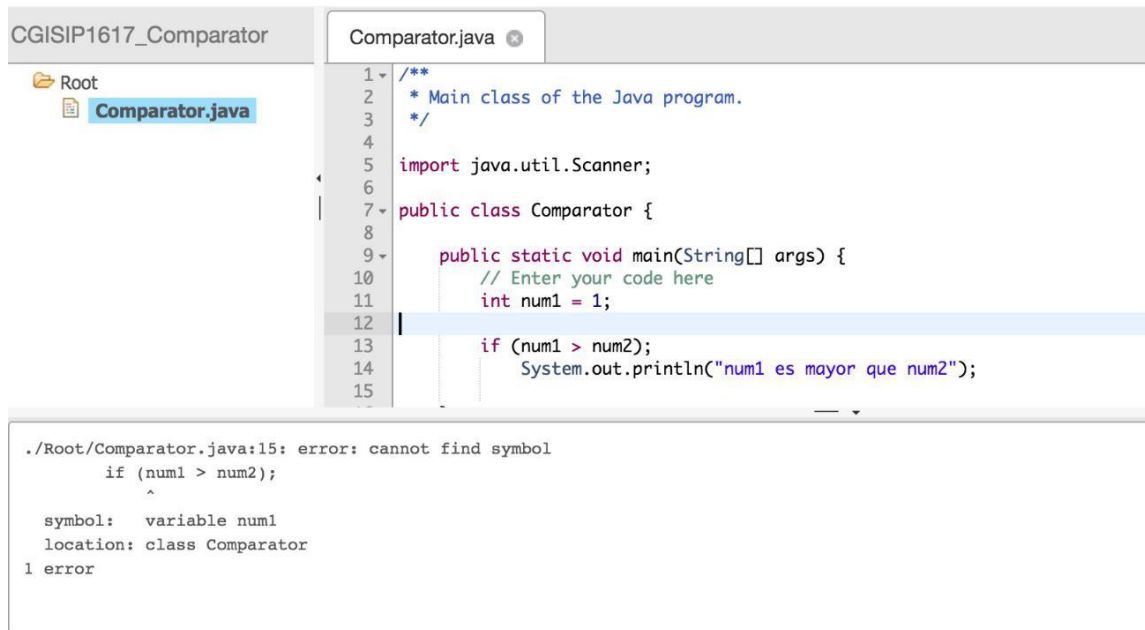
Frecuentemente cometemos el error de añadir un ';' después de especificar la condición que debe evaluarse en una sentencia if. Este error provoca que, contrariamente a lo que queremos, la sentencia if acabé en ese punto, y por tanto las sentencias que queríamos que se ejecutasen si se cumplía la condición, no dependen en ninguna medida, y se ejecutarán siempre, independientemente del valor que resulte de evaluar la condición.

```
if (num1 > num2);  
    System.out.println("num1 es mayor que num2");
```

## MENSAJES DE ERROR DEL COMPILADOR

### **error: cannot find symbol**

Uno de los mensajes de error más frecuentes que nos devuelve el compilador es el siguiente: error: cannot find symbol . Lo más habitual es que este mensaje se deba a que estamos tratando de utilizar una variable, que nos hemos declarado previamente, como en el siguiente ejemplo:



```
CGISIP1617_Comparator  Comparator.java x
Root
  Comparator.java
1  /**
2  * Main class of the Java program.
3  */
4
5  import java.util.Scanner;
6
7  public class Comparator {
8
9      public static void main(String[] args) {
10         // Enter your code here
11         int num1 = 1;
12
13         if (num1 > num2);
14             System.out.println("num1 es mayor que num2");
15     }
16 }

./Root/Comparator.java:15: error: cannot find symbol
    if (num1 > num2);
                ^
    symbol:   variable num2
    location: class Comparator
1 error
```

En el ejemplo, hemos olvidado declarar la variable num2 y cuando queremos utilizarla, el compilador, con su mensaje de error, nos informa de que no reconoce ese símbolo. En otras palabras, no conoce a nadie que se llame num2.

La solución pasa por declarar la variable antes de usarla. A partir del momento en que la declaremos, podemos usarla en cualquier otro lugar de nuestro programa (respetando las reglas que rigen el ámbito de una variable ... que seguro conoces.

### **error: variable \_\_\_\_ is already defined**

Otro de los mensajes de error típicos que recibimos al compilar es el siguiente: error: variable \_\_\_\_ is already defined. En general, este mensaje se debe a que estamos tratando de declarar de nuevo una variable, que YA hemos declarado previamente, como en el siguiente ejemplo.

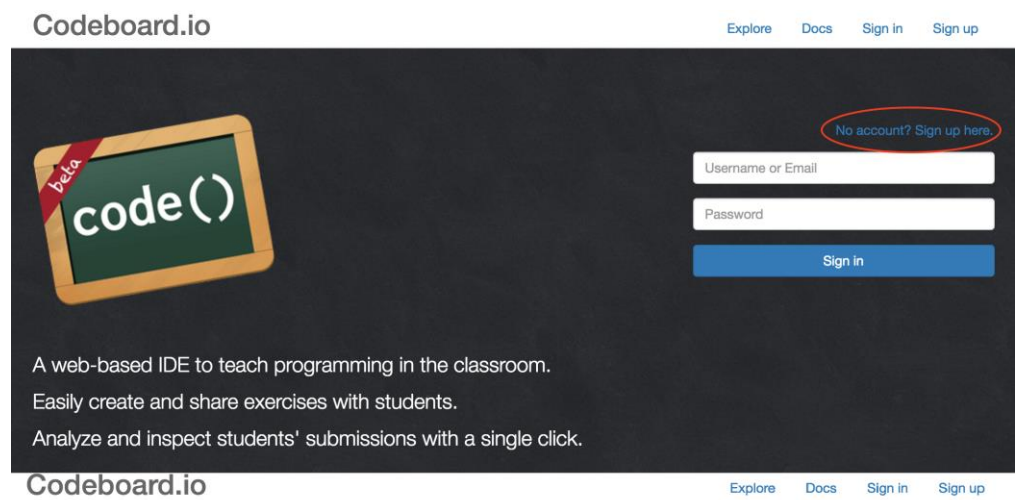
```
CGISIP1617_TimeConvorsor | TimeConvorsor.java | TimeConvorsorSlcn.java |
├── Root
│   ├── TimeConvorsor.java
│   ├── TimeConvorsorSlcn.java (h)
│   └── codeboard.json (h)
└── TimeConvorsor.java
    1  /**
    2  * Main class of the Java program.
    3  */
    4  import java.util.Scanner;
    5
    6  public class TimeConvorsor {
    7
    8  public static void main(String[] args) {
    9      System.out.println("Introduce los segundos: ");
   10      Scanner scan = new Scanner(System.in);
   11      int seconds = scan.nextInt();
   12
   13      int minutes = seconds / 60;
   14      int seconds = seconds - (seconds / 60);
   15
   16  }
   17  }
   18
./Root/TimeConvorsor.java:14: error: variable seconds is already defined in method main(String[])
    int seconds = seconds - (seconds / 60);
    ^
1 error
```

En el ejemplo, hemos declarado la variable `seconds` y le hemos asignado el valor leído por pantalla (línea 11). A partir de aquí, podríamos utilizar la variable en cualquier punto de nuestro programa (siempre de acuerdo a las reglas que rigen el ámbito de una variable).

Por ejemplo, en la línea 14 queremos actualizar dicha variable, de modo que guarde el número de segundos que restan tras convertir los segundos iniciales a minutos. El problema estriba en que, al anteponer el tipo de dato a la variable, `int seconds`, lo que en realidad estamos haciendo es declarar una nueva variable, con el mismo nombre de otra que ya existe, algo que el compilador no permite, elevando el correspondiente mensaje de error.

## Trabajando con Codeboard

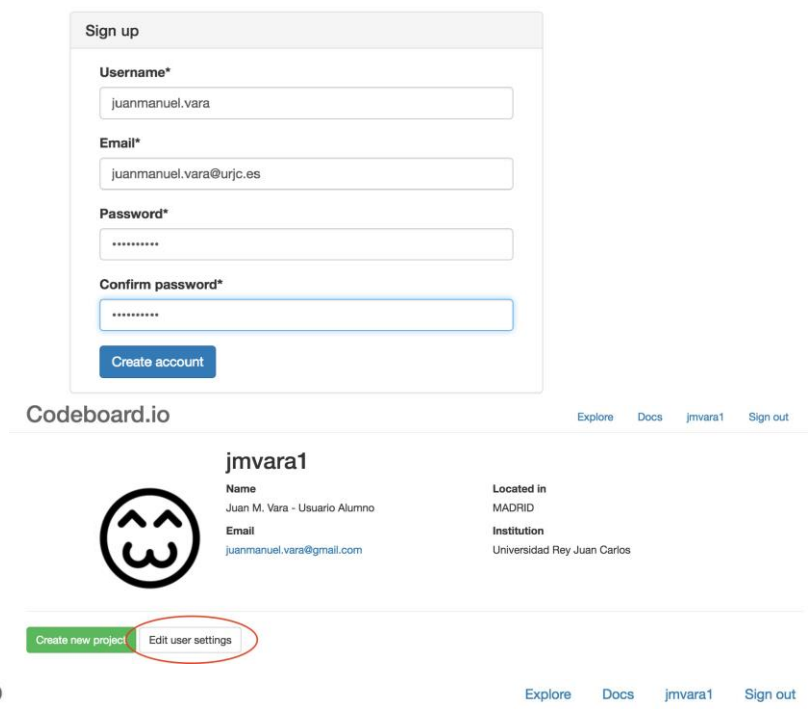
Lo primero que debemos hacer es Registrarnos en la plataforma. Para ello vistamos la URL <https://codeboard.io>. Una vez en la home del proyecto utilizamos el enlace "Sign up here." para comenzar el registro.



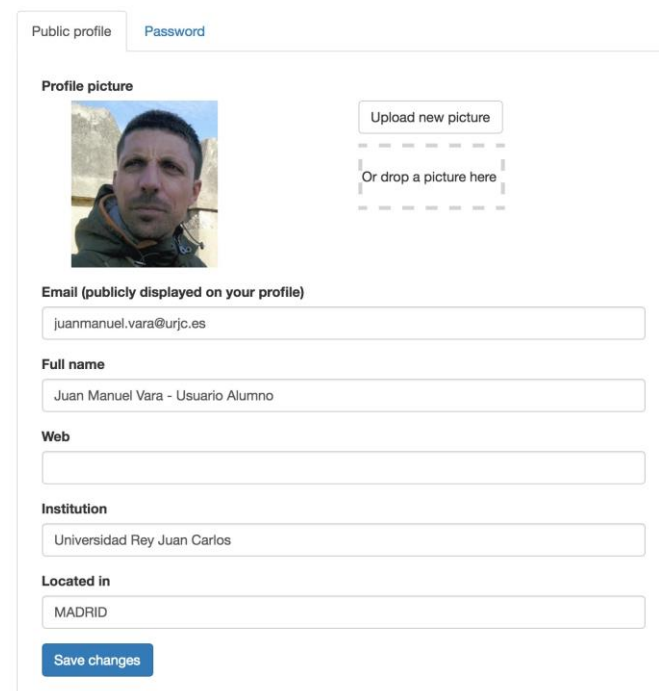
Durante el registro, utilizad por favor vuestro nombre y apellidos (separado de un punto) como nombre de usuario, y la cuenta de correo de la Universidad.

Una vez que os habéis registrado, haced clic sobre vuestro nombre de usuario en la parte superior derecha de la pantalla para acceder a vuestro perfil de usuario.

Pulsad el botón "Edit User Settings" para acceder a la página que os permite completar vuestros datos.



En esta nueva pantalla, no dejéis de especificar vuestro correo de la universidad (nombre.apellido@alumnos.urjc.es), vuestro nombre completo (o full name) e incluir una foto, como siempre, para facilitarnos el seguimiento de vuestro trabajo.



A partir de este momento, podéis utilizar Codeboard para completar los diferentes ejercicios prácticos que os plantearemos a medida que avance la asignatura.

La forma de hacerlo será siempre la misma:

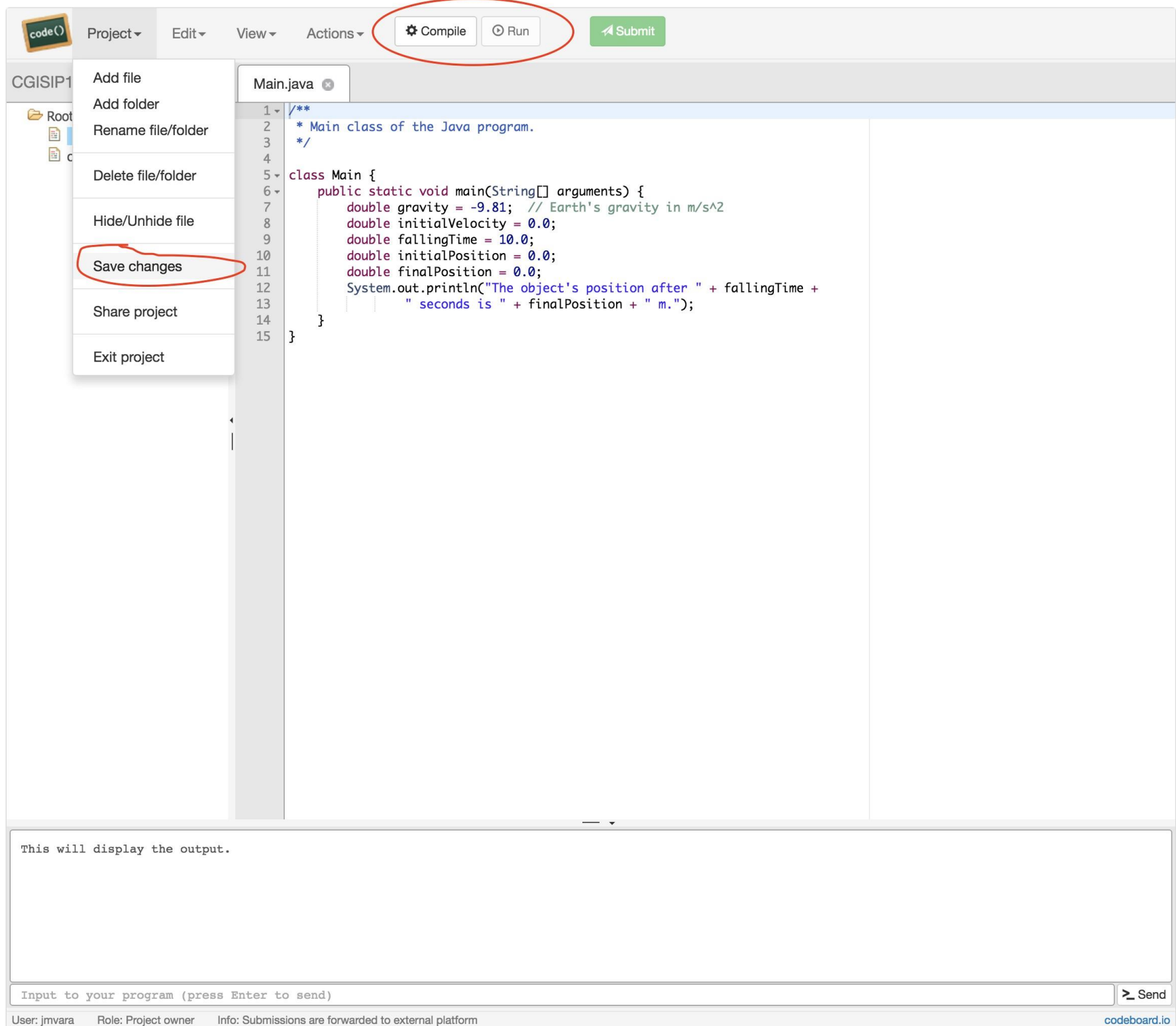
- Pinchad el enlace al ejercicio correspondiente, de entre las que iremos publicando en la pestaña Ejercicios Prácticos de la sección de Contenidos. Como resultado, se abrirá una nueva ventana de Codeboard. En dicha ventana encontraréis el proyecto Java de partida para el ejercicio en cuestión.
- Realizad las modificaciones necesarias, para alcanzar los objetivos planteados en la especificación del Ejercicio.
- Compilad y Ejecutad vuestro proyecto para comprobar, que es sintácticamente correcto (compilación) y que proporciona la funcionalidad requerida (Ejecución). Utilizad para ello los botones "Compile" y "Run" de la parte superior de la ventana.
- Guardad los cambios realizados: menú Project -> opción Save changes

GRADO EN CIENCIA, GESTION E INGENIERIA DE SERVICIOS (MADRID)  
2176 - INTRODUCCION A LA PROGRAMACION - MAÑANA A - 1Q

Página Principal > 2016-17\_2176006\_134795\_53 > Ejercicios Prácticos > Práctica 1 - GravityCalculator

Práctica 1 - GravityCalculator

Primer proyecto de la asignatura "Introducción a la Programación" del Grado en Ciencia, Gestión e Ingeniería de Servicios, curso 2016/2017



- Si no os habéis logueado previamente, en ese momento Codeboard os pedirá que lo hagáis. Utilizad para ello el usuario y contraseña con el que os registrasteis en la plataforma.

- A partir de ese momento, podéis reabrir el proyecto tantas veces como queráis, utilizando el enlace al ejercicio mencionado en el primer punto.
- Cada vez que lo hagáis, Codeboard os informará de que existe una versión modificada por vosotros de ese proyecto, y os preguntará si queréis cargar vuestra versión, o la original (la que os proporcionamos al comienzo del ejercicio).
- En general, preferiréis reabrir vuestra versión para continuar trabajando sobre ella, aunque en ocasiones, puede que prefiráis empezar de cero, reabriendo la versión original o de partida que os proporcionamos al comienzo del ejercicio.
- Finalmente, cuando consideréis que habéis resuelto correctamente el Ejercicio, debéis enviar vuestro proyecto, pulsando el botón Submit (color verde) en la ventana de Codeboard. De esta forma, vuestra solución quedará grabada en el sistema, incluyendo la fecha y hora del envío.

GRADO EN CIENCIA, GESTION E INGENIERIA DE SERVICIOS (MADRID)  
2176 - INTRODUCCION A LA PROGRAMACION - MAÑANA A - 1Q

Página Principal > 2016-17\_2176006\_134795\_53 > Ejercicios Prácticos > Práctica 1 - GravityCalculator

Práctica 1 - GravityCalculator

Primer proyecto de la asignatura "Introducción a la Programación" del Grado en Ciencia, Gestión e Ingeniería de Servicios, curso 2016/2017

