



U N I V E R S I D A D  
**COMPLUTENSE**  
M A D R I D

# Tutoriales Interactivos en el Aprendizaje de la Programación

Enrique Martín - [emartinm@ucm.es](mailto:emartinm@ucm.es)

Adrián Riesco - [ariesco@ucm.es](mailto:ariesco@ucm.es)

Fac. Informática UCM – 30 de mayo de 2017

# Introducción

# Motivación

- **Experiencia personal:** los alumnos de 1º de programación suelen tener dificultades con la materia (*al menos en la Fac. de Ciencias Matemáticas*).
- Cualquier herramienta o técnica que pueda complementar la docencia tradicional es bienvenida.
- Los **jueces** revisan automáticamente los programas de los alumnos para muchos casos → rápida retroalimentación.
  - DOMjudge (<https://www.domjudge.org>, que p.ej. es usado en la FDI en TAIS <http://tais.fdi.ucm.es>)
  - FLOP (<http://problem-g.estad.ucm.es>, usado en *Informática* de los Grados de la FCM).

# Motivación

- Sin embargo los jueces automáticos tienen alguna *limitación*:
  - Usualmente requieren programas completos.
  - Sus respuestas no son muy informativas.
  - Se ejecutan en remoto, quizá usando sistemas operativos, compiladores y versiones diferentes al alumno.
- *(Al fin y al cabo se diseñaron principalmente para competencias, no para docencia)*

# Idea

- Combinar la **evaluación automática** de código del alumno con el **aprendizaje paso a paso** de conceptos teóricos estilo tutorial.
- Origen de la idea: **swirl** para aprender R (<http://swirlstats.com>)
  - Modo consola
  - Paso a paso, distintas lecciones
  - Integrable en MOOCs

# Nuestra herramienta

- Resultado de 2 proyectos incrementales:
  - PIMCD 2015/109 (<http://eprints.ucm.es/35367/>)
  - **INNOVA-Docencia UCM (ref. 2016/35)**
- Construido sobre el prototipo realizado en el TFG de Rafael Caturla y Carlos Congosto (<http://eprints.ucm.es/38408/>), dirigido por Manuel Montenegro y Enrique Martín.

# Equipo proyecto 2015/109

- Carlos Gregorio                      DSIC, Matemáticas UCM
- Enrique Martín                      DSIC, Informática UCM
- Manuel Montenegro                      DSIC, Informática UCM
- Adrián Riesco                      DSIC, Informática UCM
- Jaime Sánchez                      DSIC, Informática UCM
- Salvador Tamarit                      DSIC, Informática UPV

# Equipo proyecto 2016/35

- Rafael Caturla                      antiguo alumno
- Carlos Congosto                    alumno Máster
- Carlos Gregorio                    DSIC, Matemáticas UCM
- Francisco J. López                  DSIC, Informática UCM
- Enrique Martín                      DSIC, Informática UCM
- Adrián Riesco                       DSIC, Informática UCM
- Jaime Sánchez                      DSIC, Informática UCM
- Salvador Tamarit                    DSIC, Informática UPV



# Nuestra herramienta

- Aplicación de **escritorio** para aprender lenguajes de programación.
- Muestra de manera **secuencial** distintos bloques:
  - Explicaciones teóricas
  - Preguntas de varias opciones
  - Pequeños ejercicios de codificación
- Para continuar con el siguiente bloque es necesario **completar** las preguntas/ejercicios.

# Nuestra herramienta

- Soporta 4 lenguajes de programación:
  - Python
  - C++ (GNU C++)
  - C# (Mono)
  - Java
- Para C++ y C# hay un soporte *experimental* para Visual Studio (usando `vcvarsXX.bat`).
- Es sencillo incorporar nuevos lenguajes, pero requiere que modificar el código de la aplicación.

# Nuestra herramienta

- Multiplataforma: mínimo Java 8, JavaFX para la interfaz gráfica.
- Gestión mediante **Maven**.
- Licencia de código abierto **MIT**.
- Disponible en GitHub:  
<https://github.com/emartinm/TutorialesInteractivos>

# **Demostración**

# Instalación y ejecución

- Instalación sencilla: descargar de GitHub o de un ZIP que prepare el profesor de la asignatura. → **descomprimir y listo.**
- Ejecución: todo el código agrupado en un único fichero JAR:
  - *Script* **.sh** (Linux/Mac) y **.bat** (Windows).
  - Directamente desde la consola con `java`.

# Configuración

- Antes de empezar a utilizar la herramienta hay que configurar dos aspectos:
  - Directorio de temas (una carpeta por lenguaje de programación soportado).
  - Ruta al compilador y/o intérprete por cada lenguaje soportado.
- Esta configuración se realiza de manera gráfica.

# Explicaciones

- Contienen texto representado mediante **Markdown** (también soporta HTML directamente):
  - Negrita, monoespaciado, itálica, etc.
  - Listas numeradas y no numeradas, anidamiento
  - Tablas
  - Enlaces que se abren en el navegador
  - Imágenes locales y remotas (incluyendo GIFs animados)
  - Soporte integrado para fórmulas LaTeX
  - Vídeos incrustados

# Preguntas de varias opciones

- De dos tipos:
  - Una sola opción válida (*radio buttons*)
  - Varias opciones válidas (*checkboxes*)
- Permiten mostrar **pistas generales** al alumno.
- Hasta que no se resuelven no se puede avanzar.



# Preguntas de codificación

- Solicitan al usuario 1 o varios fragmentos de código.
- Permiten mostrar **pistas generales** al alumno.
- Si la solución no es correcta, puede mostrar **pistas particulares** sobre el código del alumno (p.ej. para avisar de errores comunes).
- Hasta que no se resuelven no se puede avanzar.

# Preguntas de codificación

- Las preguntas de codificación se corrigen usando un **programa corrector**, que es un programa con uno o más *"huecos"*.
- El código del alumno se incrusta en esos huecos → **programa completado**.
- El programa completo se compila/interpreta, comprobando la validez del código del alumno y generando un fichero **JSON con 3 campos**:
  - `'isCorrect'`: true/false
  - `'typeError'`: mensaje sobre la naturaleza del error
  - `'Hints'`: lista de pistas

# Definición de temas

- Fichero YAML por cada tema (*ojo que YAML tiende a ser un poco molesto con espacios y tabuladores*).
- Cada tema contiene una lista de lecciones.
- Cada lección contiene una lista de elementos.
- Cada elemento está definido como un diccionario clave-valor de campos.

# Más información

- Hay varios manuales en la carpeta de documentación de la herramienta:
  - Manual del usuario  
[https://github.com/emartinm/TutorialesInteractivos/raw/master/doc/Manual\\_usuario.pdf](https://github.com/emartinm/TutorialesInteractivos/raw/master/doc/Manual_usuario.pdf)
  - Manual para crear lecciones  
[https://github.com/emartinm/TutorialesInteractivos/raw/master/doc/Manual\\_crear\\_lecciones.pdf](https://github.com/emartinm/TutorialesInteractivos/raw/master/doc/Manual_crear_lecciones.pdf)

# Conclusiones

# Aspectos interesantes

- Orientado a lecciones cortas, pequeños pasos → **motivación.**
- Usa el mismo sistema (sistema operativo y compilador) que tiene el alumno en su ordenador.
- Contenido y soluciones incluidas en el fichero descargado → no necesita conexión a Internet.
- Puede complementar a los jueces automáticos.

# Aspectos interesantes

- Aplicable a **distintas situaciones**:
  - **Introducción** a la programación desde 0: asignaciones, operaciones aritméticas, condicionales, bucles, listas, etc.
  - **Refuerzo/recordatorio**: para asignaturas que utilizan lenguajes aprendidos en otras.
  - **Extensión**: para explorar nuevos aspectos de lenguajes ya conocidos o la utilización de nuevos módulos o librerías.
  - **Adaptación**: para mostrar los detalles propios de un nuevo lenguaje a partir de otro conocido.
- Aplicable a **distintos niveles educativos**: Universidad, Formación Profesional, Bachillerato o incluso ESO.

# Ideas para el futuro

- Gestión de distintos usuarios, posiblemente mediante un servidor central que almacena sus progresos.
- Integración con el Campus Virtual → puntos adicionales a los alumnos según superen lecciones.
- Soportar más lenguajes de programación (Maude, Haskell, Scala) e incluso otros paradigmas (ensamblador, VHDL).



# ¿Mejora el aprendizaje?

- Queremos pensar que sí, pero hay que medirlo.
- Hemos solicitado un nuevo proyecto INNOVA-Docencia para el curso 2017/18:
  - Aplicar a grupos reducidos en FCM y FDI
  - Obtener indicadores objetivos y subjetivos
  - Analizar la magnitud de la mejora obtenida

# Discusión

- ¿Consideráis que la herramienta puede mejorar el aprendizaje de la programación?
- ¿La herramienta se podría aplicar en vuestras clases? ¿Lo haríais?
- ¿Qué aspectos se podrían mejorar para hacerla más útil?

# Encuesta

- Sería muy interesante recabar vuestra opinión sobre la herramienta.
- Podéis encontrar una breve encuesta en:  
<https://goo.gl/forms/5erVztzgdOPPjigw1>  
*(se tarda unos 4 minutos, son pocas preguntas)*

**¡Gracias!**