Catalyde: herramienta para enseñar programación

Catalin Stanciu, Joan Pastor, Salvador España
cstanciu@dsic.upv.es, jpastor@dsic.upv.es, sespana@dsic.upv.es
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia

Motivación

- Aprender a programar requiere del desarrollo de habilidades por medio de la práctica. Del mismo modo que no aprendemos a ir en bicicleta con una bici de carrera, también tiene sentido aprender a programar en unas condiciones que faciliten el proceso de aprendizaje.

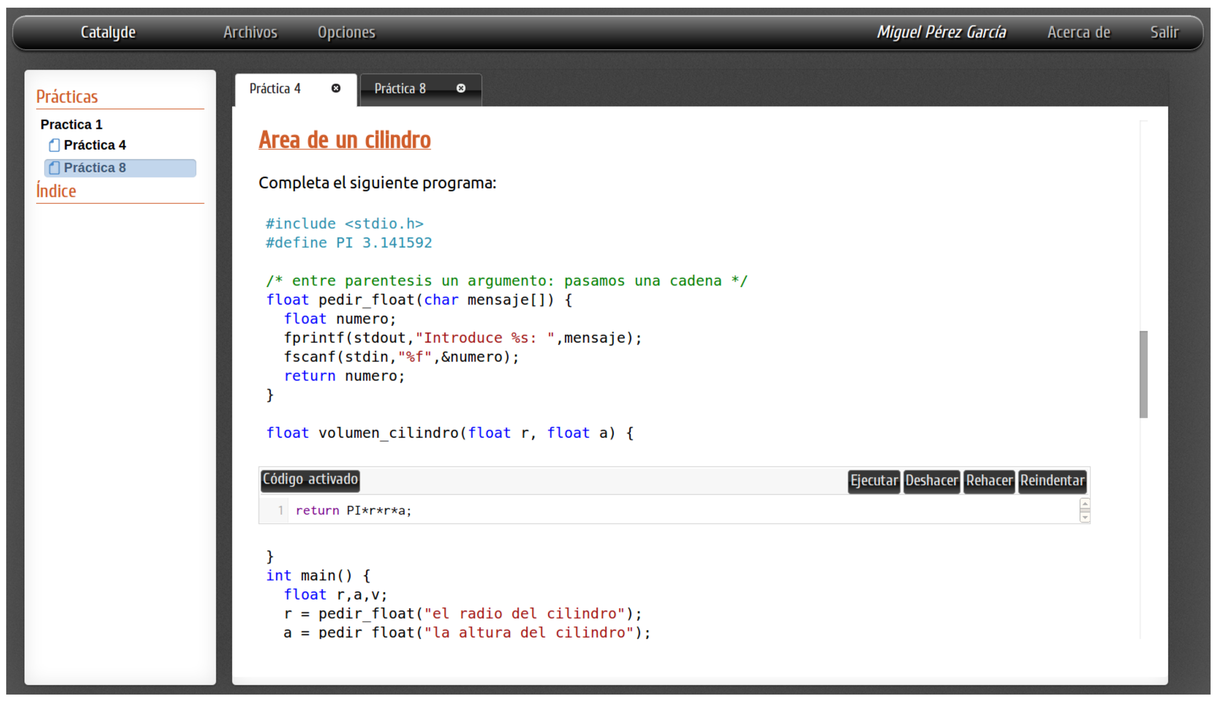
- Históricamente se ha planteado el desarrollo de lenguajes con fines pedagógicos (ej. Logo), así como entornos más orientados al aprendizaje que al uso profesional (ej. para java, utilizar Bluej en lugar de Eclipse).
- Se presenta una herramienta para facilitar el aprendizaje a la programación. Está orientada principalmente a las asignaturas de primeros cursos, en particular, para asignaturas que utilizan el lenguaje C (aunque puede adaptarse a cualquier otro lenguaje de programación como java). Se ha elegido C para poder implantar la herramienta en las asignatura de informática de los grados de ingeniería técnica industrial en la UPV (la ETSINF utiliza Java en los primeros cursos).

Situación actual

- **Por parte del profesor:** Tradicionalmente, la elaboración de las prácticas consiste en el desarrollo de un boletín, la implementación de una solución y, en algunos casos, la selección de una parte de la misma para que el alumno la complete. Finalmente el material se deja en poliformaT.
- **Por parte del alumno:** bajarse el boletín y los fragmentos de código a completar, manipular las distintas versiones que se van realizando en las carpetas de su ordenador, utilizando un entorno de desarrollo (por ejemplo, en el grado de ingeniería técnica industrial se utiliza `devcpp`) y finalmente entregar las soluciones, si así se solicita, mediante correo electrónico o vía poliformaT.
  - Normalmente el alumno no tiene acceso a una versión funcional del programa para poder probarlo y así entender más fácilmente qué es lo que se pretende resolver. La experiencia nos indica que poder ejecutar la solución ayuda enormemente a entender lo que se le pide.
  - Aunque proporcionar una solución compilada puede mejorar la comprensión del ejercicio, el alumno no podrá utilizarla para validar las distintas partes realizadas por él y normalmente no podrá saber si una parte es correcta hasta haber completado todo el ejercicio.

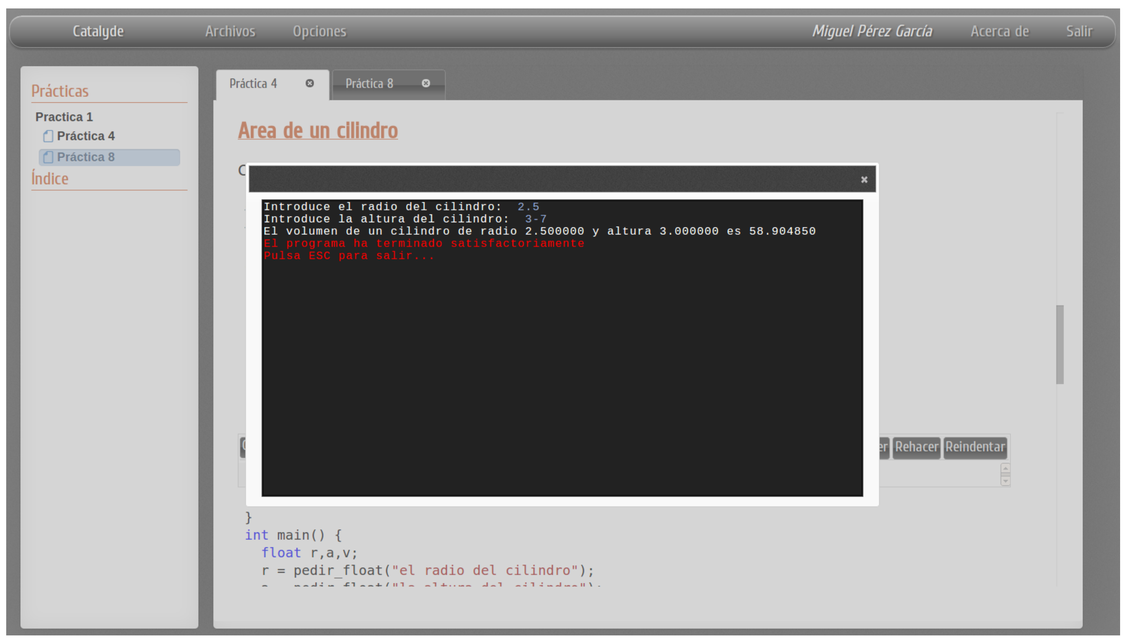
Sistema propuesto

- Se trata de una herramienta vía web en la que el alumno se autentifica y puede acceder a las prácticas elaboradas por el profesor.
- La descripción de las prácticas contiene zonas de edición de código donde el alumno puede escribir su solución. Pudiendo además, activarla y desactivarla (se utiliza la solución correcta oculta).
- El alumno puede probar en todo momento la solución oculta para entender lo que se pide o sustituir partes de ella por su propia solución.
- Existe la opción de comprobar la validez de las distintas partes realizadas por el alumno de manera independiente del programa final.
- Las diferentes partes de la práctica se pueden realizar en ejemplos sencillos y combinarse en un programa más complejo que el alumno puede probar. De esta forma, el alumno puede analizar, estudiar y probar dicho código.



¿Qué aporta esta herramienta para aprender a programar?

- Poder trabajar con los programas de manera incremental. El profesor puede dividir el código en apartados y el alumno podrá activar o desactivar su propio código de modo que pueda ejecutar el programa independientemente de que se haya completado todo.
- Centrarse en los aspectos concretos que el alumno debe aprender. Hasta ahora, el profesor proporciona un esqueleto del programa, pero no hay garantía de que el alumno respete la estructura.
- Poder validar determinados fragmentos de código mediante el uso de casos de prueba. Esta validación es parcial en la medida en que pasar unos casos de prueba no garantiza la corrección del algoritmo pero sí a la inversa: no pasar estos casos indica que hay un error. Estas técnicas se utilizan también en los oráculos de los concursos de programación.
- Mejor seguimiento y evaluación del alumno: El sistema guarda todos los pasos intermedios realizados por el alumno. Facilita la corrección de las prácticas, así como la detección de copias.
- Ejecutar el programa e interactuar con él mediante un terminal emulado vía web permite utilizar la aplicación en cursos online y facilita la realización de prácticas desde casa. También elimina la necesidad de instalar el entorno por parte de los alumnos.



Ampliaciones futuras

- Modo examen:
  - Utilizar el mismo entorno que el usado normalmente en las prácticas.
  - No hace falta tener que subir un conjunto de ficheros como ocurre con los entornos offline.
- Trabajo colaborativo: permitir que varios alumnos trabajen en grupo o que si varios alumnos se sienten en un mismo ordenador puedan compartir el trabajo realizado en dicha sesión, mientras actualmente están obligados a entrar con la cuenta de uno de ellos
- Modo evaluación y estadísticas: para que el profesor pueda navegar más cómodamente inspeccionando las prácticas realizadas por los alumnos, visualizar estadísticas, etc.
- Modo para que el profesor pueda editar e introducir los nuevos contenidos con más facilidad.

Conclusiones

- La aplicación permite focalizar a los alumnos en las diversas partes que forman un programa, centrándose así en la parte que el profesor quiera enfatizar en cada momento. Los alumnos, pueden verificar si una parte del código es correcta sin tener que completar todo el programa.
- El profesor escribe su programa y debe marcar lo que deben completar los alumnos. Los alumnos no necesitan acceder al código no relevante para aprender los conceptos requeridos.
- La aplicación ha sido diseñada para poder ser lo suficientemente flexible, para que el profesor pueda añadir funciones que verifiquen el funcionamiento del código, así como ejecutar diferentes instancias del programa con diferentes configuraciones. Por tanto, es el profesor el que decide mediante el diseño de las prácticas aquellos aspectos en los que el alumno deberá centrarse.
- Es criticable la falta de realismo respecto a un entorno de programación convencional:
  - Resulta necesario una transición progresiva a entornos más convencionales. Para ello se pedirá progresivamente ejercicios que requieran escribir un programa completo.
  - Resulta importante atraer a los alumnos con experiencias sencillas y exitosas y centrarse en la parte algorítmica. Una vez aprendido esto, saltar a un entorno real resultará relativamente sencillo.