

# Desarrollo de una plataforma virtual de simulación de sistemas eléctricos (DEVISI)

**Alberto Laso Pérez**

*Universidad de Cantabria, España*

**Sergio Bustamante Sánchez**

*Universidad de Cantabria, España*

**Raquel Martínez Torre**

*Universidad de Cantabria, España*

**Mario Mañana Canteli**

*Universidad de Cantabria, España*

**Alberto Arroyo Gutiérrez**

*Universidad de Cantabria, España*

## Resumen

Uno de los problemas de las pequeñas actividades de seguimiento o evaluación a realizar en casa por los alumnos es la facilidad con la que estas actividades pueden ser copiadas. El principal motivo es que normalmente todos los alumnos deben de resolver el mismo ejercicio. Una solución inmediata a este problema es que el docente prepare ejercicios individualizados para todos los alumnos, pero esta estrategia puede llegar a significar una carga de trabajo inasumible para el profesor. Este documento resume el funcionamiento de un sistema que genera automáticamente ejercicios de análisis de circuitos individualizados para cada alumno y corrige automáticamente los resultados sin apenas intervención del profesor. El sistema genera variantes del mismo ejercicio y, por tanto, sigue siendo susceptible de copia, pero ahora esta copia no es tan inmediata y requiere un mínimo de entendimiento de lo que se está copiando por lo que, incluso copiando, el alumno terminará adquiriendo conocimiento. Además, esto no es solo útil para la evaluación, sino que gracias a este desarrollo se pueden proporcionar fácilmente ejercicios para trabajo personal del alumnado desde su propia casa y con opción de verificación de resultados. Por último, y en una situación de pandemia como la que se ha dado, el sistema permite examinar en remoto a los alumnos de forma simultánea dificultando la copia entre ellos.

*Palabras clave: python; análisis de circuitos; spice; sistemas eléctricos; ejercicios automatizados.*

## Development of a virtual simulation platform for electrical systems (DEVISI)

### Abstract

One of the main problems of the small monitoring and evaluation exercises to be completed out of the lecture room is the easiness to copy the results from other students. The main reason for this easiness is the fact that it is common for all students to face the exact same exercise. An immediate solution for this problem is for the teacher to prepare individualised tasks for every pupil. However, this solution may imply an unaffordable amount of work for the lecturer. This document describes a system that can generate individualised circuit analysis exercises for every student and allows for automatic verification of the results with minimum effort from the teacher. The system generates variants from the same exercise and so they can still be copied, but now copying them requires a minimum understanding of what is being done. This requirement means that the students will acquire some knowledge even if they do not resolve the task by themselves. Moreover, this system is not only useful for evaluation but also for easily providing exercises for personal work out of the classroom and with the chance of automatic correction. Finally, given a situation like the pandemic, this development allows for remote simultaneous exams with barriers to cheating.

*Keywords: python; circuit analysis; spice; electrical systems; automated exercises.*

### Introducción

Con la llegada de la pandemia han surgido nuevos retos en el ámbito de la educación. Uno de los principales ha sido la adaptación a la impartición de clases no presenciales, lo cual se ha solucionado de distintas maneras, como por ejemplo la emisión de las clases por videoconferencia o la grabación de las lecciones en vídeo por parte del profesor. Sin embargo, estas soluciones no arreglan otro de los problemas asociados a la no presencialidad: las pruebas de evaluación.

Un aula permite evaluar a todos los alumnos simultáneamente con el mismo ejercicio y con unas mínimas garantías de que lo resuelven en solitario. Por desgracia, esas garantías no son fáciles de conseguir cuando la evaluación se realiza en remoto. Los alumnos disponen de muchas facilidades para copiar los ejercicios de alguno de sus compañeros. Una manera de mitigar este problema es crear un ejercicio diferente para cada alumno, pero eso puede llegar a suponer un trabajo inabordable para el docente.

Este documento propone (y pone en práctica) una solución en la que un software específico genera diferentes ejercicios para cada alumno de forma automática y corrige automáticamente el resultado, facilitando enormemente el trabajo del profesor.

### Antecedentes

Si bien el sistema resulta ahora especialmente interesante dada la situación de pandemia vivida y los posibles cambios futuros de la docencia, la idea original no está asociada a la misma y resulta igual de útil en condiciones normales de funcionamiento, ya que puede ser aplicada a ejercicios de evaluación continua virtual a través de plataformas tipo Moodle.

Originalmente se realizaban una serie de prácticas por ordenador de simulación de circuitos, componentes y sistemas eléctricos de forma presencial en un laboratorio del departamento. Estas prácticas implicaban la utilización de un software propietario que requería de una licencia/mochila USB

para su funcionamiento. Este requisito planteaba dificultades adicionales de tipo logístico en relación al horario de disponibilidad, ya que el laboratorio de simulación era utilizado también por otras asignaturas. Esto significaba que los alumnos no podían disponer normalmente del laboratorio para uso personal fuera de sus horas lectivas en unos momentos en los que el alumnado demandaba crecientemente la disponibilidad de herramientas para trabajo individual.

De forma complementaria, las asignaturas con un número elevado de alumnos requieren una gran dedicación del profesor para la corrección de ejercicios de simulación que, en muchos casos, son rutinarios. Además, con un número elevado de alumnos, resulta imposible desarrollar ejercicios individualizados. En estos casos, resulta también complejo determinar qué alumnos han copiado la solución de otros compañeros.

En base a todas estas premisas, se planteó desarrollar una herramienta virtual capaz de generar y corregir de forma automática casos individualizados de cada ejercicio con escasa o nula intervención por parte del profesor. Esta herramienta debería de ser capaz de manejar diferentes parámetros de entrada e incógnitas para cada alumno y ejercicio/caso, de manera que cada estudiante deba resolver un problema personalizado.

## *Objetivos*

El objetivo de este proyecto es desarrollar una herramienta virtual para la simulación de circuitos que pueda ser utilizada de forma remota por los alumnos mediante una interface implementada en un servidor web.

La simulación de sistemas eléctricos constituye una herramienta fundamental en ingeniería, tanto en la fase de diseño como de análisis del comportamiento de sistemas existentes. La herramienta puede ser utilizada también para la generación de ejercicios personalizados en primeros cursos de teoría de circuitos, electrotecnia y fundamentos de ingeniería eléctrica.

Se espera que la herramienta les ayude, tanto a entender mejor los conceptos básicos de teoría de circuitos como, a usar software de simulación eléctrica. Además, se pretende poder hacer una evaluación continua más personalizada debido a que tanto el sistema de generación de casos como de corrección son automáticos.

## **Desarrollo y Arquitectura**

### *Fases del proyecto*

El desarrollo del proyecto ha llevado varias fases:

F1. Estado del arte. Se han revisado los entornos de desarrollo (herramientas de programación, servidores web, etc.) necesarios para implementar las necesidades funcionales del proyecto. Se ha decidido utilizar el entorno .net junto con el servidor IIS.

F2. Análisis de herramientas de simulación. Se han revisado las herramientas de simulación disponibles, tanto de tipo privativo no gratuito (Cadence, Altium, National Instruments), privativo gratuito (LTSpice, Spice OPUS) como software abierto (QUCS). En una primera fase se decidió utilizar LTSpice y QUCS. Posteriormente, y para facilitar la integración de la herramienta de simulación con sistemas de línea de comandos, se realizó un cambio a NGSpice [1]. Este simulador utiliza también un motor SPICE de Berkeley con la ventaja de que puede ejecutarse utilizando una interfaz gráfica o la línea de comandos.

F3. Desarrollo de pruebas de evaluación. Se han seleccionado pruebas de evaluación de un extenso banco de ejercicios resueltos con soluciones disponibles en Pspice. La selección de las pruebas

se realizó atendiendo al curso y contenidos para los que se programan las actividades. Las pruebas de evaluación cuentan con una plantilla Latex de enunciado, una plantilla “.cir” de simulación y un fichero Excel de configuración.

F4. Desarrollo de una interfaz web. Se desarrolló una interfaz web que integra la entrada/salida de datos al entorno de simulación, discriminando por usuarios y con seguimiento del uso de la herramienta, que sirve tanto como vía de acceso del alumno al sistema o como herramienta de seguimiento de evaluación por parte del profesor.

F5. Integración del sistema. La integración del sistema se realiza de forma *offline*. Un conjunto de ficheros Python [2] se encargan de integrar las listas de alumnos (se utiliza el formato proporcionado por el campus virtual), así como de generar los casos personalizados y sus soluciones.

F6. Pruebas del sistema. Se han realizado pruebas básicas de funcionamiento con usuarios individuales. Tras verificar el funcionamiento, se ofreció a los alumnos de dos asignaturas participar en una prueba piloto.

### Arquitectura y funcionamiento

El sistema desarrollado se basa en un código Python que genera los casos (junto con sus soluciones) y los almacena en una base de datos SQL, la cual será consultada por una página web al efecto que los alumnos emplearán para acceder a los ejercicios. La figura 1 muestra el diagrama de bloques de la generación de los casos.

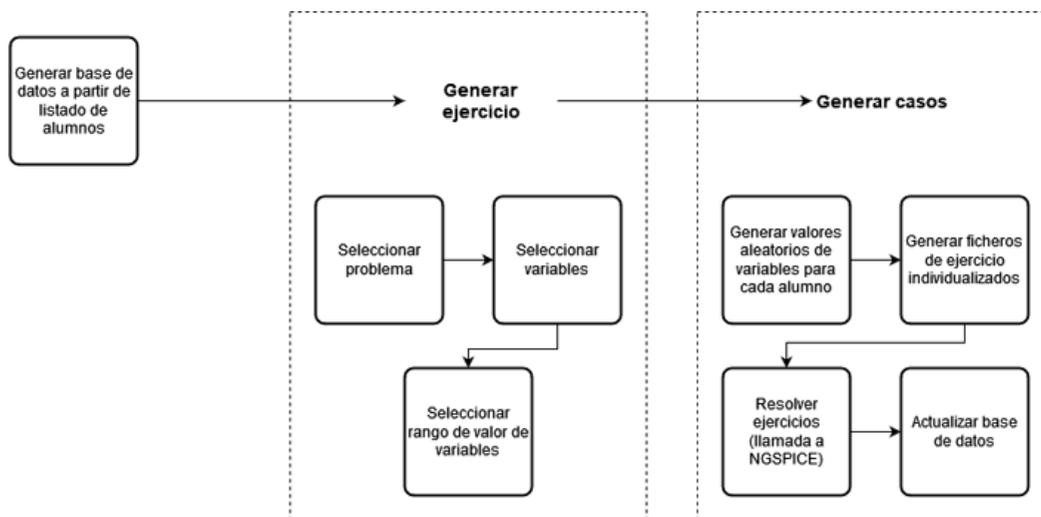


Figura 1. Generación de casos personalizados

Como se puede ver en la Figura 1, en primer lugar se genera la base de datos SQL a partir del listado de alumnos. A continuación, el docente seleccionará un problema e introducirá en el código las variables del mismo, así como sus rangos de valor aceptables. Esto último es importante para que no se acaben generando casos con valores ilógicos o que puedan causar problemas durante la resolución.

Una vez introducidos los datos, el programa se ejecuta generando valores de variables para cada alumno y ficheros de ejercicio individualizados. Los ejercicios son resueltos mediante llamadas a NGSpice y, por último, se actualiza la base de datos con toda la información.

Tras la generación de los ejercicios, estos pasan a estar disponibles para los estudiantes. La Figura 2 muestra el proceso de resolución.

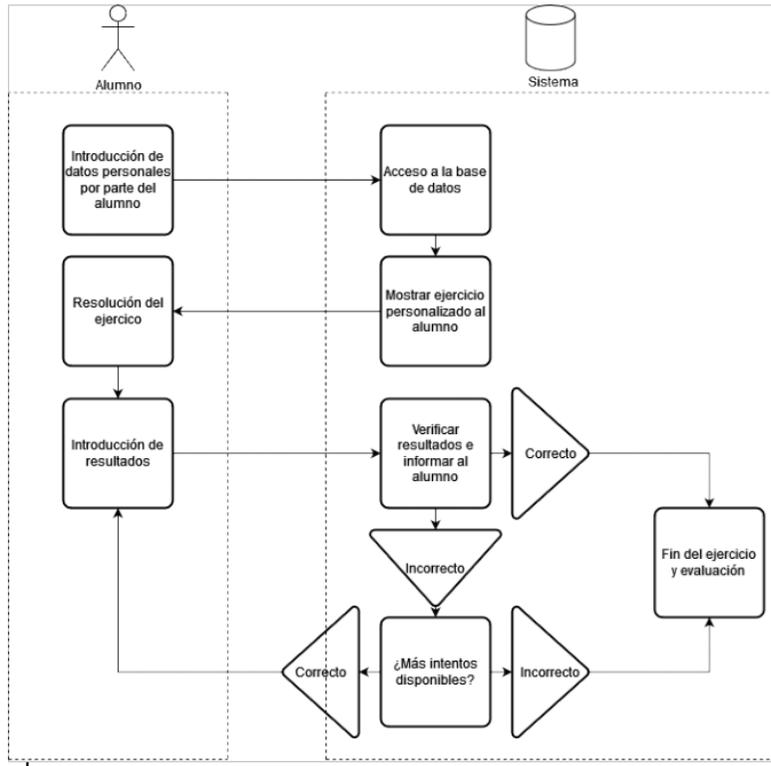


Figura 2. Proceso de resolución por parte del alumno

Una vez en la página web creada al efecto, el alumno introduce sus datos de acceso para acceder a los ejercicios que tiene disponibles. La Figura 3 muestra un ejemplo del fichero que obtiene. Tras resolver el ejercicio seleccionado, accede de nuevo para su verificación. El sistema evalúa entonces si la respuesta es o no correcta y, si procede, dará una nueva oportunidad de resolución. Para la comprobación del resultado se aplica una pequeña tolerancia al valor obtenido en simulación ya que los decimales pueden variar según la precisión aplicada durante la resolución.

*Problema: caso001*  
 Alumno: [REDACTED]  
 Asignatura: [REDACTED]  
 Fecha: [REDACTED]

El circuito de la figura 1 se encuentra en régimen permanente y alimentado en D.C. Sabiendo que los valores de los elementos activos y pasivos son:  $U_{AB}=47.0\text{ V}$ ,  $Z_1=78.0\ \Omega$ ,  $Z_2=100\ \Omega$ ,  $Z_3=89.0\ \Omega$ ,  $Z_4=1000\ \Omega$  y  $C_5=12\ \mu\text{F}$ , determinar el valor de  $I_2$ :

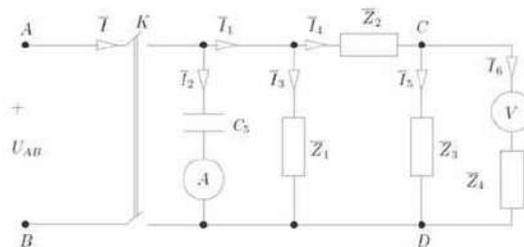


Figura 3: Ejemplo de fichero recibido por el alumno

## Resultados

En el piloto participaron un total de 102 alumnos que tuvieron la oportunidad de resolver 2 ejercicios generados independientemente para cada uno. De un total de 204 ejercicios propuestos, solo 71 fueron resueltos, lo cual representa un porcentaje ligeramente superior a un tercio. Además, 64 de los alumnos no llegaron a intentar ninguno de los ejercicios propuestos, lo cual significa que el 63% del alumnado no mostró ningún interés por la herramienta.

En la parte positiva, se puede destacar que de 33 de los 38 alumnos que mostraron interés en la iniciativa resolvieron los dos ejercicios, con lo que podemos estimar que un tercio del alumnado encontró útil el sistema propuesto. Además, estos alumnos no se limitaban a realizar un intento, sino que seguían repitiendo el ejercicio hasta dar con la solución correcta llegando en algunos casos a 5 intentos.

## Conclusiones

Se ha desarrollado y puesto en funcionamiento un sistema autónomo de generación de ejercicios personalizados para cada alumno. El sistema permite además la evaluación automática de los resultados proporcionados por los estudiantes. Sus potenciales aplicaciones son:

- Permitir a los alumnos disponer de varios ejercicios autoevaluables para trabajo personal.
- Permitir a los docentes evaluar de forma no presencial a los alumnos sin que estos puedan copiar ciegamente los resultados. En el caso de que los alumnos copien correctamente los ejercicios, habrán podido copiar solo el proceso y se habrán visto obligados a resolver para su caso particular, por lo que en el peor caso se consigue que se practique y entienda la resolución.

En cuanto a los resultados obtenidos, la respuesta ha sido bastante menor de la esperada. Pese a ello, los alumnos interesados en la aplicación si la han aprovechado al máximo, con lo que se puede hablar de un éxito parcial de la prueba piloto.

El código fuente se encuentra disponible para acceso público en un repositorio de GitHub [3].

## Agradecimientos

Este proyecto ha sido dotado con fondos del plan de innovación docente de la Universidad de Cantabria.

## Referencias

- [1] NGSpice Project website. (17 de noviembre de 2021). Recuperado de <http://ngspice.sourceforge.net/>
- [2] Python website. (17 de noviembre de 2021). Recuperado de <https://www.python.org/>
- [3] Manana, M. (17 de noviembre de 2021). *Repositorio GitHub de DEVISI*. Recuperado de <https://github.com/mmanana/DEVISI>