

Cipressus: Un sistema de gestión de contenidos para la enseñanza de sistemas digitales

Matías Micheletto

Dep. de Ing. Eléctrica y Computadoras,
IIIE Universidad Nacional del Sur - CONICET

Resumen Se presenta un software, denominado Cipressus, que permite incorporar herramientas de e-learning y autogestión para los alumnos de las cátedras de Sistemas Digitales y Diseño de Circuitos Lógicos dictadas por la Universidad Nacional del Sur para las carreras de Ingeniería Electrónica, Electricista y en Computación. Cipressus pone a disposición de los alumnos un conjunto de herramientas interactivas que lo asisten y guían en el desarrollo de sus habilidades de resolución de problemas durante el transcurso de la asignatura y resultando de esta interacción de los alumnos con el sistema, una gran cantidad de datos que permite a los docentes evaluar el desempeño del curso en general y de cada alumno en particular, destacando las habilidades individuales y preferencias de aprendizaje de cada estudiante, mediante un sistema modular y configurable.

Keywords: LCMS, TIC, PWA, Sistemas digitales, OBE.

1. Introducción

En los últimos años, se han incorporado las redes sociales más populares como medio de comunicación entre docentes y alumnos, ya que agiliza el intercambio de información referida a la asignatura. Sin embargo, el hecho de que los estudiantes acostumbran a utilizar sólo algunas redes sociales de preferencia e ignorar otras y sumado al rápido cambio de tendencias en el uso de estas plataformas, se sigue dificultando atraer la atención de los alumnos hacia el contenido, de menor o mayor relevancia, que se intenta difundir desde la cátedra. Algunas redes sociales orientan su contenido a la información multimedia o difusión de noticias, otras lo hacen mayormente a través de imágenes y también están aquellas donde predomina la mensajería de tiempo real. Esto tiene como consecuencia, que en ciertos casos la cátedra adopte varios medios de comunicación, incluyendo el correo electrónico, un medio de difusión de contenido multimedia para el envío de material de estudio y otro de mensajería de tiempo real para anuncios urgentes, por ejemplo.

Sumado al software empleado como medios de comunicación, entre docentes es común la utilización de hojas de cálculo para el registro y cómputo de las calificaciones y asistencia y el empleo de servicios de almacenamiento en la nube

o herramientas colaborativas para las tareas de evaluación de trabajos prácticos, redacción de consignas de laboratorio, exámenes, entre otras.

El framework Moodle [2], de código abierto, es adoptado por varias instituciones educativas como herramienta para docentes y alumnos y engloba muchas de las funcionalidades de las redes sociales, en el sentido de que permite la difusión de contenido multimedia, notificaciones push, organización de eventos en calendario, mensajería instantánea y combina espacios para múltiples asignaturas universitarias en un mismo sitio. Sin embargo, no siempre está disponible la posibilidad de que los docentes puedan incorporar extensiones personalizadas con herramientas propias de la asignatura que dictan, ya que se requieren conocimientos avanzados de desarrollo de software al tratarse de una aplicación *server-side*. El sistema tampoco registra ni presenta datos estadísticos sobre la actividad de los alumnos en la plataforma, y tampoco es trivial por ejemplo, obtener información sobre la descarga por parte de los alumnos del material provisto, ni algún otro tipo de *feedback*.

En este artículo se presenta el desarrollo de una aplicación web, denominada Cipressus, que tiene por objetivo reunir y combinar las funciones de todas las aplicaciones que se emplean en la gestión de un curso y en este caso particular, referido a la enseñanza de sistemas digitales. Cipressus es un proyecto modular, de código abierto y desarrollado con la intención de proveer las herramientas básicas para poner en marcha una plataforma simple y para que cada docente, con conocimientos mínimos de programación web, tenga la posibilidad de modificar e incorporar a su propio sistema, todas las herramientas necesarias.

Se muestra, en este trabajo, la incorporación de diversas tecnologías web interactivas de código abierto o de diseño propio al sistema propuesto, con la finalidad aplicar algunos conceptos de Educación Basada en Resultados (OBE, por sus siglas en inglés, Outcome-Based Learning) y de fomentar cierto nivel de trabajo independiente por parte de los alumnos, que les permita descubrir sus habilidades de resolución de problemas a partir de consignas y entornos de trabajo diseñados por el docente.

2. Trabajos relacionados

Existen numerosos trabajos en los que se propone el uso de TICs aplicados a la educación. En esta sección se mencionan aquellos que poseen características similares respecto de la propuesta del presente artículo o se aplican al estudio de sistemas digitales. A continuación se resume sus contribuciones y diferencias.

En [3] se presenta el desarrollo de un laboratorio remoto para estudiantes de teoría de control al que pueden acceder desde sus computadoras personales y realizar experimentos de control sobre una planta física instalada en el laboratorio de la universidad. El docente debe estar en línea mientras los alumnos utilizan el sistema y realizan las prácticas, pero el sistema no registra información acerca de los alumnos que se conectan, ni de qué uso le dan al sistema sino que actúa simplemente como una manera de acercar las prácticas de laboratorio a cualquier lugar donde los alumnos se encuentren.

Los trabajos [8] y [7] presentan herramientas con interfaces gráficas adaptativas que se ajustan al estilo de aprendizaje del alumno medido a partir del inventario de Felder-Silverman [4, 6] (FSLSM, por sus siglas en inglés, *Felder-Silverman Learning Style Model*). El primero es orientado a un modelo ontológico y en el segundo caso, se aplica al sistema Moodle.

En [9] se presenta un software que reúne una serie de herramientas interactivas que se denominan ICOs (por sus siglas en inglés, *Interactive Content Objects*) para el aprendizaje de sistemas digitales. El software permite a los estudiantes analizar múltiples diseños y posee herramientas que les permite entender mejor conceptos complejos del diseño digital. El sistema no dispone de identificación de los usuarios ni provee de un *feedback* para los docentes.

Por último, se mencionan dos trabajos que si bien no presentan el desarrollo de software aplicados a la educación, se analiza en éstos las ventajas de un modelo de aprendizaje activo por un lado y basado en juegos por el otro, utilizando TICs. En [5] se recolecta información de unos 225 trabajos donde se presentan datos acerca de aciertos y errores de estudiantes al ser evaluados mediante métodos tradicionales o enfocados al aprendizaje activo. Por otro lado, en [1] se analizan los resultados de aplicar una herramienta interactiva orientada a juegos denominada Kahoot y el efecto de motivación de los alumnos de ingeniería al interactuar en un entorno que fomenta cierto nivel de competencia con sus compañeros.

3. Materiales y métodos

Cypressus es una aplicación PWA (por sus siglas en inglés, Progressive Web Application) por lo que está desarrollada en lenguajes HTML, Javascript y CSS. Se trata de un sistema “serverless” que utiliza el servicio Firebase [11] para la autenticación de usuarios, que ofrece además una base de datos NoSQL, almacenamiento de archivos, mensajería y notificaciones, entre otras herramientas y de manera gratuita dentro de ciertos límites de uso [12]. Para encapsular los métodos que provee la librería de Firebase, se desarrolló un *middleware* que permite abstraer el sistema de la tecnología empleada en el *backend*. La Fig. 1 muestra un esquema simplificado de la arquitectura de los principales componentes de Cypressus.

El front-end de la aplicación utiliza una arquitectura MVC (por sus siglas, Modelo Vista Controlador) implementada con el framework AngularJS [10] y los componentes de la interfaz gráfica son provistos por el framework Materialize [15].

La aplicación web se organiza en lo que se denomina vistas, que representa el contenido dinámico que se muestra en la página principal. Cada vista tiene asociada un controlador que contiene la lógica que modela el comportamiento de dicha vista y la interacción de estos componentes se realiza por medio del *two-way data-binding* que provee Angular.

Cypressus registra para cada usuario, las vistas a las que accede y el tiempo que permanece en cada una, lo cual permite, por un lado monitorear la interac-

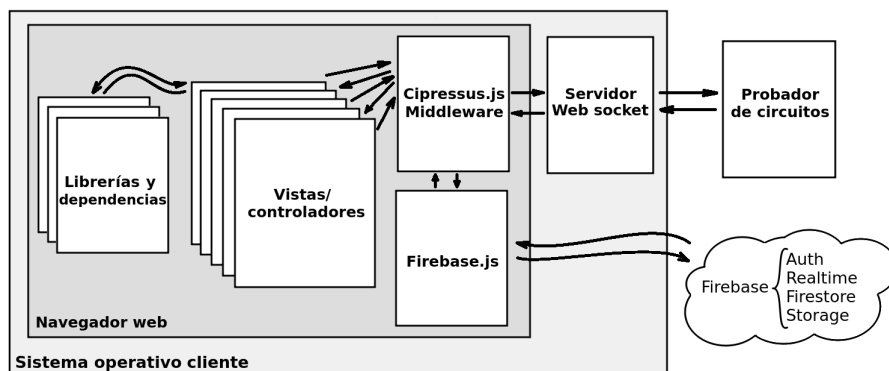


Fig. 1: Arquitectura general del sistema.

ción de los usuarios con el sistema y por el otro, iniciar la aplicación siempre en el punto donde quedó en la última sesión. Debido a que el número de usuarios conectados en simultáneo es pequeño, la base de datos permite registrar todas las acciones que los mismos realizan sin exceder los límites de transferencia de datos. Esta información puede luego ser procesada en la búsqueda de algún indicador particular sobre el desempeño del alumno, que permita al docente mejorar su didáctica en algún aspecto o incluso estimar los resultados de un examen o actividad de antemano. Las consultas a la base de datos requeridas para realizar este tipo de análisis no está actualmente implementada.

A continuación se detallan las principales funcionalidades de la aplicación y las vistas que se disponen para cada caso.

3.1. Usuarios

Los usuarios del sistema deben estar identificados ya que la interfaz gráfica se adapta al rol de cada uno. Cypressus dispone de tres tipos de roles: Visitante, Alumno o Docente. El sistema requiere de al menos un usuario con el rol de Docente, ya que actúa como administrador gestionando los roles de otros usuarios y configurando los cursos disponibles.

Cuando un usuario se registra, se le asigna el rol de Visitante mediante el cual puede acceder a las herramientas interactivas que se detallan en la Sección 3.5. Los usuarios con rol de Docente reciben una notificación del evento de registro y si corresponde a una persona que pertenece a un curso dado, se le puede asignar el rol de Alumno o de Docente, dependiendo del caso. Cuando un usuario tiene rol de Alumno, el mismo debe estar asociado a un curso particular, de manera que pueda acceder a todo el contenido relacionado a dicho curso y que se detalla en la Sección 3.3.

Cada usuario puede configurar su perfil modificando sus datos personales, número de libreta universitario o legajo docente, compañeros de comisión en

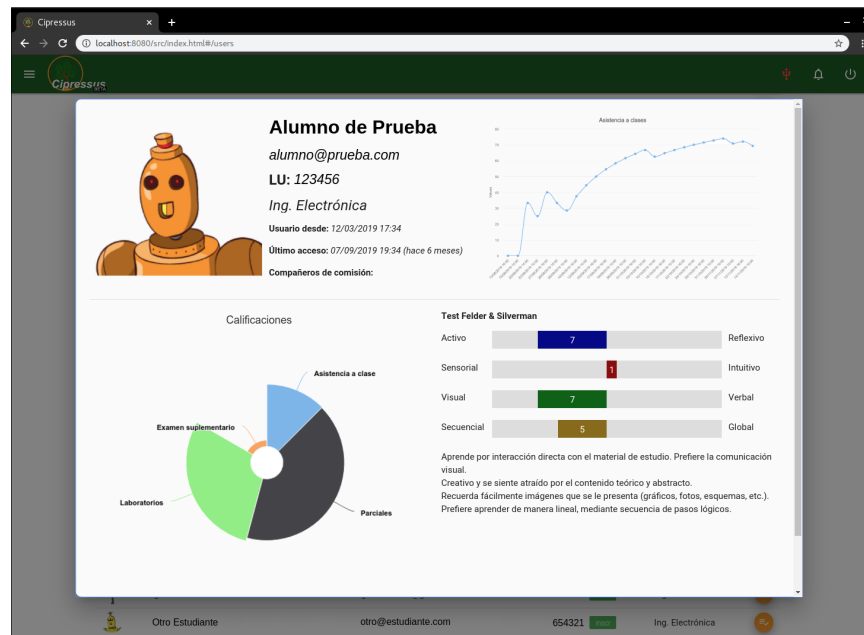


Fig. 2: Captura de pantalla del perfil de un usuario alumno.

el caso de los alumnos y foto de perfil, que inicialmente es generada mediante Robohash [17].

La Fig. 2 muestra una captura de pantalla de Cipressus con la información que ve el docente sobre el perfil de un alumno, a modo de ejemplo. Se dispone de una pequeña línea temporal que ilustra la evolución de su asistencia a clase y puede estar disponible un gráfico para ilustrar la actividad del alumno en el sistema mostrando la cantidad de accesos a las distintas herramientas. En la sección inferior de la tarjeta, se dispone un gráfico de proyección solar, generado mediante la librería HighCharts [13], que provee un panorama general de las calificaciones del estudiante. A la derecha, el gráfico de barras horizontal muestra los valores de las escalas que resultan del Test Felder y Silverman y debajo del mismo, una descripción textual del estilo de aprendizaje de preferencia del alumno.

3.2. Sistema de evaluación de alumnos

En el contexto de Cipressus, se denomina “actividad” a cada consigna propuesta por la cátedra y que el estudiante debe cumplir para obtener un incremento de su calificación, es decir, representan los resultados del modelo OBE. Dado que para cada actividad es posible evaluar diferentes aspectos, todas las actividades del curso se organizan en un esquema jerárquico y conforman un modelo abstracto denominado “árbol de actividades”. Por ejemplo, la actividad

“Laboratorio Número 1” puede estar compuesta por las sub-actividades “Implementación” e “Informe” y a la vez, la actividad “Informe” puede estar compuesta por “Presentación” y “Contenido”, donde la primer componente evalúa la prolijidad del informe presentado y la segunda, la correctitud del contenido de dicho reporte.

Los alumnos inician el curso con un puntaje inicial nulo que se va incrementando a medida que realizan las distintas actividades propuestas a lo largo del cursado de la asignatura. El puntaje obtenido por cada estudiante al finalizar el cursado puede ser utilizado para proponer a cada uno, su calificación final en la correspondiente materia.

Por otro lado, cada una de las actividades del curso tiene asociado un valor o puntaje numérico que para cada alumno será afectado por un porcentaje de calificación en escala 0 % a 100 % y que es asignado por un docente. Por ejemplo, si una actividad tiene un puntaje de 10 puntos y un determinado alumno es calificado con una nota del 70 % en la realización de dicha actividad, el alumno incrementará su calificación en 7 puntos, en escala absoluta.

Algunas actividades pueden tener asignado un vencimiento o plazo máximo de realización a partir del cual la calificación obtenida se verá afectada por el tiempo transcurrido desde dicho vencimiento hasta el cumplimiento por parte del alumno del objetivo propuesto por la actividad. El sistema permite al docente configurar, para cada actividad, una curva de decaimiento del puntaje en función del tiempo que se aplica pasado el plazo de presentación.

La estructura piramidal del árbol de actividades permite al docente distribuir la nota final del curso, que se ubicará en la raíz del árbol, entre las distintas sub-actividades a medida que se van ramificando. Si se elije un valor de 100 puntos como puntaje máximo alcanzable al final del cursado, el puntaje asociado a cada nodo del árbol representa el porcentaje que aporta la realización de la actividad correspondiente a dicho nodo a la calificación final. La Fig. 3 ejemplifica un árbol dado donde el docente deberá evaluar 10 actividades diferentes, incluyendo la asistencia a clase. En cada nodo se muestra la proporción de calificación que aporta a su nodo padre y entre paréntesis se muestra el puntaje absoluto. El algoritmo de cálculo de la calificación de los alumnos evalúa todos los nodos del árbol de actividades, ya que permite conocer el rendimiento de cada conjunto de actividades agrupadas por tema o según el esquema diseñado por la cátedra.

El docente puede reconfigurar el árbol de actividades según lo crea conveniente en todo momento, teniendo en cuenta que los puntajes de los alumnos pueden verse afectados por cada cambio.

3.3. Gestión del curso

La vista principal, llamada “home”, actúa como sección de noticias donde los usuarios con rol de Docente pueden realizar publicaciones con contenido multimedia embebido y referido a cada curso, por lo que sólo los alumnos inscriptos en dicho curso pueden ver el contenido publicado y realizar comentarios al pie de la nota.

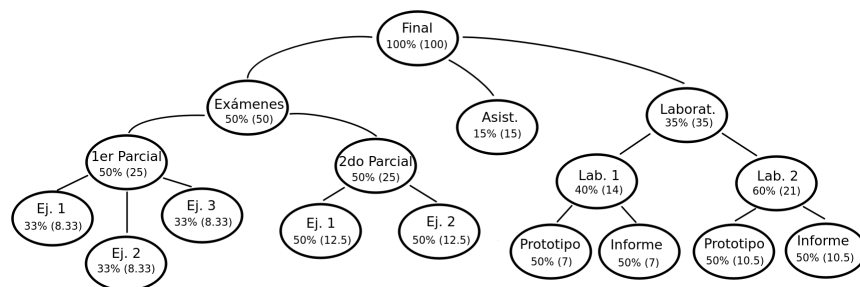


Fig. 3: Un árbol de evaluación de actividades con sus puntajes.

Los alumnos disponen de un tablero o “dashboard” interactivo con información de las actividades de la asignatura realizadas y por realizar, puntajes de cada una, porcentaje de asistencia a clase, próximos eventos y también puede visualizar, en un gráfico de barras, los puntajes obtenidos por el resto del curso, de forma anónima. Esto último permite comparar su desempeño con el de sus compañeros y si bien no se detalla información de a cuáles usuarios corresponde cada calificación, tiene el objetivo de incentivar al alumno a completar actividades en tiempo y forma para mantener un buen rendimiento.

Cada curso dispone de una lista de eventos que se muestran en un calendario para conocer las fechas de cada clase teórica o práctica, consultas particulares, exámenes y fechas de vencimiento de presentación de informes u otras actividades. Los usuarios Docentes pueden editar el calendario en todo momento.

Si bien la asistencia a clase por parte de los alumnos no es obligatoria en las asignaturas en las que se emplea el sistema actualmente, se computa en cada clase la lista de alumnos presentes para disponer, en cierta manera, de información relacionada al compromiso de cada alumno con la cátedra. El plantel docente de la cátedra puede decidir si destinar una proporción de la nota ponderado por el porcentaje de asistencia a clase o no. Para evaluar la asistencia, el sistema identifica, de acuerdo a la hora actual, el evento de calendario que está transcurriendo en ese momento, ya sea una clase teórica, práctica, consulta o examen. La interfaz gráfica muestra dos listas de ítems correspondientes a los alumnos presentes y ausentes, de manera que tildando sobre cada ítem, se alterna el estado ausente por presente y viceversa.

Luego de cada actividad de laboratorio, los alumnos deben redactar y presentar un informe a modo de reporte técnico del diseño propuesto para cada consigna. El sistema cuenta con una sección para poder subir archivos asociados a una actividad determinada y que luego el docente deberá evaluar. Dado que en general la entrega de informes tiene un plazo de vencimiento, el sistema notifica al alumno el tiempo restante para el cumplimiento del objetivo y en caso de haber transcurrido la fecha límite, se muestra el puntaje actualizado por la función de decaimiento. Como los informes se realizan en comisiones de dos o más alumnos, cuando uno de los integrantes de la comisión realiza la entrega de su informe y luego de que uno de los docentes evalúe e ingrese la calificación

de la actividad, el sistema computa la nota para todos los integrantes por igual, ya que los usuarios alumnos pueden estar vinculados como integrantes de una comisión.

3.4. Evaluación de estilos de aprendizaje

El nivel de aprendizaje alcanzado por un estudiante en clase depende en parte de su habilidad innata y preparación previa, pero también de la compatibilidad entre su estilo de aprendizaje y el estilo de enseñanza del docente [4].

El inventario FLSM permite conocer el estilo de aprendizaje de preferencia de un estudiante por medio de un cuestionario *multiple-choice* de 44 preguntas de dos opciones cada una. El inventario expresa el resultado en cuatro escalas o dimensiones cuyos rangos van de -11 a 11 puntos con incrementos de 2 puntos.

Cada una de las 44 preguntas suma o resta un punto a una escala dada dependiendo de cuál de las dos opciones se eligió como respuesta. Para evaluar el test se puede emplear un algoritmo sencillo que consiste en crear cuatro matrices de puntajes, una para cada escala, llamadas *AR* (Activo-Reflexivo), *SI* (Sensorial-Intuitivo), *VV* (Visual-Verbal) y *SG* (Secuencial-Global). Si se indica con $p : p \in [1, 44]$ a cada fila de cada matriz correspondiente a cada una de las 44 preguntas y con $c : c \in [1, 2]$ a cada una de las dos columnas correspondientes a cada opción de respuesta, entonces, el valor $M_{p,c}$, donde $M : M \in [AR, SI, VV, SG]$, indica el puntaje que aporta la opción c de la pregunta p a la escala M . A modo de ejemplo, si $M = AR$, $AR_{5,1} = -1$ significa que la opción 1 de la pregunta 5, descuenta un punto a la escala *AR*. Por otro lado supongamos que r es el arreglo de respuestas del individuo, entonces r_p puede tomar el valor 1 ó 2 dependiendo de cuál de las dos opciones eligió el individuo como respuesta a la pregunta p . De este modo, la fórmula de evaluación de cualquier escala M resulta:

$$\text{escala}(M) = \sum_{p=1}^{44} M_{p,r_p}. \quad (1)$$

Los usuarios con rol de alumno deben responder el inventario FSLM, pero no es un requisito para utilizar el sistema, por lo que en cada inicio de sesión de la aplicación se le recuerda al usuario que debe completar el cuestionario hasta que decida hacerlo. Al finalizar se muestra el resultado y éste queda accesible en la sección de consulta de perfil. Para evaluar el cuestionario a partir del arreglo de respuestas que el usuario ingresa, Cipressus emplea la ecuación 1.

3.5. Herramientas interactivas

El sistema contiene un simulador de circuitos lógicos basado en el proyecto SimcirJS desarrollado en Javascript que puede ser integrado en cualquier página web [18]. La librería permite definir nuevos componentes además del paquete básico que provee el desarrollador. Por ejemplo, en el caso de Cipressus

se incorporó un componente de salida que reproduce notas musicales de diversos instrumentos con cada flanco ascendente de sus ocho entradas digitales. Esto permite implementar un secuenciador de audio en el mismo simulador, lo cual resulta una actividad entretenida y motivante para los alumnos.

El circuito que se está simulando se modela mediante un objeto javascript que contiene una lista de los identificadores de los componentes y otra lista con las conexiones entre los pines numerados de cada componente. Esta manera de representar los circuitos es muy eficiente en términos de almacenamiento y permite guardar cualquier simulación en la base de datos del sistema, incluso ser compartida entre usuarios. Esta funcionalidad se aprovecha para proponer actividades que consisten en la entrega de simulaciones por parte de los alumnos a partir de una consigna propuesta por los docentes. La Fig. 4 muestra una captura de pantalla de la vista que contiene esta herramienta donde se puede ver la descomposición en distintas ventanas de los diferentes bloques constitutivos del circuito.

Otra de las extensiones que se añadieron al simulador es la posibilidad de calcular automáticamente las expresiones lógicas de las funciones de salida de un circuito combinatorial implementado en el mismo simulador. Para lograr esto se definieron dos nuevos componentes para el simulador que modelan entradas y salidas digitales externas, es decir que comunican el estado lógico de las mismas por fuera del motor de simulación. Luego, evaluando la salida del circuito para cada combinación de estados posibles de sus entradas, se construye la tabla de verdad del circuito combinatorial y mediante el método Quine-McCluskey se determinan las expresiones mínimas de las funciones de salida. Esta herramienta le permite al alumno verificar de manera rápida y directa si el circuito combinatorial construido en el simulador verifica el funcionamiento deseado. La mayor limitación del simulador es la incompatibilidad con dispositivos de pantalla táctil, lo cual obliga al usuario a emplear esta herramienta sólo en computadora personal o bien utilizando un dispositivo de señalización como el *mouse*.

Además del simulador de circuitos se dispone de un Mapa de Karnaugh interactivo que permite construir funciones lógicas de forma gráfica, seleccionando los casilleros del Mapa para alternar sus valores lógicos 1, 0 o “don’t care”. La librería para implementar esta herramienta fue desarrollada por docentes de la Universidad de Marburgo y está actualmente disponible on-line en su sitio web [14].

3.6. Probador de circuitos digitales

Las asignaturas en las que se aplica el sistema Cipressus, contienen en su programa la realización de actividades de laboratorio en la que los alumnos deben diseñar e implementar circuitos lógicos utilizando circuitos integrados de la serie 7400. Tradicionalmente se emplearon “probadores de circuitos digitales” de ocho canales disponibles como instrumental de laboratorio de la universidad. Estos probadores consisten en módulos que disponen de un arreglo de interruptores para controlar el estado de sus salidas lógicas y un arreglo de leds para visualizar el estado de las entradas lógicas. De esta manera es posible verificar

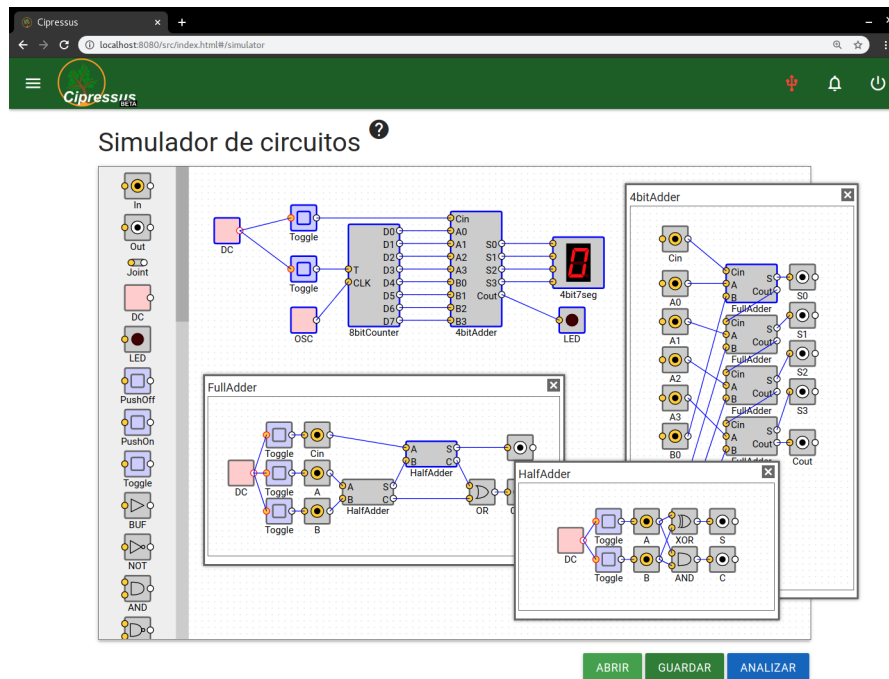


Fig. 4: Captura de pantalla del simulador integrado.

el funcionamiento de circuitos digitales combinacionales o secuenciales configurando las entradas con los interruptores del probador y verificando el estado de las salidas mediante los leds indicadores.

Como parte del mismo proyecto se desarrolló un modelo de probador basado en Arduino Nano y UNO y que para ambos casos consiste en una plaqueta tipo *shield* que permite conectar los módulos de los microcontroladores Arduino a una plaqueta de prototipos (protoboard) donde se construye el circuito digital que se desea testear. Los modelos tanto Arduino Nano como UNO poseen al menos ocho pines para ser utilizados como entradas digitales y otros ocho pines de salidas digitales lo cual equipara al modelo de probador antiguo, pero además provee alimentación de 5V desde su puerto USB a la plaqueta de prototipos, lo cual resulta ventajoso si se pretende utilizar este probador fuera del entorno del laboratorio, donde no hay disponibles fuentes de alimentación de dicho voltaje.

El microcontrolador puede comunicarse con la computadora personal por medio de un puerto serie emulado por USB, lo que posibilita controlar salidas del probador y medir sus entradas por software, es decir, realizar la verificación del funcionamiento de un circuito físicamente implementado en plaqueta de prototipos, vía software. Para que Cypressus pueda comunicarse con el probador, se desarrolló un *script* para NodeJS [16] que comunica el puerto serie de la computadora con un servidor local WebSocket, de manera que la aplicación

que se ejecuta en el navegador web, se conecta por medio de WebSockets con el script y éste a la vez con el probador por medio del puerto serie. Los detalles de este esquema pueden observarse en el diagrama de la Fig. 1.

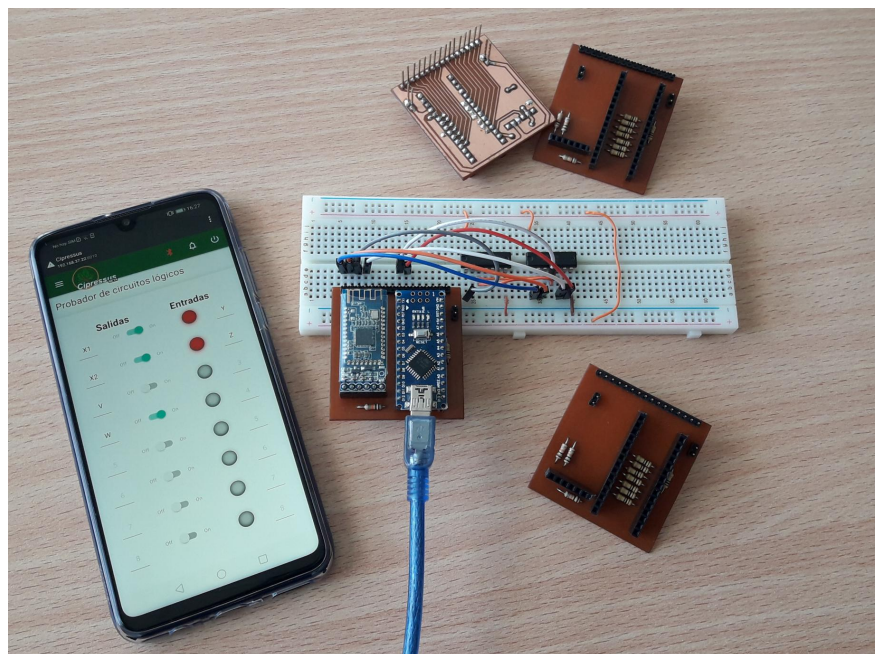


Fig. 5: Probador de circuitos digitales.

La posibilidad de vincular el probador con la aplicación web permite que los alumnos puedan realizar la verificación del funcionamiento de sus circuitos implementados y la presentación de los mismos sin la supervisión de un docente, incluso trabajando desde sus respectivos hogares. Además, es posible diseñar simulaciones de ciertas situaciones de la vida real que requieran del diseño de un circuito combinacional o secuencial, como ser un proceso industrial básico o el control de luces de semáforo en una intersección, donde el probador actúa como interfaz del circuito implementado por el alumno y la simulación del problema que está resolviendo.

El probador digital desarrollado, también dispone de un módulo bluetooth, de manera que puede operar en forma inalámbrica y gracias a las extensiones de algunos navegadores web, es posible realizar la conexión de manera directa, sin utilizar otro software externo a Cipressus, lo cual es óptimo para verificar el funcionamiento de un circuito empleando sólo un teléfono inteligente.

En la fotografía de la Fig. 5 se muestra una implementación del probador con Arduino Nano y un módulo Bluetooth BLE 4.0 AT-09 conectado a un circuito combinacional implementado con dos circuitos integrados de compuertas lógicas

de la serie 7400. El control del probador se realiza desde la aplicación web y puede ejecutarse desde un smartphone, como se aprecia en la misma imagen.

4. Resultados y discusión

La primera versión del software Cipressus fue puesta en línea en marzo de 2019 y hasta la fecha de redacción del presente artículo ha sido utilizada por 25 alumnos y cuatro docentes en tres cursos cuatrimestrales.

4.1. Impacto

Las estadísticas del sistema indican que los momentos de mayor actividad es durante las horas previas al horario de clase de la asignatura, en donde los alumnos revisan la actualización del material de estudio y los últimos anuncios publicados por los profesores.

La Fig. 6 muestra los porcentajes de uso de plataformas y dispositivos con los cuales se acceden mayormente al sistema y puede observarse que los alumnos emplean teléfonos inteligentes y computadoras prácticamente en iguales cantidades. El navegador web más utilizado es Google Chrome con 92.2% de accesos contra el 7.6% de los casos en los que se emplea Mozilla Firefox.

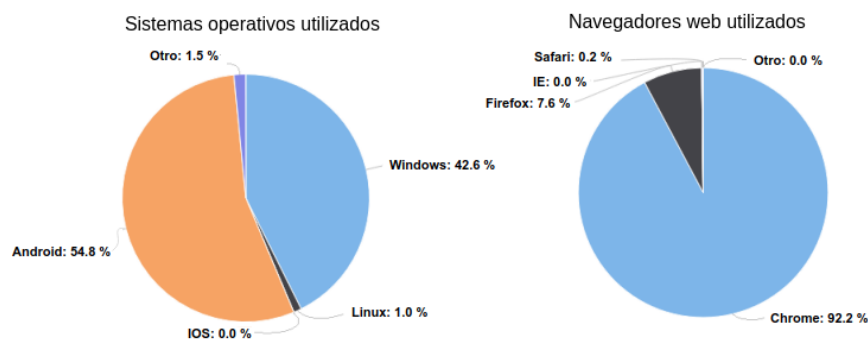


Fig. 6: Proporción de sistemas operativos y navegadores utilizados por los usuarios para acceder al sistema.

No se dispone de un registro histórico de calificaciones ni de un marco comparativo que permita obtener conclusiones válidas acerca de si existe realmente una mejora en términos de rendimiento académico de los alumnos al introducir las tecnologías educativas propuestas en el presente trabajo.

La principal diferencia comportamental que se observa en el grupo de alumnos, es la tendencia a estar mayormente pendientes de la evolución de su calificación, ya que a diferencia del uso de Moodle, del correo electrónico o las redes sociales donde las calificaciones eran actualizadas por los docentes dos o

tres veces a lo largo de todo el cuatrimestre, mediante el sistema Cipressus, los estudiantes disponen de esta información en todo momento y sus calificaciones se actualizan, en promedio, a razón de dos o tres veces por semana, dependiendo de la participación de cada uno en actividades de evaluación automática o de las veces que los docentes evalúan actividades entregadas. En cierta medida esto incentiva a los alumnos a completar los objetivos propuestos por la cátedra y a esforzarse por mantener un rendimiento dentro de la media del curso.

Los resultados de las encuestas que realiza la Universidad sobre las cátedras en las que funciona Cipressus evidencian un buen nivel de conformidad por parte de los alumnos con la herramienta y la metodología de enseñanza en general.

4.2. Perfil promedio del alumno

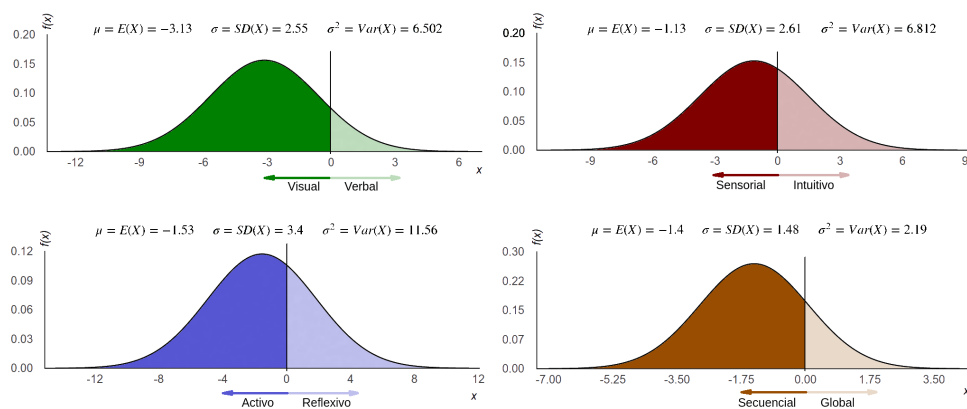


Fig. 7: Medias y desvíos de cada escala del Inventario Felder-Silverman.

De los 25 alumnos que han finalizado el cursado de la asignatura, sólo 20 han completado el Test de Felder-Silverman. Los resultados promedios y distribuciones se muestran en la Fig. 7, donde el estilo de aprendizaje que predomina, presenta las siguientes características:

- Aprende por interacción directa con el material de estudio. Prefiere la comunicación visual.
- Detallista y práctico con preferencia por hechos concretos y aplicaciones del mundo real.
- Recuerda fácilmente imágenes que se le presenta (gráficos, fotos, esquemas, etc.).
- Prefiere aprender de manera lineal, mediante secuencia de pasos lógicos.

Si bien este perfil promedio presenta una dispersión considerable en cada escala, los resultados del test de cada curso y los valores promedios históricos

de todos los usuarios de Cipressus se emplean para orientar la didáctica de las clases en función del estilo de enseñanza de preferencia mayoritaria.

5. Conclusiones y futuros pasos

La propuesta que se presenta en este trabajo hace énfasis en dos aspectos principales, por un lado en la incorporación de herramientas educativas digitales o tecnologías ad hoc, que hoy en día se encuentran al alcance de cualquier docente y por el otro, las ventajas de combinar estas tecnologías con elementos de OBE. El software permite crear un entorno de aprendizaje compuesto por el material de estudio (libros, audio, videos, etc.), las herramientas interactivas (cuestionarios, simuladores, calculadoras, juegos, etc.) y los objetivos esperados que el estudiante debe cumplir (ejercicios, exámenes, laboratorios, etc.) y que pueden verificarse vía software sin la permanente intervención del docente.

Cipressus es un sistema en continuo desarrollo que, dada su estructura modular, fácilmente permite incorporar nuevas herramientas basadas en tecnologías web sin afectar su funcionalidad original. Simplifica a los docentes la tarea de mantener todo el material referido a la cátedra en un mismo sitio, requiriendo únicamente un teléfono inteligente o computadora personal con conexión a internet. El hecho de que el sistema admite la interacción entre usuarios alumnos y docentes en todo momento y permite supervisar en forma remota el aprendizaje de los conceptos impartidos por la cátedra, logra una notable mejora en la eficiencia de la actividad docente.

Vale mencionar que en términos de desarrollo queda una gran cantidad de tareas por realizar que involucran mejoras en la experiencia de usuario, incorporación de nuevas funcionalidades y corrección de errores. Actualmente y en algunos casos, cuando se requiere de la modificación de cierta información de la base de datos, ya sea que haya sido cargada por error de un usuario u otro motivo, es necesario realizar la edición desde la consola web en forma manual, por lo que se busca independizar completamente, al usuario administrador, del *back-end* del sistema y que todas las acciones puedan llevarse a cabo utilizando únicamente la aplicación propiamente dicha. Se espera también, en el futuro próximo, lograr que el sistema pueda utilizarse, al menos en parte, sin requerir de una conexión a internet.

Finalmente, una vez que el sistema acumule un número importante de usuarios, se buscará definir estadísticas que permita a los docentes estimar el rendimiento y habilidades de cada alumno en base a su perfil de aprendizaje e interacción con el sistema, de manera que sea posible enfocar los esfuerzos en asistir a aquellos estudiantes que encuentren mayores dificultades y también mejorar el nivel general de enseñanza impartida, que podrá evaluarse desde las métricas que arroje el sistema.

Referencias

1. Criollo-C, S., Luján-Mora, S.: Encouraging Student Motivation Through Gamification in Engineering Education, pp. 204–211. Springer, Cham (04 2019)

2. Dougiamas, M., Taylor, P.: Moodle: Using learning communities to create an open source course management system. In: Lassner, D., McNaught, C. (eds.) Proceedings of EdMedia + Innovate Learning 2003. pp. 171–178. Association for the Advancement of Computing in Education (AACE), Honolulu, Hawaii, USA (2003), <https://www.learntechlib.org/p/13739>
3. Fabregas, E., Farias, G., Dormido-Canto, S., Dormido, S., Esquembre, F.: Developing a remote laboratory for engineering education. Computers & Education 57(2), 1686–1697 (September 2011), <https://www.learntechlib.org/p/50770>
4. Felder, R.: Learning and teaching styles in engineering education. Journal of Engineering Education -Washington- 78, 674–681 (01 1988)
5. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P.: Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences 111(23), 8410–8415 (2014), <https://www.pnas.org/content/111/23/8410>
6. Graf, S., Viola, S.R., Leo, T., Kinshuk: In-depth analysis of the felder-silverman learning style dimensions. Journal of Research on Technology in Education 40(1), 79–93 (2007), <https://doi.org/10.1080/15391523.2007.10782498>
7. Kolekar, S.V., Pai, R.M., M.M., M.P.: Adaptive user interface for moodle based e-learning system using learning styles. Procedia Computer Science 135, 606 – 615 (2018), <http://www.sciencedirect.com/science/article/pii/S1877050918315229>, the 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life
8. Rani, M., Nayak, R., Vyas, O.: An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowledge-Based Systems 90, 33 – 48 (2015), <http://www.sciencedirect.com/science/article/pii/S0950705115003779>
9. Wuttke, H.D., Hutschenreuter, R., Sukiennik, D., Henke, K.: Interactive content objects for learning digital systems design. In: Auer, M.E., Tsiatsos, T. (eds.) Mobile Technologies and Applications for the Internet of Things. pp. 59–69. Springer International Publishing, Cham (2019)
10. Angular framework MVW para front-end. <https://angularjs.org/>, accessed: 2020-03-10
11. Firebase servicio de back-end serverless. <https://firebase.google.com/>, accessed: 2020-03-10
12. Planes y precios de Firebase. <https://firebase.google.com/pricing?authuser=0>, accessed: 2020-03-10
13. HighCharts, gráficos interactivos JavaScript. <https://www.highcharts.com/>, accessed: 2020-03-10
14. Applet para resolver Mapas de Karnaugh-Veitch. <https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/code/karnaughmap/>, accessed: 2020-03-10
15. Materialize framework para GUIs web responsive. <https://materializecss.com/>, accessed: 2020-03-10
16. NodeJS, entorno de ejecución javascript. <https://nodejs.org/es/>, accessed: 2020-03-10
17. RoboHash, generador de avatares. <https://robohash.org/>, accessed: 2020-03-10
18. SimCirJS simulador de circuitos JavaScript. por Kazuhiko Arase. <https://kazuhikoarase.github.io/simcirjs/>, accessed: 2020-03-10