

DaVinci 1.0, Simulador de Sistemas Dinámicos Bidimensionales

Valeria I. Bertossi¹, Sonia P. Pastorelli¹
¹ Departamento de Materias Básicas, FRSF, UTN
Lavaise 610, 3000 Santa Fe
{vbertossi, spastorelli}@frsf.utn.edu.ar

Resumen. DaVinci 1.0 es un objeto de aprendizaje interactivo desarrollado en el entorno Descartes 5 por Valeria Bertossi como proyecto final de la carrera Ingeniería en Sistemas de Información. Consiste en un software de simulación que permite visualizar, mediante una animación, el campo vectorial o direccional del modelo matemático de un sistema dinámico bidimensional (autónomo o no) de variable continua, la órbita solución para las condiciones iniciales elegidas por el usuario y las gráficas representativas de las variables de estado. Su finalidad es la de ser utilizado como recurso didáctico digital para favorecer la comprensión de los sistemas dinámicos durante el desarrollo de la unidad “Ecuaciones Diferenciales” de la asignatura “Análisis Matemático II” de la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional.

Palabras Clave: Objeto de Aprendizaje Interactivo, Sistemas Dinámicos, Comprensión.

1 Introducción

En Ingeniería son habituales las situaciones en las que es necesario encontrar la solución de un sistema de ecuaciones diferenciales y si ello no es posible, al menos predecir el comportamiento a largo plazo de las mismas. En su labor profesional, es fundamental que el ingeniero tenga dominio de esta temática, pero es en la instancia de su formación académica donde dificultades de comprensión que merecen ser atendidas salen a la luz.

Sobrados son los ejemplos que exhiben los exámenes de Análisis Matemático II (AM II), en los que los alumnos hacen prevalecer tediosos métodos analíticos, habilidades de rutina aplicadas mecánicamente, que conducen incluso a resultados visiblemente incorrectos, sin detectar siquiera que su respuesta es imposible, producto de un error algebraico menor, un signo, un error en una suma o una distracción a la hora de hacer las cuentas. Dichas evaluaciones ponen de manifiesto desempeños que están basados en conocimientos y modos de pensar no disciplinarios, que en algunos casos llegan al extremo de ser intuitivos, poco reflexivos y no estructurados. No son integradores ni críticos, revelan que la validación del trabajo propio no es autorregulada, sino que depende de autoridad externa y demuestran un uso poco flexible de conceptos a los que les cuesta transferir a otros contextos. Seguramente esta forma de proceder replica y se deriva del tratamiento cuantitativo que hacen los libros de texto y pone de manifiesto un conocimiento ingenuo, ritual e inerte, que caracteriza a la categoría más baja de la comprensión [1], [2].

Es consabida en el ámbito académico la influencia de la tecnología en pos de propiciar el aprendizaje y la enseñanza de la matemática. Stone Wiske [2] afirma que puede perfeccionar y enriquecer los desempeños de comprensión de diversas maneras; y, por su parte, el NCTM concuerda en que “proporciona visualizaciones convenientes, exactas y dinámicas permitiéndole a los estudiantes ampliar la gama y calidad de sus investigaciones matemáticas” [3], a la vez que declara que “los docentes deben estar preparados para determinar cuándo y cómo sus estudiantes pueden utilizar las herramientas tecnológicas con mayor eficacia” [4].

1.1 Una necesidad a satisfacer

Bajo un enfoque cualitativo, y en línea con los intereses del proyecto de investigación “El Uso de Sistemas Algebraicos de Cómputos (SAC), Análisis de su Incidencia en la Comprensión de Matemática en Carreras de Ingeniería de la FRSF” la cátedra de AM II ha realizado algunas experiencias en el aula y en talleres usando el SAC Mathematica. Pero esta facultad cuenta con unas pocas licencias, lo que constituye un obstáculo para que los alumnos lo utilicen libremente. Además, la producción de las visualizaciones es llevada a cabo por el docente, y no por los estudiantes, porque el dominio de la sintaxis requerida impone una agenda inviable.

De la indagación en los anales de los últimos diez congresos EMCI se han encontrado publicaciones [5], [6], [7], [8] que dan cuenta del uso que se hace de las TIC en la enseñanza de ecuaciones diferenciales. Sin embargo,

estas comunicaciones se limitan a socializar diferentes herramientas informáticas, tales como Maxima, Octave, Maple, MatLab, Mathematica, Graphmatica, Graph, entre otras, empleadas para visualizar el campo direccional o vectorial de una EDO al cual se le superpone la gráfica de la solución particular para un valor inicial, obtener raíces de un polinomio para la ecuación característica y resolver sistemas de EDO usando comandos específicos. Esto muestra a las claras que junto a las convenciones del lenguaje matemático se suman las del propio software (SW) utilizado. En síntesis, el alumno debe codificar en un lenguaje de programación con la consecuente carga extra de aprender su sintaxis, situación que agudiza la dificultad cognitiva de hacer foco en lo conceptual.

Lo expuesto evidenció la necesidad de un SW educativo que satisfaga no sólo los requerimientos de funcionalidad, sino también de usabilidad y disponibilidad libre y gratuita. Así es que, en calidad de integrante del proyecto de investigación mencionado ut supra y estudiante de Ingeniería en Sistemas de Información, se propuso, como proyecto final de carrera (PFC), el desarrollo de una aplicación interactiva capaz de realizar simulaciones de modelos matemáticos de sistemas dinámicos bidimensionales de variable continua, lineales, a coeficientes constantes, autónomos y no autónomos, para utilizarla como recurso didáctico digital durante el dictado de la unidad “Ecuaciones Diferenciales” de AM II.

2 DaVinci 1.0




La construcción de *DaVinci 1.0* se llevó a cabo en dos fases bien diferenciadas. Una de índole preliminar, en la que se realizó una preselección y análisis de plataformas de desarrollo de SW. Y la segunda, consistió en el proceso de desarrollo de la aplicación propiamente dicha en el entorno seleccionado en la fase anterior.

2.1 Análisis de plataformas de desarrollo de SW

Las particularidades del producto de SW que se planteó en el PFC requerían de una plataforma específica para aplicaciones matemáticas, de licencia libre y gratuita, con soporte de las funciones del cálculo, resolución de sistemas de ecuaciones diferenciales y capacidad gráfica para visualización de planos cartesianos, curvas y campos vectoriales y direccionales. Se preseleccionaron Descartes y Geogebra por tratarse de herramientas de autor, es decir, están especialmente diseñadas para la elaboración de recursos didácticos interactivos con capacidad multimedia. Freemath, Octave y Scilab son muy similares entre sí ya que todas están inspiradas en MatLab. El último release de Freemath data del 31/06/2013; el de Octave es del 06/08/2014 y el de Scilab, del 10/02/2014. Debido a la similitud entre estas últimas se optó por evaluar Octave debido a su actualización más reciente.

En la tabla 1 se consigna información general de las tres plataformas que fueron evaluadas para la posterior selección de una de ellas.




Tabla 1. Plataformas para el desarrollo de aplicaciones matemáticas.

	 Geogebra	 Descartes	 Octave
<i>Versión</i>	4.4	5	3.8
<i>SO</i>	Windows/Mac/Linux	Windows/Mac/Linux	Windows/Mac/Linux
<i>Licencia</i>	GNU GPL	EUPL v.1.1	GNU GPL
<i>Lenguaje</i>	Java	Java	C++
<i>Web</i>	http://www.geogebra.org	http://arquimedes.matem.unam.mx/Descartes5/distribucion/	http://www.gnu.org/software/octave/

El proceso de selección se llevó a cabo mediante dos actividades complementarias: la *valoración cuantitativa* y la *valoración cualitativa*. Este esquema de trabajo se diagramó con la intención de que cada actividad subsane las falencias de la otra.

Valoración cuantitativa. La tabla 2 compendia la calificación de cada plataforma, asignada en base a una comparación objetiva y como resultado de sumar una unidad por cada recurso o funcionalidad de interés ofrecido de manera inmediata y directa por la plataforma, esto es, que no requiera ser emulado mediante algún artificio o suplantado por una serie de procedimientos. Si bien se reconoce que éste es un método de puntuación muy elemental, puesto que carece de la ponderación de cada criterio, pretende restringir las posibilidades de subjetividad en la actividad de *valoración cualitativa* y constituye un primer elemento a tener en cuenta en la decisión final de selección.

Tabla 2. Valoración cuantitativa de plataformas para el desarrollo de aplicaciones matemáticas.

Criterio de benchmarking			
<i>Ayuda:</i> Foros/tutoriales/manuales	1	1	1
<i>Programación:</i> Interfaz gráfica de programación; Manejo de excepciones; Estructuras de control: If-Else, Do-While, For, Repeat-Until	1	2	5
<i>Instalación:</i> Se libera al alumno de la instalación del intérprete	1	1	0
<i>Funcionalidad núcleo:</i> Graficador de campo vectorial; Escalamiento de los vectores del campo; Normalización de los vectores del campo; Configuración de la densidad del campo; Graficador de campo direccional; Implementación de algoritmo de resolución numérica de sistemas dinámicos autónomos; Implementación de algoritmo de resolución numérica de sistemas dinámicos no autónomos	1	6	3
<i>Facilidades para diseño de la GUI:</i> Botón; Lista desplegable, Campo editable; Control gráfico; Pulsador; Checkbox; Tab; Tooltip; Ayuda contextual; Zoom del espacio asignado a la representación gráfica; Desplazamiento del plano coordenado dentro del espacio asignado a la representación gráfica; Visualización de más de un plano coordenado en la misma ventana	9	12	3
<i>Tipos de objetos y atributos:</i> Implementación del objeto gráfico “Vector”; Implementación del objeto gráfico “Segmento”; Implementación del objeto gráfico “Punto”; Estilos aplicables a los objetos (grosor, color, transparencia); Capacidad para manipular la razón de aspecto entre los ejes coordenados	5	4	4
<i>Otras funcionalidades:</i> Soporte de animaciones; Inserción de imágenes (png/gif/jpg); Exportación a páginas “Web”; Inserción de enlaces URL; Reproducción de videos	3	5	1
	Σ 21	31	17

Valoración cualitativa. Se evaluó, mediante un procedimiento top-down, la pertinencia y relevancia para el desarrollo del objeto de aprendizaje de cada uno de los elementos de comparación entre plataformas, explorando en primera instancia las capacidades prioritarias macro, para luego adentrarse en facilidades específicas. Se hizo foco en el núcleo de funcionalidad a la que da soporte cada plataforma, luego se analizó su potencialidad de programación y se procedió a explorar el esfuerzo requerido por parte del usuario para ejecutar la aplicación resultante. De esta terna de elementos considerados quedó desestimada Octave como herramienta de desarrollo factible, debido a que sólo admite interacción por línea de comandos; lo que implica que el usuario debe tener instalado el intérprete (para lo cual se requiere que previamente lo descargue de la página web oficial) y ejecute la aplicación desde el entorno Octave. En cambio, en Descartes y Geogebra, la comunicación entre la aplicación y el intérprete es transparente para el usuario porque se realiza mediante una sentencia de código; además, la ejecución de la aplicación se efectúa en el contexto de un navegador web. Circunscripta ahora la tarea a decidirse entre Descartes y Geogebra, la evaluación se centró en la potencialidad para diseño de la GUI (Graphic User Interface), factor primordial para cubrir necesidades de usabilidad teniendo en cuenta que los principales usuarios son los alumnos. De tal análisis se concluyó que para el diseño del layout general (esquema de distribución de los elementos dentro de la interfaz), Descartes es más flexible que Geogebra. En cuanto a la exportación a páginas web, Geogebra requiere de varios pasos; mientras que en la otra plataforma el proceso es transparente para el programador porque el editor Descartes genera directamente el archivo html. Además, con miras a la producción futura de otros materiales didácticos hipermediales, Descartes provee otros dos recursos: la posibilidad de enlaces a otras páginas web y la reproducción de videos que podrían utilizarse como instrumento motivacional en la introducción de una unidad. Un aspecto a destacar, que también incidió en la elección definitiva de la herramienta de desarrollo, es haber establecido contacto con uno de los diseñadores funcionales de Descartes. El trato directo para el esclarecimiento de dudas, solución de problemas, e inmediatez en las respuestas a la hora de la capacitación en el uso de la plataforma se ha considerado una ventaja sobre la utilidad que pudieran ofrecer los foros.

2.2 Proceso de desarrollo de SW

Requisitos funcionales. Las funcionalidades aportadas al usuario (alumno o docente) se ilustran en la Fig. 1. Son denominadas “casos de uso” y se codifican de la siguiente manera: CU_{xx} *NombreFuncionalidad*. Donde, CU es el acrónimo de “Caso de Uso”, xx es un n° identificatorio y *NombreFuncionalidad* es un nombre significativo de la funcionalidad aportada por el caso de uso. En la nomenclatura se utilizó el acrónimo de “Sistema Autónomo” y “Sistema No Autónomo”, SA y SNA respectivamente.

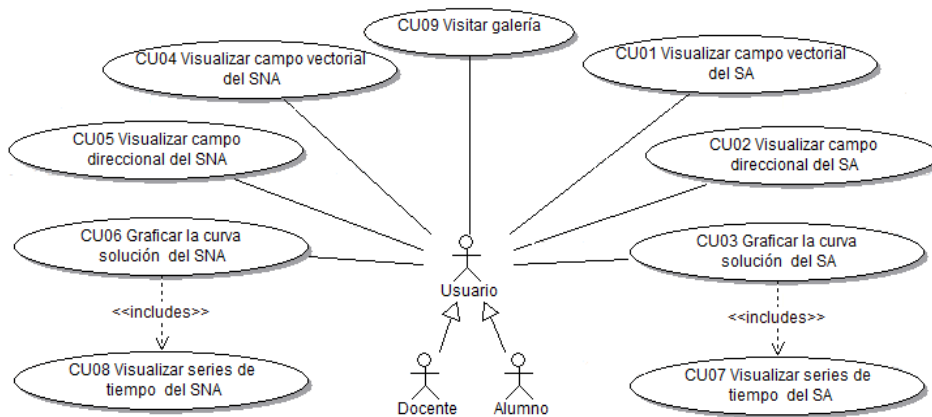


Fig. 1. Diagrama de casos de uso que modela las funcionalidades provistas por *DaVinci 1.0*.

- *CU01 Visualizar campo vectorial del SA* y *CU02 Visualizar campo direccional del SA*: Grafican el campo vectorial y el campo de pendientes, respectivamente, asociado al sistema autónomo ingresado por el usuario, cuya forma genérica es la correspondiente a las ecuaciones (1).

$$\begin{cases} x'(t) = f(x, y) \\ y'(t) = g(x, y) \end{cases} \quad (1)$$

- *CU03 Graficar la curva solución del SA*: Muestra la curva plana representativa de la solución del sistema autónomo a partir de las condiciones iniciales ingresadas por el usuario, cuya expresión genérica es (2).

$$\begin{cases} x'(t) = f(x, y) \\ y'(t) = g(x, y) \\ x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases} \quad (2)$$

- *CU04 Visualizar campo vectorial del SNA* y *CU05 Visualizar campo direccional del SNA*: Grafican el campo vectorial y el campo de pendientes, respectivamente, asociado al sistema no autónomo ingresado por el usuario, cuya forma genérica es la correspondiente a las ecuaciones (3).

$$\begin{cases} x'(t) = f(x, y, t) \\ y'(t) = g(x, y, t) \end{cases} \quad (3)$$

- *CU06 Graficar la curva solución del SNA*: Muestra la curva plana representativa de la solución del sistema no autónomo a partir de las condiciones iniciales y del instante de tiempo que haya ingresado el usuario, cuya expresión genérica es (4).

$$\begin{cases} x'(t) = f(x, y, t) \\ y'(t) = g(x, y, t) \\ x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases} \quad (4)$$

- *CU07 Visualizar series de tiempo del SA*: Grafica las curvas representativas de las variables de estado del sistema autónomo ingresado por el usuario, cuya expresión genérica es (2).
- *CU08 Visualizar series de tiempo del SNA*: Grafica las curvas representativas de las variables de estado del sistema no autónomo ingresado por el usuario, cuya expresión genérica es (4).
- *CU09 Visitar galería*: Ofrece una biblioteca de ejemplos de sistemas autónomos que contempla las distintas soluciones que refieren a diferentes tipos de equilibrio (estables, asintóticamente estables e inestables), según los autovalores que posea la matriz del sistema: complejos con parte real negativa, complejos con parte real positiva, imaginarios puros, reales negativos y distintos, reales positivos y distintos, reales de distinto signo, nulos, un autovalor real negativo y otro nulo, un autovalor real positivo y otro nulo, un único autovalor negativo, un único autovalor positivo. También ofrece una biblioteca de ejemplos de sistemas no autónomos con las clases de autovalores citados precedentemente y diferentes tipos de equilibrio.

Requisitos no funcionales. Son los atributos de calidad que reúne el SW.

- *Portabilidad*: Soporta multiplataforma, es decir, diferentes sistemas operativos. La capa de portabilidad es

ofrecida por la Máquina Virtual de Java y la aplicación es ejecutada a través de un navegador Web.

- *Extensibilidad*: Admite el agregado de futuras funcionalidades de manera modular.
- *Disponibilidad*: La aplicación está operacional en el Campus Virtual de la UTN-FRSF.
- *Usabilidad*: Según el estándar ISO 9241-11:1998 [9], “la usabilidad es el grado en el que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con eficacia, eficiencia y satisfacción en un determinado contexto de uso”. En general, los aspectos que contempla el SW para facilitarle al usuario una experiencia satisfactoria en el uso del objeto de aprendizaje, fundamentalmente al usuario-alumno, son:
 - Interactividad: todas las funcionalidades provistas hacen uso intensivo de la interacción a través del teclado y mouse, que es la esencia de un SW didáctico.
 - Carácter intuitivo: de fácil aprendizaje a través de una interfaz cuyo diseño gráfico y de interacción resultan familiares a los alumnos, tanto desde el punto de vista del lenguaje matemático como del tecnológico.
 - Asistencia: sistema de ayuda sensible al contexto.
 - Simpleza: la interfaz es “limpia”, sin acumulación de objetos; muestra nada más que lo necesario para una comprensión correcta.
 - Mínimo contenido textual: las indicaciones son mínimas y efectivas.
 - Consistencia interna.
 - El contenido icónico de la interfaz satisface las dimensiones semántica, sintáctica y funcional:
 - Semántica: Cada ícono guarda relación entre imagen y significado, es unívoco (no genera confusiones por error de concepto o tipo de resolución) y representa el mensaje de manera concisa.
 - Sintáctica: Todos los íconos de la aplicación comparten un conjunto de aspectos formales coherentes.
 - Funcional: Cada ícono guarda relación con el contexto aplicado para facilitar su legibilidad.

Requisitos específicos. La aplicación tiene dos modos de operación:

- *Modo Ejemplo*: aporta todas las funcionalidades sobre un conjunto de ejemplos definidos por defecto de sistemas lineales de EDO homogéneos e inhomogéneos, que el usuario no podrá modificar.
- *Modo Edición*: permite al usuario editar un sistema de EDO, lineal o no lineal, homogéneo o inhomogéneo, y aporta todas las funcionalidades sobre dicho sistema.

Arquitectura. La arquitectura de un SW es el conjunto de elementos que lo conforman y la manera en que se interrelacionan para satisfacer los atributos de calidad especificados. Técnicamente, *DaVinci 1.0* es un applet, es decir, está diseñado para ser embebido en una página html y operar en entornos Web. Para su ejecución, debe invocar al intérprete Descartes, que está alojado en el servidor Arquímedes de la UNAM (Universidad Nacional Autónoma de México). En la Fig. 2 puede apreciarse una perspectiva del producto de SW que se socializa en esta contribución.

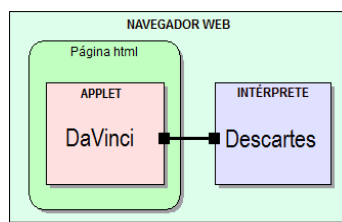


Fig. 2. Perspectiva del producto de SW *DaVinci 1.0*.

A los efectos de representar los elementos arquitectónicamente relevantes, se utilizan distintas vistas de la arquitectura para ofrecer un panorama de la estructura del sistema.

- *Vista Física*: Es el despliegue de los diferentes elementos de SW y hardware que hacen posible el funcionamiento de *DaVinci 1.0*. A un alto nivel de abstracción, se consideró conveniente representar el mapeo físico de los componentes de SW con el hardware recurriendo a estereotipos (Fig. 3) que simplifiquen la comprensión de esta vista.
- *Vista Lógica*: Al menor nivel de abstracción con el que es posible representar la arquitectura, la Fig. 4 exhibe, desde un punto de vista estático, los componentes de SW más importantes que dan soporte a la funcionalidad que *DaVinci 1.0* provee al usuario final. Los que se muestran en rosa son componentes de desarrollo propio, mientras que los contruidos por terceros se muestran en azul.
 - Interfaz de usuario: Es un componente relevante de la arquitectura debido a la cualidad altamente interactiva del objeto de aprendizaje. Incluye todos los elementos necesarios para la interacción del usuario como para la presentación de los resultados obtenidos de los paquetes graficadores.
 - Graficador de campos: tiene la responsabilidad de graficar el campo vectorial y direccional de un

sistema de ecuaciones diferenciales bidimensional, autónomo o no. Usa el componente de SW *Campo Vectorial* y el paquete *Intérprete Descartes*.

- Graficador de curvas: es responsable de graficar la trayectoria solución del sistema dinámico para las condiciones iniciales ingresadas por el usuario, así como las curvas representativas de las variables de estado. Usa el componente *Integrador Numérico* y el paquete *Intérprete Descartes*.
- Librerías de terceros: incluye los componentes de SW codificados por terceros que son utilizados por los paquetes graficadores: *Campo Vectorial* (grafica campos vectoriales estacionarios) e *Integrador Numérico* (resuelve numéricamente un sistema de EDO bidimensional homogéneo, lineal o no). Están disponibles, respectivamente, en el directorio de distribución de Descartes 5 en el servidor Arquímedes de la UNAM: <http://arquimedes.matem.unam.mx/Descartes5/macros/g2d/vectores/campVect> y <http://arquimedes.matem.unam.mx/Descartes5/macros/g2d/calculos/RungeKutta>
- Intérprete Descartes: es un componente, también creado por terceros, disponible en <http://arquimedes.matem.unam.mx/Descartes5/distribucion/>, que interpreta los objetos cartesianos codificados para que el usuario pueda visualizarlos e interactuar.

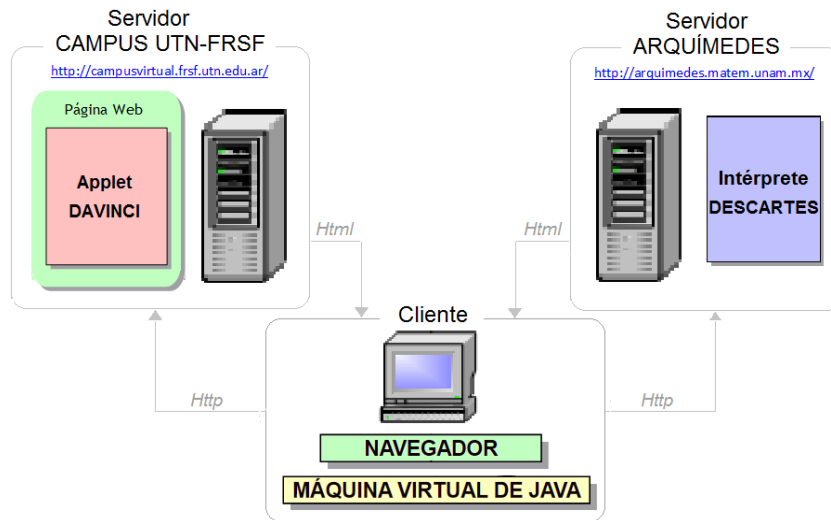


Fig. 3. Vista Física de la arquitectura de *DaVinci 1.0*.

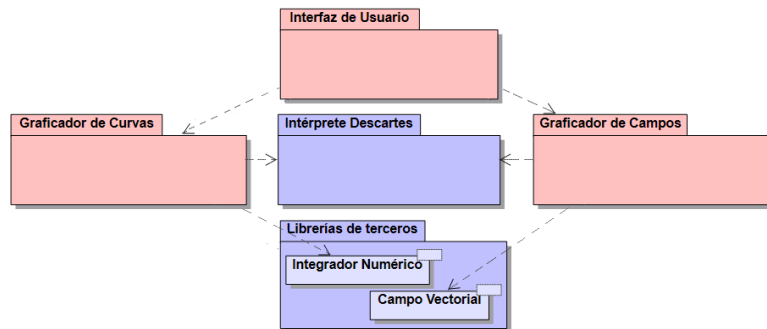


Fig. 4. Vista Lógica de la arquitectura de *DaVinci 1.0*

Diseño de la GUI. La Fig. 5 muestra el esquema general de la interfaz gráfica de usuario y en las Fig. 6 a 8 y 12 a 14 del apartado 2.3 puede apreciarse la pantalla de *DaVinci 1.0*. Los distintos elementos que allí aparecen se explican a continuación:

- *Plano de Fase:* Escena donde se visualiza un plano coordenado con la gráfica del campo vectorial/direccional y la trayectoria solución del sistema dinámico.
- *Plano de las Series de Tiempo:* Escena donde se visualiza un plano coordenado con las gráficas representativas de las variables de estado $x(t)$ e $y(t)$.
- *Ecuaciones diferenciales:* Región de edición que contiene dos campos de texto para las ecuaciones diferenciales $x'(t)$ e $y'(t)$.
- *Controles A:* Campos numéricos y selectores de opciones cuya acción tiene efecto sobre lo que se visualiza tanto en el Plano de las Series de Tiempo como en el Plano de Fase.

- *Controles B*: Conjunto de botones y pulsadores que inciden sobre el Plano de las Series de Tiempo.
- *Controles C*: Conjunto de botones y pulsadores cuya acción tiene efecto sobre el Plano de Fase y sobre los modos de operación de la aplicación.
- *Controles D*: Conjunto de controles relacionados con las animaciones (animar, pausar, reiniciar, etc.).
- *Información de estado*: Región destinada a brindar información de retroalimentación al usuario:
 - Mensajes de error y de advertencia
 - Indicación del Modo en el que está operando el sistema
 - Valor actualizado de la variable independiente t durante la ejecución de las simulaciones.

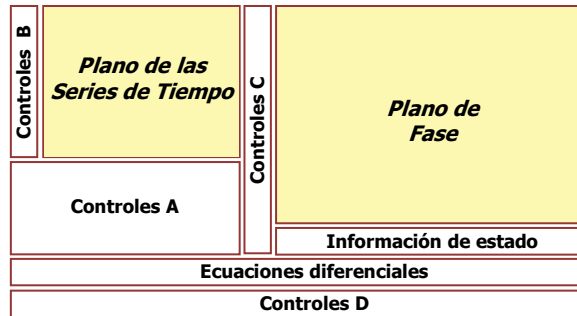


Fig. 5. Diseño de la interfaz gráfica de usuario.

2.3 Uso de DaVinci en el aula

Las demostraciones en el aula descritas en este apartado pueden ser llevadas a cabo de dos formas, incluso, simultáneamente: el docente utiliza cañón para proyectar la aplicación en la pizarra blanca y/o los alumnos llevan sus notebooks a clase y, reuniéndose en grupos, efectúan ellos mismos sus experimentaciones. Las posibilidades de uso no se agotan en los casos aquí expuestos. Durante el tratamiento de los temas, una amplia y rica variedad de ejemplos irán surgiendo de la espontaneidad del pedagogo y educando sin temor a equivocarse con los ensayos de prueba y error, situación facilitada por el carácter interactivo de *DaVinci 1.0*.

Cuando se desarrolla la teoría cualitativa, el docente puede plantear varios ejercicios para determinar la trayectoria (bajo la interpretación de línea de flujo) que seguirá una partícula que se suelta en el punto $P(x_0, y_0)$ quedando sujeta al campo de velocidad estacionario (5) y pedirle a los alumnos que calculen los autovalores de la matriz del sistema, los relacionen con la forma de la órbita, asocien las gráficas del Plano de las Series de Tiempo con la forma que tienen las ecuaciones de la solución general, caractericen el equilibrio del sistema autónomo y clasifiquen los puntos de equilibrio. Las Fig. 6 y 7 ejemplifican equilibrios estables y asintóticamente estables, respectivamente, de sistemas homogéneos.

$$\vec{v} = (x'; y') \quad (5)$$

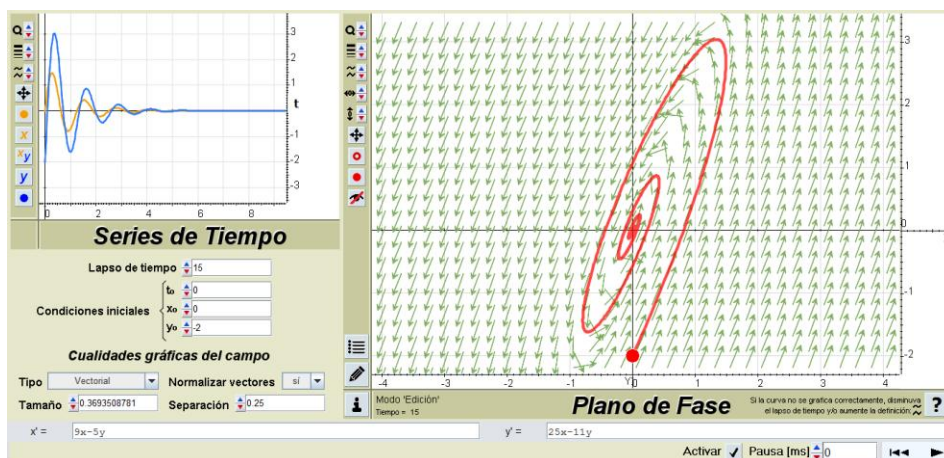


Fig. 6. Matriz del sistema con autovalores complejos con parte real negativa.

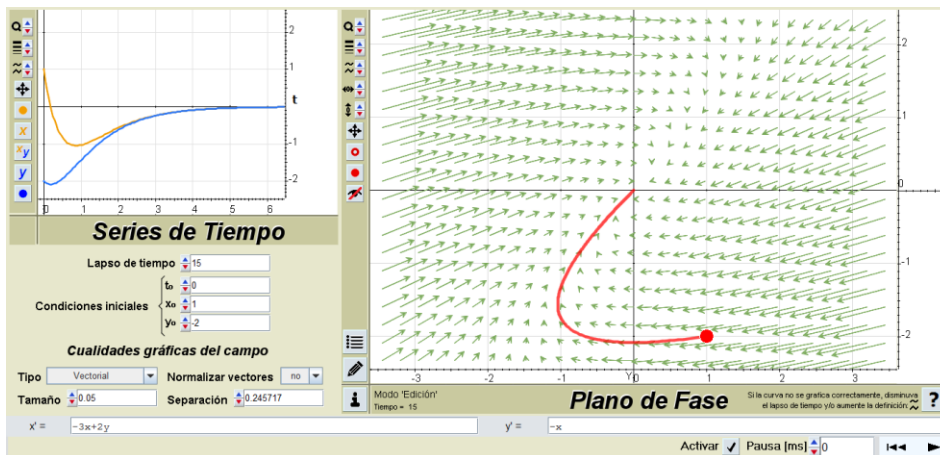


Fig. 7. Matriz del sistema con autovalores reales negativos y distintos.

Con el ejercicio de la Fig. 8, el docente puede hacer notar que si el equilibrio es inestable, cualquiera sea (x_0, y_0) (el alumnado lo puede verificar moviendo con el mouse en distintas direcciones el punto inicial de la trayectoria), si t tiende a infinito, entonces también $x(t)$ e $y(t)$ tenderán, en valor absoluto, a infinito (Fig. 9 y 10) y la partícula se alejará del origen (Fig. 11).

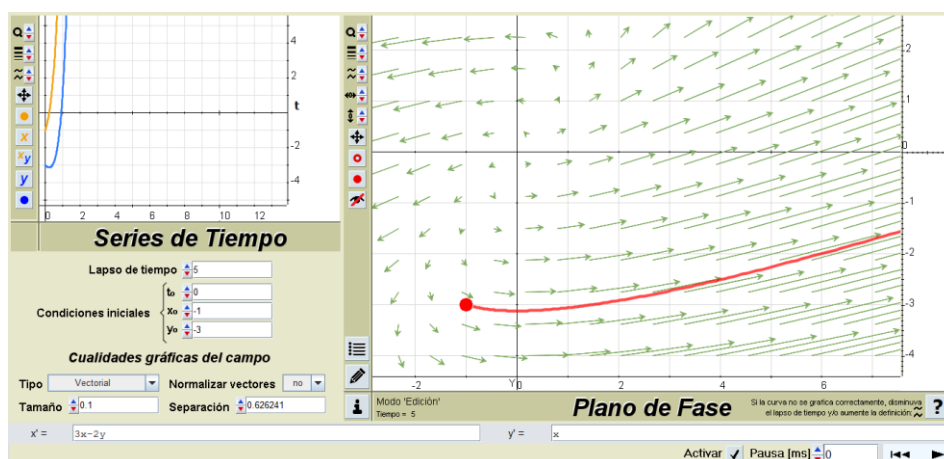


Fig. 8. Matriz del sistema con autovalores reales de distinto signo.

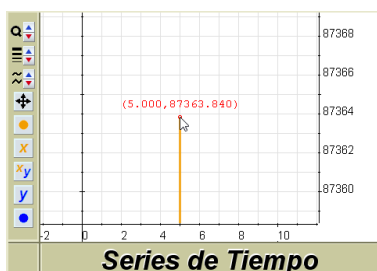


Fig. 9. $x(5)$.

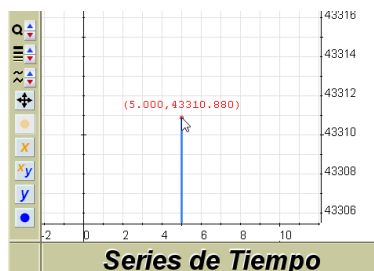


Fig. 10. $y(5)$.



Fig. 11. Partícula cuando $t = 5$.

Las simulaciones son fundamentales para aprehender el concepto de sistema no autónomo. Cuando el docente explica que este tipo de sistemas tiene órbitas dependientes del valor de t_0 , los alumnos pueden verificarlo graficando la órbita del sistema (6) mediante una animación, particularizando para $t_0 = 0$ (Fig. 12), $t_0 = 2$ (Fig. 13) y $t_0 = -2$ (Fig. 14). El campo no se muestra en ninguna de las figuras porque varía dinámicamente y debe apreciarse mientras la simulación es ejecutada.

$$\begin{cases} x' = 0,1x - y - 2 \\ y' = 5x - 0,2y + t \end{cases} \quad \begin{matrix} x(t_0) = 1 \\ y(t_0) = -1 \end{matrix} \quad (6)$$

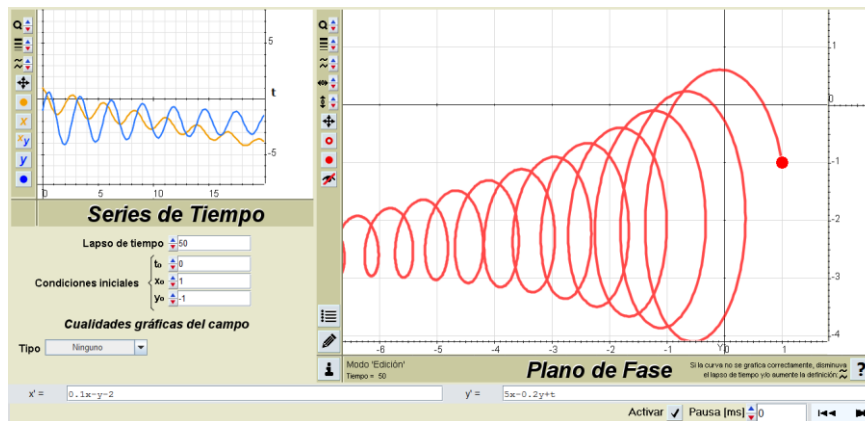


Fig. 12. Trayectoria solución para $t_0 = 0$.

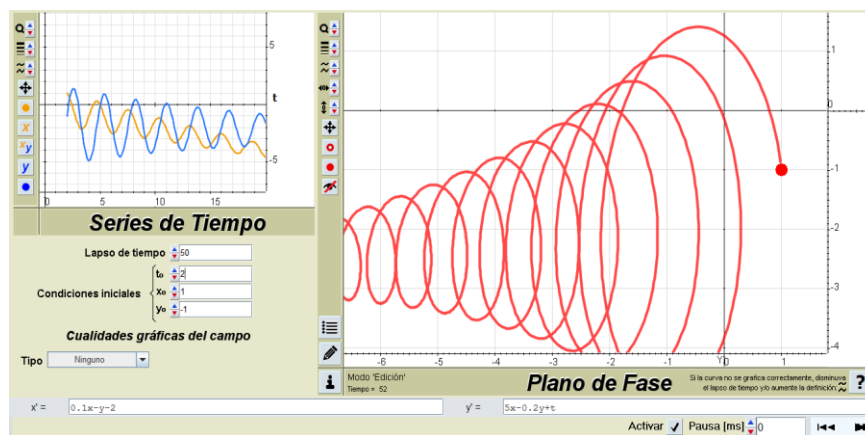


Fig. 13. Trayectoria solución para $t_0 = 2$.

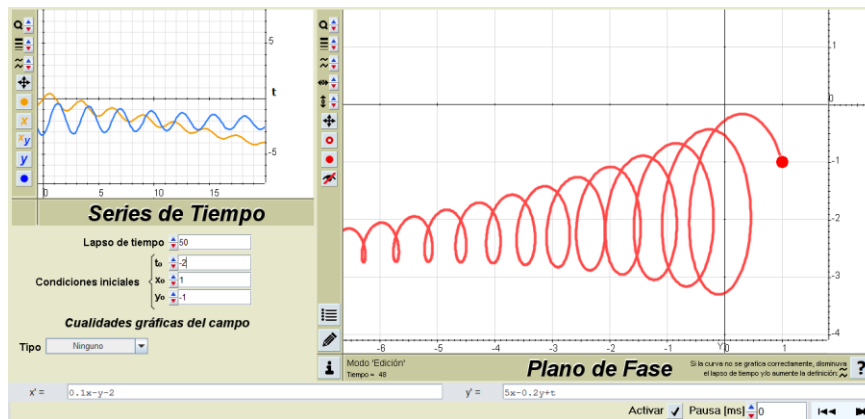


Fig. 14. Trayectoria solución para $t_0 = -2$.

3 Conclusiones y trabajos futuros

La aplicación presentada en esta comunicación aporta a la cátedra AM II una herramienta de simulación alineada con el objetivo consignado en su programa sintético dentro del plan de estudio de todas las carreras de Ingeniería [10], [11], [12], [13], [14] de la UTN: “capacitar al estudiante en el uso de herramientas computacionales que permitan la solución de los problemas de análisis, la presentación gráfica asociada a ellos y la simulación de

modelos planteados con ecuaciones diferenciales. [...] Se usarán en las prácticas paquetes de computación que permitan cálculos numéricos y simbólicos con capacidad gráfica. En el caso de ecuaciones diferenciales se instruirá al alumno en el uso de un paquete interactivo que permita la simulación y el análisis de los resultados”.

Las expectativas de la cátedra son muy optimistas. Se considera que el uso de este recurso informático permitirá optimizar el aprovechamiento de los tiempos académicos, mejorará los desempeños de comprensión actuales, e incluso servirá como un instrumento para determinar si los problemas de comprensión habituales son superados. Además, la aplicación ya está subida al campus de la facultad en el espacio asignado a la asignatura para ser utilizada durante la cursada 2015, de modo que, con unos pocos y muy simples pasos de instalación, los alumnos dispondrán de una herramienta interactiva a la que pueden acceder de forma gratuita y que les permitirá hacer sus propios ensayos y comprobaciones tanto en sus casas como en el aula, junto al docente.

Respecto del proceso de desarrollo de la aplicación en sí, cabe aclarar que se tuvo especial cuidado en el diseño de la interfaz. Ésta fue la actividad que implicó el mayor desafío dentro de la etapa de diseño. Se buscó armonía en las proporciones de los elementos que la componen, se pensó cuidadosamente en la distribución de los campos de entrada, botones, colores y diseño de los íconos; de modo que no sólo cuente con un componente estético visualmente agradable, sino que el layout contribuya a una navegabilidad lo más simple e intuitiva posible, máxime teniendo en cuenta que los principales usuarios serán alumnos de segundo año de ingeniería y lo que se pretende es que hagan foco en los aspectos conceptuales de los sistemas dinámicos y no que tengan que sortear escollos al usar un recurso informático cuya finalidad es, justamente, servirles de apoyo al estudio.

Como trabajo futuro, se planea agregar la funcionalidad de devolver los autovalores del sistema dinámico ingresado por el usuario. Además, se está pensando en realizar las adaptaciones necesarias para usar un intérprete Descartes diferente, desarrollado en Javascript, de manera que la aplicación pueda ser ejecutada en tablets y smartphones. Las intenciones de este cambio apuntan a que los alumnos puedan utilizar la herramienta en el aula a través de sus celulares, sin necesidad de llevar sus notebooks.

Referencias

1. Pastorelli, S.: Uso de los sistemas algebraicos de cómputo como recurso didáctico para favorecer la comprensión de los tópicos matriz pseudoinversa y uso de los cuadrados mínimos en el estudio de los sistemas lineales en el ciclo básico universitario. Tesis de maestría en didácticas específicas mención matemática. Facultad de Hds. y Cs. UNL (2007)
2. Stone Wiske, M.; Franz, K.; Breit, L.: *Enseñar para la Comprensión con Nuevas Tecnologías*. Paidós, pp. 87 (2006)
3. National Council of Teachers of Mathematics: The use of technology in the learning and teaching of mathematics. http://www.hpedsb.on.ca/ec/services/cst/elementary/math/documents/nctm_use_of_technology.pdf. (2003). Accedido el 11 de Mayo de 2015
4. National Council of Teachers of Mathematics: Technology in Teaching and Learning Mathematics. <http://www.nctm.org/Standards-and-Positions/Position-Statements/Technology-in-Teaching-and-Learning-Mathematics/>. (2011). Accedido el 11 de Mayo de 2015
5. Cafferata Ferri, S.; Campillo, A.; Srour, Y.: La modelización de problemas y su implementación en las clases de matemática. *Actas del XVII EMCI Nacional y IX EMCI Internacional*, D:\Trabajos Area 2. Aplicaciones de la Matemática\Cafferata Ferri, S._ Campillo, A._ Srour, Y. - LA MODELIZACIÓN DE PROBLEMAS Y SU IMPLEMENTACIÓN.pdf (2012)
6. Haye, E.; Córdoba, L.: Uso de recursos informáticos en temas de ecuaciones diferenciales. *Actas XVII EMCI Nacional y IX EMCI Internacional*, D:\Trabajos Area 3. Experiencias de Cátedra\Haye, E._ Córdoba, L. - USO DE RECURSOS INFORMATICOS EN TEMAS DE ECUACIONES DIFERENCIALES.pdf (2012)
7. Cuadros, P.; Millán, Z.; Gil, Y.: Una clase de EDO con uso de nuevas tecnologías. *Actas del XIV EMCI Nacional y VI EMCI Internacional*, D:\extensos\3_UNA CLASE DE EDO CON USO DE NUEVAS TECNOLOGÍAS.pdf (2008)
8. Priemer, N.; Lazarte, G.: Campos de pendientes en la solución de ecuaciones diferenciales. *Actas del XIII EMCI Nacional y V EMCI Internacional*, D:\HTML\extensos\Inv_educative\Inv_Educ_\086-Campos_de_pendientes_en_la_solucion_de_EDOs-Extenso.pdf (2006)
9. ISO 9241-11:1998 “Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability”
10. UTN. <http://csu.rec.utn.edu.ar/docs/php/salida.php3?tipo=ORD&numero=1030&anio=0&facultad=CSU>. Accedido el 11 de Mayo de 2015
11. UTN. <http://csu.rec.utn.edu.ar/docs/php/salida.php3?tipo=ORD&numero=1026&anio=0&facultad=CSU>. Accedido el 11 de Mayo de 2015
12. UTN. <http://csu.rec.utn.edu.ar/docs/php/salida.php3?tipo=ORD&numero=1114&anio=0&facultad=CSU>. Accedido el 11 de Mayo de 2015
13. UTN. <http://csu.rec.utn.edu.ar/docs/php/salida.php3?tipo=ORD&numero=1027&anio=0&facultad=CSU>. Accedido el 11 de Mayo de 2015
14. UTN. <http://csu.rec.utn.edu.ar/docs/php/salida.php3?tipo=ORD&numero=1150&anio=0&facultad=CSU&pagina=1>. Accedido el 11 de Mayo de 2015