

EST 2016, 19° Concurso de Trabajos Estudiantiles

Objeto Digital Educativo para Simular Modelos Matemáticos de Sistemas Autónomos y No Autónomos

Valeria Bertossi¹Directora del Proyecto de Investigación y del Trabajo Final: Mg. Sonia Pastorelli¹
Directora del Trabajo Final: Dra. Mariel Ale²¹Dpto. Materias Básicas – FRFSF – UTN, Lavaise 610, Santa Fe (S3004EWB), Argentina
{vbertossi, spastorelli}@frsf.utn.edu.ar²CIDISI – FRFSF – UTN, Lavaise 610, Santa Fe (S3004EWB), Argentina
male@frsf.utn.edu.ar

Abstract. El Trabajo Final de carrera que se socializa en esta comunicación se llevó a cabo en el marco de un proyecto de investigación educativa que analizó la incidencia de los sistemas algebraicos de cómputo en la comprensión de la Matemática en carreras de ingeniería. Empleando tecnología Java se desarrolló un Objeto Digital Educativo (ODE) cuya finalidad es promover la exploración a cargo del alumno de objetos de conocimiento inherentes, no sólo a la ingeniería, sino a una variedad ciencias: los Sistemas Autónomos (SA) y No Autónomos (SNA). Este ODE permite simular modelos matemáticos bidimensionales de tiempo continuo y le facilita al estudiante el acercamiento a conceptos medulares de la teoría cualitativa de las ecuaciones diferenciales. Conceptos que irán construyendo en su estructura cognitiva la red semántica de conocimientos previos que conformará el andamiaje necesario para abordar cursos superiores y que, en su futuro desempeño profesional, deberá dominar con pericia para resolver problemas ingenieriles.

Keywords: Objeto Digital Educativo, ODE, Sistemas Autónomos, SA, Sistemas No Autónomos, SNA

1 Introducción

En Ingeniería son habituales las situaciones en las que es necesario encontrar la solución de un sistema de ecuaciones diferenciales que modela determinado fenómeno de la realidad; y si ello no es posible, al menos predecir su comportamiento a largo plazo. En el análisis cualitativo interesa el tratamiento conceptual de la estabilidad de estos sistemas; esto es, estudiar cómo difieren las soluciones bajo pequeñas modificaciones en las condiciones iniciales. Dado que en la práctica es difícil conocerlas con certeza, cobra relevancia el estudio de la estabilidad, de modo que se tengan garantías de que cambios mínimos en las condiciones iniciales no generen comportamientos cualitativos diferentes en la solución encontrada. En su labor profesional, el ingeniero deberá dominar esta temática, pero es en la instancia de su formación académica donde dificultades de comprensión que merecen ser atendidas salen a la luz.

adfá, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

Una profusión de exámenes de Análisis Matemático II (AM II) de las carreras de ingeniería de la Universidad Tecnológica Nacional – Facultad Regional Santa Fe (UTN–FRSF) evidencia que los alumnos aplican habilidades de rutina mecánicamente y no detectan siquiera que arriban a respuestas imposibles, a veces a causa de algún error algebraico menor. Ello revela desempeños basados en conocimientos y modos de pensar no disciplinarios, no autorregulados ni integradores, con dificultades para transferir los conceptos 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].

Es consabida en el ámbito académico la influencia de la tecnología en el aprendizaje de la Matemática. Stone Wiske [2] afirma que puede perfeccionar y enriquecer de diversas maneras los desempeños de comprensión. A su vez, el NCTM (National Council of Teachers of Mathematics) sostiene que los docentes deben estar preparados para determinar la oportunidad y el modo en que sus estudiantes pueden sacar mayor provecho de las herramientas tecnológicas [3].

En tal sentido, se plantearon los siguientes objetivos para el Trabajo Final (TF) que se presenta en esta contribución: 1) investigar el uso de TIC en la enseñanza de ecuaciones diferenciales en la educación superior; 2) seleccionar una plataforma de software (SW) libre y gratuita, específica para aplicaciones matemáticas y 3) desarrollar en dicha plataforma un ODE que pueda utilizarse como recurso didáctico interactivo en la unidad “Ecuaciones Diferenciales” de la asignatura AM II. Dicho ODE debe simular el comportamiento de un SA o un SNA bidimensional de tiempo continuo, a coeficientes constantes, cuyo modelo matemático es conocido.

Un SA de las características mencionadas puede modelarse a través del sistema de ecuaciones diferenciales ordinarias (1), donde x e y son funciones del tiempo; en tanto que (2) hace lo propio con un SNA, pero en este caso, las ecuaciones diferenciales dependen explícitamente de la variable independiente t , que representa el tiempo.

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

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

2 Análisis de Secuencias de Enseñanza y de Recursos Didácticos

En el desarrollo de la teoría cualitativa de las ecuaciones diferenciales, la cátedra de AM II hace uso intensivo de las interpretaciones gráficas provistas por SW. Las visualizaciones constituyen un apoyo primordial en el tratamiento conceptual de la estabilidad (se trate de SA o SNA), la discusión de la forma de la órbita y de la gráfica de las variables de estado, dado que en algunos casos no es posible hallar analíticamente la solución o se necesita de numerosos teoremas y tediosos cálculos concretos.

En el anexo I se compendia el análisis comparativo de los tres recursos digitales utilizados como herramientas facilitadoras de la comprensión del alumno y como

apoyo a la construcción de aprendizaje significativo: el sistema algebraico de cómputo Mathematica de Wolfram, un ODE de la Universidad de Belgrano [4] y un ODE de Wolfram Demonstration Project [5]. Pero las características interesantes al pedagogo que reúne cada uno de estos recursos no lo ofrece el otro, lo que le obliga a intercambiar entre uno y otro durante una secuencia didáctica para capitalizar las ventajas y obviar las desventajas del que esté utilizando en el momento.

Esta etapa incluyó también la indagación en publicaciones de los últimos diez congresos de Enseñanza de Matemática en Carreras de Ingeniería sobre el uso de TIC en la enseñanza de ecuaciones diferenciales. Sin embargo, las contribuciones [6] [7] [8] [9], se limitan a socializar herramientas tales como Maxima, Octave, Maple, MatLab, Mathematica, Graphmatica, y Graph, entre otras. En síntesis, para resolver sus trabajos prácticos el alumno debe codificar en un lenguaje de programación, con la consecuente carga extra de sumar a las convenciones del lenguaje matemático las del propio SW, lo cual agudiza la dificultad cognitiva de hacer foco en lo conceptual.

3 Análisis de Plataformas de Desarrollo de SW




Las particularidades del producto de SW a desarrollar en el presente TF 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 ordinarias y capacidad gráfica para visualizar planos cartesianos, curvas y campos vectoriales y direccionales.

Como plataformas factibles para desarrollar el ODE 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.

Otras herramientas exploradas fueron Freemat, Octave y Scilab. Pero como son muy similares entre sí porque todas están inspiradas en MatLab, se optó por evaluar Octave ya que al momento de realizar la inspección contaba con las actualizaciones más recientes: el último release de Freemat databa del 31/06/2013; el de Octave era del 06/08/2014 y el de Scilab, del 10/02/2014.

En la tabla 1 se consigna información general de las tres plataformas preseleccionadas. Posteriormente, se realizó la elección definitiva a través de un proceso de selección que consistió en dos actividades complementarias, de modo que cada una de ellas subsane las falencias de la otra: la “valoración cuantitativa” y la “valoración cualitativa”.

Tabla 1. Plataformas preseleccionadas 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>Lenguaje</i>	Java	Java	C++
<i>Licencia</i>	GNU GPL	EUPL v.1.1	GNU GPL
<i>Web</i>	[10]	[11]	[12]

3.1 Valoración Cuantitativa

El anexo II compendia la puntuación de cada plataforma asignada en base a una comparación objetiva, en la que se sumó una unidad por cada recurso o funcionalidad de interés que ofrece de manera inmediata y directa; esto es, que la característica objeto de comparación no requiera ser emulada mediante artificios de programación o alcanzada aplicando una serie de procedimientos adicionales. Si bien se reconoce que éste es un método de calificación muy básico (porque carece de la ponderación de cada criterio) pretendió restringir las posibilidades de subjetividad en la actividad inmediata siguiente: la “valoración cualitativa”. El producido de esta actividad fue una calificación de 21 puntos para Geogebra; 31 para Descartes y 17 para Octave.

3.2 Valoración Cualitativa

Mediante un procedimiento top-down se evaluó la pertinencia y relevancia para el desarrollo del ODE de los diferentes elementos objeto de comparación entre plataformas, explorando en primera instancia las capacidades prioritarias macro (núcleo de funcionalidad a la que se da soporte, potencialidad de programación, esfuerzo requerido por parte del usuario para ejecutar la aplicación resultante), para luego adentrarse en facilidades específicas. Considerando la terna de elementos prioritarios macro quedó desestimada Octave ya que sólo admite interacción por línea de comandos, el usuario debe descargar el intérprete de la página web oficial, instalarlo y recién luego ejecutar la aplicación dentro del propio entorno. En cambio, en Descartes y Geogebra, la comunicación intérprete–aplicación es transparente para el usuario y la ejecución se efectúa en el contexto de un navegador, brindando mayor portabilidad.

Circunscripta la tarea a decidirse entre Descartes y Geogebra, se hizo foco en la potencialidad para diseño de la GUI (Graphic User Interface), factor primordial para cubrir necesidades de usabilidad. En este sentido, se observó que Descartes es más dúctil que Geogebra.

Otra consideración tenida en cuenta fue 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.

Con miras a la producción futura de material didáctico hipermedial, se observó que Descartes reúne otras dos características adicionales: enlaces a otras páginas web y reproducción de videos que podrían utilizarse como instrumento motivacional al inicio 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, hecho que se ha considerado una ventaja a la hora de la capacitación en el uso de la plataforma por sobre la utilidad que pudieran ofrecer foros y tutoriales. En esta instancia del TF, la comunidad Descartes venía desarrollando una plataforma en JavaScript; pero como estaba en versión beta se optó por codificar el ODE en el entorno Descartes 5, versión estable que utiliza tecnología Java.

4 Desarrollo del ODE

Requisitos funcionales estables, definidos sin ambigüedad y con muy baja probabilidad de volatilidad; TF de carácter unipersonal; conocimiento del dominio del problema de su única integrante por desempeñarse como auxiliar de segunda en AM II durante dos ciclos lectivos consecutivos y existencia de canales de comunicación fluidos y versátiles con el resto del plantel docente de la cátedra (circunstancia que reduce al mínimo el riesgo de imprecisión en las interpretaciones) conforman un conjunto de factores concurrentes que respaldan la elección del modelo de ciclo de vida en cascada como metodología de desarrollo de SW. Un plan de gestión de riesgos fue incorporado al plan de proyecto del TF para minimizar el impacto negativo de eventuales escenarios no deseados.

4.1 Especificación de Requisitos de SW

Los apartados de esta subsección compilan los ítems más relevantes del documento de especificación de requisitos de SW, que fue estructurado según los lineamientos del estándar ISO/IEC/IEEE 29148:2011 [13].

Perspectiva del Producto

El ODE fue denominado “DaVinci”. Técnicamente, es un applet de Java que se embebe en una página html y, en el contexto de un navegador Web, invoca al intérprete Descartes que se aloja en el servidor Arquímedes de la Universidad Nacional Autónoma de México para ejecutar los objetos cartesianos codificados. En la Figura 1 puede apreciarse una perspectiva del producto de SW.

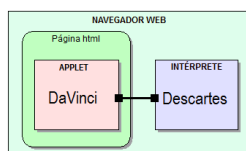


Fig. 1. Perspectiva del producto de SW.

Funciones del Producto

La Figura 2 ilustra, mediante un diagrama de casos de uso, los requisitos funcionales que se describen a continuación (en el anexo III se ofrece la especificación de los casos de uso más importantes):

- CU01 y CU02: grafican en el plano de fase el campo estacionario vectorial y de pendientes (o direccional), respectivamente, asociado a un SA.
- CU03 y CU06: muestran dinámicamente en el plano de fase el trazado de la órbita representativa de la solución del SA y SNA, respectivamente, a partir de condiciones iniciales y durante el lapso de tiempo de observación elegidos por el usuario.
- CU04 y CU05: muestran en el plano de fase la dinámica del campo no estacionario

- vectorial y de pendientes (direccional), respectivamente, asociado a un SNA.
- CU07 y CU08: grafican en el plano de las series de tiempo, a medida que éste transcurre, las curvas representativas de las variables de estado de un SA y de un SNA, respectivamente.
 - CU09: ofrece una galería de ejemplos de SA discriminados según los autovalores que posee la matriz del sistema, de modo que el alumno pueda caracterizar su equilibrio: estable, asintóticamente estable o inestable. Provee, además, un set de ejemplos de SNA que combina cada tipo de equilibrio con autovalores pertenecientes a diferentes conjuntos numéricos (reales, imaginarios y complejos).

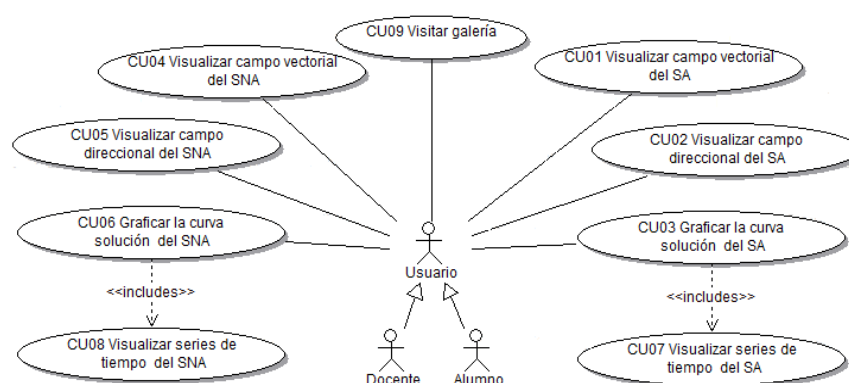


Fig. 2. Requisitos funcionales del ODE “DaVinci”

Características del Usuario

- Alumno: estudiante de segundo año de las carreras de Ingeniería de la UTN-FRSF. Se lo califica como usuario principiante, con dominio de las herramientas tecnológicas en general. Utilizará la aplicación al tiempo que irá adquiriendo el conocimiento del dominio durante el cursado de la unidad “Ecuaciones diferenciales”. Tendrá acceso a todas las funcionalidades de la herramienta.
- Docente: usuario calificado, con alto nivel de conocimiento y experiencia. Tendrá acceso a todas las funcionalidades del sistema y actuará como facilitador del alumno en el aprendizaje de la herramienta.

Restricciones

El ODE debe implementarse en la plataforma de licencia libre y gratuita Descartes por tratarse de una herramienta de autoría optimizada para la elaboración de unidades didácticas autocontenidas, tutoriales y otros materiales educativos.

Atributos de Calidad del SW

- Portabilidad: soportar multiplataforma para el acceso a través de un navegador.

- Extensibilidad: admitir el agregado de futuras funcionalidades de manera modular.
- Disponibilidad: estar operacional en el Campus Virtual de la UTN – FRSE.
- Usabilidad: atendiendo al estándar ISO 9241-11:1998 “Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability” [14], los aspectos a contemplar para facilitarle al usuario–alumno una experiencia satisfactoria son: interactividad, carácter intuitivo, ayuda sensible al contexto, interfaz limpia, mínimo contenido textual, consistencia interna y alto contenido icónico que satisfaga las dimensiones semántica, sintáctica y funcional.

Requisitos Específicos

El ODE debe operar en dos modos:

- Modo Ejemplo: aportar todas las funcionalidades sobre una galería de ejemplos definidos por defecto, que el usuario no estará habilitado a modificar.
- Modo Edición: permitir al usuario editar, según su interés, otras ecuaciones diferenciales para someter a análisis el nuevo sistema ingresado.

4.2 Descripción de la Arquitectura

A los efectos de representar los elementos arquitectónicamente relevantes se siguieron los lineamientos del Modelo de vistas “4+1” [15]. A un alto nivel de abstracción, se consideró conveniente representar el mapeo físico de los componentes de software con el hardware recurriendo a estereotipos que simplifiquen la comprensión de la vista Física. Bajando un poco el nivel de abstracción, se presenta la vista de Desarrollo siguiendo el patrón MVC. La vista Lógica, a un nivel de abstracción aún menor que la anterior, muestra, desde un punto de vista estático, los paquetes de diseño y clases más importantes que soportan la funcionalidad que el sistema provee al usuario final. No es procedente incorporar la vista de Proceso debido a que no existen mecanismos de sincronización ni concurrencia que deban ser contemplados por la arquitectura. La vista de Caso de Uso puede apreciarse en el apartado “Funciones del Producto” de la subsección 4.1.

Vista Física

Al más alto nivel de abstracción, la Figura 3 describe la arquitectura a través del mapeo físico de los componentes de SW con el hardware.

Vista de Desarrollo

En la Figura 4 se pueden apreciar los elementos que reúne el Modelo, la Vista y el Controlador:

- Núcleo funcional: implementa el conocimiento del dominio.
- Objetos de interacción: entidades perceptibles y manipulables por el usuario (campos de texto, controles gráficos, botones, pulsadores, menús desplegables).
- Escena: presentación visible al usuario de los resultados de sus acciones.

- Controlador de diálogo: controla el secuenciamiento lógico de las acciones del usuario. La lógica de la interacción, por ejemplo, determina cuándo un objeto de la interfaz es accesible al usuario o no.
- Administrador de recursos: controla los recursos involucrados en la interacción, por ejemplo, ventanas.

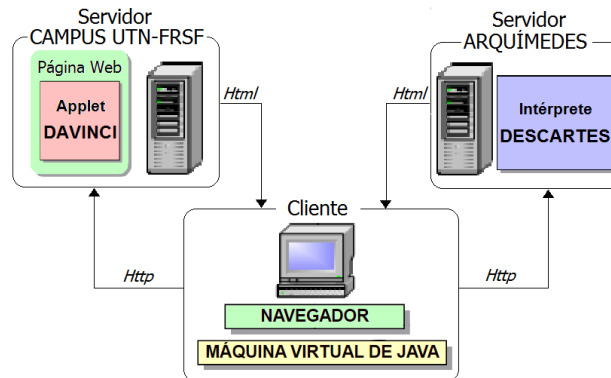


Fig. 3. Vista Física de la arquitectura.

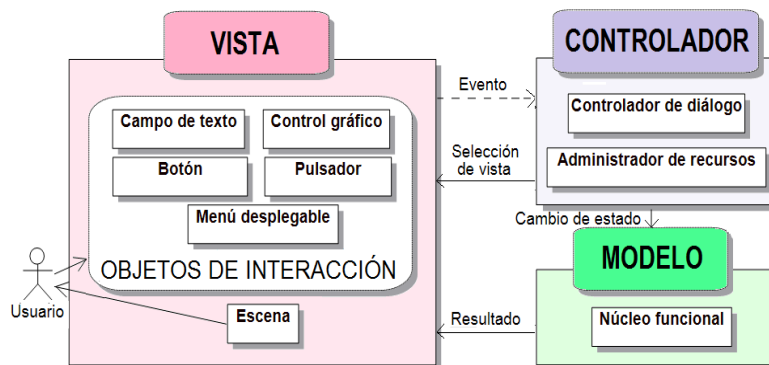


Fig. 4. Vista de Desarrollo de la Arquitectura.

Vista Lógica

La Figura 5 exhibe los paquetes de diseño más importantes. Los que se muestran en rosa son componentes de desarrollo propio; en tanto que los azules están construidos por terceros.

- Interfaz de usuario: incluye los elementos necesarios para permitir la interacción del usuario y para presentarle los resultados obtenidos de los paquetes graficadores.
- Graficador de campos: su responsabilidad es graficar el campo vectorial y el campo direccional de un sistema de ecuaciones diferenciales bidimensional, autónomo o no, lineal o no. Usa el componente de SW “Campo Vectorial” y el paquete

“Intérprete Descartes”.

- Graficador de curvas: a partir de las condiciones iniciales y el lapso de tiempo establecidos por el usuario, es responsable de graficar la trayectoria solución del sistema dinámico y las curvas representativas de sus variables de estado. Usa el componente “Integrador Numérico” y el paquete “Intérprete Descartes”.
- Librerías de terceros: incluye los componentes de SW “Campo Vectorial” [16] e “Integrador Numérico” [17], que trabajan sólo con SA. El primero grafica el campo vectorial asociado al SA y el segundo lo resuelve numéricamente.
- Intérprete Descartes: interpreta los objetos cartesianos codificados para que el usuario pueda visualizarlos e interactuar.

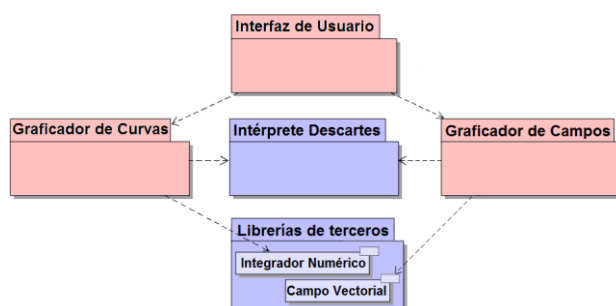


Fig. 5. Vista Lógica de la Arquitectura.

4.3 Diseño

Los entregables de la etapa de diseño se resumen en la Figura 6. La parte (a) muestra la estructura del ODE “DaVinci” a través de un diagrama de clases. La parte (b) esboza el esquema general de la GUI, que contiene los siguientes elementos:

- Plano de Fase: plano coordenado donde se grafica el campo vectorial/direccional y la órbita del sistema dinámico modelado por las ecuaciones diferenciales.
- Ecuaciones diferenciales: región de edición del modelo matemático
- Plano de las Series de Tiempo: plano coordenado donde se muestra el comportamiento de las variables de estado durante el lapso de tiempo establecido.
- Controles A: región de campos editables y selectores que permiten configurar los parámetros del simulador.
- Controles B: conjunto de botones y pulsadores que inciden sobre lo que se visualiza en el Plano de las Series de Tiempo.
- Controles C: conjunto de botones y pulsadores que hacen lo propio sobre el Plano de Fase.
- Controles D: permiten ejecutar las simulaciones, pausarlas y establecer su rapidez.
- Información de estado: zona donde se presentan mensajes de retroalimentación.

La Figura 6 (c) describe la dinámica de la aplicación en sus dos formas de operación: “Modo Ejemplo” y “Modo Edición” (en el anexo IV se ofrece un diagrama de transi-

ción de estados detallado).

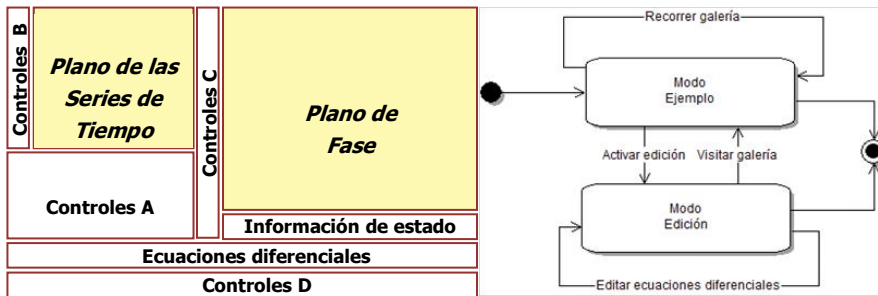
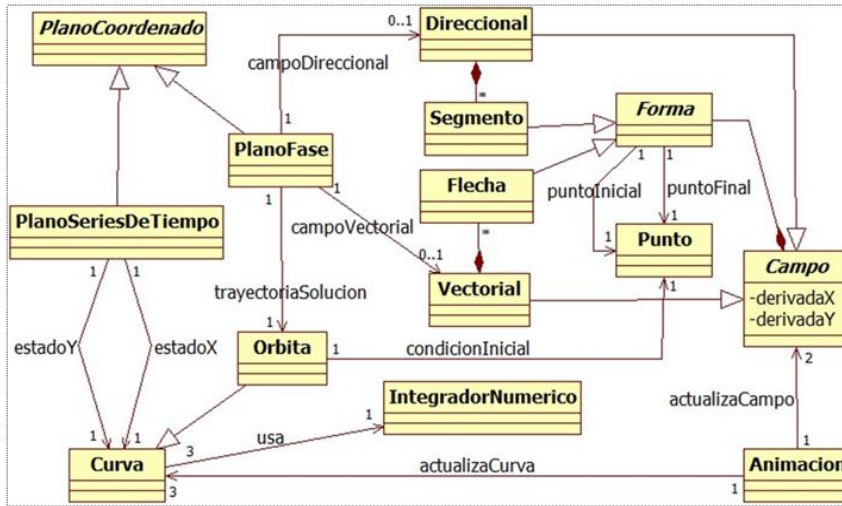


Fig. 6. . Entregables de la etapa de Diseño

5 Uso de DaVinci en el Aula

Las experiencias en el aula, algunas de las cuales se describen en esta sección, pueden realizarse de dos formas, incluso simultáneamente: el docente proyecta "DaVinci" con cañón y/o los alumnos usan sus notebooks para hacer sus propias exploraciones.

Para advertir que la solución de un SA es independiente de las condiciones iniciales se puede simular el modelo (3) desde el instante 0, 2 y -2 (Figura 7a, 7b y 7c respectivamente); y comprobar que si las trayectorias correspondieran a tres móviles, éstos recorrerían el mismo camino pero terminarían en distintas posiciones.

$$\begin{cases} x' = 2x - y; & x(t_0) = 1 \\ y' = 5x - 2y; & y(t_0) = -1 \end{cases} \quad (3)$$

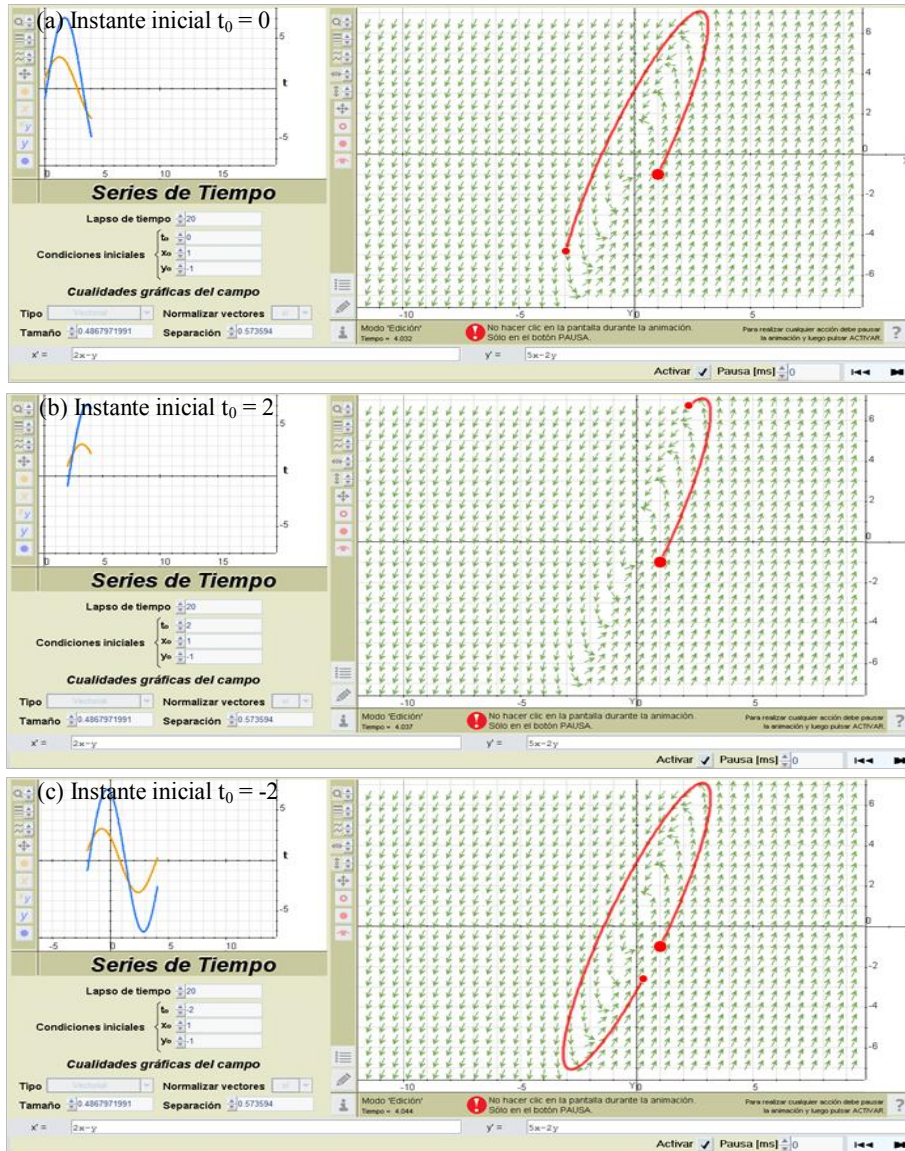


Fig. 7. Trayectorias descritas por tres móviles luego de transcurridas 4 unidades de tiempo.

Considerando el modelo (4) del movimiento armónico simple de una masa m unida a un resorte de constante elástica k (sin fuerza de fricción), la variable y representa la velocidad de m y x su posición. Con una simulación se analiza el comportamiento del resorte que el usuario–docente va pausando para sacar conclusiones (Figura 8).

$$\begin{cases} x' = y & ; x_0 = 0 \\ y' = -k/m x & ; y_0 = 1; \quad k/m = 1 \end{cases} \quad (4)$$

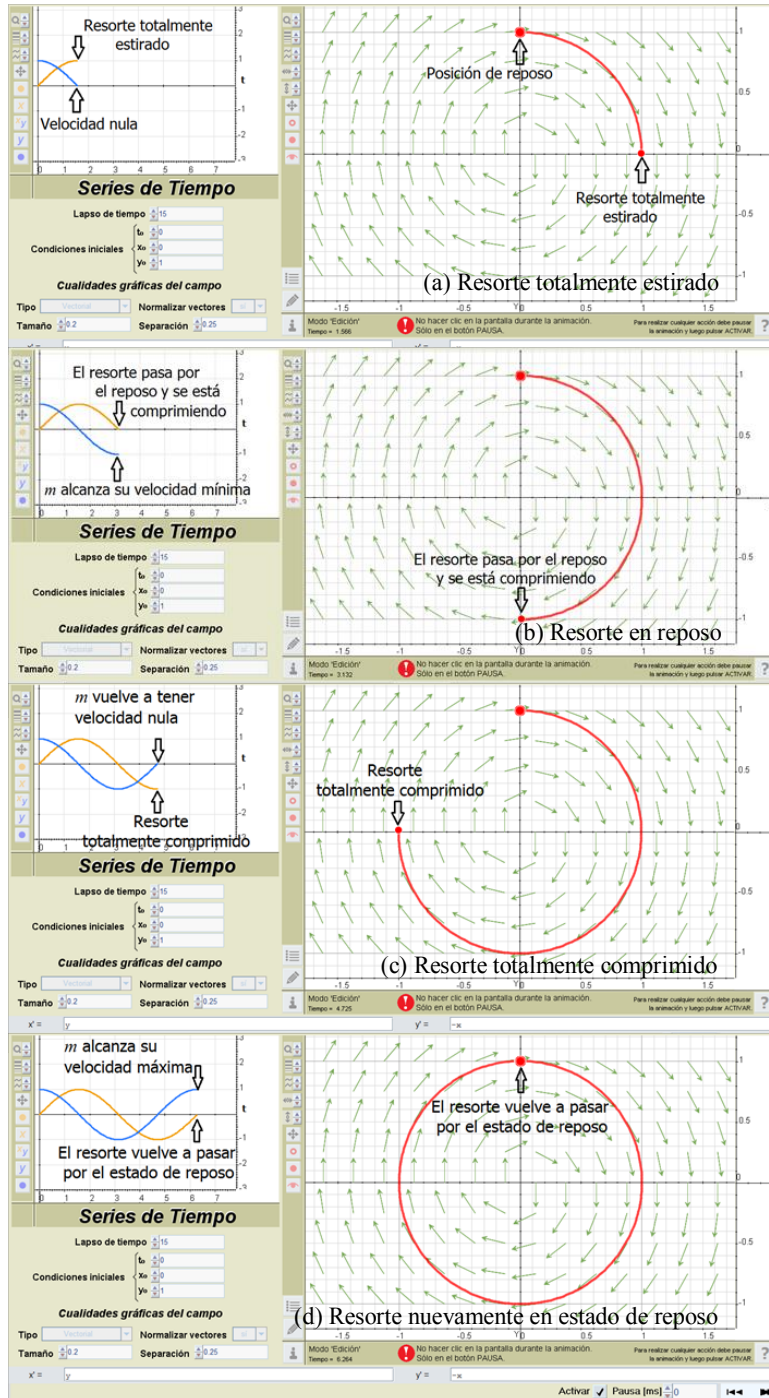


Fig. 8. Movimiento armónico simple.

Un ejemplo de SA con autovalores complejos de parte real negativa se muestra en la Figura 9. El usuario–alumno puede manipular el punto inicial y verificar que cualquiera sea éste, a medida que t aumenta, tanto x como y siempre tenderán a cero.

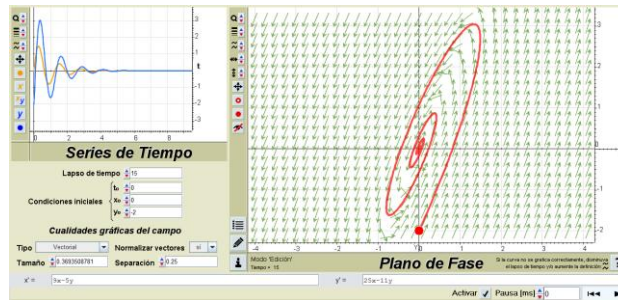


Fig. 9. SA con autovalores complejos de parte real negativa.

Para interpretar que los SNA tienen órbitas dependientes de las condiciones iniciales, el usuario–alumno puede analizar la órbita del SNA (5) que se grafica si el instante inicial es 0, 2 ó -2 (Figura 10a, 10b y 10c respectivamente). En ningún caso se muestra el campo vectorial porque se requiere del dinamismo de la simulación para apreciar que varía con el transcurso del tiempo.

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

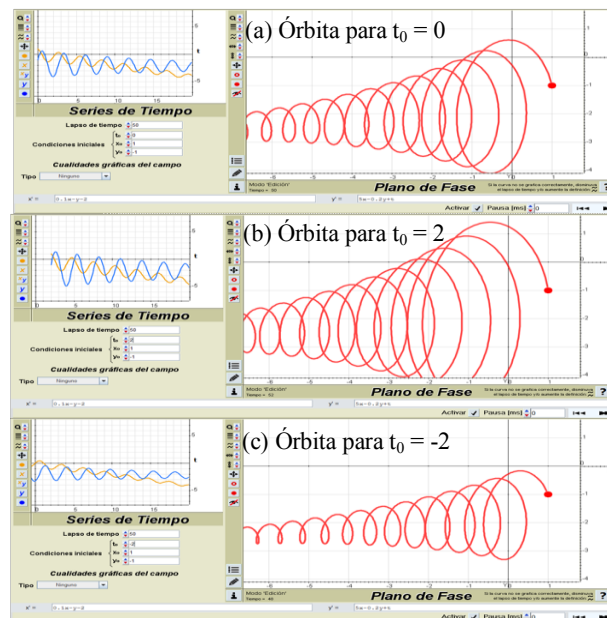


Fig. 10. Órbita de un SNA para instantes iniciales diferentes.

6 Conclusiones y líneas futuras

Este TF aporta a la cátedra de AM II de la UTN-FRSF una herramienta de simulación alineada con el objetivo consignado en su programa sintético dentro del plan de estudio de todas las carreras de Ingeniería 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”. Hacia fines de 2015 se realizó una experiencia en el aula con muy buenos resultados: permitió optimizar el aprovechamiento de los tiempos académicos durante el desarrollo de la unidad “Ecuaciones diferenciales”, contribuyó a facilitar la comprensión de los alumnos en clase y, actualmente, está siendo utilizado por una tesista de maestría, integrante del mismo proyecto de investigación educativa en el que estuvo enmarcado este TF, como un instrumento para determinar si los problemas de comprensión habituales son superados.

Como es sabido, Oracle fue imponiendo progresivamente cambios en las normas de seguridad de Java. Las continuas actualizaciones en las que se embarcó comenzaron a generar ciertas incompatibilidades entre Descartes 5 y los navegadores.

A partir de abril de 2015 fue necesario que el usuario importe en el panel de control de Java el certificado de confianza de Descartes disponible en el servidor Arquímedes de la UNAM. Por otro lado, en setiembre del mismo año, para una navegación más segura, rápida y estable, la versión 45 de Chrome deshabilitó por defecto los plugins que utilizan NPAPI (Netscape Plugin Application Programming Interface), entre ellos, el que “DaVinci” necesita. Si bien otros navegadores, permiten la ejecución de applets Java, lanzan una advertencia de vulnerabilidad que obliga al usuario a dar su conformidad cada vez que desee ejecutar “DaVinci”. Entre tanto, la comunidad Descartes siguió trabajando y liberó una versión estable del intérprete en JavaScript, lo que permitió dar solución a los problemas mencionados. Ésta es una de las ventajas de estar respaldados por una comunidad que colaborativamente actualiza las herramientas de SW para que las aplicaciones heredadas continúen su ciclo de vida.

Dado que todos los navegadores en los distintos sistemas operativos van adaptándose gradualmente al estándar HTML5, es sensato tener presente que cada uno de ellos tiene diferentes niveles de estandarización y que es factible encontrar comportamientos que dependen no sólo del navegador en cuestión, sino de su versión, de la versión del sistema operativo en el que esté ejecutando e, incluso, de la distribución de éste. Y, como es de esperar que todos vayan confluyendo hacia lo establecido, se decidió migrar el ODE a la tecnología JavaScript. Tarea que conllevó mínimos cambios en la interfaz y una leve disminución de la performance en las simulaciones con respecto a la versión en Java. Aprovechando las actividades realizadas en la fase de mantenimiento se incorporó una nueva funcionalidad: calcular y mostrar los autovalores de SA para que el alumno disponga de un elemento adicional a la hora de caracterizar su equilibrio [18]. Pero ésta no fue la única razón por la que se decidió llevar

adelante la migración; “DaVinci” debía dar cabal cumplimiento al requisito de SW definido entre sus atributos de calidad: la portabilidad. El nuevo intérprete, al ser compatible HTML5, permite que los ODE funcionen en cualquier tipo de plataforma independientemente del sistema operativo que porte: PC de escritorio, notebook, net-book, tablet o smartphone.

Actualmente, se sigue mejorando la interfaz (layout, botones y controles de mayor tamaño, diseño responsivo, etc.) para que “DaVinci” ofrezca una mejor experiencia a los usuarios de dispositivos táctiles que ocupan las aulas universitarias de hoy en día.

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 Humanidades y Ciencias UNL (2007)
2. Stone Wiske, M., Franz, K., Breit, L.: Enseñar para la Comprensión con Nuevas Tecnologías. Buenos Aires. Paidós (2006)
3. NCTM: Strategic Use of Technology in Teaching and Learning Mathematics. http://www.nctm.org/uploadedFiles/Standards_and_Positions/Position_Statements/Strategic%20Use%20of%20Technology%20July%202015.pdf (2015) (Accedido el 31/05/2016)
4. <http://www.ub.edu.ar/catedras/ingenieria/AnalisisIII/ECDIF/ECDIF1/sisauto1.htm> (Accedido el 31/05/2016)
5. <http://demonstrations.wolfram.com/TwoDimensionalLinearSystems/> (Accedido el 31/05/2016)
6. Cafferata Ferri, S., Campillo, A., Srour, Y.: La modelización de problemas y su implementación en las clases de matemática. Acta XVII EMCI Nacional. y IX EMCI Internacional (2012)
7. Haye, E., Córdoba, L.: Uso de recursos informáticos en temas de ecuaciones diferenciales. Acta XVII EMCI Nacional y IX EMCI Internacional (2012)
8. Cuadros, P., Millán, Z., Gil, Y.: Una clase de EDO con uso de nuevas tecnologías. Actas XIV EMCI Nacional y VI EMCI Internacional (2008)
9. Priemer, N., Lazarte, G.: Campos de pendientes en la solución de ecuaciones diferenciales. Acta XIII EMCI Nacional y V EMCI Internacional (2006)
10. <http://www.geogebra.org> (Accedido el 31/05/2016)
11. <http://arquimedes.matem.unam.mx/Descartes5/distribucion/> (Accedido el 31/05/2016)
12. <http://www.gnu.org/software/octave/> (Accedido el 31/05/2016)
13. ISO/IEC/IEEE 29148:2011. Systems and software engineering — Life cycle processes — Requirements engineering
14. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en> (Accedido el 31/05/2016)
15. Kruchten, P.: The 4+1 View Model of Architecture. IEEE Software, Vol. 12, n° 6, (1995) 42–50
16. <http://arquimedes.matem.unam.mx/Descartes5/macros/g2d/vectores/campVect> (Accedido el 31/05/2016)
17. <http://arquimedes.matem.unam.mx/Descartes5/macros/g2d/calculos/RungeKutta> (Accedido el 31/05/2016)
18. http://proyectodescartes.org/miscelanea/materiales_didacticos/DAVINCI-JS/index.html (Accedido el 31/05/2016)