

Jornadas de Automática

Nuevas herramientas cliente-servidor para la enseñanza en modelado, simulación y análisis de datos: IODA y WebEJS

Esquembre, Francisco^a, Sáenz, Jacobo^b, Chacón, Jesús^c, Dormido-Canto, Sebastián^b

^aFacultad de Matemáticas, Universidad de Murcia, Campus de Espinardo, 300071, Murcia, Spain

^bETSI Informática, UNED, 28040, Madrid, Spain.

^cFacultad de Ciencias Físicas, UCM, 28040, Madrid, Spain

To cite this article: Esquembre, Francisco, Sáenz, Jacobo, Chacón, Jesús, Dormido, Sebastián. 2025. New client-server web tools for teaching modelling, simulation, and data analysis. *Jornadas de Automática*, 46. <https://doi.org/10.17979/ja-cea.2025.46.12215>

Resumen

El acceso universal y generalizado a Internet en el ámbito educativo está llevando a la transformación de herramientas informáticas en plataformas cliente-servidor que se pueden usar desde cualquier dispositivo con acceso a la red. En este trabajo introducimos dos de estas implementaciones. WebEJS es la evolución centrada en Internet del conocido Easy JavaScript Simulations, que antes requería un ordenador y una máquina virtual Java. IODA es una herramienta gráfica cliente-servidor para el análisis guiado de datos que, creada con fines de investigación, puede ser útil para cursos introductorios de datos para estudiantes de física. Ambas herramientas no sólo facilitan el acceso universal a las tecnologías docentes existentes, sino que abren nuevas posibilidades a profesores e investigadores educativos.

Palabras clave: Educación, Modelado, Simulación, Análisis de datos

New client-server web tools for teaching modelling, simulation, and data analysis: IODA y WebEJS

Abstract

The universal and widespread access to the Internet in education is leading to the transformation of computer tools into client-server platforms that can be used from any device with access to the network. In this paper we introduce two of these implementations. WebEJS is the Internet-centric evolution of the well-known Easy JavaScript Simulations, which previously required a computer and a Java virtual machine. IODA is a graphical client-server tool for guided data analysis that, created for research purposes, can be useful for introductory data courses for physics students. Both tools not only facilitate universal access to existing teaching technologies, but also open up new possibilities for teachers and educational researchers.

Keywords: Education, Modelling, Simulation, Data Analytics

1. Introducción

La popularización del acceso a Internet, especialmente desde las instituciones educativas, junto con el rápido aumento del número de estudiantes que poseen dispositivos electrónicos con acceso a Internet (tabletas, teléfonos inteligentes, etc.), ha generado un cambio de las herramientas educativas, que necesitaban de un ordenador para usarse, hacia plataformas puramente basadas en Internet. Estos

sistemas siguen generalmente una arquitectura cliente-servidor, en la cual la interfaz gráfica de usuario (GUI) del cliente está separada, tanto en su diseño como en su implementación, del motor computacional.

El cliente, que se ejecuta en el dispositivo del estudiante, está programado utilizando los estándares de Internet y solo requiere el uso de un navegador web moderno. Dado que todos los dispositivos incluyen al menos un navegador de este tipo, no es necesario realizar ninguna instalación ni abonar tarifas

de licencia. Además, los clientes están diseñados para ser intuitivos, ofrecen documentación inmediata mediante redirección a otras páginas web y son de bajo consumo de recursos, ya que las tareas intensivas en CPU son gestionadas por el servidor. Adicionalmente, al dejar más recursos liberadores en el dispositivo de los usuarios, esto permite la utilización de entornos web interactivos, enriquecidos y personalizados (Marquez-Barja 2018).

Estas características facilitan la implementación rápida de las herramientas en los dispositivos de los estudiantes; el único requisito es una conexión a Internet adecuada. Al otro lado de la red, se encuentran uno o más servidores, que actúan como los encargados de la plataforma. Estos servidores pueden ser programados en cualquier lenguaje de programación y emplear plataformas o herramientas sofisticadas, tales como bases de datos, bibliotecas de software avanzadas, programas que generan páginas web personalizadas para los estudiantes, herramientas de inteligencia artificial o algoritmos de análisis del comportamiento de los estudiantes en tiempo real. En su forma más básica, el servidor típicamente se encarga de autenticar a los estudiantes, generar las páginas HTML del cliente y responder a las acciones de los estudiantes.

Esta separación resulta particularmente adecuada para un equipo compuesto por educadores y programadores. Los educadores tienen la responsabilidad de diseñar y especificar los contenidos que el estudiante verá y con los que interactuará en el cliente, mientras que los programadores tienen plena libertad para utilizar cualquier lenguaje de desarrollo que consideren necesario para trabajar de manera eficiente en el lado del servidor, sin las limitaciones impuestas por la plataforma de despliegue.

En este trabajo presentamos dos implementaciones que utilizan este tipo de arquitectura y que pueden ser utilizadas para la enseñanza en el ámbito de las ciencias. Los siguientes apartados contienen una descripción de ambas herramientas, primero desde un punto de vista de su estructura y después describiendo sus funcionalidades.

2. IODA y WebEJS

IODA (Input-Output Data Analysis) y WebEJS (Web Easy JavaScript Simulations) se presentan en esta sección como diferentes implementaciones de herramientas cliente-servidor que abren nuevas posibilidades para los docentes y los investigadores relacionados con la educación en ciencias. Estas herramientas pueden usarse en el aula, para tareas de enseñanza, tanto como para programar funciones avanzadas, siendo entre otras cosas posible: realizar análisis de datos, análisis de aprendizaje e instrucción guiada por inteligencia artificial.

Para facilitar aún más el uso y la colaboración, ambas herramientas han sido desarrolladas haciendo uso de contenedores, en concreto Docker. En el ámbito de la educación en ciencias, el software de código abierto Docker se ha consolidado como una herramienta poderosa para facilitar la enseñanza y el aprendizaje de habilidades computacionales (Tobarrá 2020). Al permitir la creación de entornos de ejecución aislados, reproducibles y fácilmente distribuibles, Docker ofrece una solución eficaz para ejecutar

código, compartir experimentos y garantizar la coherencia entre distintos sistemas operativos. Esta tecnología permite a los docentes definir entornos personalizados con todas las dependencias necesarias, lo cual resulta especialmente útil en simulaciones científicas (Canesche 2023). Al reducir los problemas técnicos relacionados con la configuración del entorno, Docker contribuye a que estudiantes y profesores se concentren en los contenidos científicos propiamente dichos (Dakkak 2016).

2.1. IODA

La primera de las implementaciones proviene de un proyecto de investigación en curso, cuyo objetivo es proporcionar a los científicos, en campos donde es necesario el análisis de grandes cantidades de datos, un acceso sencillo a enormes repositorios remotos, así como su análisis y visualización utilizando esta misma arquitectura cliente-servidor.

Los distintos componentes de la arquitectura de IODA pueden usarse y configurarse de múltiples maneras, incluyendo una o varias máquinas e involucrando actores adicionales como gestores de contenedores. En la Figura 1, se muestra un ejemplo donde aparecen todos los componentes involucrados separados en función de sus cometidos en el sistema, esta separación puede ser física, una máquina para cada componente, o solo lógica, cada componente en un contenedor:

- Usuario: La persona que se desea hacer uso de la herramienta.
- Interfaz de Usuario (UI): El entorno gráfico personalizado que permite al usuario crear los grafos dirigidos, Figura 2.
- Coordinador: Servidor web que suministra la UI al usuario, gestiona sus interacciones y comunica eventos. Se encarga también de comunicar al ejecutor cuando hay un grafo que ejecutar.
- Ejecutor: Recibe el grafo, crea un plan de ejecución y se encarga de gestionar que se lleve a cabo enviando tareas a los nodos y recibiendo sus respuestas. Puede haber varios ejecutores disponibles.
- Nodos: Computadoras cuya tarea es ejecutar las tareas y devolver los resultados. Puede ser un solo nodo, o toda una red de ellos.

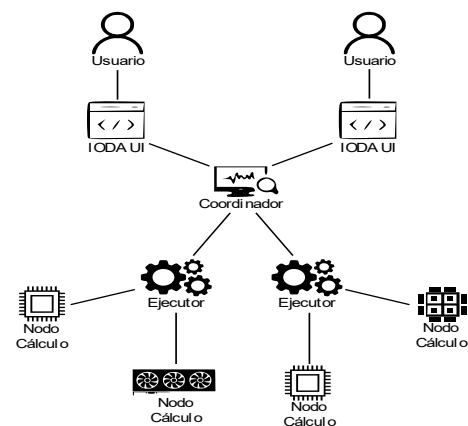


Figura 1: Arquitectura de IODA en un ejemplo donde solo hay dos ejecutores y 2 nodos por ejecutor.

Uno de los objetivos de esta herramienta es que sus usuarios puedan combinar las rutinas y bibliotecas estándar existentes con su propio código orientado a la investigación, para analizar y visualizar datos. La arquitectura mostrada y su naturaleza, permite a los usuarios crear, de manera gráfica, grafos dirigidos de elementos dedicados (Esquembre 2023). Estos elementos representan un algoritmo determinado, que toma algunos datos de entrada y produce ciertos resultados, y son proporcionados por la herramienta a partir de una lista de elementos desarrollados por expertos, que incluye algoritmos típicos de la ciencia de datos y otros específicos de otras disciplinas. El usuario puede, desde el cliente, crear nuevos análisis seleccionando ciertos elementos, configurando sus propiedades y conectando adecuadamente sus entradas y salidas. El servidor puede luego ejecutar el grafo invocando los diferentes algoritmos en nodos computacionales dedicados, y presentar los resultados en formato HTML.

La Figura 2 muestra un grafo típico creado en el navegador web de un cliente, donde pueden apreciarse el grafo dirigido, así como los paneles de herramientas para la edición, control y personalización de los elementos involucrados.

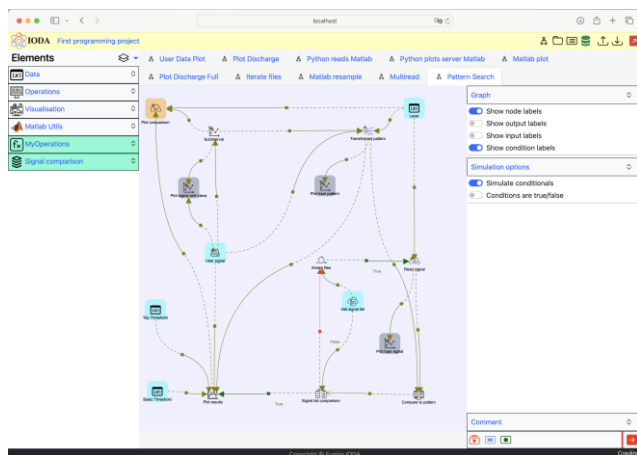


Figura 2: IODA. Captura de pantalla de un navegador con un grafo de IODA siendo utilizados en una tablet.

2.1. WebEJS

El segundo desarrollo, WebEJS, es una implementación cliente-servidor de la galardonada herramienta de modelado y autoría EJS (Esquembre 2004). EJS constituye una parte importante de la colección OpenSourcePhysics (Christian 2011) en la biblioteca digital ComPADRE, que incluye cientos de simulaciones computacionales listas para usar, de nivel de secundaria y universidad, en el ámbito de la física, las cuales pueden ser adoptadas y adaptadas (mediante EJS) por los instructores con habilidades mínimas en programación.

De la misma manera que en la primera implementación, el servidor WebEJS está preparado para lanzarse en un contenedor Docker (Esquembre 2022), permitiendo así un despliegue sencillo y rápido, tanto para su uso local como para que sea accesible desde Internet. Actualmente, en la

propia página web de WebEJS, Figura 3, pueden verse tres servidores públicos accesibles, dos de ellos en la Universidad de Murcia y otro en Singapur



Figura 3: Captura de pantalla del índice de la página web de WebEJS (<https://webajs.iwant2study.org>) donde se aprecian tres enlaces para acceder los servidores de WebEJS correspondientes.

La Figura 4 muestra la interfaz gráfica de usuario (GUI) de WebEJS en un navegador. Esta interfaz ofrece la misma combinación de simplicidad y potencia de EJS y facilita el acceso a potentes tecnologías de red.

En esta nueva implementación de WebEJS, el uso de EJS con los estudiantes ya no requiere la adquisición, configuración ni mantenimiento de un laboratorio de ordenadores personales dedicado. Es posible utilizar dispositivos de Internet básicos (como tabletas) tanto en clase como en casa.

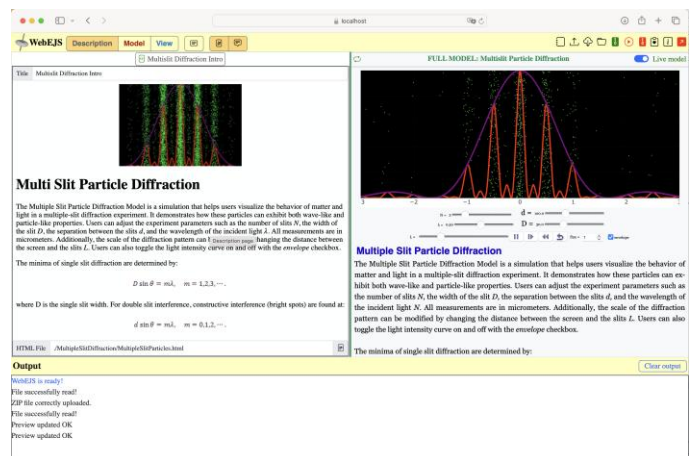


Figura 4: WebEJS. Captura de pantalla de un navegador con Web EJS siendo utilizados en una tablet.

3. Conclusiones

WebEJS e IODA son implementaciones cliente-servidor que ofrecen medios para compartir recursos, aplicaciones y conocimientos, de una forma sencilla e intuitiva. Además, ofrecen todas las ventajas de las arquitecturas cliente-servidor (Pautasso 2009):

- Usar interfaces de usuario livianas, dejando la carga computacional al servidor

- Uso en la mayoría de los navegadores web, y por lo tanto en la mayor parte de los dispositivos, pudiendo acceder a un mismo proyecto activo desde diferentes dispositivos sin dificultades.
- El desacoplamiento del cliente del servidor, pudiendo por tanto realizar su mantenimiento y actualizaciones de forma independiente.

Adicionalmente, el uso de contenedores Docker en ambas implementaciones provee de múltiples ventajas relacionadas con dos factores principales: la velocidad de despliegue y la homogeneidad de las soluciones ofrecidas a los usuarios o administradores de las máquinas.

Por un lado, gracias a la nueva implementación de WebEJS los estudiantes pueden utilizar cualquier dispositivo con acceso a Internet, tanto en clase como en casa sin depender de una instalación particular. Adicionalmente, dado que EJS lleva muchos años usándose, los estudiantes pueden optar entre:

- Crear sus propias aplicaciones, usando la interfaz de WebEJS.
- Acceder a otras aplicaciones ya creadas por otras personas, por ejemplo, otros estudiantes o docentes.
- Acceder a los recursos online existentes en la colección OpenSourcePhysics.

Por otro lado, la plataforma IODA permite: la ejecución de algoritmos escritos en diferentes lenguajes de programación, la combinación de distintos nodos computacionales diferentes, así como un acceso controlado a bases de datos y el uso de hardware especializado en una forma simplificada.

IODA se encuentra actualmente en su periodo de desarrollo y beta-testing y se está utilizando para albergar una pequeña comunidad experimental. Esta comunidad está construyendo una amplia variedad de elementos comunes para el análisis y visualización de datos, además de instalar ejecutores y nodos de computación utilizando diferentes enfoques y ofreciendo una amplia gama de servicios. Por lo tanto, aunque algunas de las funciones aún no han sido implementadas en la plataforma, como el análisis del aprendizaje y la asistencia por IA, su arquitectura permite que los tecnólogos educativos añadan dichas características.

Por todo lo dicho anteriormente, este trabajo constituye al mismo tiempo, una presentación de ambas herramientas y

una llamada a la colaboración para los investigadores interesados en utilizarlas.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Agencia Nacional en los proyectos, PID2022-137680OB-C31, and PID2022-137680OB-C32.

Referencias

- Canesche, Michael & Leissa, Roland & Pereira, Fernando. (2023). Preparing Reproducible Scientific Artifacts using Docker. 10.48550/arXiv.2308.14122.
- Christian, W., Esquembre, F., and Barbato, L. 2011. Open Source Physics. *Science*, 334:1077–1078.
- Dakkak, A. Pearson, C. and Hwu, W. "WebGPU: A Scalable Online Development Platform for GPU Programming Courses," *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Chicago, IL, USA, 2016, pp. 942-949, doi: 10.1109/IPDPSW.2016.63.
- Esquembre, F., 2004. Easy Java Simulations: a software tool to create scientific simulations in java. *Computer Physics Communications* 156, 2:199–204.
- Esquembre, F., Chacón, J. Saenz, J., Vega, J., Dormido-Canto, S. 2023. A programmable web platform for distributed access, analysis, and visualization of data. *Fusion Engineering and Design* 197: 114049.
- Esquembre, F., Clemente-García, F., Chacón, J. (September 2022) WebEJS: A fully Web-based version of EasyJavaScript Simulations. 25th International Workshop on Multimedia in Physics Teaching and Learning, Wroclaw, Poland.
- J. M. Marquez-Barja, N. Kaminski and L. A. Dasilva, "Assessing the Impact of User Interface Abstraction on Online Telecommunications Course Laboratories," in *IEEE Access*, vol. 6, pp. 50394-50403, 2018, doi: 10.1109/ACCESS.2018.2868904.
- Pautasso, Cesare & Wilde, Erik. (2009). Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. 18th International World Wide Web Conference. 911-920. 10.1145/1526709.1526832.
- Tobarra, L., Robles-Gómez, A., Pastor, R., Hernández, R., Duque, A., & Cano, J. (2020). Students' Acceptance and Tracking of a New Container-Based Virtual Laboratory. *Applied Sciences*, 10(3), 1091. <https://doi.org/10.3390/app10031091>
- WebEJS. <https://www.um.es/fem/wikis/runwebejs/index.html>
- Docker. *Docker* <https://www.docker.com/>