

1/10/2021



IO 6: TUTORIAL DE DISEÑO TÉCNICO E IMPLEMENTACIÓN DE UN WEBLAB — INTRODUCCIÓN



Co-funded by
the European Union

1. INTRODUCCIÓN

Los laboratorios web (weblabs) han demostrado ser una herramienta fantástica para la enseñanza y el aprendizaje, especialmente, como complemento de los laboratorios prácticos tradicionales en los campos técnico, de ingeniería y ciencia. Desafortunadamente, diseñar e implementar un weblab no suele ser una tarea fácil ni rápida, lo que hace que este tipo de recursos sean difíciles de encontrar en muchos cursos universitarios. Por lo tanto, se requiere un esfuerzo para educar a nuestros educadores sobre cómo crear weblabs, para que puedan agregarlos a sus cursos. El proyecto RELAB es consciente de esta carencia, y su Intellectual Output 6 (al que pertenece este documento) aborda este problema.

1.1. Objetivo

Este manual es parte de una serie de documentos que fueron creados con la intención de dar algunas instrucciones muy específicas sobre cómo crear un weblab e implementarlo en un curso en línea para hacerlo accesible para los estudiantes. De esta manera, nuestra intención y esperanza es que más profesores en Europa y en todo el mundo se sientan cómodos para desarrollar sus propios weblabs y utilizarlos en sus cursos con el fin de mejorar su enseñanza.

1.2. Público objetivo

El público objetivo de esta serie de manuales son profesores de asignaturas técnicas, de ingeniería y ciencias que desean incluir más actividades de experimentación en sus cursos sin las restricciones que imponen los laboratorios prácticos, pero no tienen una idea clara de cómo o por dónde empezar.

Una suposición hecha en este documento es que el lector ya sabe lo que es un laboratorio basado en la web, un laboratorio en línea o un laboratorio remoto, y tiene una ligera idea de cuáles son las partes que forman parte de él.

1.3. Alcance

Si bien los weblabs se pueden desarrollar utilizando muchas tecnologías, herramientas y enfoques diferentes (muchos de los cuales son igualmente válidos en muchos aspectos), estos manuales fueron concebidos para ser lo más cortos y simples posible para tratar de introducir el desarrollo de weblabs de una manera amigable.

Las herramientas propuestas en estos documentos para el desarrollo de weblabs, y para las que se pueden encontrar instrucciones de uso específicas, son todas gratuitas y de código abierto, siguiendo las directivas y el espíritu europeos y facilitando su acceso, adopción y adaptabilidad.

2. VISTA GENERAL Y HERRAMIENTAS

Un weblab generalmente se compone de un cliente, una aplicación basada en la web y algunos activos del lado del servidor. Ya sea que los activos sean una computadora que ejecuta un software de simulación o uno conectado a una planta real (como el ejemplo de la Figura 1), algunas tareas siempre se realizan en un laboratorio remoto (el lado del servidor). Por lo tanto, normalmente se requieren al menos dos herramientas o tecnologías de software para preparar un weblab.

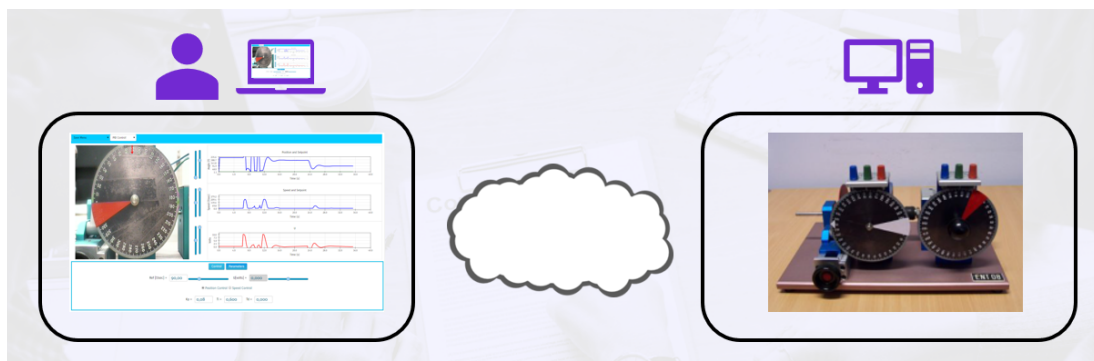


Figura 1. Arquitectura básica de un weblab

En aras de la simplicidad, estos manuales se centran solo en el uso de las siguientes herramientas de software.

Para el lado del cliente:

- Easy Java/Javascript Simulations (EjsS) – Es una herramienta gratuita y de código abierto especialmente creada para hacer que la tarea de desarrollar simulaciones y aplicaciones web para weblabs sea lo más fácil posible. El resultado de un proyecto EjsS es una aplicación web HTML5 (como la que se muestra en la figura anterior) que se puede integrar fácilmente en los cursos en línea.

Para el lado del servidor:

- LabVIEW – El software de National Instruments es uno de los más comunes utilizados en universidades y centros de investigación para operar y controlar dispositivos físicos de laboratorio, sistemas y plantas de experimentación. Por muy bueno que sea, este es un software comercial que requiere pagar una licencia, y no todos los grupos o instituciones pueden permitírselo.
- Python: este es un lenguaje de programación gratuito conocido y poderoso que también permite conexiones con muchos dispositivos de hardware, como Arduino, Raspberry y muchos más. Dado que estos manuales presentan guías para diseñar e implementar weblabs tanto con LabVIEW como con Python, los usuarios siempre tendrán una opción disponible a su alcance para seguirlo.

Los siguientes documentos que se centran en el uso de cada una de estas herramientas para su parte correspondiente en el desarrollo e implementación de weblabs proporcionan los enlaces para descargar los recursos y el software necesarios que se requieren.

LOS CUATRO PASOS

Hay cuatro pasos principales relacionados con el diseño, desarrollo y uso de weblabs. Este documento en general, y esta sección en particular, solo los aborda de manera general, dando una breve introducción a cada uno de estos pasos. Sin embargo, cuatro manuales más forman parte de esta serie de documentos, cada uno de ellos abordando cada uno de estos pasos con más detalle, y dando una guía paso a paso sobre cómo abordar estas tareas con las herramientas propuestas. Antes de leerlos, se recomienda primero revisar las líneas aquí para obtener una idea general y una visión general de todo el proceso.

Los cuatro pasos mencionados anteriormente son: 1) el desarrollo e implementación del lado del servidor, 2) el desarrollo e implementación del lado del cliente, 3) la implementación de la aplicación weblab en un curso en línea y 4) problemas de seguridad y comunicación. Las secciones next introducen cada uno de ellos y contextualizan el resto de los manuales de esta serie dentro de su ámbito particular.

2.1. Desarrollo e implementación del lado del servidor

El desarrollo del lado del servidor para un weblab generalmente tiene que resolver la conexión de una computadora con hardware real, en forma de actuadores, sensores, etc. Esto se puede hacer a través de una variedad de soluciones de hardware (Arduino, Raspberry, tarjetas de adquisición de datos, PLC, etc.) y software (LabVIEW, Beckhoff, Matlab, Python, etc.).

Estos manuales no entran en la parte de hardware, ya que esto depende en gran medida del tipo de experimento que esté preparando y de los recursos que pueda tener disponibles o a su alcance. Para la parte de software, los documentos de esta serie se centran en dos soluciones (LabVIEW y Python), ambas igualmente válidas y fáciles de conectar a una aplicación web generada con EjsS.

Aquellos familiarizados con LabVIEW y que pueden usar el software, probablemente preferirían esta opción, ya que hay buenas posibilidades de que el hardware de laboratorio que necesitan para conectarse ya tenga una interfaz y / o API de LabVIEW. Se recomienda a cualquier otra persona utilizar la solución Python, ya que presenta un alto grado de interoperabilidad y conectividad, por lo que es ideal para la mayoría de las aplicaciones.

2.2. Desarrollo e implementación del lado del cliente

El desarrollo del lado del cliente incluye la creación de la aplicación web HTML5 que permitirá a los estudiantes interactuar con los activos de laboratorio remotos. Esto generalmente incluye botones, controles deslizantes, gráficos, campos de entrada y / o una transmisión de video de lo que está sucediendo en la sala de laboratorio.

Cualquier programación HTML + Javascript + CSS podría darle una aplicación HTML5 de weblab como resultado, pero la cantidad de trabajo para programar esto desde cero es considerable. Otras soluciones, como el uso de Unity o motores similares amigables con la web, también son posibles, pero el mismo problema permanece. Estos manuales se centrarán en el uso de EjsS

para esta tarea, ya que está especialmente diseñado para ello y cualquiera puede descargarlo y utilizarlo de forma gratuita.

2.3. Despliegue

Si bien el trabajo requerido en el lado del servidor es imprescindible para configurar un weblab, es la aplicación web cliente la que debe implementarse en un curso en línea para que sea accesible a los estudiantes y les permita realizar sus tareas de laboratorio.

Una vez más, muchas soluciones son posibles para esto, pero estos manuales se centran en uno en particular con la idea de no ser generales y vagos, sino específicos y claros. La guía que encontrará en el manual para el tercer paso toma Moodle como el LMS objetivo en el que implementar las aplicaciones web HTML5 creadas con EjsS, y utiliza el conjunto de complementos EJSApp para facilitar la labor de integración de estas aplicaciones en un curso de Moodle.

2.4. Seguridad y comunicaciones

Por lo general, los problemas de seguridad y comunicaciones con los weblabs son extremadamente importantes y, a veces, difíciles de abordar. Un manual completo sobre cómo resolver estos problemas está fuera del alcance de esta serie de manuales, pero el último documento proporciona una breve descripción del problema y algunas posibles soluciones.