

18/12/2021



IO 6: TUTORIAL DE DISEÑO TÉCNICO E IMPLEMENTACIÓN – IMPLEMENTACIÓN



Co-funded by
the European Union

1. CARACTERÍSTICAS DE LAS APLICACIONES WEB DE EjsS INTEGRADAS EN LA PLATAFORMA MOODLE CON EJSAPP

Web aplicaciones desarrolladas en EjsS (se recomienda encarecidamente utilizar siempre la última versión disponible de EjsS, que puede obtener aquí: <https://gitlab.com/EjsS/tool>) que están incrustadas en el entorno del Sistema de Gestión del Aprendizaje Moodle, ya que las actividades de EJSApp pueden adquirir las siguientes características y funcionalidades:

- Agregue una nueva actividad de aprendizaje en un curso de Moodle basado en aplicaciones web (EJSApp).
- Integración de soporte multilingüe de aplicaciones web con Moodle.
- Guardar archivos (.json, .blk, .rec, .txt, .jpg ...), desde la aplicación web hasta el repositorio de archivos privados de Moodle y leyéndolos / cargándolos, desde este repositorio, a la aplicación web.
- Cargue los estados iniciales de una aplicación web (archivos en formato json).
- Grabación y reproducción de interacciones con aplicaciones web (archivos en formato .rec).
- Permitir el uso de un lenguaje de programación visual para interactuar y comunicarse con aplicaciones web.
- Configurar una aplicación web para usarla como una interfaz remota que se conecta a hardware real.

Este manual describe cómo obtener todas las características de integración enumeradas anteriormente. El Capítulo 2 de este documento explica cómo se deben preparar las aplicaciones web en EjsS para obtener el funcionamiento adecuado en Moodle, mientras que el Capítulo 3 muestra cómo configurar las actividades de Moodle EJSApp. Si necesita ayuda con las actividades de Moodle en general a nivel de usuario, recuerde que esta plataforma es solo una extensión de Moodle, por lo que una visita a la documentación oficial de Moodle realmente ayuda: <https://docs.moodle.org/en/Activities>

2. PREPARACIÓN DE APLICACIONES WEB EN EjsS PARA SU USO EN MOODLE

En primer lugar, vale la pena mencionar que esta guía se limita a la preparación de aplicaciones EjsS en modo JavaScript, ya que la versión de Java está en desuso. Por lo tanto, lo primero es comprobar que EjsS se ha iniciado en este modo y, si no, hacerlo. La Figura 1 muestra la pestaña *Básico* en la ventana de la consola de EjsS, desde la cual se puede seleccionar el lenguaje de programación JavaScript.

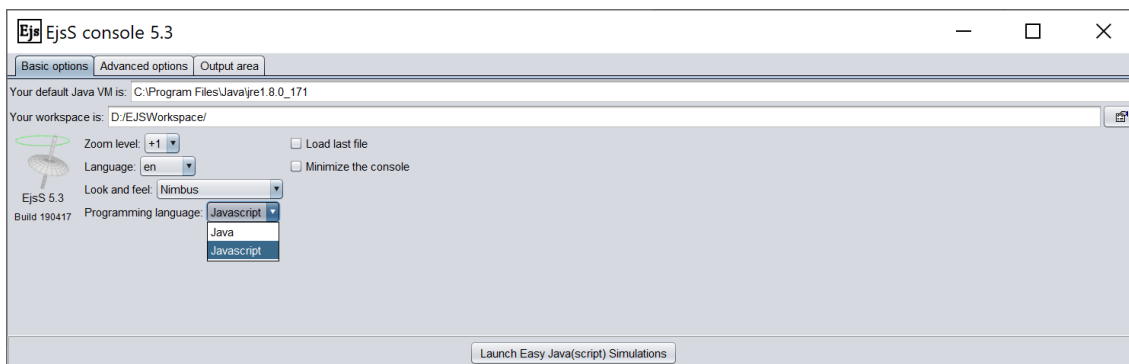


Figura 1. Configuración de JavaScript como modo predeterminado/lenguaje de programación para trabajar con EjsS

Si necesita un tutorial para EjsS, puede encontrar un manual para comenzar a aprender sobre cómo usar este programa aquí: https://www.um.es/fem/EjsWiki/uploads/Download/EjsS_Manual.pdf Mientras que el Capítulo 2 está dedicado al modo Java de EjsS y el Capítulo 4 se centra en convertir aplicaciones Java a aplicaciones JavaScript, puede omitir con seguridad los dos capítulos y leer el resto del documento.

2.1. Soporte multilingüe

La configuración de language L en las aplicaciones web E jsS integradas en Moodle con EJApp se vincula automáticamente al idioma seleccionado en un entorno web de Moodle cuando la aplicación web se ha preparado utilizando el soporte nativo en varios idiomas ofrecido por EjsS. Para preparar una aplicación en EjsS con este soporte para múltiples idiomas, basta con seguir las instrucciones dadas en este video-tutorial: <https://www.um.es/fem/Ejs/WebCasts/en/EJSTranslationFacilities/media/EJSTranslationFacilities.mp4>

Nota importante: Es un requisito que el idioma predeterminado de la aplicación web (es decir, el idioma de los textos que están escritos en los elementos de la vista EjsS) sea el inglés. Then, se pueden agregar traducciones adicionales. Lo contrario (hacer, por ejemplo, el español el idioma predeterminado y agregar la traducción al inglés más tarde) hace que este último no se muestre en Moodle.

2.2. Guardar y cargar archivos

Es importante seguir algunas líneas de guíade diseño de las interfaces de usuario que sean similares para todas las aplicaciones. El menú de guardado/lectura de archivos se ha colocado tradicionalmente en la parte superior de las aplicaciones, como se muestra en la Figura 2.

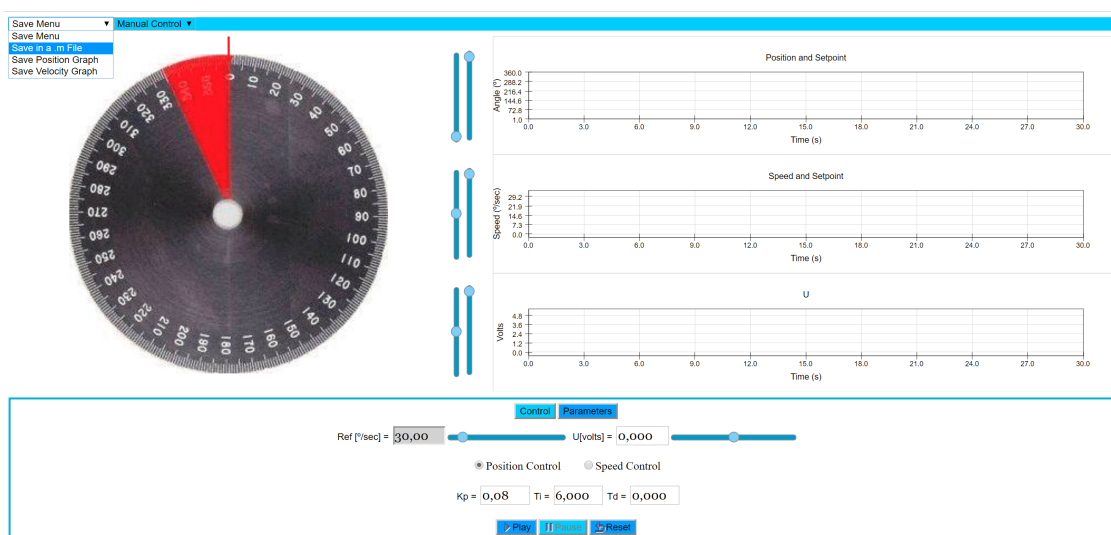


Figura 2. Menú y botones para guardar/cargar archivos

La Tabla 1 ofrece una lista de métodos nativos de EjsS para guardar y cargar diferentes tipos de archivos. Estos métodos se pueden utilizar en las acciones invocadas cuando se seleccionan las opciones del menú superior, para guardar y cargar archivos.

Tabla 1. Métodos para guardar y cargar archivos

Hay cuatro métodos que permiten enviar un archivo desde la aplicación web EjsS al repositorio de archivos privados de Moodle:

- Método `_saveText(String FileName, String FileType, String FileContent)` , para guardar un archivo de texto que corrige automáticamente, según `FileType`, su extensión.
- Método `_saveImage(String FileName, String ViewElementName)`, para guardar una imagen de captura de pantalla de la interfaz de usuario de la aplicación web (determinada por el nombre del elemento View en EjsS)..
- Método `_saveState(String FileName)`, para guardar en un archivo .json la lista completa de variables `f` (y sus valores) que definen el estado de las simulaciones de una aplicación web.
- Método `_saveVariables(String FileName, List<String> vars)`, para guardar sólo las variables indicadas (parámetro `vars`) y sus valores en un archivo .json.

Y hay tres formas de recuperar los datos almacenados en archivos desde una aplicación web EjsS:

- Método `_readText(String FileName, String FileType, String VariableName)`, para leer el contenido de un archivo de texto sin formato y escribirlo en una variable denominada

VariableName. Al leer archivos de texto sin formato, *FileType* debe establecerse en 'texto'.
Nota importante: para usar esto desde Moodle, *FileName* debe ser null, ya que la aplicación mostrará un menú para permitir que el usuario seleccione de qué archivo desea leer datos.

- Método *_readState(null)*, para leer y cargar la lista completa de variables de una simulación de aplicación web desde un archivo .json.
- Método *_readVariables(null, List<String> vars)*, para leer y cargar una lista de variables seleccionadas de una simulación de aplicación web desde un archivo .json.

Todos los métodos de guardado abren una ventana modal como la que se muestra en la Figura 3.a, mientras que todos los métodos de carga abren una como la que se muestra en la Figura 3.b. No es necesario crear estas ventanas usted mismo, ya que se crean automáticamente mediante los métodos de guardado y carga.

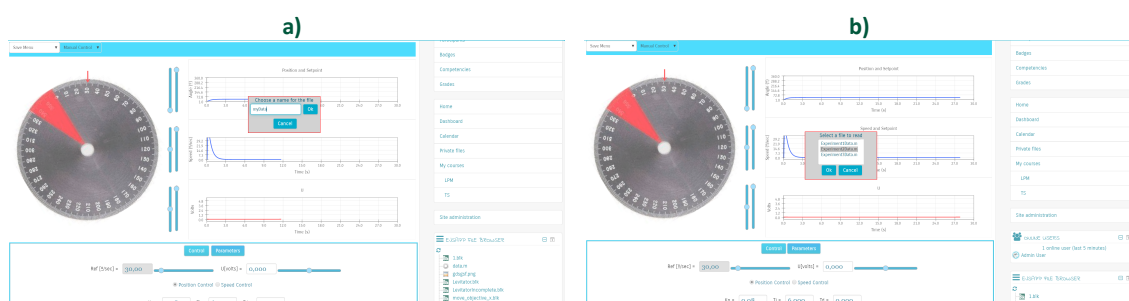


Figura 3. Ventanas modales creadas por los métodos de guardado y carga para escribir/leer archivos

2.3. Laboratorio remoto

Si la aplicación EjsS está destinada a ser un laboratorio remoto, entonces se debe hacer más trabajo para conectar la aplicación HTML5 con el software / hardware en el laboratorio. Si desea utilizar nuestro Protocolo de interoperabilidad remota (RIP) para esto, lea la **Guía del usuario de ELEMENTOs RIP** para aprender a configurar y preparar su aplicación EjsS y las Guías **del usuario del servidor RIP** para aprender a preparar su dispositivo de laboratorio para aceptar conexiones de clientes RIP. Hay dos implementaciones (y sus correspondientes guías) para el servidor RIP: LabVIEW y Python.

Finalmente, si está utilizando un sistema ENLARGE para habilitar las comunicaciones con laboratorios dentro de su LAN institucional y para administrar las conexiones de laboratorio remotas, también debe usar el elemento ENLARGE Proxy EjsS, para lo cual debe leer la **Guía del usuario de ENLARGE Proxy Element**.

3. CONFIGURACIÓN Y USO DE EJSAPP ACTIVITIES EN MOODLE

3.1. Agregar un EjsApp a MOODLE

La figura 4 muestra el formulario web para agregar o editar una actividad con una aplicación EjsS (EJSApp). Los únicos dos fields que deben rellenarse son el nombre de la actividad y el archivo .zip que encapsula la aplicación EjsS.

El campo para el nombre de la actividad se puede rellenar utilizando el HTML nomenclature que permite a Moodle filtrar el texto en función del idioma seleccionado por el usuario. De esta forma, se puede establecer el nombre de la actividad tanto en inglés como en español, por ejemplo (ver Tabla 2) y se mostrará en el idioma adecuado para cada usuario.

Tabla 2. Código para establecer un texto en Moodle para que admita el filtro multilingüe

```
<span class="multilang" lang="es">Laboratorio Virtual del Servo Motor</span><span class="multilang" lang="en">Laboratorio Virtual del Servomotor</span>
```

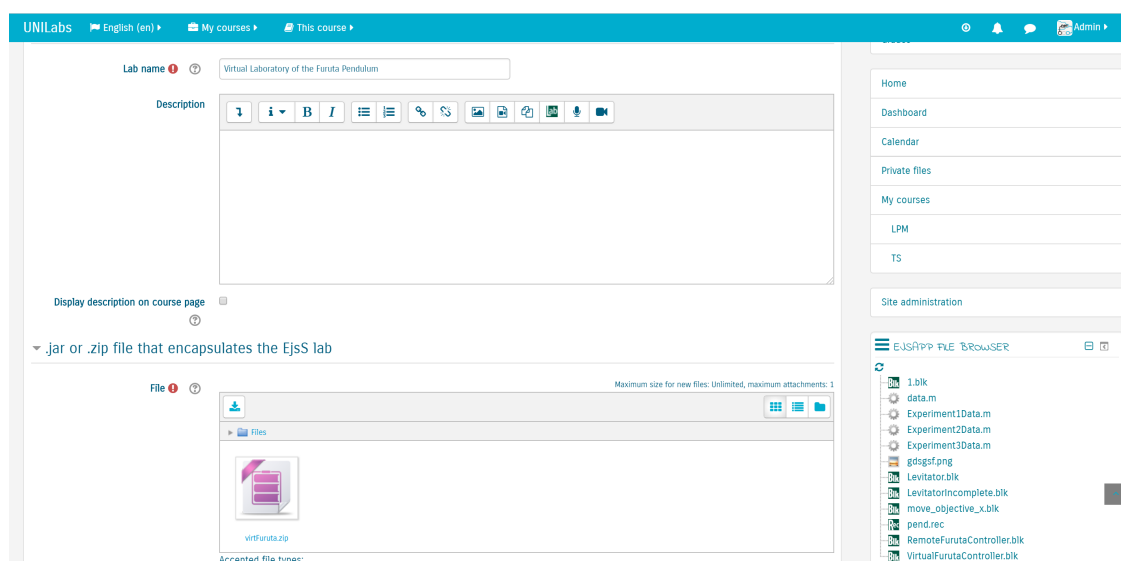


Figura 4. Agregar una actividad de la aplicación EjsS en MOODLE

Una vez establecido el nombre de la actividad y cargado el archivo EjsS .zip, se pueden configurar varias opciones en la actividad. Las siguientes secciones pasan por estas opciones.

3.2. Carga de un estado inicial

Cuando se crea una actividad EJSApp en Moodle, es posible agregar un archivo .json que contenga un estado particular del laboratorio. Una vez hecho esto, el sistema carga automáticamente la aplicación en el estado contenido en el archivo .json. Para hacer esto,

simplemente agregue el archivo .json a la sección "*archivo .json con el estado que se leerá cuando se cargue este laboratorio de EjsS*" del formulario de adición / edición (Figura 5) de la actividad de EJSApp.

Esta opción permite cargar un estado diferente al que está establecido por defecto en la aplicación EjsS. Esto, a su vez, se puede utilizar para proponer diferentes escenarios o casos de uso predefinidos (con un valor didáctico especial) con el mismo sistema de laboratorio.

Para generar estos estados y almacenarlos en archivos .json, utilice el método `_saveState()` introducido en la Tabla 1.

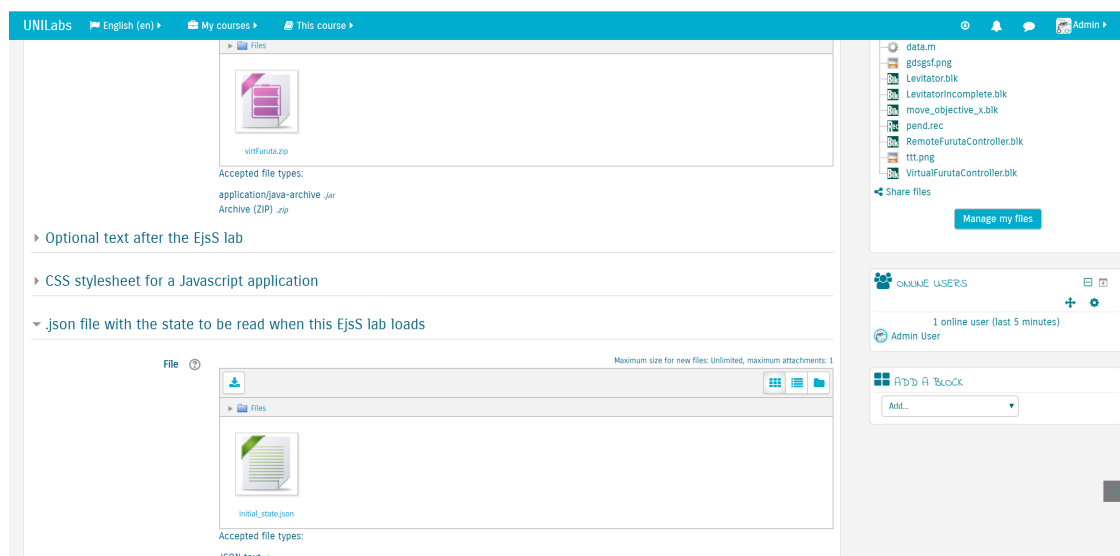


Figura 5. Carga de un estado inicial en y actividad de EJSApp

3.3. Carga de una interacción grabada

También es posible agregar un archivo de texto con extensión .rec que contenga un script con las instrucciones para reproducir automáticamente las interacciones realizadas por un usuario con la aplicación EjsS. Esto se puede hacer en la sección "*archivo .rec con la grabación que se ejecutará cuando se cargue este laboratorio de EjsS*" del formulario de adición / edición, similar a los dos anteriores.

Esta opción permite cargar y reproducir automáticamente las interacciones realizadas por un usuario con la app EjsS, similar a un vídeo pero con el modelo de simulación o las comunicaciones remotas de laboratorio ejecutándose en tiempo real. Esto se puede utilizar como demostraciones, tutoriales, etc.

Para registrar estas interacciones y almacenarlas en archivos .rec, utilice el bloque del explorador de archivos EJSApp.

3.4. Interacción de programación visual con la aplicación web

Se puede usar un lenguaje de programación visual basado en Blockly para permitir a los usuarios interactuar con la aplicación EjsS a través de la codificación de bloques. Esto abre muchas posibilidades, acercando la experimentación a una experiencia sandbox. La Figura 7 muestra un ejemplo de configuración, donde las diferentes categorías de bloques de programación se pueden agregar o no a la biblioteca de bloques que los usuarios tendrían disponibles para trabajar con la aplicación EjsS.

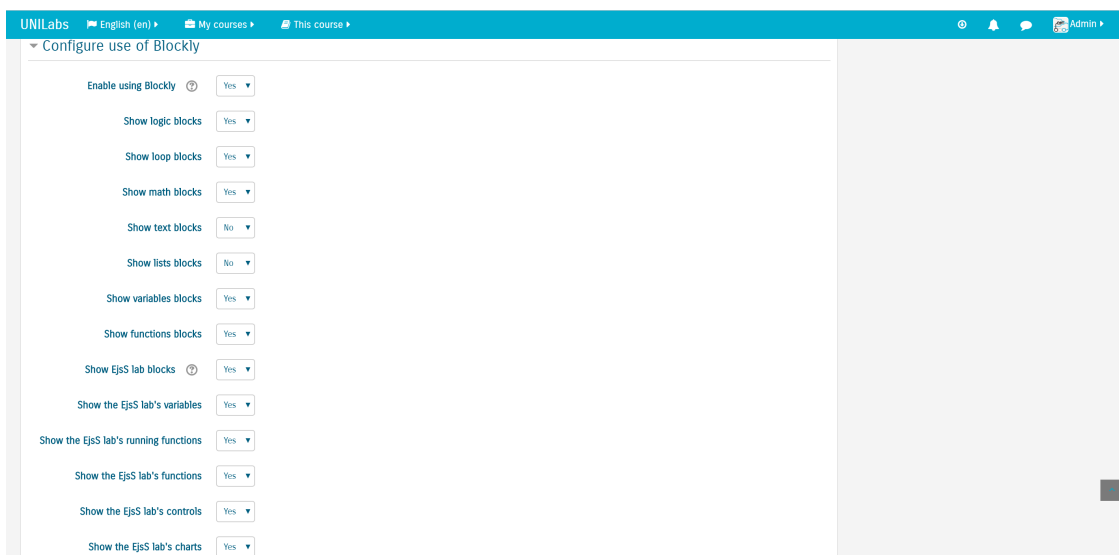


Figura 6. Habilitación de la programación visual con Blockly en actividades de EjsS

Además, es posible añadir un archivo .blk si queremos que se cargue un programa de bloqueo inicial cuando un usuario entra en la actividad de EJSApp, como se muestra en la figura 8.

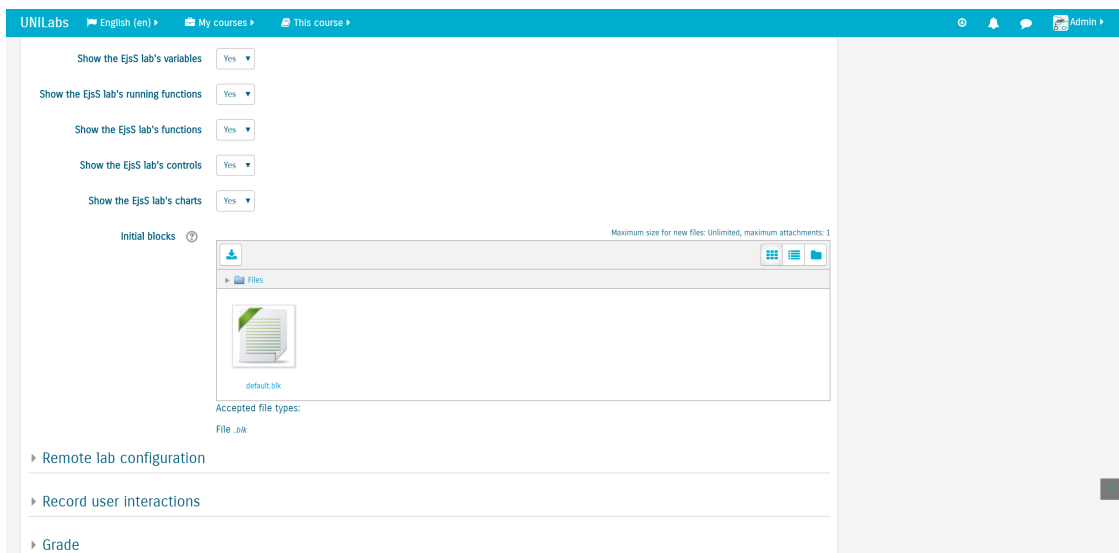


Figura 7. Habilitación de la programación visual con Blockly en actividades de EjsS

La Figura 9 muestra un ejemplo de una aplicación EjsS cuya actividad EJSApp se configuró para habilitar la interacción de programación visual y cargar un programa de bloque inicial.

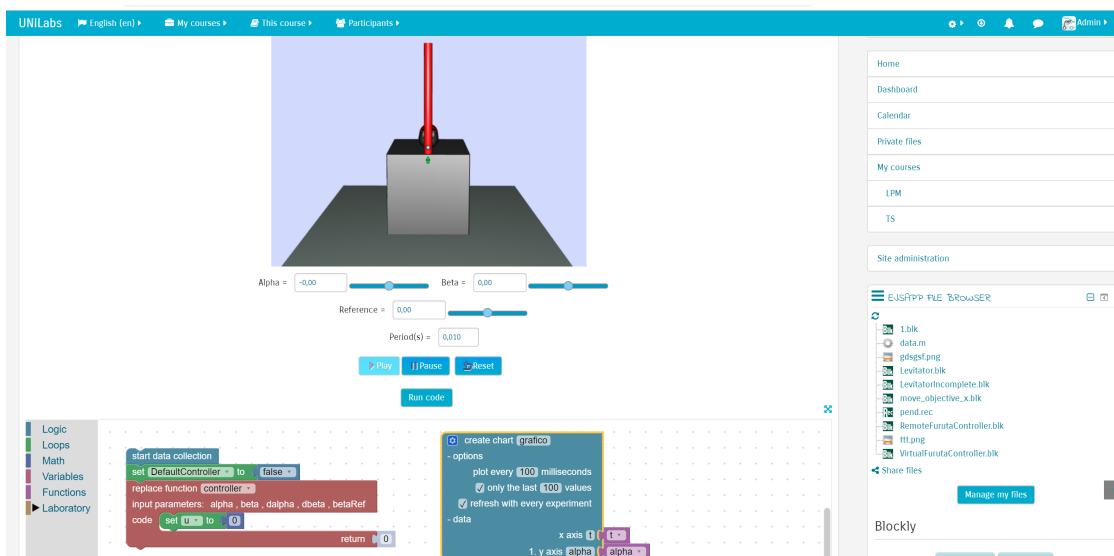


Figura 8. Una actividad de EJSApp con el lenguaje de programación visual habilitado y un programa de bloque predeterminado

3.5. Laboratorios remotos

Al agregar un lab remoto en una actividad de EJSApp, es necesario especificarlo en la sección "Configuración de laboratorio remoto" dentro del formulario de agregar/editar. Esta parte está fuertemente vinculada al bloque Remlab Manager, pero cuando dicho bloque se ha instalado, configurado y utilizado para definir un laboratorio remoto, los identificadores de estos laboratorios ya definidos aparecerán aquí. Entonces, es tan simple como cambiar el "¿Sistema experimental remoto?" en el formulario de adición/editación a "Sí" y seleccionando el identificador de experiencia correcto en el menú desplegable "Identificador de práctica" (consulte la Figura 10).

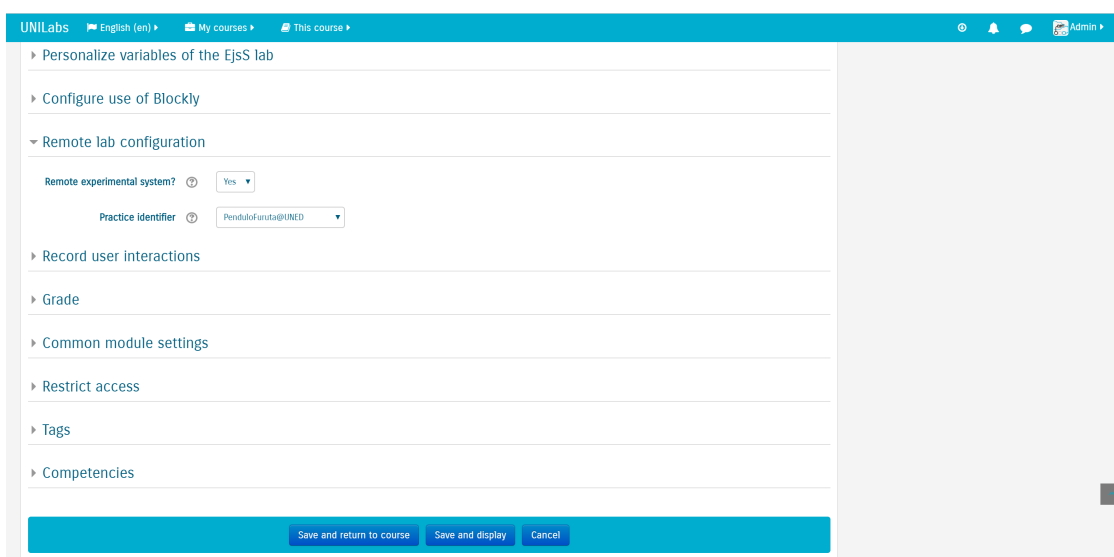


Figura 9. Configuración y actividad de EJSApp como laboratorio remoto