

18/12/2021



ТУТОРИЈАЛ ЗА ПРОЈЕКТОВАЊЕ И  
ТЕХНИЧКУ ИМПЛЕМЕНТАЦИЈУ ВЕБ  
ЛАБОРАТОРИЈА – РАЗВОЈ ВЕБ  
АПЛИКАЦИЈЕ



Co-funded by  
the European Union

## 1. КАРАКТЕРИСТИКЕ EJSApp ВЕБ АПЛИКАЦИЈА УГРАЂЕНИХ У МООДЛЕ ПЛАТФОРМУ СА EjsS

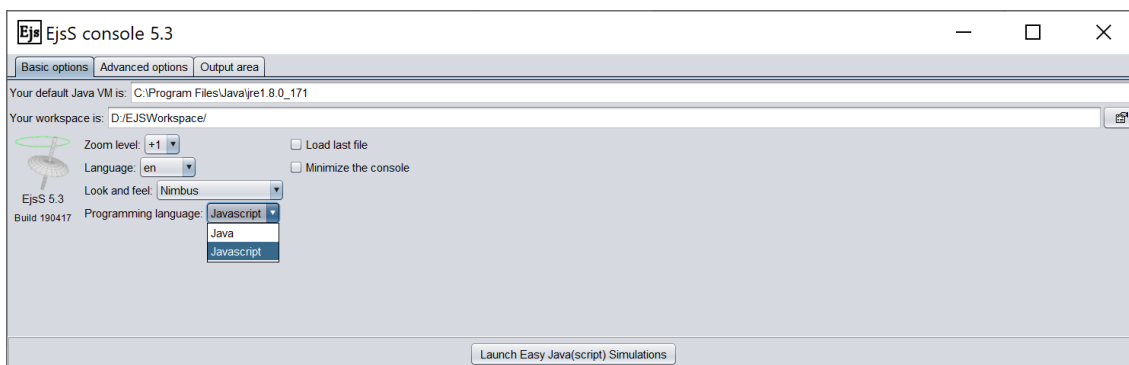
Веб апликације развијене у EJSApp (препоручљиво је да увек користите најновију доступну верзију EjsS, коју можете наћи овде: <https://gitlab.com/EjsS/tool>) који су уграђени у *Moodle Learning Management System* (Моодле) окружење као EJSApp активности могу добити следеће карактеристике и функционалности:

- Додавање нове активности учења у Моодле курс заснован на веб апликацијама (EJSApp).
- Интеграција вишејезичне подршке за веб апликације са Моодле-ом.
- Чување датотека (.json, .blk, .res, .txt, .jpg ...), из веб апликације у Моодле-ово приватно складиште датотека и читање/учитавање из овог спремишта у веб апликацију.
- Учитавање почетних стања за веб апликацију (датотеке у .json формату).
- Снимање и репродукција интеракција са веб апликацијама (датотеке у .res формату).
- Омогућавање употребе визуелног програмског језика за интеракцију и комуникацију са веб апликацијама.
- Конфигурисање веб апликације да је користи као удаљени интерфејс који се повезује са стварним хардвером.

Овај приручник описује како да добијете све функције интеграције наведене изнад. Поглавље 2 овог документа објашњава како веб апликације треба да буду припремљене у EjsS -у да би правилно функционисале у Моодле-у, док Поглавље 3 показује како да конфигуришете Моодле EJSApp активности. Ако вам је потребна помоћ са Моодле активностима на нивоу корисника, запамтите да је ова платформа само продужетак Моодле-а, па посета Моодле званичној документацији може бити од помоћи: <https://docs.moodle.org/en/Activities>

## 2. ПРИПРЕМА ВЕБ АПЛИКАЦИЈЕ У EjsS -у ЗА ЊИХОВО КОРИШЋЕЊЕ У МООДЛЕ-у

Прво, вреди напоменути да је овај водич ограничен на припрему EjsS апликација у JavaScript -у, пошто Јава верзија није у употреби. Стога је прва ствар да проверите да ли је EjsS покренут у овом моду и, ако не, да то урадите. На слици 1 приказана је картица Основне опције (*Basic options*) у прозору конзоле EjsS -а, из које се JavaScript програмски језик може изабрати.



Слика 1. Постављање JavaScript т-а као подразумеваног режима/програмског језика за рад са EjsS -ом

Ако вам је потребан водич за EjsS, приручник за почетак учења о томе како да користите овај програм можете пронаћи овде:

<https://vvv.um.es/fem/Ejs/VebCasts/en/EJSTranslationFacilities/media/EJSTranslationFacilities.mp4> док је Поглавље 2 посвећено је Јава режиму EjsS -а и Поглавље 4 се фокусира на претварање Јава апликација у JavaScript апликације, можете безбедно да прескочите ова два поглавља и прочитајте остатак документа.

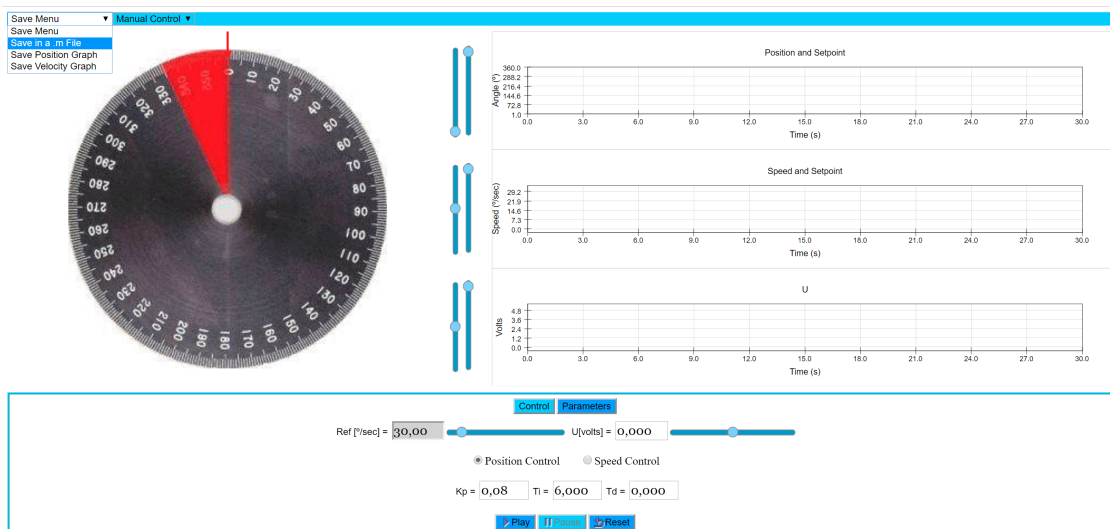
### 2.1. Подршка за више језика

Поставка језика у EjsS веб апликацијама интегрисаним у Моодле са EJSSApp се аутоматски повезује са изабраним језиком у Моодле веб окружењу, када је веб апликација припремљена користећи изворни језик у оквиру вишејезичне подршке коју нуди EjsS. Да припремите апликацију у EjsS -у са овом подршком за више језика, само пратите упутства дата у овом видео туторијалу: <https://www.um.es/fem/Ejs/WebCasts/en/EJSTranslationFacilities/media/EJSTranslationFacilities.mp4>

Важна напомена: Услов је да подразумевани језик веб апликације (тј. језик текстова који су написани у елементима EjsS) је енглески. Затим се могу додати додатни преводи. Супротно (давање, на пример, шпанског језика као подразумеваног језика и додавање енглеског превода касније) чини ово последње да се не приказује у Моодле-у.

### 2.2. Чување и учитавање података

Важно је пратити неке смернице за дизајн корисничких интерфејса који су слични за све апликације. Мени (*Menu*) за чување/читање датотека традиционално се налази на врху апликација, као што је приказано на слици 2.



Слика 2. Мени и тастери за чување/учитавање датотека

Табела 1 нуди листу EjsS изворних метода за чување и учитавање различитих типова датотека. Ове методе могу бити користи се у радњама које се позивају када су изабране опције горњег менија, за чување и учитавање датотека.

Табела 1. Методе чувања и учитавања датотека

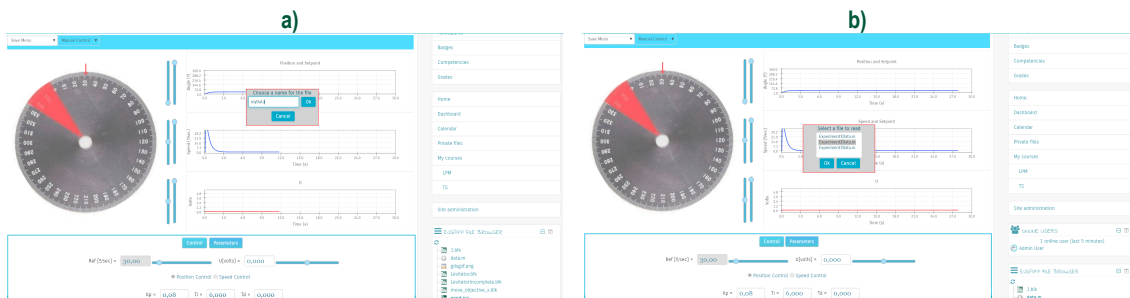
Постоје четири методе које омогућавају слање датотеке из EjsS веб апликације у Моодле-ово приватно складиште датотека:

- Method `_saveText(String FileName, String FileType, String FileContent)`, да сачувате текстуалну датотеку која ће аутоматски поправити, у складу са `FileType`, њену екстензију.
- Method `_saveImage(String FileName, String ViewElementName)`, да бисте сачували слику екрана корисничког интерфејса веб апликације (одређено именом елемента `View` у EjsS).
- Method `_saveState(String FileName)`, да бисте у `.json` датотеку сачували комплетну листу променљивих (и њихових вредности) које дефинишу стање симулација веб апликације.
- Method `_saveVariables(String FileName, List<String> vars)`, да сачувате само назначене променљиве (`vars` parameter) и њихове вредности у `.json` датотеци.

Постоје три начина да повратите податке ускладиштене у датотекама из EjsS веб апликације:

- Method `_readText(String FileName, String FileType, String VariableName)`, за читање садржаја обичне текстуалне датотеке и записивање у променљиву под називом `VariableName`. Када читате обичне текстуалне датотеке, `FileType` треба да буде подешен на `"Text"`. **Важна напомена: да бисте ово користили из Моодле-а, `FileName` мора бити `null`, јер ће апликација показати мени који омогућава кориснику да одабере из које датотеке жели да чита податке.**
- Method `_readState(null)`, за читање и учитавање комплетне листе варијабли симулације веб апликације из `.json` датотеке.
- Method `_readVariables(null, List<String> vars)`, за читање и учитавање листе изабраних варијабли симулације веб апликације из `.json` датотеке.

Све методе чувања отварају модални прозор као што је приказан на слици 3.а, док све методе учитавања отварају прозор као што је приказано на слици 3.б. Не морате сами да креирате ове прозоре, јер се аутоматски креирају методама чувања и учитавања.



Слика 3. Модални прозори креирани методом чувања и учитавања за писање/читање датотека

### 2.3. Лабораторија са удаљеним приступом

Ако је EjsS апликација замишљена да буде лабораторија са удаљеним приступом, потребно је даље радити на повезивању *HTML5* апликација са софтвером/хардвером у лабораторији. Ако желите да користите наш протокол интероперабилности на даљину (RIP) за ово, прочитајте Упутство за кориснике RIP елемента да научите како да конфигуришете и припремите своју EjsS апликацију и упутства за кориснике RIP сервера да бисте научили како да припремите свој лабораторијски уређај да прихвати везе са RIP клијентом. Постоје две имплементације (и њихови одговарајући водичи) за RIP сервер: LabVIEW и Python.

Коначно, ако користите систем ENLARGE да бисте омогућили комуникацију са лабораторијама унутар ваше установе LAN и да управљате даљинским лабораторијским конекцијама, онда такође морате да користите *ENLARGE Proxy EjsS Element*, за који би требало да прочитате *ENLARGE Proxy EjsS Element* корисничко упутство.

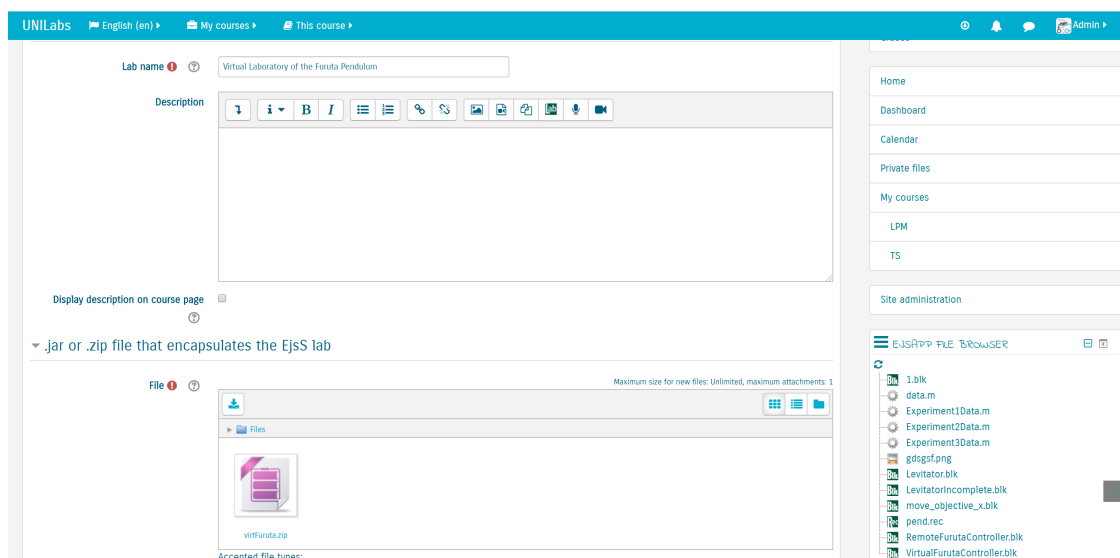
### 3. КОНФИГУРИСАЊЕ И КОРИШЋЕЊЕ EJSApp АКТИВНОСТИ У МООДЛЕ-у

#### 3.1. Додавање EjsS апликације у МООДЛЕ

Слика 4 приказује веб образац за додавање или уређивање активности помоћу EjsS апликације (EJSApp). Једина два поља која треба попунити су назив активности и .zip датотеку која обухвата EjsS апликацију. Поље за назив активности може се попунити коришћењем HTML номенклатуре која омогућава Моодле-у да филтрира текст у зависности од језика који је изабрао корисник. На овај начин можете подесити назив активности оба, на енглеском и шпанском, на пример (погледајте табелу 2) и биће приказана на одговарајућем језику за сваког корисника.

Табела 2. Код за постављање текста у Моодле-у тако да подржава вишејезични филтер

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



Слика 4. Додавање активности EjsS апликације у МООДЛЕ

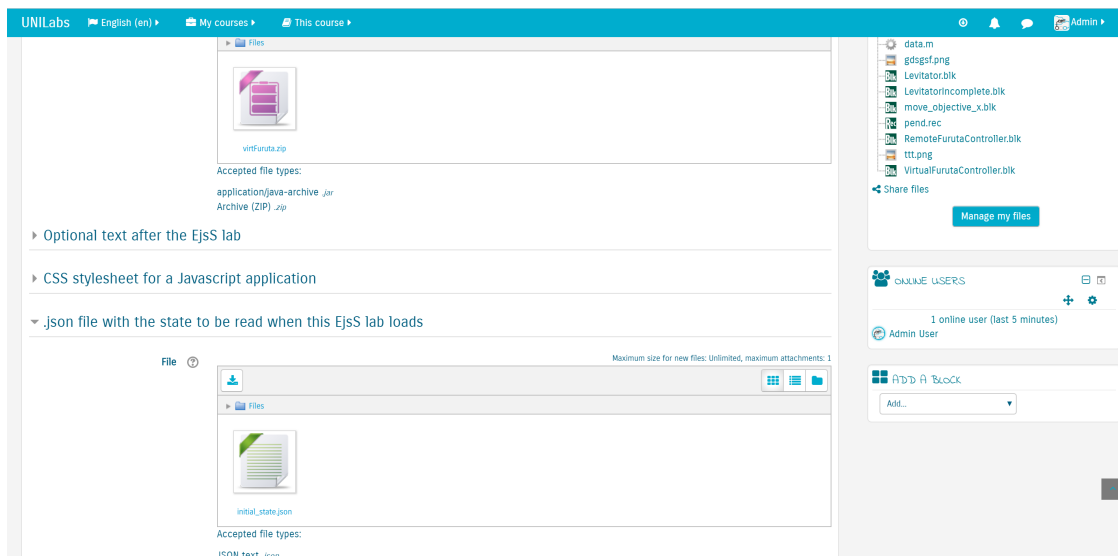
Након што се установи назив активности и отпремљена EjsS .zip датотека, неколико опција може се конфигурирати у активности. Следећи одељци пролазе кроз ове опције.

#### 3.2. Учитавање почетног стања

Када се EJSApp активност креира у Моодле-у, могуће је додати .json датотеку која садржи одређено стање лабораторије. Када се то уради, систем ће аутоматски учитати апликацију у стању садржаном у .json фајлу. Да бисте то урадили, само додајте .json датотеку у *“.json file with the state to be read when this EjsS lab loads”* одељак обрасца за додавање/уређивање (слика 5) активности EJSApp.

Ова опција омогућава учитавање различитог стања од онога које је подразумевано постављено у EjsS апликацији. Ово се може користити за предлагање различитих сценарија или унапред дефинисаних случајева употребе (са посебном дидактичком вредношћу) са истим лабораторијским системом.

Да бисте генерисали ова стања и сачували их у .json датотекама, користите метод `_saveState()` представљен у табели 1.



Слика 5. Учитавање почетног стања у и EJSApp активност

### 3.3. Учитавање снимљене интеракције

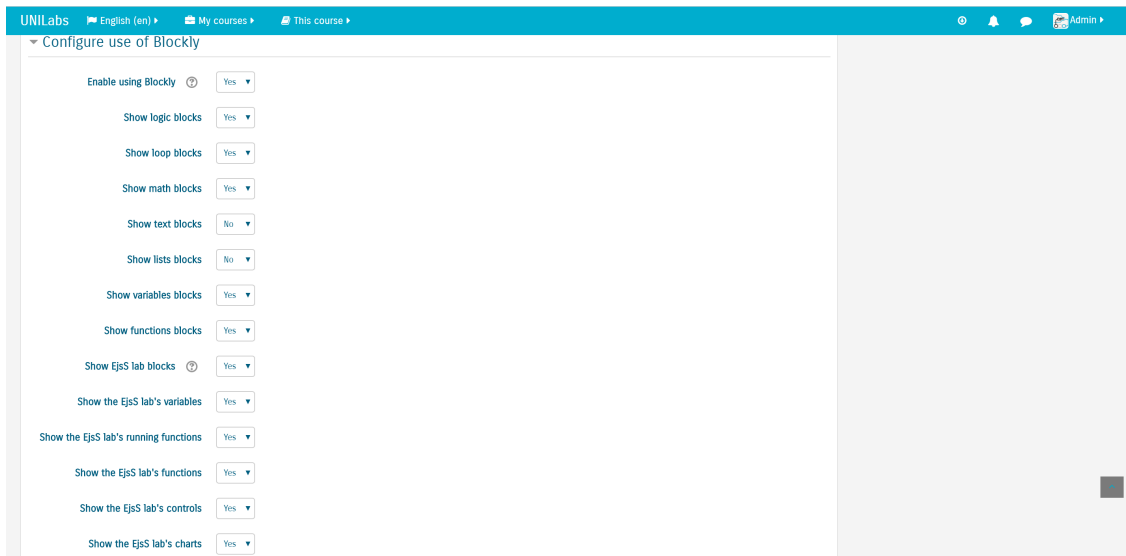
Такође је могуће додати текстуалну датотеку са екстензијом .rec која садржи скрипту са упутствима за аутоматски репродукују интеракције које је извршио корисник са апликацијом EjsC. Ово се може урадити у *“.rec file with the recording to be run when this EjsS lab loads”* одељку обрасца за додавање/уређивање, слично као два претходна.

Ова опција омогућава аутоматско учитавање и репродукцију интеракције коју је корисник извршио са EjsS апликацијом, она је слична видео снимку али са симулационим моделом или комуникацијама удаљене лабораторије у реалном времену. Оне се могу користити као демонстрације, туторијали и тако даље. Да бисте снимили ове интеракције и сачували их у .rec фајловима, користите блок EJSApp File Browser Block.

### 3.4. Интеракција визуелног програмирања са веб апликацијом

База визуелног програмског језика Blockly се може користити да омогући корисницима интеракцију са апликацијом EjsS кроз блок кодирање. Ово отвара многе могућности, приближавајући експериментисање sandbox искуству.

Слика 7 приказује пример конфигурације, где могу бити додате различите категорије програмских блокова у библиотеку блокова које би корисници имали на располагању за рад са EjsS апликацијом.



Слика 6. Пример конфигурације Blockly са EjsS активностима

Поред тога, могуће је додати .блок датотеку ако желимо да се почетни блок програм учита када корисник улази у активност EJSApp, као што је приказано на слици 8.

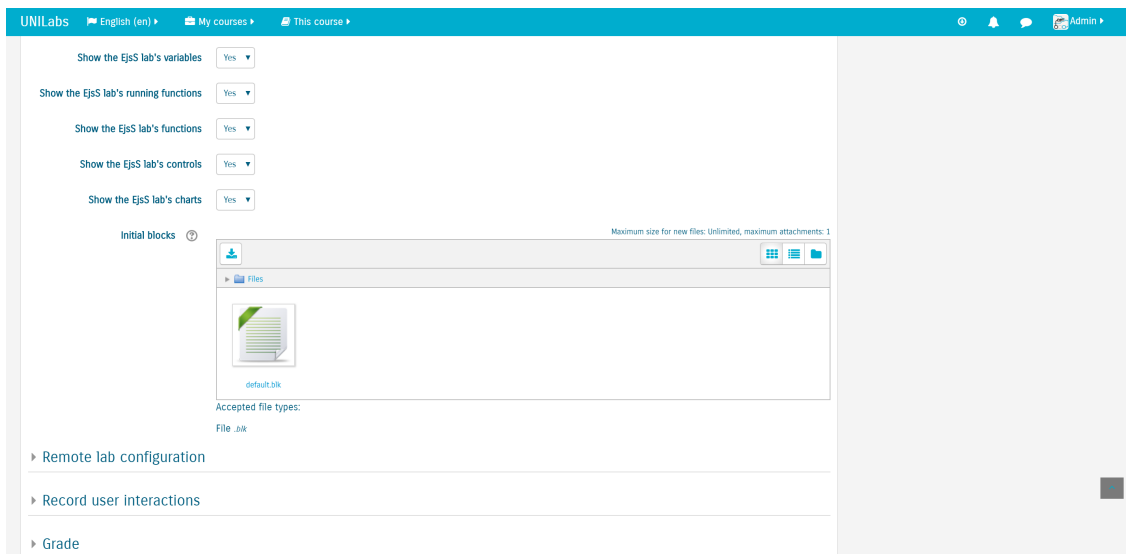
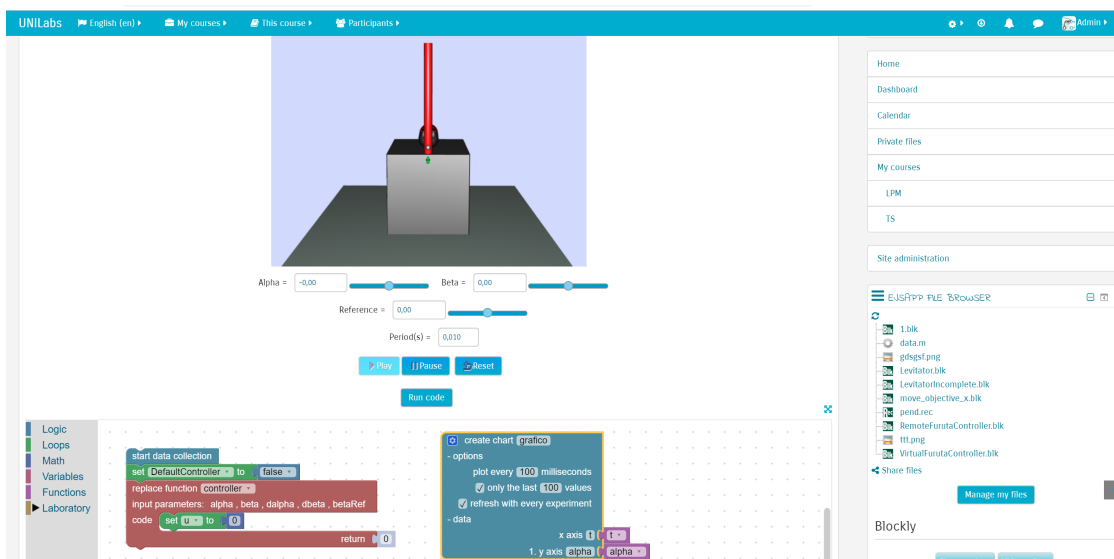


Figure 7. EJSApp активност са омогућеним визуелним програмским језиком и подразумеваним блок програмом

Слика 9 приказује и пример EjsS апликације чија је EJSApp активност конфигурирана да омогући визуелни приказ програмирање интеракције и учитавање почетног блок програма.

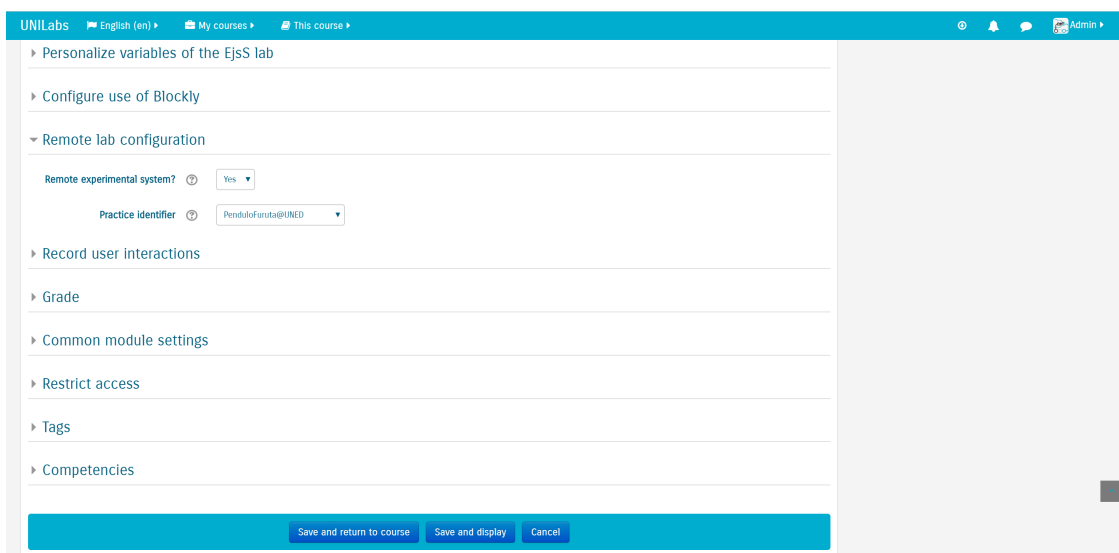




Слика 8. EJSApp активност са омогућеним визуелним програмским језиком и подразумеваним блок програмом

### 3.5. Лабораторија са удаљеним приступом

Када додајете лабораторију са удаљеним приступом у активност EJSApp, потребно је то одабрати у *“Remote lab configuration”* у оквиру обрасца за додавање/уређивање. Овај део је у великој мери повезан са блоком *Remlab Manager*, ако је овај блок инсталиран, конфигурисан и коришћен за дефинисање удаљене лабораторије, идентификатори за њих ће већ бити дефинисани и лабораторије ће се појавити овде. Потребно је то урадити на следећи начин у делу *“Remote experimental system?”* у оквиру падајућег менија у обрасцу за додавање/уређивање одабрати „Yes“ поред тога изабрати прво искуство у оквиру *“Practice identifier”* падајућег менија (погледајте слику 10).



Слика 9. Конфигурисање EJSApp активности као лабораторије са удаљеним приступом