



## SISTEMAS WEB

### 2. Práctica: Buscar Información en eGela

#### OBJETIVO

Usando el IDE PyCharm, debéis programar un cliente web en Python que acceda al curso en eGela para:

- 1.- Descargar los archivos PDF de la asignatura "Sistemas Web" de eGela, organizados en carpetas por temas o secciones.
- 2.- (Opcional) Crear un documento "tareas.csv" con todas las tareas a realizar en el curso, la fecha de entrega y el enlace correspondiente.

El cliente web deberá cumplir la siguiente condición:

- Para la implantación y mantenimiento de la sesión http con el servidor de eGela, las cabeceras Cookie y Set-Cookie se gestionarán de forma bruta (**no** se puede utilizar *requests.Session()*)

#### ENTREGABLES

El resultado de la práctica está compuesto por dos entregables:

- **Burp**: un documento pdf con las capturas de pantalla de las solicitudes que hay que realizar al servidor de eGela para establecer la sesión HTTP, mostrando el nombre del usuario en la última.
- **Python**: cliente web que descarga los archivos PDF de la asignatura y genera el documento *tareas.csv*.

Cada entregable deberán estar **perfectamente identificado** con una cabecera que incluya:

- Nombre y apellidos del estudiante
- Asignatura y grupo
- Fecha de entrega
- Nombre de la tarea
- Breve descripción del entregable

Los entregables deben subir a la tarea disponible en eGela al finalizar la práctica para cada uno de los grupos.

## PASOS E INSTRUCCIONES PARA LA PRÁCTICA

### 1. Identificar las peticiones que debe realizar el cliente web utilizando el monitor de red del navegador.

Utiliza el navegador, solicitar la página de inicio de sesión en eGela, autenticarte en la misma e identificar las solicitudes que se realizan hasta que aparece la página de este curso (Sistemas Web). Para ello utiliza el monitor de red de las herramientas de desarrollo del navegador ([https://developer.mozilla.org/en-US/docs/Tools/Network\\_Monitor](https://developer.mozilla.org/en-US/docs/Tools/Network_Monitor)).

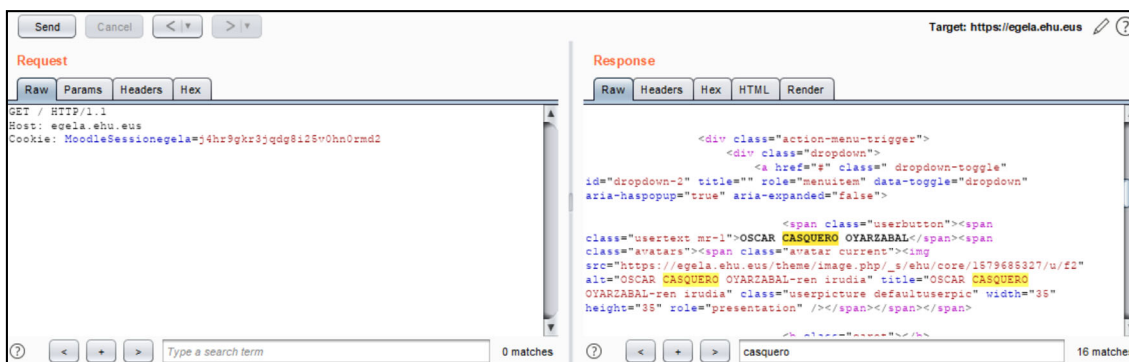
Nota: la solicitud de la página principal de inicio <https://egela.ehu.eus/> no se tendrá en cuenta. El proceso comenzará solicitando la hoja de login <https://egela.ehu.eus/login/index.php>

**Pregunta:** ¿qué vais a utilizar para hacer las solicitudes? ¿Requests o selenium+geckodriver?

### 2.- Utilizando Burp, conocer el proceso de implementación de la sesión HTTP con el servidor de eGela

Los pasos para llevar a cabo este apartado se dieron en la clase del 08-02-2024 (<https://egela.ehu.eus/mod/resource/view.php?id=7348480>).

Para acreditar la correcta ejecución del procedimiento de implementación, en la respuesta que se devuelve al realizar la última solicitud, deben figurar vuestro nombre y apellidos en el contenido (indicativo de que se ha autenticado correctamente).



### 3.- Programar en Python, utilizando la librería requests, una sesión HTTP con el servidor de eGela.

Basado en el paso anterior, llevaréis a cabo el proceso de implementación en Python. Por lo tanto, tenéis que programar cuatro parejas de solicitudes y respuestas cumpliendo las siguientes condiciones:

- El programa se llamará desde el terminal siguiendo la siguiente estructura:  
`>python eGela_PDF_downloader.py usuario "NOMBRE APELLIDO".` Por ejemplo:

**`>python eGela_PDF_downloader.py bcpalgun "LUZ ALVAREZ"`**



- La contraseña se solicitará utilizando la librería **getpass**.
- Introduciréis manualmente el valor del URI de la primera solicitud ("**hardcoded**"). **Los URIs de las siguientes solicitudes no pueden ser hardcoded, es decir, los valores de los URIs de estas solicitudes deben estar parametrizados** (salir de la respuesta anterior).
- Extraer el valor del parámetro de **logintoken** parseando el HTML (no se puede introducir manualmente).
- Los valores de las cabeceras de **cookies** deben estar parametrizados en función de las respuestas anteriores (no se pueden introducir manualmente).
- El programa no leerá todo el valor de la cabecera Set-Cookie, sino sólo la parte que contenga el modelo **MoodleSessionegela=n03hvgma567kq290985634afsp**. Esta parte es la que incluiréis en la cabecera de Cookie.
- En relación a cada solicitud-respuesta, **en la terminal de ejecución del programa se imprimirá de forma ordenada la siguiente información:**
  - **En la solicitud**, en la primera línea hay que imprimir el método y el URI (completo). Si la solicitud tiene contenido, imprimir ésta en la siguiente línea.
  - En cuanto a **la respuesta**, en la primera línea hay que imprimir el status y la descripción. En las siguientes las cabeceras location y set-cookie (si están)
- En la cuarta solicitud, que devuelve la página de usuario con la lista de asignaturas, se realizará la comprobación de la **autenticación buscando el nombre y apellido** del usuario en HTML (esta búsqueda se puede realizar directamente en la cadena de texto HTML; es decir, no se debe utilizar BeautifulSoup).
  - **Si la autenticación no es correcta**, el programa termina y sale.
  - **Si la autenticación se realiza correctamente**, el mensaje que lo indica se imprime y el programa se detiene hasta que el usuario pulsa cualquier tecla.
  -

*4.- Localizar cada uno de las secciones (temas, pestañas) y sus PDFs, así como las tareas que aparecen eGela de esta asignatura en la estructura del HTML utilizando el bookmarklet "Visual Source Chart".*

A continuación, se realizará una quinta solicitud de la página de **eGela de esta asignatura**. Para ello, parseando el HTML de la anterior solicitud (4ª solicitud que muestra la lista de asignaturas de cada alumno), **se buscará el URI del enlace de esta asignatura** (Sistemas Web).

Al acceder al aula virtual de la asignatura y se deberá:

- Localiza las secciones y sus URIs. Tener en cuenta que la **sección destacada** es diferente
- Identificar los documentos pdf en la página de cada sección.
- Identificar las tareas dentro de las páginas.

**Nota:** para identificar los pdfs no se puede utilizar el atributo src completo de la imagen del icono de los pdf (<https://egela.ehu.eus/theme/image.php/ehu/core/1709188990/f/pdf>). El número del enlace cambia si se realizan algunos cambios.



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea



SISTEMEN INGENIERITZA ETA AUTOMATIKA SAILA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

## Sistemas Web

[Página Principal](#) / [Mis cursos](#) / [Estudios de Grado](#) / [20230\\_363\\_GIIGSI30\\_27702\\_01](#) / [Secciones](#) / [SERVIDOR WEB](#)

General	Recursos SW	HTTP	WEB SCRAPING	SERVIDOR WEB	OAuth	Práctica 1	Práctica 2	Práctica 3
Actividad 4								

### Semana 5



2024-02-23 SERVIDOR-INTRODUCCIÓN Y EJEMPLOS

- Introducción a Tomcat
- Crear Proyecto Web
- Primer servlet - web.xml - Metodos Request y Response



holaMundoCabeceras.java

### Semana 6



2024-02-23 RECOGIDA DE DATOS

- Recogida de datos con formato formulario



HolaMundoCache.java

[Analiza este servlet](#)



2024-02-29 SEPARAR VISTA-LOGICA - REDIRECCIONES

#### 4. Realizar el cliente Python que descarga los PDF.

Utilizando la librería BeautifulSoup, descargar los pdfs organizados dentro de una carpeta organizados en subcarpetas según las secciones.

NOTA: recibir un 200 OK no indica que hayas recibido el PDF.