

## ESCUELA DE INGENIERÍA DE BILBAO

### EXAMEN EXTRAORDINARIO DE GRADO - 2020/06/17

#### ASIGNATURA: Sistemas Web

#### GRADO: Ingeniería Informática de Gestión y Sistemas de Información

#### Nombre y apellidos:

##### Notas:

- Duración examen: 60 min.
- La pregunta 9 vale 2 puntos. El resto de preguntas valen 1 punto cada una.
- Algunas preguntas de test tienen más de una respuesta correcta. Para que la pregunta sea considerada correcta se deben marcar únicamente todas las respuestas correctas.
- Este examen constituye el 60% de la nota final. Tal y como se indica en la guía docente de la asignatura, "Es necesario aprobar el examen escrito para aprobar la asignatura".

1. Cuando se emplea el método *POST* para enviar datos...

- Hay que indicar la longitud de la petición en la cabecera *Content-Length*.
- La respuesta puede ser de tipo 302/303, por ejemplo: cuando un usuario hace login en eGela.
- Generalmente, los datos se incluyen en el cuerpo del mensaje de la petición.
- El tamaño de los datos depende de la MTU de la capa inferior, es decir, del tamaño máximo del campo de datos de la capa TCP.
- Hay que indicar el tipo de los datos (formulario, json, etc) en la cabecera *Content-Type*.

2. Dado el siguiente fichero de despliegue:

```
<servlet>
<servlet-name>DoLoginServlet</servlet-name>
  <servlet-class>DoLogin</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>DoLoginServlet</servlet-name>
  <url-pattern>/do/Login</url-pattern>
</servlet-mapping>
```

¿Cuál de las siguientes URI será una petición correcta al servlet de nombre *DoLoginServlet*?

- <http://mydomain.com/myapp/DoLogin>
- <http://mydomain.com/myapp/DoLoginServlet>
- <http://mydomain.com/myapp/DoLogin/DoLoginServlet>
- <http://mydomain.com/myapp/DoLoginServlet/DoLogin>
- Ninguna de las anteriores

3. A través de un formulario embebido dentro de una página HTML, un usuario envía unos datos a una aplicación alojada en Google App Engine. El servidor devuelve el código de respuesta *405 method not allowed*. ¿Dónde radica el problema?
- a. Los nombres de los campos indicados en el formulario no coinciden con los nombres de las variables indicadas en el servidor.
  - b. Falta indicar la URI de acceso en el fichero *app.yaml*.
  - c. *NullPointerException* a la hora de obtener la cabecera *Method-Type*.
  - d. El método indicado en el formulario es *POST*, pero el método de la clase encargada de gestionar dicha petición es *get()*.
  - e. El método indicado en el formulario es *GET*, pero el método de la clase encargada de gestionar dicha petición es *post()*.
4. El objeto **HttpServletRequest**...
- a. Permite transformar un documento XML en una petición HTTP.
  - b. Permite realizar peticiones HTTP desde código javascript.
  - c. Permite realizar peticiones HTTP cuyas respuestas contienen un cuerpo del mensaje que está formateado únicamente en XML.
  - d. Es el objeto que se utiliza en los métodos *doGet()* y *doPost()* de Tomcat para referenciar una petición HTTP.
  - e. Es el objeto que se utiliza en los métodos *get()* y *post()* de Google App Engine para referenciar una petición HTTP.
5. ¿En OAuth 2.0, a quién identifican las variables *consumer\_key* y *consumer\_secret*?
- a. Al usuario.
  - b. Al programador.
  - c. A la aplicación.
  - d. A la instancia del servidor de aplicaciones.
  - e. Todas las anteriores.
6. Para crear una aplicación web en Google App Engine que únicamente accede al API de Dropbox... (elige la respuesta más adecuada)
- a. Sólo es necesario registrar la aplicación en Dropbox.
  - b. Es necesario crear un proyecto en Google Cloud y registrar la aplicación en Dropbox.
  - c. Es necesario crear un proyecto en Google Cloud y unas credenciales para la aplicación tanto en Google como en Dropbox.
  - d. Es necesario crear un proyecto en Google Cloud, iniciar App Engine en dicho proyecto y crear unas credenciales para la aplicación en Dropbox.
  - e. La *callback\_uri* o *redirect\_uri* utilizada a la hora de registrar la aplicación en Dropbox debe coincidir con la indicada a la hora de crear las credenciales en Google.

7. Con la ayuda de la documentación adjunta del API de Gmail, escribe la petición HTTP que permite recuperar los mensajes no leídos con remitente [sw2019@ehu.es](mailto:sw2019@ehu.es).

HTTP request

GET <https://www.googleapis.com/gmail/v1/users/userId/messages>

Parameters

Parameter name	Value	Description
<b>Path parameters</b>		
userId	string	The user's email address. The special value <code>me</code> can be used to indicate the authenticated user.
<b>Optional query parameters</b>		
includeSpamTrash	boolean	Include messages from SPAM and TRASH in the results. (Default: <code>false</code> )
labelIds[]	list	Only return messages with labels that match all of the specified label IDs.
maxResults	unsigned integer	Maximum number of messages to return.
pageToken	string	Page token to return a specific page of results in the list.
q	string	Only return messages matching the specified query. Supports the same query format as the Gmail search box. For example: <code>"from:someuser@example.com rfc822msgid:&lt;somemsgid@example.com&gt; is:unread"</code> . Parameter cannot be used when accessing the api using the gmail.metadata scope.

Authorization

This request requires authorization with at least one of the following scopes ([read more about authentication and authorization](#)).

```

main2.app
import webapp2

class MainHandler(webapp2.RequestHandler):
    ...
app = webapp2.WSGIApplication([
    ('/GoogleLogin', MainHandler),
], debug=True)

main2.app
import webapp2

class MainHandler(webapp2.RequestHandler):
    ...
class CurrentTime(webapp2.RequestHandler):
    ...
app = webapp2.WSGIApplication([
    ('/horaMain', MainHandler),
    ('/horaActual', CurrentTime),
], debug=True)
  
```

Respuesta:

GET /gmail/v1/users/me/messages?q=from%3ASW2019%40gmail.com+is%3Aunread HTTP/1.1  
 Host: www.googleapis.com  
 Authorization: Bearer

8. Completa el siguiente diagrama de secuencia.

