

Tarea No. 8. Metodos

Barrera Pérez Carlos Tonatihu
Profesor: José Asunción Enríquez Zárate
Web Application Development
Grupo: 3CM9

24 de febrero de 2019

Índice

1. Introducción	3
2. Desarrollo	3
3. Conclusiones	5

1. Introducción

En esta tarea se trabajaron tres métodos diferentes los cuales en un inicio pareciera que hacen lo mismo, sin embargo esto no tiene porque ser del todo cierto, es por esto que se explicara el como funciona cada uno de estos métodos y el funcionamiento que realizan.

2. Desarrollo

```
1 // El metodo recibe los parametros de request y response
2 public void metodoUno(HttpServletRequest request ,
   HttpServletRequest response)
3 {
4     // Se trata de obtener una sesion asociada con el
   request
5     // Si dicha sesion no existe se crea una ya que el
   parametro que se pasa
6     // al metodo getSession es true
7     HttpSession s = request.getSession(true);
8     // Ya que tenemos la sesion introducimos un
   atributo de nombre "nombre"
9     // y con el valor "valor"
10    s.setAttribute("nombre", "valor");
11 }
```

Código 1: Método Uno

```
1 // El metodo recibe los parametros de request y response
2 public void metodoDos(HttpServletRequest request ,
   HttpServletRequest response)
3 {
4     /* Se trata de obtener una sesion asociada con el
   request
5     * Si dicha sesion no existe se crea una ya que el
   parametro que se pasa
6     * al metodo getSession es true
7     */
8     HttpSession s = request.getSession(true);
9     /* Ya que la sesion es valida se intenta remover el
   atributo "nombre"
```

```

10      * si no existe este atributo no pasa nada
11      */
12      s.removeAttribute("nombre");
13      /*
14      * Verificamos si la sesion no es nula ,
15      * lo cual siempre es verdadero
16      * debido a lo explicado con anterioridad
17      */
18      if (s != null){
19          /* Ahora se invalida la sesion por lo cual
20          * ya no se podra utilizar
21          * y se tendra que crear una nueva si es que se
necesita
22          */
23          s.invalidate();
24      }
25  }

```

Código 2: Método Dos

```

1  /* El metodo recibe los parametros de request y response
2  * el metodo tiene por tipo de retorno void , sin embargo en
    el metodo
3  * se regresan datos de tipo boolean , lo cual es un error
4  */
5  public void metodoTres(HttpServletRequest request ,
    HttpServletResponse
6  response) {
7      /* Se trata de obtener una sesion asociada con el
    request
8      * Si dicha sesion no existe no se crea y retorna
    null
9      * ya que el parametro que se pasa
10     * al metodo getSession es false
11     */
12     HttpSession s = request.getSession(false);
13     /*Si la sesion es nula regresamos false*/
14     if (s == null)
15         return false;
16     else {
17         /* Se trata de obtener el atributo "nombre", en el
    caso
18     * de que no exista se obtiene un null y por ende el

```

```
metodo
19     * regresa false en caso contrario regresa true
20     * si su valor de "nombre" no es nulo
21     */
22     return s.getAttribute("nombre") != null;
23 }
24 }
```

Código 3: Método Tres

3. Conclusiones

El analizar distintos tipos de situaciones en el desarrollo de aplicaciones es importante, esta tarea es una muestra de ello. El realizar este ejercicio de análisis permite reforzar el aprendizaje que se tiene.