

LINT REPORT

G-98:

Isabel Arrans Vega

José Manuel Bejarano Pozo

Francisco Javier Beltrán Rabadán

Matthew Bwye Lera

Pablo Colmenero Capote

Guillermo López Rosado

Contenido

Primer análisis de SonarLint..... 2

 Corrección de Bad Smell: Administrator Threshold Show Service..... 2

 Corrección del método obsoleto getHours en AdministratorDashBoardShowService 3

Respecto a posteriores análisis de SonarLint..... 3

 Corrección de Bad Smell: Manager Task Create Service..... 4

 Corrección de Bad Smell: Administrator SpamWord Update Service..... 4

 Corrección de Bad Smell: Administrator Threshold Update Service..... 5

 Corrección de Bad Smell: Manager Task Delete Service..... 5

Primer análisis de SonarLint

Al ejecutar el primer análisis de SonarLint sobre el código de la entrega anterior, nos aparecieron 3 malos olores de código en las clases AdministratorThresholdShowService y AdministratorDashBoardShowService.

Items		
Resource	Date	Description
Administrato		Remove this use of "getHours"; it is deprecated.
Administrato		Remove this use of "getHours"; it is deprecated.
Administrato		Immediately return this expression instead of assigning it to the temporary variable "result".

Corrección de Bad Smell: Administrator Threshold Show Service

```
@Override
public boolean authorise(final Request<Treshold> request) {
    assert request != null;

    final boolean result = true;

    return result;
}
```

Para corregir el mal olor procedente de esta clase, se eliminó la variable result del método authorise y se devolvió, directamente, el valor true. Quedando el código final

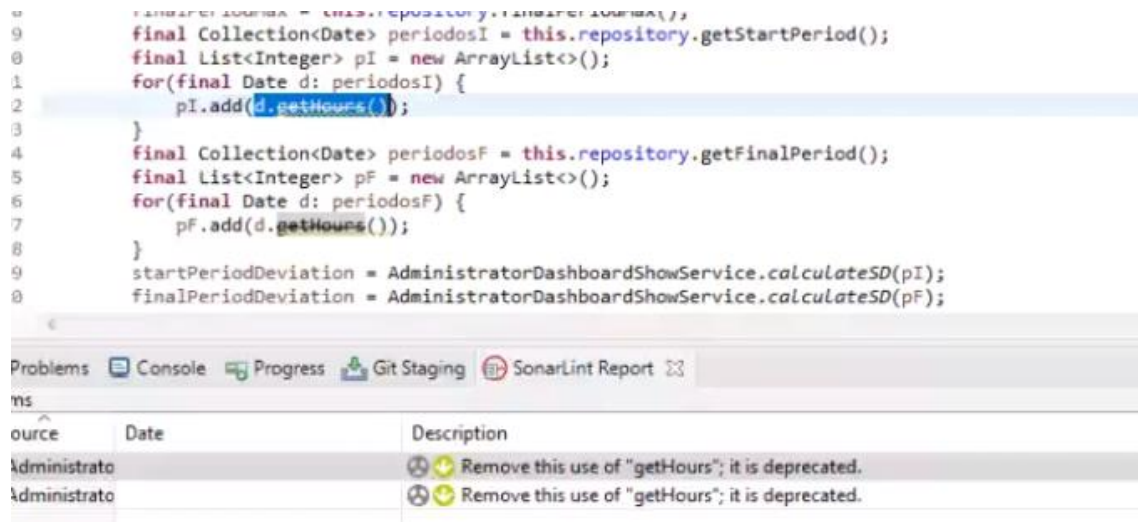
de este método como se muestra en la siguiente imagen:

```

26
27 @Override
28 public boolean authorise(final Request<Treshold> request) {
29     assert request != null;
30     return true;
31 }
32

```

Corrección del método obsoleto getHours en
AdministratorDashBoardShowService



Para poder prescindir del método obsoleto `getHours`, transformamos el tipo `Date` a tipo `Calendar` y almacenamos las horas obtenidas a partir de una nueva instancia de `calendar`. Repetimos esta solución para las dos ocurrencias de este mal olor. El código resultante, tras aplicar los cambios, fue el siguiente para el primero de los fallos:

```

1  startPeriodMax = this.repository.startPeriodMax();
2  finalPeriodMax = this.repository.finalPeriodMax();
3  final Collection<Date> periodosI = this.repository.getStartPeriod();
4  final List<Integer> pI = new ArrayList<>();
5  for (final Date d: periodosI) {
6      Calendar c = Calendar.getInstance();
7      c.setTime(d);
8      pI.add(c.get(Calendar.HOUR));
9  }

```

Respecto a posteriores análisis de SonarLint

En versiones posteriores de Acme-Planner, tras realizar el análisis hemos detectado nuevos olores de fácil resolución.

Corrección de Bad Smell: Manager Task Create Service

Nos encontramos con un mal olor del tipo **Remove this useless assignment to local variable "fin" and "inicio"**, debido a que estamos instanciando e inicializando dos variables con las cuales después no se transportan ni procesan, es decir, no tienen ningún tipo de utilidad:

```
@Override
public Task instantiate(final Request<Task> request) {
    assert request != null;

    Task result;
    Date inicio;
    Date fin;

    inicio = new Date(System.currentTimeMillis() - 1);
    fin = new Date(System.currentTimeMillis() - 1);

    result = new Task();
    result.setPublica(true);
    result.setTitulo("Titulo");
    result.setCargaTrabajo(1.);
    result.setDescripcion("Descripcion");
    result.setEnlace("https://clockify.me/tracker");

    Principal principal;
    principal = request.getPrincipal();

    result.setManager(this.repository.findOneManagerById(principal.getActiveRoleId()));
}
```

Para solucionarlo, únicamente las eliminamos:

```
@Override
public Task instantiate(final Request<Task> request) {
    assert request != null;

    Task result;

    result = new Task();
    result.setPublica(true);
    result.setTitulo("Titulo");
    result.setCargaTrabajo(1.);
    result.setDescripcion("Descripcion");
    result.setEnlace("https://clockify.me/tracker");

    Principal principal;
    principal = request.getPrincipal();

    result.setManager(this.repository.findOneManagerById(principal.getActiveRoleId()));

    return result;
}
```

Corrección de Bad Smell: Administrator SpamWord Update Service

Nos encontramos con un mal olor del tipo **Add a nested comment explaining why this method is empty**, debido a que dentro de este servicio existen métodos vacíos:

```
@Override
public void unbind(final Request<Spamword> request, final Spamword entity, final Model model) {
}
```

Para solucionar este mal olor, lo que hacemos es añadir un comentario indicando el motivo por el cual esta vacío.

```
@Override
public void unbind(final Request<Spamword> request, final Spamword entity, final Model model) {
    //There is nothing to unbind in this service
}
```

Corrección de Bad Smell: Administrator Threshold Update Service

De manera idéntica al anterior, nos hayamos ante un mal olor del tipo **Add a nested comment explaining why this method is empty:**

```
@Override
public void unbind(final Request<Threshold> request, final Threshold entity, final Model model) {
}
```

Aplicamos la misma solución añadiendo un comentario:

```
@Override
public void unbind(final Request<Threshold> request, final Threshold entity, final Model model) {
    //There is nothing to unbind in this service
}
```

Corrección de Bad Smell: Manager Task Delete Service

Nos encontramos con un mal olor del tipo **Complete the task associated to this TODO comment**, debido a que dentro de este servicio existen métodos vacíos que tienen un comentario TODO:

```
@Override
public void bind(final Request<Task> request, final Task entity, final Errors errors) {
    // TODO Auto-generated method stub
}

@Override
public void unbind(final Request<Task> request, final Task entity, final Model model) {
    // TODO Auto-generated method stub
}
```

Para solucionar este mal olor lo que hacemos es eliminar dicho comentario, pero por consecuencia de esto aparece de nuevo el mal olor anterior por lo cual, aplicamos la misma solución para este también:

```
@Override
public void bind(final Request<Task> request, final Task entity, final Errors errors) {

    //There is nothing to bind in this service

}

@Override
public void unbind(final Request<Task> request, final Task entity, final Model model) {

    //There is nothing to unbind in this service

}
```