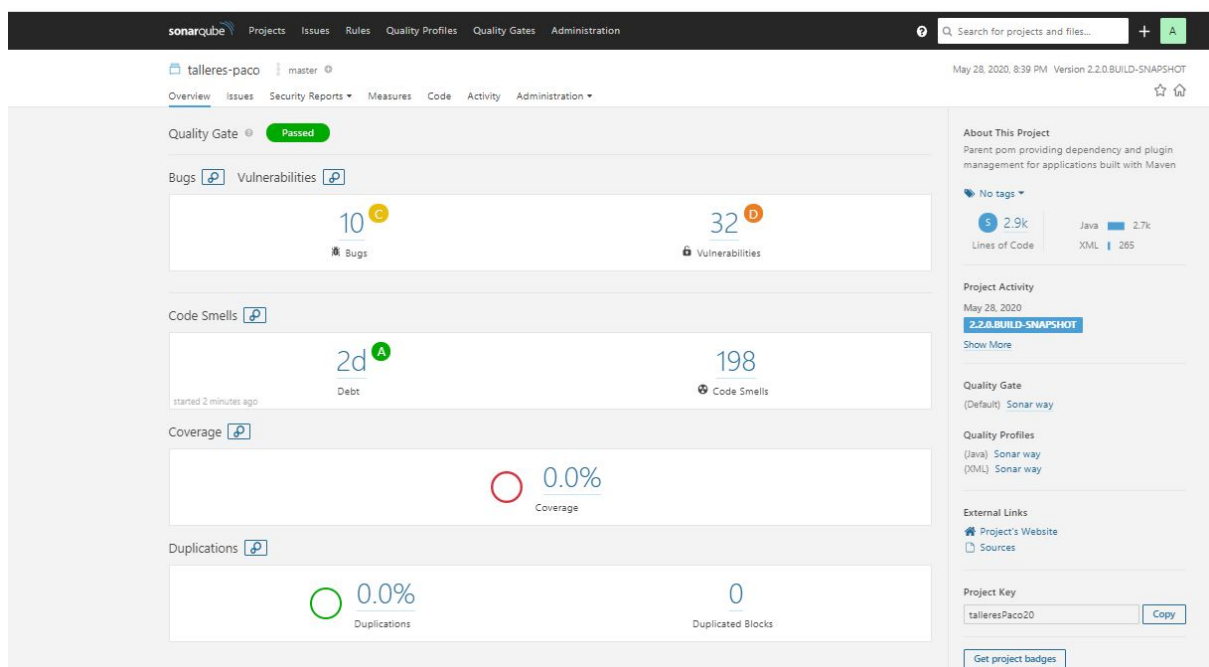


Diseño y pruebas II

Refactoring

Grupo G1-15

Utilizando SonarQube, inicialmente encontramos los siguientes bugs, vulnerabilidades y malos olores:



Hemos corregido los siguiente malos olores:

- *Replace this if-then-else statement by a single return statement*

src/.../samples/talleres/web/AveriaController.java

Replace this if-then-else statement by a single return statement. ***	last month ▾ L74 🔗 📄
🔗 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort	👤 clumsy
Replace this if-then-else statement by a single return statement. ***	2 months ago ▾ L83 🔗 📄
🔗 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort	👤 clumsy
Replace this if-then-else statement by a single return statement. ***	last month ▾ L102 🔗 📄
🔗 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort	👤 clumsy
Replace this if-then-else statement by a single return statement. ***	last month ▾ L113 🔗 📄
🔗 Code Smell 🟡 Minor 🔵 Open Not assigned 2min effort	👤 clumsy

61 / 66 issues

src/.../samples/talleres/web/VehiculoController.java

the regular expression `^[a-z][a-zA-Z0-9]*$`.

Code Smell Minor

src/.../samples/talleres/web/VehiculoControl...

Replace this if-then-else statement by a single return statement.

Code Smell Minor

Return of boolean expressions should not be wrapped into an "if-then-else" statement

```

59 private boolean comprobarIdentidad(final Principal principal, final int vehiculoId) {
60     boolean res = false;
61     Vehiculo vehiculo = this.vehiculoService.findVehiculoById(vehiculoId);
62     if (this.clienteService.findByIdByUsername(principal.getName()).equals(vehiculo.getClient().getId())) {
63         return true;
64     } else {
65         return false;
66     }
67 }

```

Replace this if-then-else statement by a single return statement. ***

Code Smell Minor Open Not assigned 2min effort

2 months ago L61 clumsy

En la última captura se puede apreciar cómo hemos resuelto los cuatro malos olores que aparecen listados arriba.

```

private boolean comprobarIdentidad(final Principal principal, final int vehiculoId) {
    boolean res = false;
    Vehiculo vehiculo = this.vehiculoService.findVehiculoById(vehiculoId);
    if (this.clienteService.findByIdByUsername(principal.getName()).equals(vehiculo.getClient().getId())) {
        res = true;
    }
    return res;
}

```

- Use the "equals" method if value comparison was intended

src/.../samples/talleres/web/CitaController.java

Use the "equals" method if value comparison was intended. ***

last month L96 cert, cwe

Use the "equals" method if value comparison was intended. ***

17 days ago L155 cert, cwe

Use the "equals" method if value comparison was intended. ***

17 days ago L210 cert, cwe

Use the "equals" method if value comparison was intended. ***

2 months ago L283 cert, cwe

Use the "equals" method if value comparison was intended. ***

17 days ago L298 cert, cwe

```

202
203 floco... @GetMapping("/cliente/citas/{citaId}") // cita en detalle
204 floco... public String showCliCitaDetalle(final Principal principal, @PathVariable("citaId") final int citaId, final Map<String, Obje
205 abrga... Cita cita = this.citaService.findCitaById(citaId);
206 Vehiculo vehiculo = cita.getVehiculo();
207 floco...
208 abrga... model.put("cita", cita);
209 floco... model.put("vehiculo", vehiculo);
210 floco... if (cita.getClient().getId() != this.clienteService.findByIdByUsername(principal.getName())) { //comprobar que no est
211 davso... return "redirect:/cliente/citas";
212 }
213 abrga... return "citas/citaEnDetalle";
214 }

```

Use the "equals" method if value comparison was intended. ***

17 days ago L210 cert, cwe

207	207		
208	208		
209	209		
210	-		
210	+		
211	211		
212	212		
213	213		

Este es un bug que encontramos que en nuestro caso no afectaba a la funcionalidad. Aún así, decidimos resolverlo.

```

32
33 @GetMapping("/cliente/citas/{citaId}") // cita en detalle
34 public String showCliCitaDetalle(final Principal principal, @PathVariable("citaId") final int citaId, final Map<String, Object> model) {
35     Cita cita = this.citaService.findCitaById(citaId);
36     Vehiculo vehiculo = cita.getVehiculo();
37
38     model.put("cita", cita);
39     model.put("vehiculo", vehiculo);
40     if (!cita.getCiente().getId().equals(this.clienteService.findByIdByUsername(principal.getName()))) { //comprobar que no esta accedien
41         return "redirect:/cliente/citas";
42     }
43     return "citas/citaEnDetalle";
44 }

```

- Add at least one assertion to this test case

src/.../samples/talleres/web/UITest/ClienteCitaCreaCancelaUITest.java
Add at least one assertion to this test case.
26 days ago
L129
Code Smell
Blocker
Open
Not assigned
10min effort
tests

158 this.driver.findElement(By.linkText("Pedir Cita")).click(); 159 this.driver.findElement(By.linkText("Primero escoge tu vehiculo")).click(); 160 this.driver.findElement(By.linkText("Elegir Vehiculo")).click(); 161 - this.driver.findElement(By.id("fechaCita")).click(); 162 - this.driver.findElement(By.id("30")).click(); 163 this.driver.findElement(By.id("descripcion")).clear(); 164 - this.driver.findElement(By.id("descripcion")).sendKeys("ff"); 165 166 this.driver.findElement(By.xpath("//button[@type='submit']")).click(); 167 168 @AfterEach	158 this.driver.findElement(By.linkText("Pedir Cita")).click(); 159 this.driver.findElement(By.linkText("Primero escoge tu vehiculo")).click(); 160 this.driver.findElement(By.linkText("Elegir Vehiculo")).click(); 161 + this.driver.findElement(By.id("descripcion")).click(); 162 163 this.driver.findElement(By.id("descripcion")).clear(); 164 + this.driver.findElement(By.id("descripcion")).sendKeys("jjjjj"); 165 + this.driver.findElement(By.id("fechaCita")).click(); 166 + this.driver.findElement(By.id("fechaCita")).clear(); 167 + this.driver.findElement(By.id("fechaCita")).sendKeys("22/11/2029 10:10"); 168 + this.driver.findElement(By.xpath("//form[@id='add-cita-form']/div[2]/div")).click(); 169 170 this.driver.findElement(By.xpath("//button[@type='submit']")).click(); 171 172 try { 173 Assertions.assertEquals("2029-11-22T10:10", 174 this.driver.findElement(By.xpath("//table[@id='citasTable']/tbody/tr[3]/td[2]")).getText()); 175 } catch (Error e) { 176 this.verificationErrors.append(e.toString()); 177 } 178 179 try { 180 Assertions.assertEquals("2345FCL", 181 this.driver.findElement(By.xpath("//table[@id='citasTable']/tbody/tr[3]/td[5]")).getText()); 182 } catch (Error e) { 183 this.verificationErrors.append(e.toString()); 184 } 185 186 } 187 188 @AfterEach
--	---

En la última captura se puede apreciar como se soluciona el Code Smell indicado en la captura inmediatamente anterior. Este Code Smell era de tipo Dispensable, ya que hacía que el test fuera un Lazy Test (ya que no podía ser una Lazy Class solo por un test).

- Replace this use of System.out by a logger

Replace this use of System.out or System.err by a logger.
2 months ago
L248
Code Smell
Major
Open
Not assigned
10min effort
bad-practice, cert

Replace this use of System.out or System.err by a logger.
2 months ago
L294
Code Smell
Major
Open
Not assigned
10min effort
bad-practice, cert



Noncompliant Code Example

```
System.out.println("My Message"); // Noncompliant
```

Compliant Solution

```
logger.log("My Message");
```

<pre> 57 - private final CitaService citaService; 58 - private final MecanicoService mecanicoService; 59 - private final VehiculoService vehiculoService; 60 - private final ClienteService clienteService; 61 - 62 - private static final String VIEWS_MEC_UPDATE_FORM = "citas/citaMecUpdate"; 63 - private static final String VIEWS_CLIENTE_CREATE_OR_UPDATE_FORM = "citas/crearCita"; 64 -- </pre>	<pre> 60 + private final CitaService citaService; 61 + private final MecanicoService mecanicoService; 62 + private final VehiculoService vehiculoService; 63 + private final ClienteService clienteService; 64 65 + private static final String VIEWS_MEC_UPDATE_FORM = "citas/citaMecUpdate"; 66 + private static final String VIEWS_CLIENTE_CREATE_OR_UPDATE_FORM = "citas/crearCita"; 67 + private static final Logger LOGGER = Logger.getLogger(CitaController.class.getName()); -- </pre>
<pre> 247 if (result.hasErrors()) { 248 - System.out.println(result.getAllErrors());; 249 - 250 return CitaController.VIEWS_CLIENTE_CREATE_OR_UPDATE_FORM; 251 252 } else { </pre>	<pre> 267 if (result.hasErrors()) { 268 + for (int i = 0; i < result.getAllErrors().size(); i++) { 269 + String error = result.getAllErrors().get(i).toString(); 270 + CitaController.LOGGER.warning(error); 271 + } 272 return CitaController.VIEWS_CLIENTE_CREATE_OR_UPDATE_FORM; 273 274 } else { </pre>

En las 2 últimas capturas se puede apreciar como se soluciona el Code Smell indicado en las otras 2 capturas.

- *Use already defined constant "views_cliente_create_or_update_form instead of duplicating its value:*

<pre> 36 @Controller 37 public class VehiculoController { 38 39 private final VehiculoService vehiculoService; 40 41 private final ClienteService clienteService; 42 43 44 - private final CitaService citaService; 45 - 46 - private final MecanicoService mecanicoService; 47 - 48 49 - private static final String VIEWS_CLIENTE_VEHICULO_CREATE_OR_UPDATE_FORM = "vehiculos/crearVehiculo"; 50 51 - private static final String VIEWS_CLI_UPDATE_FORM "vehiculos/vehiculoUpdate"; 52 53 </pre>	<pre> 34 @Controller 35 public class VehiculoController { 36 37 + private static final String REDIRECT_CLIENTE_VEHICULOS = "redirect:/cliente/vehiculos"; 38 + 39 private final VehiculoService vehiculoService; 40 41 private final ClienteService clienteService; 42 43 + private final CitaService citaService; 44 45 + private final MecanicoService mecanicoService; 46 47 + private static final String VIEWS_CLIENTE_VEHICULO_CREATE_FORM = "vehiculos/crearVehiculo"; 48 49 + private static final String VIEWS_CLI_UPDATE_FORM "vehiculos/vehiculoUpdate"; 50 51 </pre>
--	--

- **Eliminar la excepción dedicada DataAccess:**

<div> <div>11</div> <div>src/main/java/org/springframework/samples/talleres/repository/VehiculoRepository.java</div> </div>	
<div> <div>18</div> <div>19 import java.util.Collection;</div> <div>20</div> <div>21 - import org.springframework.dao.DataAccessException;</div> <div>22 import org.springframework.samples.talleres.model.Vehiculo;</div> <div>23</div> <div>24 /**</div> </div>	<div> <div>18</div> <div>19 import java.util.Collection;</div> <div>20</div> <div>21 import org.springframework.samples.talleres.model.Vehiculo;</div> <div>22</div> <div>23 /**</div> </div>
<div> <div>34</div> <div>35 public interface VehiculoRepository {</div> <div>36</div> <div>37 - void save(Vehiculo vehiculo) throws DataAccessException;</div> <div>38 - Collection<Vehiculo> findVehiculoByClienteId(Integer clienteId) throws</div> <div>39 - DataAccessException;</div> <div>40 - Vehiculo findVehiculoById(int vehiculoId) throws DataAccessException;</div> <div>41</div> <div>42 }</div> <div>43</div> </div>	<div> <div>33</div> <div>34 public interface VehiculoRepository {</div> <div>35</div> <div>36 + void save(Vehiculo vehiculo);</div> <div>37 + Collection<Vehiculo> findVehiculoByClienteId(Integer clienteId);</div> <div>38 + Vehiculo findVehiculoById(int vehiculoId);</div> <div>39</div> <div>40 }</div> <div>41</div> <div>42 }</div> </div>
<div> <div>7</div> <div>src/main/java/org/springframework/samples/talleres/service/VehiculoService.java</div> </div>	
<div> <div>6</div> <div>7 import javax.validation.Valid;</div> <div>8</div> <div>9 import org.springframework.beans.factory.annotation.Autowired;</div> <div>10 import org.springframework.samples.talleres.model.Vehiculo;</div> <div>11 import org.springframework.samples.talleres.repository.VehiculoRepository;</div> <div>12 import org.springframework.stereotype.Service;</div> </div>	<div> <div>6</div> <div>7 import javax.validation.Valid;</div> <div>8</div> <div>9 import org.springframework.samples.talleres.model.Vehiculo;</div> <div>10 import org.springframework.samples.talleres.repository.VehiculoRepository;</div> <div>11 import org.springframework.stereotype.Service;</div> </div>
<div> <div>24</div> <div>25 } // public VehiculoService(final VehiculoRepository vehiculoRepository) {</div> <div>26</div> <div>27 - @Transactional(readOnly = true)</div> <div>28 - public Vehiculo findVehiculoById(final Integer id) throws DataAccessException {</div> <div>29 - return this.vehiculoRepository.findVehiculoById(id);</div> <div>30 - }</div> <div>31 - @Transactional(readOnly = true)</div> <div>32 - public Collection<Vehiculo> findVehiculosByClienteId(final Integer idCliente) throws</div> <div>33 - DataAccessException {</div> <div>34 - return this.vehiculoRepository.findVehiculoByClienteId(idCliente);</div> <div>35 - }</div> <div>36 - public void saveVehiculo(@Valid final Vehiculo vehiculo) throws DataAccessException {</div> <div>37 - this.vehiculoRepository.save(vehiculo);</div> <div>38 - }</div> </div>	<div> <div>23</div> <div>24 } // public VehiculoService(final VehiculoRepository vehiculoRepository) {</div> <div>25</div> <div>26 + @Transactional(readOnly = true)</div> <div>27 + public Vehiculo findVehiculoById(final Integer id) {</div> <div>28 + return this.vehiculoRepository.findVehiculoById(id);</div> <div>29 + }</div> <div>30 + @Transactional(readOnly = true)</div> <div>31 + public Collection<Vehiculo> findVehiculosByClienteId(final Integer idCliente) {</div> <div>32 + return this.vehiculoRepository.findVehiculoByClienteId(idCliente);</div> <div>33 + }</div> <div>34 + public void saveVehiculo(@Valid final Vehiculo vehiculo) {</div> <div>35 + this.vehiculoRepository.save(vehiculo);</div> <div>36 + }</div> <div>37 + }</div> </div>

- **Replace this persistent entity with a simple POJO or DTO object:**

<div> <div>138</div> <div>@GetMapping(value = "/cliente/vehiculos/crear") //el cliente quiere crear un</div> <div>vehiculo</div> </div>	
<div> <div>139</div> <div>- public String vehiculoCreation(final Cliente cliente, final Map<String, Object></div> <div>140</div> <div>- model) {</div> <div>141</div> <div>142 - Vehiculo vehiculo = new Vehiculo();</div> <div>143 - model.put("vehiculo", vehiculo);</div> <div>144 - return "vehiculos/crearVehiculo";</div> <div>145 - }</div> </div>	<div> <div>134</div> <div>@GetMapping(value = "/cliente/vehiculos/crear") //el cliente quiere crear un</div> <div>vehiculo</div> </div>
<div> <div>146</div> <div>- @PostMapping(value = "/cliente/vehiculos/crear") //el cliente crea el vehiculo</div> <div>147</div> <div>- public String vehiculoCreation(final Principal principal, @Valid final Vehiculo</div> <div>148</div> <div>- vehiculo, final BindingResult result, final Map<String, Object> model) throws</div> <div>149</div> <div>- DataAccessException {</div> <div>150</div> <div>151 - if (result.hasErrors()) {</div> <div>152 - System.out.println(result.getAllErrors());</div> <div>153 - return VehiculoController.VIEWS_CLIENTE_VEHICULO_CREATE_OR_UPDATE_FORM;</div> <div>154 - }</div> <div>155 - Integer idCliente =</div> <div>156 - this.clienteService.findByIdByUsername(principal.getName());</div> <div>157 - Collection<Vehiculo> results =</div> <div>158 - this.vehiculoService.findVehiculosByClienteId(idCliente);</div> <div>159 - vehiculo.setCliente(this.clienteService.findClienteById(idCliente));</div> <div>160 - vehiculo.setActivo(true);</div> <div>161 - results.add(vehiculo);</div> <div>162 - this.vehiculoService.saveVehiculo(vehiculo);</div> </div>	<div> <div>141</div> <div>+ @PostMapping(value = "/cliente/vehiculos/crear") //el cliente crea el vehiculo</div> <div>142</div> <div>+ public String vehiculoCreation(final Principal principal, @Valid final Vehiculo coche,</div> <div>143</div> <div>+ final BindingResult result, final Map<String, Object> model) {</div> <div>144</div> <div>145 + if (result.hasErrors()) {</div> <div>146 + //System.out.println(result.getAllErrors());</div> <div>147 + return VehiculoController.VIEWS_CLIENTE_VEHICULO_CREATE_FORM;</div> <div>148 + }</div> <div>149 + Integer idCliente = this.clienteService.findByIdByUsername(principal.getName());</div> <div>150 + Collection<Vehiculo> results =</div> <div>151 + this.vehiculoService.findVehiculosByClienteId(idCliente);</div> <div>152 + coche.setCliente(this.clienteService.findClienteById(idCliente));</div> <div>153 + coche.setActivo(true);</div> <div>154 + results.add(coche);</div> <div>155 + this.vehiculoService.saveVehiculo(coche);</div> </div>

- **Define a constant instead of duplicating this literal “redirect:/cliente/vehiculos” 3 times.**

<pre> 162 model.put("results", results); 163 - return "redirect:/cliente/vehiculos"; 164 } 165 } 166 167 @@ -177,7 +173,7 @@ public String updateVehiculo(@PathVariable("vehiculoId") final int vehiculoId, f 178 } 179 @PostMapping(value = "/cliente/vehiculos/{vehiculoId}/edit") //el cliente 180 - public String updateVehiculo(@Valid final Vehiculo vehiculoEditado, final 181 BindingResult result, @PathVariable("vehiculoId") final int vehiculoId, final Principal 182 principal, final ModelMap model) throws DataAccessException{ 183 if (!this.comprobarIdentidad(principal, vehiculoId)) { 184 return "exception"; 185 } 186 @@ -195,7 +191,7 @@ public String updateVehiculo(@Valid final Vehiculo vehiculoEditado, final Bindin 197 this.vehiculoService.saveVehiculo(vehiculoAntiguo); 198 - return "redirect:/cliente/vehiculos"; 199 } </pre>	<pre> 158 model.put("results", results); 159 + return REDIRECT_CLIENTE_VEHICULOS; 160 } 161 } 162 163 @PostMapping(value = "/cliente/vehiculos/{vehiculoId}/edit") //el cliente actualiza 164 el vehiculo 165 + public String updateVehiculo(@Valid final Vehiculo vehiculoEditado, final 166 BindingResult result, @PathVariable("vehiculoId") final int vehiculoId, final Principal 167 principal, final ModelMap model) { 168 if (!this.comprobarIdentidad(principal, vehiculoId)) { 169 return "exception"; 170 } 171 this.vehiculoService.saveVehiculo(vehiculoAntiguo); 172 + return REDIRECT_CLIENTE_VEHICULOS; 173 } </pre>
<pre> 220 } 221 @PostMapping(value = "/cliente/vehiculos/{vehiculoId}/disable") //el cliente da de 222 baja el vehiculo 223 - public String deshabilitarVehiculoForm(@PathVariable("vehiculoId") final int 224 vehiculoId, final Principal principal, final ModelMap model) throws DataAccessException { 225 if (!this.comprobarIdentidad(principal, vehiculoId)) { 226 return "exception"; 227 } 228 @@ -236,7 +232,7 @@ public String deshabilitarVehiculoForm(@PathVariable("vehiculoId") final int veh 239 this.vehiculoService.saveVehiculo(vehiculo); 240 - return "redirect:/cliente/vehiculos"; 241 } </pre>	<pre> 216 } 217 @PostMapping(value = "/cliente/vehiculos/{vehiculoId}/disable") //el cliente da de 218 baja el vehiculo 219 + public String deshabilitarVehiculoForm(@PathVariable("vehiculoId") final int 220 vehiculoId, final Principal principal, final ModelMap model) { 221 if (!this.comprobarIdentidad(principal, vehiculoId)) { 222 return "exception"; 223 } 224 this.vehiculoService.saveVehiculo(vehiculo); 225 + return REDIRECT_CLIENTE_VEHICULOS; 226 } </pre>

Análisis del proyecto después de refactorizar partes del código:

