

Proyecto Adóptame

Fundamentos de Ingeniería de Software

Jorge Esteban Gómez Zuluaga

Gerardo Elián Martínez Ramírez

André Sebastian Landinez Forero

Sara Rodríguez Urueña

Luis Moreno Sandoval

Pontificia Universidad Javeriana

1. Introducción

Objetivo del proyecto

- Desarrollar una página web para gestionar procesos de adopción de mascotas.
- Implementar metodologías ágiles para organizar y planificar las actividades del equipo de desarrollo.
- Garantizar la calidad del software a través de patrones de diseño y buenas prácticas de desarrollo.

Contexto y relevancia

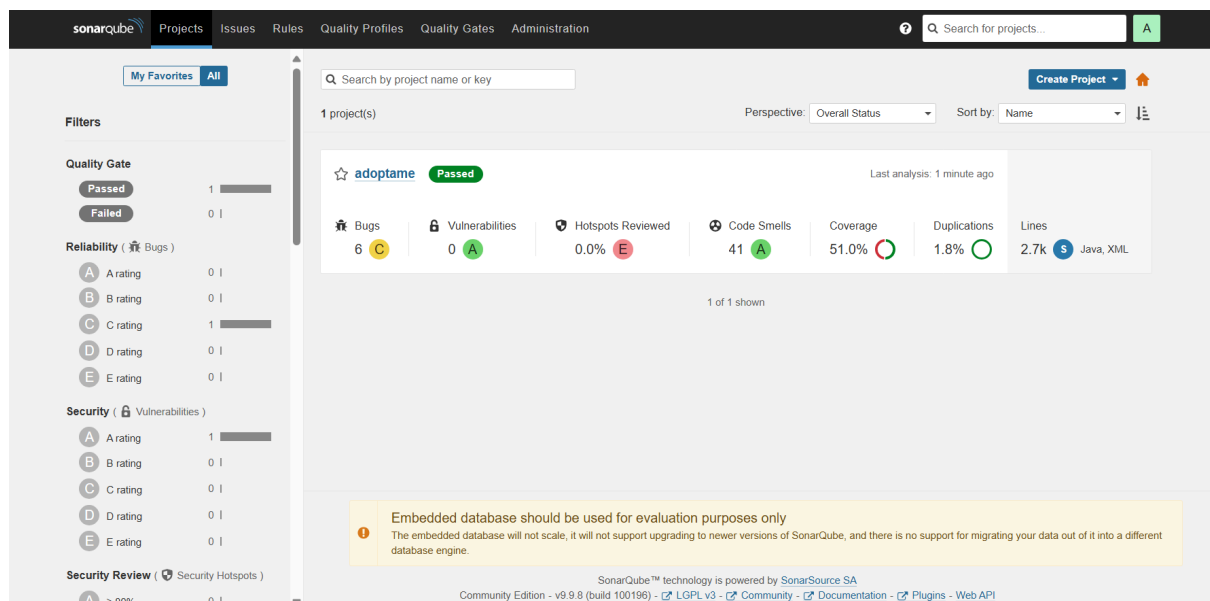
El presente proyecto consiste en el desarrollo de un sistema de adopción de mascotas, el cual busca facilitar la conexión entre personas interesadas en adoptar y las organizaciones que gestionan animales disponibles. El desarrollo se realizó aplicando metodologías ágiles, permitiendo la entrega continua de valor y adaptándose a las necesidades del cliente.

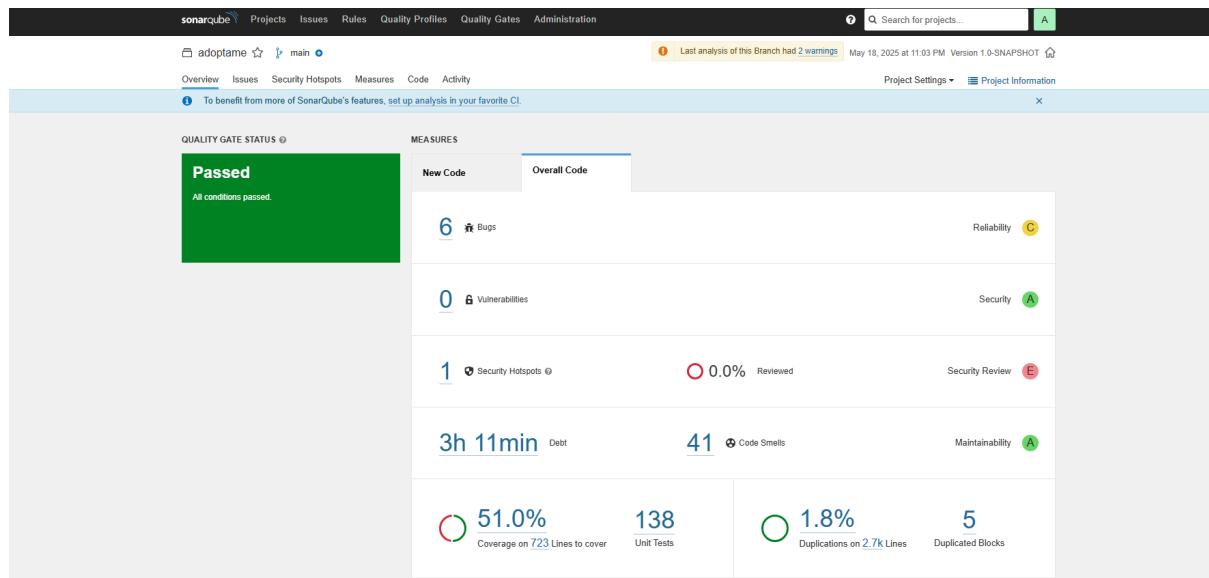
2. Informe

Desarrollo

2.1 Implementación métricas de calidad

SONARQUBE:





En el desarrollo del sistema de adopción de mascotas se implementaron herramientas de apoyo para evaluar la calidad del código y mejorar la comunicación con los usuarios. La primera imagen corresponde al análisis del proyecto en SonarQube, una plataforma utilizada para medir la calidad del software. En ella se observa que el proyecto cumple con todas las condiciones establecidas en el Quality Gate, es decir, no presenta errores críticos ni vulnerabilidades que afecten su estabilidad o seguridad. El análisis indica que no hay errores ni vulnerabilidades, y que se han detectado 41 “code smells”, los cuales son recomendaciones para mejorar el estilo y la mantenibilidad del código. La cobertura de pruebas alcanza un 51%, lo que significa que más de la mitad del código está siendo ejecutado por pruebas automatizadas. Además, el porcentaje de duplicación es bajo (1.8%), y el tiempo estimado para corregir los aspectos señalados es de 3 horas y 11 minutos, lo cual representa una deuda técnica manejable. En conjunto, estas métricas reflejan un código con buen nivel de calidad, mantenibilidad aceptable y una base sólida para escalar el sistema.

La segunda imagen muestra la información del número telefónico proporcionado por la plataforma Twilio, el cual fue configurado e integrado en el sistema para permitir el envío de notificaciones por SMS. Este número, con código de área de Estados Unidos, tiene habilitadas las capacidades de voz, mensajería SMS, mensajes multimedia (MMS) y fax. La integración de Twilio se realizó mediante el uso de su SDK para Java, lo cual permitió automatizar el envío de mensajes de texto a

los usuarios del sistema. Estos mensajes son utilizados para confirmar registros, adopciones y otros eventos relevantes dentro de la plataforma. El uso de esta funcionalidad mejora significativamente la comunicación con los usuarios finales, brindándoles notificaciones en tiempo real y fortaleciendo la experiencia de adopción digital.

ISSUES

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Q

Search for projects...

A

adoptame

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

1 / 47 issues4h 11min effort

My IssuesAll

Filters

Issues in new code

Type

- Bug6
- Vulnerability0
- Code Smell41

Severity

- Blocker0
- Critical2
- Major13
- Minor13
- Info19

Scope

Resolution

Status

Security Category

Creation Date

Language

Rule

Tag

Directory

File

src/main/java/com/fito_2025_g6/auth/CustomUserDetails.java

Replace this instanceof check and cast with 'Instanceof Adoptant adoptant'

Code SmellMinorOpenNot assigned5min effortComment

58 minutes agoL73java16

Replace this instanceof check and cast with 'Instanceof Refuge refuge'

Code SmellMinorOpenNot assigned5min effortComment

58 minutes agoL77java16

Replace this instanceof check and cast with 'Instanceof Administrator administrator'

Code SmellMinorOpenNot assigned5min effortComment

58 minutes agoL81java16

src/_icon/fito_2025_g6/controller/AdministratorController.java

Replace this lambda with method reference 'ResponseEntity::ok'

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL42java8

src/_icon/fito_2025_g6/controller/AdoptantController.java

Replace this lambda with method reference 'ResponseEntity::ok'

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL59java8

Call 'Optional.isPresent()' or 'Optional.isEmpty()' before accessing the value.

BugMajorOpenNot assigned10min effortComment

58 minutes agoL128cwe

src/_icon/fito_2025_g6/controller/ApplicationController.java

Replace this lambda with method reference 'ResponseEntity::ok'

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL52java8

Call 'Optional.isPresent()' or 'Optional.isEmpty()' before accessing the value.

BugMajorOpenNot assigned10min effortComment

58 minutes agoL60cwe

adoptame

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

My IssuesAll

Filters

Issues in new code

Type

- Bug6
- Vulnerability0
- Code Smell41

Severity

- Blocker0
- Critical2
- Major13
- Minor13
- Info19

Scope

Resolution

Status

Security Category

Creation Date

Language

Rule

Tag

Directory

File

Bulk Change

1 / 47 issues4h 11min effort

Replace this lambda with method reference 'Optional::isPresent()' or 'Optional::isEmpty()' before accessing the value.

59 minutes agoL96

cwe

src/_com/fis/_2025_g6/controller/DonationController.java

Replace this lambda with method reference 'ResponseEntity::ok'.

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL49

java8

Replace this lambda with method reference 'Optional::isPresent()' or 'Optional::isEmpty()' before accessing the value.

59 minutes agoL57

cwe

src/_com/fis/_2025_g6/controller/FormController.java

Replace this lambda with method reference 'ResponseEntity::ok'.

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL42

java8

src/_com/fis/_2025_g6/controller/PetController.java

Replace this lambda with method reference 'ResponseEntity::ok'.

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL45

java8

Call 'Optional::isPresent()' or 'Optional::isEmpty()' before accessing the value.

59 minutes agoL64

cwe

src/_com/fis/_2025_g6/controller/RefugeController.java

Replace this lambda with method reference 'ResponseEntity::ok'.

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL46

java8

Call 'Optional::isPresent()' or 'Optional::isEmpty()' before accessing the value.

59 minutes agoL109

cwe

adoptame

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

My IssuesAll

Filters

Issues in new code

Type

- Bug6
- Vulnerability0
- Code Smell41

Severity

- Blocker0
- Critical2
- Major13
- Minor13
- Info19

Scope

Resolution

Status

Security Category

Creation Date

Language

Rule

Tag

Directory

File

Bulk Change 1 Issue(s)

1 / 47 issues4h 11min effort

Define a constant instead of duplicating this literal "SMS enviado a" 3 times.

Code SmellCriticalOpenNot assigned8min effortComment

1 hour agoL41

design

src/_com/fis/_2025_g6/controller/UserController.java

Replace this lambda with method reference 'ResponseEntity::ok'.

Code SmellMinorOpenNot assigned2min effortComment

1 day agoL45

java8

src/_java/com/fis/_2025_g6/entity/Application.java

Make "state" transient or serializable.

Code SmellCriticalOpenNot assigned30min effortComment

4 days agoL45

cwe, serialization

src/_com/fis/_2025_g6/notification/AppNotifier.java

Replace this use of System.out or System.err by a logger.

Code SmellMajorOpenNot assigned10min effortComment

2 days agoL15

bad-practice, cert, owasp-a3

src/_com/fis/_2025_g6/notification/EmailNotifier.java

Replace this use of System.out or System.err by a logger.

Code SmellMajorOpenNot assigned10min effortComment

2 days agoL15

bad-practice, cert, owasp-a3

src/_com/fis/_2025_g6/notification/SMSNotifier.java

Replace this use of System.out or System.err by a logger.

Code SmellMajorOpenNot assigned10min effortComment

2 days agoL16

bad-practice, cert, owasp-a3

src/_java/com/fis/_2025_g6/service/ApplicationService.java

Make this final field static too.

Code SmellMinorOpenNot assigned2min effortComment

23 hours agoL16

convention

src/_java/com/fis/_2025_g6/service/CustomUserDetailsService.java

adoptame

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

My IssuesAll

Filters

Issues in new code

Type

- Bug6
- Vulnerability0
- Code Smell41

Severity

- Blocker0
- Critical2
- Major13
- Minor13
- Info19

Scope

Resolution

Status

Security Category

Creation Date

Language

Rule

Tag

Directory

File

Bulk Change 2 Issue(s)

1 / 47 issues4h 11min effort

src/_java/com/fis/_2025_g6/service/PetService.java

Replace this lambda with method reference 'Refuge.class::cast'.

Code SmellMinorOpenNot assigned2min effortComment

4 days agoL35

java8

src/main/java/com/fis/_2025_g6/util/CoverageReportReader.java

Replace this use of System.out or System.err by a logger.

Code SmellMajorOpenNot assigned10min effortComment

23 hours agoL20

bad-practice, cert, owasp-a3

Replace this use of System.out or System.err by a logger.

Code SmellMajorOpenNot assigned10min effortComment

23 hours agoL22

bad-practice, cert, owasp-a3

Define and throw a dedicated exception instead of using a generic one.

Code SmellMajorOpenNot assigned20min effortComment

23 hours agoL27

cert, cwe, error-handling

src/test/java/com/fis/_2025_g6/AdministratorControllerTest.java

Remove this 'public' modifier.

Code SmellInfoOpenNot assigned2min effortComment

1 day agoL30

unit, tests

src/test/java/com/fis/_2025_g6/AdministratorServiceTest.java

Remove this 'public' modifier.

Code SmellInfoOpenNot assigned2min effortComment

1 day agoL22

unit, tests

src/test/java/com/fis/_2025_g6/AdoptantControllerTest.java

Remove this 'public' modifier.

Code SmellInfoOpenNot assigned2min effortComment

1 day agoL34

unit, tests

The screenshots show the SonarQube interface for a project named 'adoptame'. The top screenshot displays a list of 14 issues, all categorized as 'Code Smell' with a severity of 'Info'. The bottom screenshot displays a similar list of 14 issues, also categorized as 'Code Smell' with a severity of 'Info'. Both screenshots include a sidebar with filters for Type, Severity, Scope, Resolution, Status, Security Category, Creation Date, Language, Rule, Tag, Directory, and File. The main panel displays the details of the selected issues, including the file path, the issue description, and the recommended action to 'Remove this \'public\' modifier.'

Durante el análisis realizado en SonarQube se encontraron varias observaciones relacionadas con la calidad del código, específicamente del tipo “code smell”. Estas no representan errores de ejecución, pero sí afectan aspectos importantes como la legibilidad, mantenibilidad y buenas prácticas. La mayoría de las recomendaciones tienen que ver con el uso adecuado de estructuras de control y funciones de Java.

Se detectaron varios casos donde se utiliza `Optional.get()` sin verificar previamente si el valor está presente, lo cual podría causar errores en tiempo de ejecución. Por eso, se sugiere hacer la validación con `isPresent()` antes de acceder al dato. También se encontró que se está usando `System.out.println` de forma directa, y se recomienda reemplazarlo por un sistema de logs más estructurado que permita un mejor control, sobre todo si el proyecto llega a producción. Además, hay advertencias por la redefinición innecesaria de constantes, el uso

Aunque la mayoría de estas observaciones son de nivel menor o moderado, es importante tenerlas en cuenta para reducir la deuda técnica y mantener un código más limpio y fácil de mantener. Corregir estos detalles no solo mejora la calidad del sistema, sino que también facilita futuras modificaciones, pruebas y el trabajo colaborativo entre los integrantes del equipo.

sonarcube

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

Q

Search for projects...

A

adoptame

☆

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

Project Overview

Reliability

Security

Security Review

Maintainability

Coverage

Duplications

Size

Complexity

Issues

adoptame

Risk

Color: Worse of Reliability Rating and Security Rating Size: Lines of Code

A

B

C

D

E

See the data presented on this chart as a list

Project Overview	
▼ Reliability ?	
Overview	
Overall	
Bugs	6
Rating	C
Remediation Effort	1h
▼ Security ?	
Overview	
Overall	
Vulnerabilities	0
Rating	A
Remediation Effort	0

Project Overview	
▼ Reliability ?	
Overview	
Overall	
Bugs	6
Rating	C
Remediation Effort	1h
▼ Security ?	
Overview	
Overall	
Vulnerabilities	0
Rating	A
Remediation Effort	0

▼ Coverage	
Overview	
Overall	
Coverage	51.0%
Lines to Cover	723
Uncovered Lines	351
Line Coverage	51.5%
Conditions to Cover	134
Uncovered Conditions	69
Condition Coverage	48.5%
Tests	
Unit Tests	138
Errors	0
Failures	0
Skipped	0
Success	100%
Duration	20s

▼ Duplications	
Overview	
Overall	
Density	1.8%
Duplicated Lines	62
Duplicated Blocks	5
Duplicated Files	3
▼ Size	
Lines of Code	2,741
Lines	3,379
Statements	535
Functions	210
Classes	77
Files	78
Comment Lines	14
Comments (%)	0.5%

El análisis de métricas generales del proyecto permitió obtener una visión completa del estado de calidad del código. En el gráfico de dispersión se observa una distribución de archivos según su cobertura de pruebas, nivel de riesgo y deuda técnica. La mayoría de los componentes aparecen en la parte inferior izquierda del gráfico, lo que indica que tienen baja deuda técnica y un riesgo bajo. Solo algunos elementos presentan niveles de cobertura moderados con algo de deuda técnica, pero ninguno se encuentra en zonas críticas que comprometan el proyecto.

En cuanto a los indicadores detallados, se identificaron seis bugs clasificados con una calificación C en confiabilidad, lo cual representa un nivel intermedio que podría mejorar con algunas correcciones puntuales. Por el lado de la seguridad, el análisis no detectó vulnerabilidades, por lo que se asignó una calificación A. La cobertura del código alcanzó un 51%, lo que indica que un poco más de la mitad del total de líneas están siendo cubiertas por pruebas automatizadas. Además, los tests registran una tasa de éxito del 100%, sin errores ni fallos. En cuanto a duplicaciones, el porcentaje es bajo, con un 1.8% de líneas duplicadas, lo cual es un buen indicador de limpieza del código.

Por último, en cuanto al tamaño del proyecto, se reportan más de 2.700 líneas de código distribuidas en 77 clases, 210 funciones y 535 instrucciones, lo que da una idea clara del volumen y estructura del sistema. En general, los resultados muestran que el proyecto se encuentra en un estado saludable, aunque con oportunidades de mejora en la cobertura y la corrección de ciertos bugs detectados.

MÉTRICAS: COMPLEJIDAD CICLOMÁTICA

▼ Complexity ?	
Cyclomatic Complexity	286
Cognitive Complexity	74
▼ Issues	
Issues	47
Open Issues	47
Reopened Issues	0
Confirmed Issues	0
False Positive Issues	0
Won't Fix Issues	0

Cyclomatic Complexity 286

auth	31
config	10
controller	115
dto	1
entity	9
factory	4
notification	7
repository	0
service	84
state	12
util	12
Adoptame.java	1
AdoptionStatus.java	0
ApplicationStatus.java	0

... ..

adoptame / src / main/java/com/fis_2025_g6 / auth

View as Tree

↑ ↓ to select files ← → to navigate

3 files

Cyclomatic Complexity 31

CustomUserDetails.java	18
JwtAuthFilter.java	7
JwtUtil.java	6

3 of 3 shown

[adoptame](#) / [src](#) / [main/java/com/fis_2025_g6](#) / config

View as

Tree

↑

↓

 to select files

←

→

 to navigate

3 files

Cyclomatic Complexity 10

OpenApiConfig.java	1
SecurityConfig.java	8
TwilioConfig.java	1

3 of 3 shown

[adoptame](#) / [src](#) / [main/java/com/fis_2025_g6](#) / controller

View as

Tree

↑

↓

 to select files

←

→

 to navigate

11 files

Cyclomatic Complexity 115

AdministratorController.java	7
AdoptantController.java	20
ApplicationController.java	12
AuthController.java	8
DonationController.java	10
EmailController.java	4
FormController.java	8
PetController.java	12
RefugeController.java	21
SMSController.java	5
UserController.java	8

11 of 11 shown

[adoptame](#) / [src](#) / [main/java/com/fis_2025_g6](#) / factory

View as

Tree

↑

↓

 to select files

←

→

 to navigate

4 files

Cyclomatic Complexity 4

AdministratorFactory.java	1
AdoptantFactory.java	1
RefugeFactory.java	1
UserFactory.java	1

4 of 4 shown

[adoptame](#) / [src](#) / [main/java/com/fis_2025_g6](#) / notification

View as

Tree

↑

↓

 to select files

←

→

 to navigate

5 files

Cyclomatic Complexity 7

AppNotifier.java	2
EmailNotifier.java	2
Notifier.java	0
NotifierDecorator.java	2
SMSNotifier.java	1

5 of 5 shown

adoptame / src / main/java/com/fis_2025_g6 / service

View as

Tree

↑

↓

to select files

←

→

to navigate

12 files

Cyclomatic Complexity 84

AdministratorService.java

5

AdoptantService.java

10

ApplicationService.java

19

CustomUserDetailsService.java

2

DonationService.java

8

EmailService.java

2

FormService.java

6

NotificationService.java

2

PetService.java

11

RefugeService.java

10

SMSService.java

1

UserService.java

8

49 of 49 shown

adoptame / src / main/java/com/fis_2025_g6 / state

View as

Tree

↑

↓

to select files

←

→

to navigate

5 files

Cyclomatic Complexity 12

ApplicationState.java

0

ApprovedState.java

3

CanceledState.java

3

PendingState.java

3

RejectedState.java

3

5 of 5 shown

En total se reportó una complejidad de 286, que está distribuida en varias capas del sistema. El paquete que más acumula es controller, con 115, seguido por service, que tiene 84. Esto tiene sentido porque ahí es donde está concentrada la mayor parte de la lógica del sistema, como el manejo de adopciones, formularios, usuarios, refugios, entre otros. Dentro del paquete controller, noté que las clases AdoptantController, PetController y ApplicationController son las que más complejidad tienen, por lo que serían buenos candidatos para dividir o simplificar en el futuro si es necesario.

También revisé los paquetes con menor complejidad, como auth, config, notification, factory y state, que tienen valores bajos y están bien estructurados, ya que su función es más específica y directa. Además de los valores de complejidad, SonarQube también detectó 47 issues relacionadas con esta métrica. Todas están abiertas, pero no hay ningún caso marcado como falso positivo o reabierto. Aunque en general la complejidad no es crítica, me parece importante tenerla en cuenta para

futuras mejoras, sobre todo en los métodos que tienen muchas condiciones anidadas o que podrían volverse difíciles de mantener con el tiempo.

RESUMEN DEL CÓDIGO

adoptame

main

Last analysis of this Branch had 2 warnings

May 18, 2025 at 11:03 PM

Version 1.0-SNAPSHOT

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

Search for files...

to select files

to navigate

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
adoptame							
src	2,518	6	0	41	1	51.0%	2.0%
src/pom.xml	223	0	0	0	0	—	0.0%

2 of 2 shown

COBERTURA DE LOS TESTS (JaCoCo)

adoptame

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.fis_2025_g6.controller	<div></div>	61%	<div></div>	40%	39	119	117	334	14	80	0	11
com.fis_2025_g6.service	<div></div>	76%	<div></div>	79%	21	88	25	152	17	75	0	12
com.fis_2025_g6.factory	<div></div>	13%		n/a	4	8	25	29	4	8	0	4
com.fis_2025_g6.state	<div></div>	26%		n/a	11	16	12	20	11	16	2	4
com.fis_2025_g6.auth	<div></div>	82%	<div></div>	67%	9	33	6	64	3	19	0	3
com.fis_2025_g6.notification	<div></div>	58%		n/a	3	8	8	19	3	8	0	4
com.fis_2025_g6	<div></div>	91%		n/a	1	4	2	12	1	4	0	3
Total	869 of 2,494	65%	59 of 128	53%	88	276	195	630	53	210	2	41

adoptame > com.fis_2025_g6.controller

com.fis_2025_g6.controller

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ApplicationController	<div></div>	37%	<div></div>	16%	6	12	21	39	3	9	0	1
RefugeController	<div></div>	53%	<div></div>	11%	10	21	20	47	2	12	0	1
AdoptantController	<div></div>	56%	<div></div>	30%	9	20	19	49	2	10	0	1
PetController	<div></div>	50%	<div></div>	50%	3	13	17	35	2	11	0	1
DonationController	<div></div>	46%	<div></div>	75%	3	10	12	26	2	8	0	1
AuthController	<div></div>	69%	<div></div>	70%	3	9	15	37	0	3	0	1
EmailController	<div></div>	48%	<div></div>	33%	3	6	9	19	1	3	0	1
UserController	<div></div>	82%	<div></div>	100%	1	8	4	25	1	7	0	1
FormController	<div></div>	95%	<div></div>	100%	1	8	0	23	1	7	0	1
AdministratorController	<div></div>	100%	<div></div>	100%	0	7	0	19	0	6	0	1
SMSController	<div></div>	100%	<div></div>	100%	0	5	0	15	0	4	0	1
Total	522 of 1,363	61%	45 of 76	40%	39	119	117	334	14	80	0	11

adoptame > com.fis_2025_g6.service

com.fis_2025_g6.service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
PetService		63%		100%	3	13	8	24	3	12	0	1
ApplicationService		80%		62%	7	19	3	35	4	14	0	1
AdoptantService		59%		100%	4	10	4	14	4	9	0	1
CustomUserDetailsService		11%		0%	2	3	5	6	1	2	0	1
RefugeService		68%		100%	3	10	3	14	3	9	0	1
UserService		88%		100%	1	8	1	12	1	7	0	1
DonationService		88%		100%	1	8	1	12	1	7	0	1
FormService		100%		100%	0	6	0	10	0	5	0	1
AdministratorService		100%		100%	0	5	0	7	0	4	0	1
EmailService		100%		n/a	0	2	0	9	0	2	0	1
SMSService		100%		n/a	0	2	0	4	0	2	0	1
NotificationService		100%		n/a	0	2	0	5	0	2	0	1
Total	145 of 611	76%	5 of 24	79%	21	88	25	152	17	75	0	12

adoptame > com.fis_2025_g6.auth

com.fis_2025_g6.auth

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
CustomUserDetails		62%		50%	6	18	3	21	3	12	0	1
JwtAuthFilter		89%		83%	2	8	3	20	0	2	0	1
JwtUtil		100%		75%	1	7	0	23	0	5	0	1
Total	41 of 241	82%	9 of 28	67%	9	33	6	64	3	19	0	3

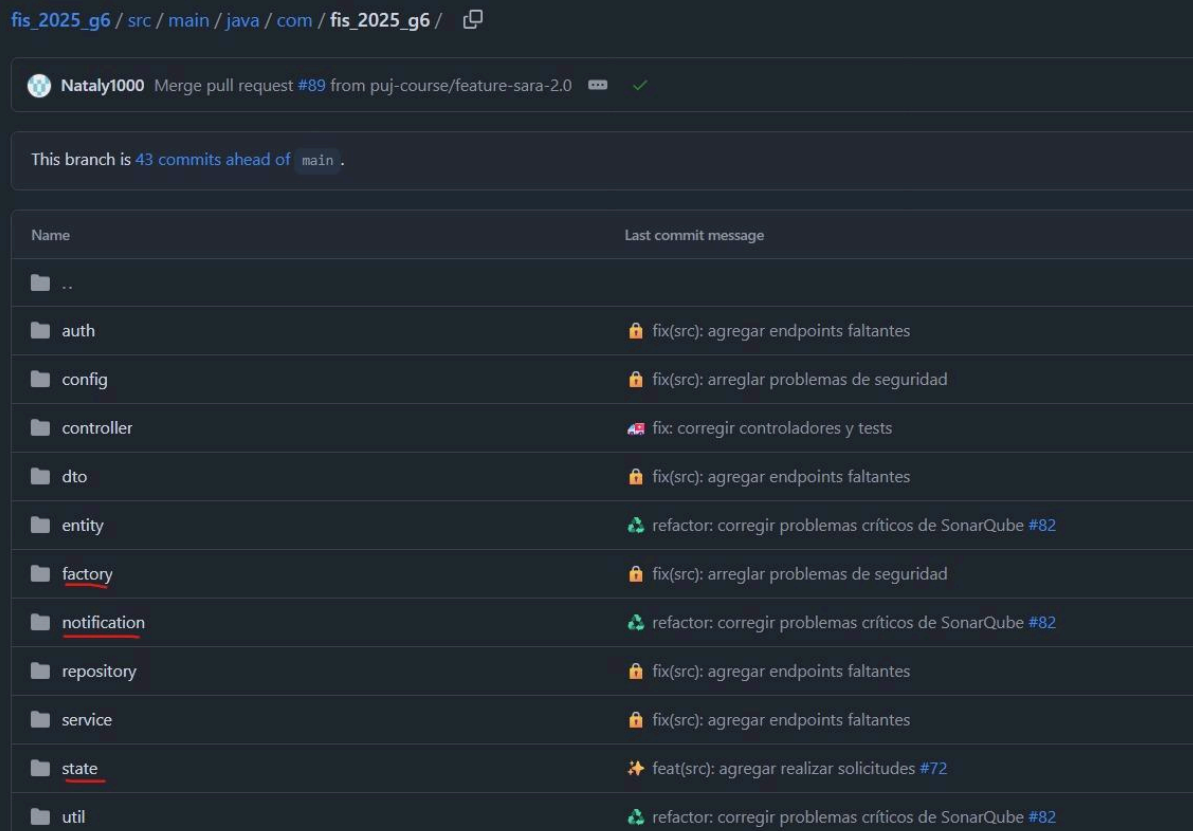
En el análisis de cobertura de pruebas generado mediante JaCoCo, se observa que el proyecto alcanza una cobertura general del 65%, lo cual indica que más de la mitad del código es ejecutado por pruebas automatizadas. Esta cobertura no es uniforme en todos los paquetes del sistema. En el caso del paquete controller, que agrupa los distintos controladores de la aplicación, la cobertura es baja en varias clases. Algunos controladores, como PetController o RefugeController, presentan niveles aceptables de cobertura, pero otros, como EmailController y FormController, no están siendo cubiertos por ninguna prueba, lo que evidencia la necesidad de mejorar esa parte del sistema.

Por otro lado, el paquete service presenta mejores resultados. Varias clases, como NotificationService, UserService y RefugeService, alcanzan un 100% de cobertura, lo que refleja que las pruebas han sido diseñadas y aplicadas de forma adecuada en esa capa. También se destaca el paquete auth, que maneja funciones relacionadas con la autenticación, con valores de cobertura superiores al 80%, lo cual garantiza cierta confiabilidad en esa funcionalidad crítica.

Este análisis permitió identificar qué partes del código están suficientemente probadas y cuáles requieren refuerzo. Aunque el porcentaje general de cobertura se considera aceptable, todavía existen

áreas que deben ser atendidas, en especial aquellas que contienen lógica relevante para el sistema y que actualmente no cuentan con pruebas que verifiquen su correcto funcionamiento.

2.2 Código y patrones



The screenshot shows a GitHub repository interface for 'fis_2025_g6'. At the top, it indicates a merge pull request #89 from 'puj-course/feature-sara-2.0' by user 'Nataly1000'. Below this, a status bar shows 'This branch is 43 commits ahead of main'. The main content is a table with two columns: 'Name' and 'Last commit message'. The table lists various directories and their most recent commit details, including commit type (fix, refactor, feat), commit message, and a link to the commit.

Name	Last commit message
..	
auth	fix(src): agregar endpoints faltantes
config	fix(src): arreglar problemas de seguridad
controller	fix: corregir controladores y tests
dto	fix(src): agregar endpoints faltantes
entity	refactor: corregir problemas críticos de SonarQube #82
factory	fix(src): arreglar problemas de seguridad
notification	refactor: corregir problemas críticos de SonarQube #82
repository	fix(src): agregar endpoints faltantes
service	fix(src): agregar endpoints faltantes
state	feat(src): agregar realizar solicitudes #72
util	refactor: corregir problemas críticos de SonarQube #82

Durante el desarrollo del proyecto se implementaron algunos patrones de diseño que ayudaron a mantener una estructura organizada, modular y fácil de escalar.

Name
..
AdministratorFactory.java
AdoptantFactory.java
RefugeFactory.java
UserFactory.java

Uno de los patrones utilizados fue el patrón Factory, que se encuentra en el paquete factory. Este patrón permitió centralizar la creación de objetos como usuarios, administradores y adoptantes, evitando repetir lógica de instanciación y facilitando el desacoplamiento entre clases.


















Name
..
ApplicationState.java
ApprovedState.java
CanceledState.java
PendingState.java
RejectedState.java

También se hizo uso del patrón State, aplicado en el paquete state, con el fin de manejar los diferentes estados de las solicitudes de adopción. Gracias a este patrón, fue posible representar cada estado como una clase independiente (por ejemplo, pendiente, aprobada o rechazada), lo que hizo que el código fuera más limpio y fácil de mantener, evitando el uso excesivo de estructuras condicionales.

Name
..
AppNotifier.java
EmailNotifier.java
Notifier.java
NotifierDecorator.java
SMSNotifier.java

Por otro lado, en el paquete notification se implementó el patrón Decorator, que sirvió para extender de forma flexible el sistema de notificaciones. Este patrón permitió combinar diferentes tipos de notificación (como correo y SMS) sin necesidad de modificar las clases base. Con esta implementación, el sistema queda preparado para integrar fácilmente nuevos canales de comunicación en el futuro, manteniendo el principio de abierto/cerrado.

2.3 Pruebas unitarias

 AdministratorControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 AdministratorServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 AdoptantControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 AdoptantServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 ApplicationControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 ApplicationServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 AuthControllerTest.java	🐛 fix: corregir controladores y tests
 CustomUserDetailsTest.java	✅ feat(test): agregar pruebas unitarias #81
 DonationControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 DonationServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 FormControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 FormServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 JwtAuthFilterTest.java	✅ feat(test): agregar pruebas unitarias #81
 JwtUtilTest.java	✅ feat(test): agregar pruebas unitarias #81
 PetControllerTest.java	✅ feat(test): agregar pruebas unitarias #81
 PetServiceTest.java	✅ feat(test): agregar pruebas unitarias #81
 RefugeControllerTest.java	✅ feat(test): agregar pruebas unitarias #81

@Test

```
void testFindAll() throws Exception {
    Administrator admin1 = new Administrator();
    admin1.setId(1L);
    admin1.setUsername("admin1");
    admin1.setEmail("admin1@example.com");
    admin1.setPassword("password123");
    admin1.setPhoneNumber("+123456789");
    admin1.setAddress("Calle A");

    Administrator admin2 = new Administrator();
    admin2.setId(2L);
    admin2.setUsername("admin2");
    admin2.setEmail("admin2@example.com");
    admin2.setPassword("password456");
    admin2.setPhoneNumber("+987654321");
    admin2.setAddress("Calle B");

    when(administratorService.findAll()).thenReturn(List.of(admin1, admin2));

    mockMvc.perform(get("/admins")
        .with(user("admin").roles("ADMIN"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$").isArray())
        .andExpect(jsonPath("$[0].username").value("admin1"));
}
```

@Test

```
void testFindById_Found() throws Exception {
    Administrator admin = new Administrator();
    admin.setId(1L);
    admin.setUsername("admin1");
    admin.setEmail("admin1@example.com");
    admin.setPassword("password123");
    admin.setPhoneNumber("+123456789");
    admin.setAddress("Calle A");

    when(administratorService.findById(1L)).thenReturn(Optional.of(admin));

    mockMvc.perform(get("/admins/1")
        .accept(MediaType.APPLICATION_JSON)
        .with(user("admin").roles("ADMIN")))
        .andExpect(status().isOk())
        .andExpect(content().contentTypeCompatibleWith(MediaType.APPLICATION_JSON))
        .andExpect(jsonPath("$.username").value("admin1"));
}
```

@Test

```
void testCreate() throws Exception {
    AdministratorDto dto = new AdministratorDto();
    dto.setUsername("newadmin");
    dto.setEmail("newadmin@example.com");
    dto.setPassword("securePass123");
    dto.setPhoneNumber("+123456789");
    dto.setAddress("Address 123");
}
```

```

@Test
void testCreateAdministrator() {
    Administrator admin = new Administrator();
    admin.setUsername("admin1");
    admin.setEmail("admin1@example.com");
    admin.setPassword("adminpass");

    when(administratorRepository.save(any(Administrator.class))).thenReturn(admin);

    Administrator result = administratorService.create(admin);

    assertNotNull(result);
    assertEquals("admin1", result.getUsername());
}

@Test
void testFindAllAdministrators() {
    List<Administrator> mockList = List.of(new Administrator(), new Administrator());
    when(administratorRepository.findAll()).thenReturn(mockList);

    List<Administrator> result = administratorService.findAll();

    assertNotNull(result);
    assertEquals(2, result.size(), "Debe devolver 2 administradores");
}

@Test
void testFindById_Exists() {
    Administrator admin = new Administrator();
    admin.setId(1L);
    when(administratorRepository.findById(1L)).thenReturn(Optional.of(admin));
}

@Test
void testFindAll() throws Exception {
    Adoptant a1 = new Adoptant();
    a1.setId(1L);
    a1.setUsername("adopt1");
    a1.setAdoptantName("Adoptante Uno");

    Adoptant a2 = new Adoptant();
    a2.setId(2L);
    a2.setUsername("adopt2");
    a2.setAdoptantName("Adoptante Dos");

    when(adoptantService.findAll()).thenReturn(List.of(a1, a2));

    mockMvc.perform(get("/adoptantes")
        .with(user("refugeUser").roles("REFUGE"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$").isArray())
        .andExpect(jsonPath("$[0].username").value("adopt1"))
        .andExpect(jsonPath("$[1].adoptantName").value("Adoptante Dos"));
}

```

```

@Test
@WithMockUser(roles = {"REFUGEE"})
void testFindAll() throws Exception {
    Application app1 = new Application();
    app1.setId(1L);
    Application app2 = new Application();
    app2.setId(2L);

    when(applicationService.findAll()).thenReturn(List.of(app1, app2));

    mockMvc.perform(get("/solicitudes").accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.length()").value(2));
}

@Test
void testFindById_Found() throws Exception {
    Application app = new Application();
    app.setId(1L);

    when(applicationService.findById(1L)).thenReturn(Optional.of(app));

    mockMvc.perform(get("/solicitudes/1").accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.id").value(1));
}

@Test
void testFindAll() throws Exception {
    Donation d1 = new Donation();
    Donation d2 = new Donation();
    when(donationService.findAll()).thenReturn(List.of(d1, d2));

    mockMvc.perform(get("/donaciones")
        .with(user("adminUser").roles("ADMIN"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.length()").value(2));
}

@Test
void testFindById_Found() throws Exception {
    Donation donation = new Donation();
    donation.setId(1L);
    donation.setAmount(50.0);

    when(donationService.findById(1L)).thenReturn(Optional.of(donation));

    mockMvc.perform(get("/donaciones/1")
        .with(user("adminUser").roles("ADMIN"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.id").value(1))

```

`@Test`

```
void testFindApplicationsByPet_Found() throws Exception {  
    Pet pet = new Pet();  
    pet.setId(1L);  
  
    Application app1 = new Application();  
    Application app2 = new Application();  
  
    pet.setApplications(List.of(app1, app2));  
  
    when(petService.findById(1L)).thenReturn(Optional.of(pet));  
  
    mockMvc.perform(get("/mascotas/1/solicitudes")  
        .with(user("refugeUser").roles("REFUGE"))  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isOk())  
        .andExpect(jsonPath("$").isArray())  
        .andExpect(jsonPath("$.length()").value(2));  
}
```

`@Test`

```
void testFindApplicationsByPet_NotFound() throws Exception {  
    when(petService.findById(1L)).thenReturn(Optional.empty());  
  
    mockMvc.perform(get("/mascotas/1/solicitudes")  
        .with(user("refugeUser").roles("REFUGE"))  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isNotFound());  
}
```

```

@Test
void testFindById_Found() throws Exception {
    Refuge refuge = new Refuge();
    refuge.setId(1L);
    refuge.setUsername("refuge1");
    refuge.setRefugeName("Refugio Uno");

    when(refugeService.findById(1L)).thenReturn(Optional.of(refuge));

    mockMvc.perform(get("/refugios/1")
        .with(user("adoptanteUser").roles("ADOPTANT"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.username").value("refuge1"))
        .andExpect(jsonPath("$.refugeName").value("Refugio Uno"));
}

```

```

@Test
void testFindById_NotFound() throws Exception {
    when(refugeService.findById(1L)).thenReturn(Optional.empty());

    mockMvc.perform(get("/refugios/1")
        .with(user("adoptanteUser").roles("ADOPTANT"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isNotFound());
}

```

```

@Test
void testFindById_Found() throws Exception {
    User adoptant = new User();
    adoptant.setId(1L);
    adoptant.setUsername("adopt1");

    when(userService.findById(1L)).thenReturn(Optional.of(adoptant));

    mockMvc.perform(get("/usuarios/1")
        .with(user("adminUser").roles("ADMIN"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.username").value("adopt1"));
}

```





```

@Test
void testFindById_NotFound() throws Exception {
    when(userService.findById(1L)).thenReturn(Optional.empty());

    mockMvc.perform(get("/usuarios/1")
        .with(user("adminUser").roles("ADMIN"))
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isNotFound());
}

```

2.4 Integración servicio SMS

Number	Friendly Name	Capabilities				Active Configuration
		Voice	SMS	MMS	Fax	
+1 857 376 3394 ▲ US, US	(857) 376-3394					Voice Webhook to POST: https://demo.twilio.com/welcome/voice/ Messaging Webhook to POST: https://demo.twilio.com/welcome/sms/reply/

Como parte del desarrollo del sistema de adopción de mascotas, se implementó la funcionalidad de envío de mensajes de texto (SMS) mediante la plataforma Twilio, con el fin de mejorar la comunicación con los usuarios. Esta integración permite que el sistema envíe notificaciones automáticas por SMS en eventos clave, como el registro de nuevos usuarios o la confirmación de adopciones. Para lograr esto, se utilizó el SDK oficial de Twilio para Java y se desarrolló una clase llamada SMS Service, encargada de autenticar la conexión con Twilio y ejecutar el envío de mensajes. El número desde el cual se envían los mensajes fue proporcionado por Twilio y corresponde al +1 857 376 3394, con capacidades habilitadas para envío de SMS, llamadas de voz y mensajes multimedia. El número de teléfono destino no se codifica directamente en el sistema, sino que se obtiene dinámicamente desde la base de datos a través de la clase UsuarioDAO, que consulta la tabla de usuarios y retorna el número asociado al ID correspondiente. Este proceso garantiza que las notificaciones lleguen correctamente al destinatario. Adicionalmente, se tomaron medidas de seguridad para proteger las credenciales de acceso a Twilio, reemplazando los valores sensibles por variables de entorno y evitando que estos se incluyeran en el repositorio del proyecto, cumpliendo con las políticas de GitHub en cuanto al escaneo de secretos. Finalmente, aunque actualmente no se han desarrollado pruebas automatizadas para esta funcionalidad, se contempla incluirlas en versiones futuras del sistema utilizando técnicas de prueba con objetos simulados.

2.5 Postmortém

Fecha: 19/05/2025

Participantes: Jorge Esteban Gómez Zuluaga, Gerardo Elián Martínez Ramírez, André Sebastian Landinez Forero, Sara Rodríguez Urueña.

Objetivo: Evaluar el desarrollo general del proyecto, identificando los principales aciertos, errores y oportunidades de mejora, con el fin de aplicar aprendizajes en proyectos futuros.

Aciertos

- El equipo mantuvo una buena comunicación y disposición para colaborar en todo momento.
- Se logró completar exitosamente todos los entregables, incluso en momentos de presión.
- El trabajo en equipo fue clave para superar los bloqueos técnicos que se presentaron en etapas tempranas.
- Hubo una buena actitud de aprendizaje, lo cual permitió mejorar progresivamente la calidad del código y las decisiones técnicas.
- Se cumplieron los objetivos generales del proyecto, demostrando compromiso y responsabilidad por parte de todos los miembros.

Errores

- En algunas ocasiones, la entrega de tareas se demoró más de lo previsto, lo cual generó que otras fases tuvieran que completarse en menor tiempo.
- Al inicio del proyecto, algunos componentes no funcionaban a la primera, lo cual llevó a que el equipo se estancara momentáneamente en ciertos puntos.
- Al inicio, la ejecución de ciertas funcionalidades resultó más compleja de lo previsto, provocando estancamientos y retrabajos.
- En algunos momentos hubo dependencia de una o dos personas para tareas clave, lo cual afectó la fluidez del avance.
- La estimación de tiempos para ciertas tareas fue demasiado optimista y no contempló posibles contratiempos.

Áreas de Mejora

- Establecer una planificación más detallada, con entregas intermedias claras, para evitar acumulación de trabajo hacia el final.
- Mejorar la distribución de responsabilidades, asegurando que más de un miembro pueda continuar una tarea si hay bloqueos.
- Realizar revisiones técnicas más frecuentes para detectar errores o malentendidos antes de que escalen.

Acciones Concretas para Proyectos Futuros

- Incorporar reuniones semanales de revisión y ajuste de tiempos, para redistribuir tareas según la carga real de trabajo.
- Planificar una semana de margen antes de la entrega final para correcciones y pruebas generales.
- Acordar puntos de control más frecuentes y ligeros.
- Promover la documentación técnica mínima durante el desarrollo para facilitar el traspaso de tareas.
- Establecer acuerdos desde el inicio sobre el manejo de entregas tardías o bloqueos.

Resultados

Se realizó la página web con las funciones necesarias y propuestas en el proyecto desde un inicio, se tuvo en cuenta cada punto propuesto para realizar a detalle lo necesario para que el proyecto fuera un éxito.

Posibles errores

Se tuvieron algunos inconvenientes al inicio con problemas de funcionamiento pero al final fueron solucionados totalmente para que de esa manera el proyecto lograra funcionar.

3. Mejoras al procedimiento

Claridad en las instrucciones

Las instrucciones fueron claras, se tuvieron en cuenta todas las retroalimentaciones dadas.

Tiempo asignado

Se tuvo un tiempo suficiente para realizar todos los puntos correspondientes.

Material de apoyo

Se proporcionaron guías y referencias, además las presentaciones y laboratorios realizados ayudaron en el proceso.

Interacción

Se tuvo varios espacios para poder realizar las preguntas correspondientes además de tener disponible el correo.

Explicaciones claras

El monitor explicó los conceptos de forma clara, asegurando que se comprendiera la metodología.

Relación entre teoría y práctica

El proyecto estuvo bien alineado con los temas vistos en clase, permitiendo aplicar los conocimientos adquiridos.

4. Conclusiones

El proyecto de adopción de mascotas logró cumplir con los objetivos propuestos, aplicando cada punto clave. El uso de cada indicación dada dio como resultado un proyecto exitoso donde al final se logró cumplir con lo necesario y propuesto desde un inicio, para obtener un resultado como se había previsto desde un comienzo.