

Memoria del Trabajo Final - Integración Continua

Alba Aguilar López

<https://moviecards-aguilar.azurewebsites.net/>

<https://moviecards-service-aguilar.azurewebsites.net>

<https://moviecards-pre-aguilar.azurewebsites.net/>

<https://github.com/aaguillo14>

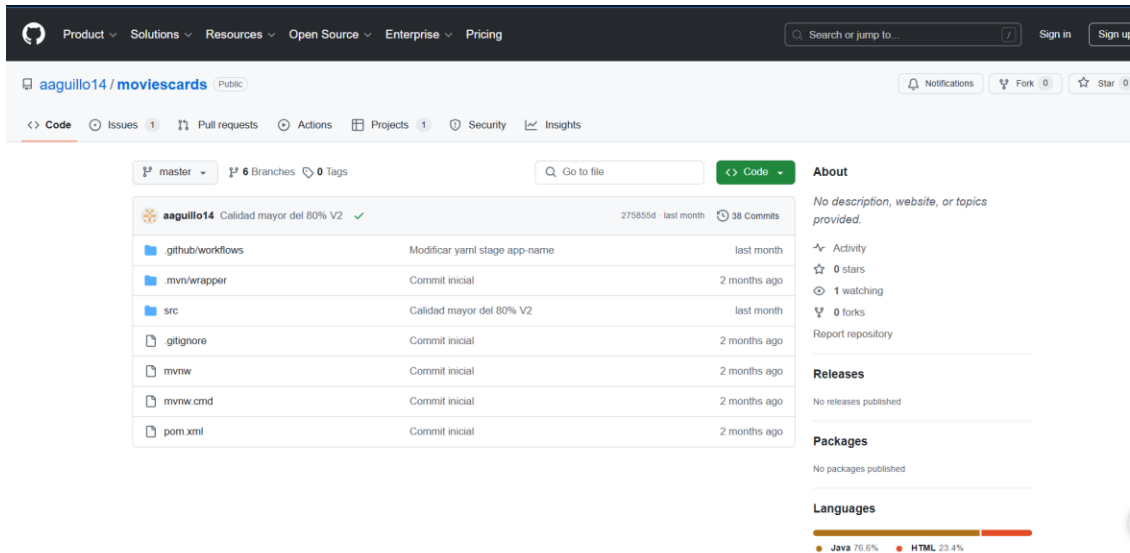
1. Introducción

Este documento detalla el desarrollo y despliegue de la aplicación web "moviecards" y su servicio asociado "moviecards-service", siguiendo los principios de Integración Continua (CI) y Desarrollo Ágil. El objetivo principal ha sido aplicar las prácticas aprendidas en la asignatura "Integración Continua en el Desarrollo Ágil" del Máster Universitario en Desarrollo Ágil de Software para la Web.

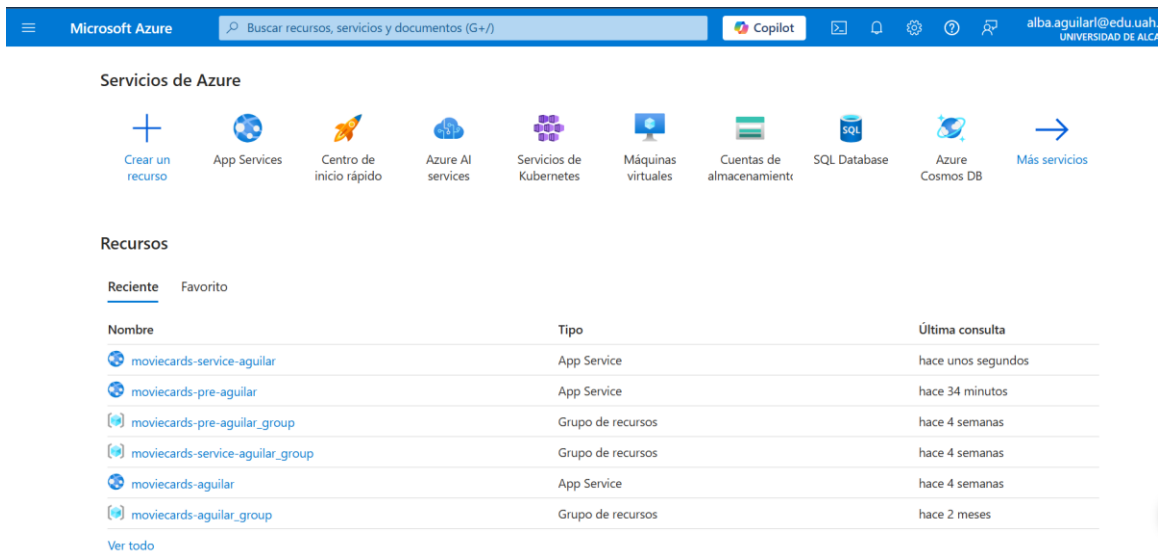
2. Infraestructura y Herramientas Utilizadas

Para el desarrollo y despliegue de la aplicación se han utilizado las siguientes herramientas y servicios:

- GitHub: Para la gestión de versiones y flujos de trabajo CI/CD.



- Azure App Services: Para el despliegue de la aplicación y el servicio.



- GitHub Actions: Para la creación y ejecución de los flujos de integración continua.

Product Solutions Resources Open Source Enterprise Pricing

Search or jump to... 7 Sign in Sign up

aagullo14 / moviecards Public

Notifications Fork 0 Star 0

Code Issues 1 Pull requests Actions Projects 1 Security Insights

Actions

All workflows

CI

Management

Caches

Deployments

All workflows

Showing runs from all workflows

Filter workflow runs

39 workflow runs

Event	Status	Branch	Actor
Calidad mayor del 80% V2	CI #37: Commit 275855d pushed by aagullo14	master	last month 2m 11s
Calidad mayor del 80%	CI #36: Commit 1583582 pushed by aagullo14	master	last month 9m 16s
Calidad menos 5 problemas criticos	CI #35: Commit 9094912 pushed by aagullo14	master	last month 23m 28s
Cambios pruebas funcionales Dead Date	CI #34: Commit b19aeb1 pushed by aagullo14	master	last month 6m 45s

- SonarQube: Para el análisis de la calidad del código.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... A

My Favorites All

Search by project name or key

Create Project

1 project(s) Perspective: Overall Status Sort by: Name

moviecards Passed Last analysis: 27 days ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	2 B	- A	53 A	0.0%	0.0%	645 x9 Java, XML

1 of 1 shown

Filters

Quality Gate

Passed 1

Failed 0

Reliability (Bugs)

A rating 1

B rating 0

C rating 0

D rating 0

E rating 0

Security (Vulnerabilities)

A rating 0

B rating 0

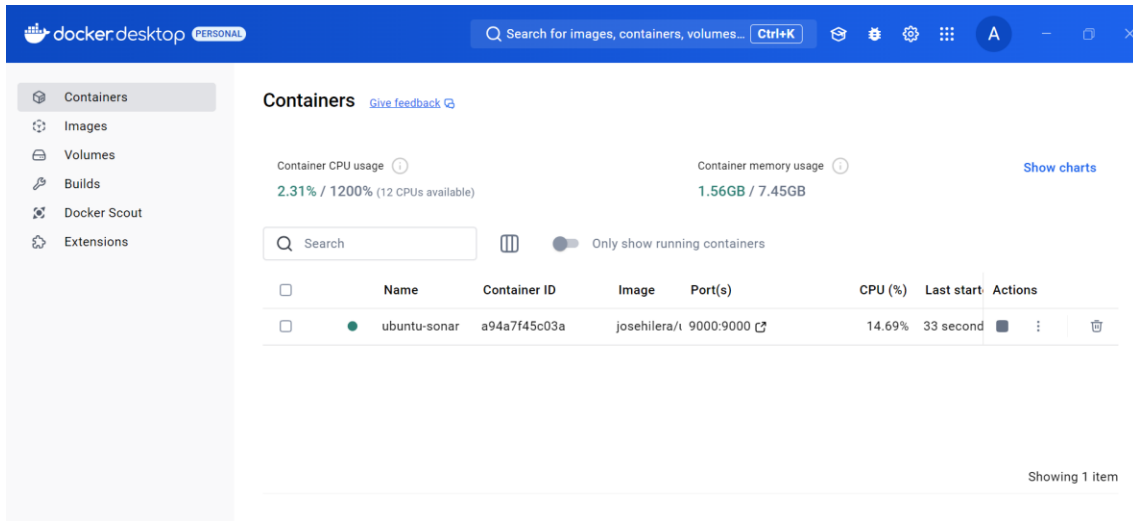
C rating 0

D rating 1

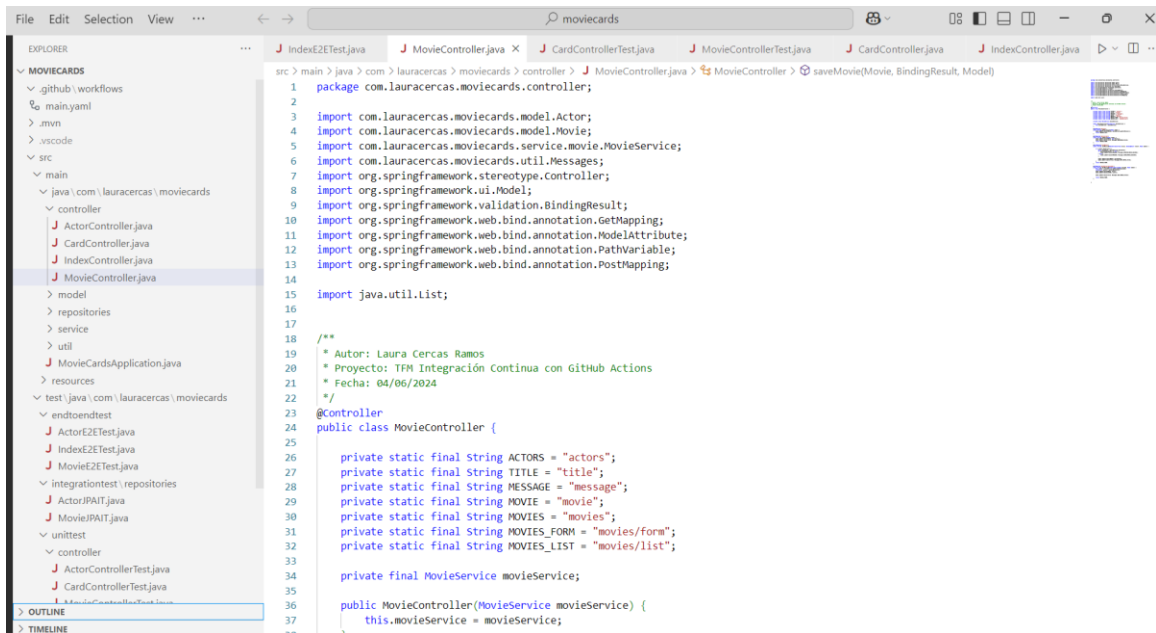
E rating 0

Security Review (Security Hotspots)

- Docker: Para la configuración de entornos y servicios.



- Visual Studio Code: Como entorno de desarrollo.
- Maven: Para la gestión de dependencias y construcción del proyecto.



3. Desarrollo de la Aplicación

La aplicación "moviecards" se ha desarrollado siguiendo las indicaciones del Tema 5 de la asignatura. Inicialmente, se creó un repositorio en GitHub donde se configuraron los flujos de trabajo para compilación, pruebas, control de calidad y despliegue. Se utilizó Spring Boot como framework base y se aseguró la correcta integración y despliegue automático en Azure.

Posteriormente, se separó el backend creando el servicio "moviecards-service", desplegado en Azure y expuesto para ser consumido por aplicaciones externas. La aplicación principal fue modificada para conectarse a este servicio.

El proceso de desarrollo se dividió en varias etapas clave:

- **Configuración del Repositorio:** Se configuró un repositorio para la aplicación "moviecards" y otro para el servicio "moviecards-service", aplicando buenas prácticas de control de versiones.
- **Configuración de GitHub Actions:** Se implementaron flujos de trabajo CI/CD utilizando GitHub Actions para asegurar que cada cambio pasara por pruebas automáticas antes del despliegue.
- **Desarrollo del Backend:** En el repositorio "moviecards-service" se construyó el backend utilizando Spring Boot, exponiendo una API REST para la gestión de actores y películas.
- **Separación del Servicio:** Se refactorizó la aplicación original para que consumiera los datos a través de la API del servicio separado, utilizando RestTemplate para las llamadas HTTP.
- **Actualización del Modelo de Datos:** Se modificó el código del repositorio "moviecards-service" para añadir un nuevo atributo en la clase Actor, llamado deadDate, que contiene la fecha de fallecimiento del actor. Esta modificación se desplegó en Azure en la misma URL del servicio existente.
- **Modificación de la Aplicación Web:** La aplicación "moviecards" se actualizó para permitir al usuario introducir la fecha de fallecimiento al crear un nuevo actor. Además, se añadió una nueva columna en la lista de actores para mostrar esta información.
- **Entorno de Pre-Producción:** se ha añadido al workflow del proyecto un nuevo trabajo, entre los trabajos "qa" y "deploy", llamado "stage" para desplegar la aplicación en un entorno de pre-producción
- **Pruebas y Validación:** Se ajustaron las pruebas unitarias y de integración para contemplar el nuevo atributo deadDate, asegurando que los nuevos cambios no introdujeran errores.
- **Despliegue:** Tras validar los cambios, se desplegaron tanto la aplicación como el servicio en Azure, asegurando su correcto funcionamiento, calidad y accesibilidad.

4. Proceso Detallado Paso a Paso

4.1. Resultado al realizar la práctica del tema 5

4.1.1. Proyecto de GitHub

The screenshot shows the GitHub interface for the repository 'aagui14 / moviescards'. The repository is public and has 6 branches and 0 tags. The main branch is 'master'. The repository has 2758554 commits and 38 commits in total. The repository is described as 'Calidad mayor del 80% V2'. The repository has 0 stars, 1 watching, and 0 forks. The repository is described as 'No description, website, or topics provided'. The repository has 0 releases and 0 packages published. The repository has 0 languages published. The repository has 0 languages published.

File	Commit	Time
github/workflows	Modificar yml stage app-name	last month
mvnw/wrapper	Commit inicial	2 months ago
src	Calidad mayor del 80% V2	last month
gitignore	Commit inicial	2 months ago
mvnw	Commit inicial	2 months ago
mvnw.cmd	Commit inicial	2 months ago
pom.xml	Commit inicial	2 months ago

4.1.2. Ramas con cada Issue

The screenshot shows a Kanban board for the 'moviescards' project. The board has three columns: 'Todo', 'In Progress', and 'Done'. The 'Todo' column has two items: 'moviescards #3: Añadir color de fondo a la página principal' and 'moviescards #4: Añadir color al título de la página principal'. The 'In Progress' column is empty. The 'Done' column is empty. The board has a search bar and a filter by keyword or by field. The board has a 'Discard' button and a 'Save' button.

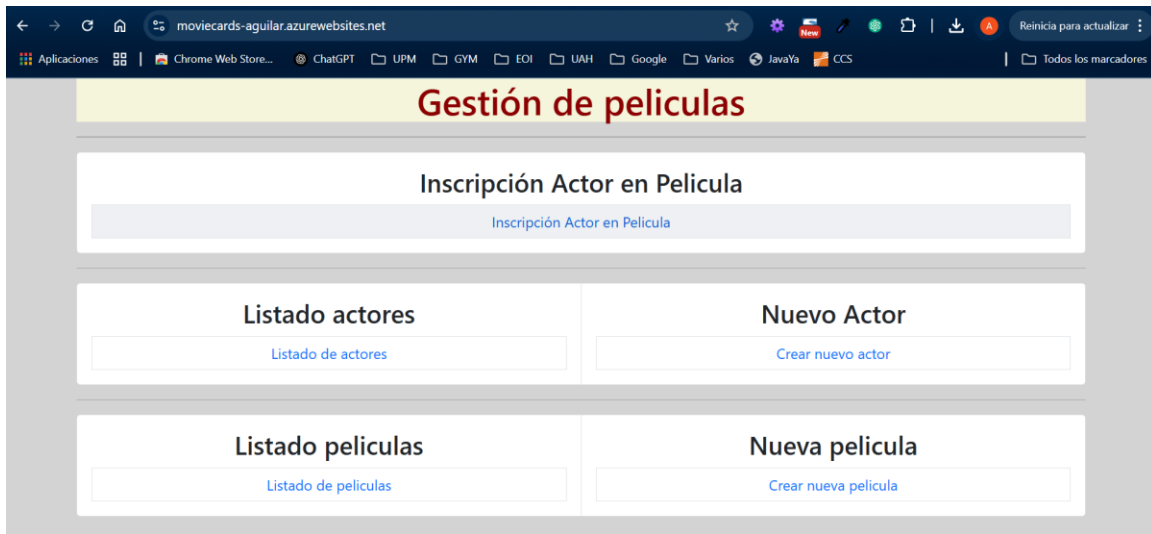
4.1.3. Creación de las ramas de git

```
C:\moviecards>git branch añadir-color-fondo

C:\moviecards>git branch añadir-color-titulo

C:\moviecards>git branch
añadir-color-fondo
añadir-color-titulo
* master
```

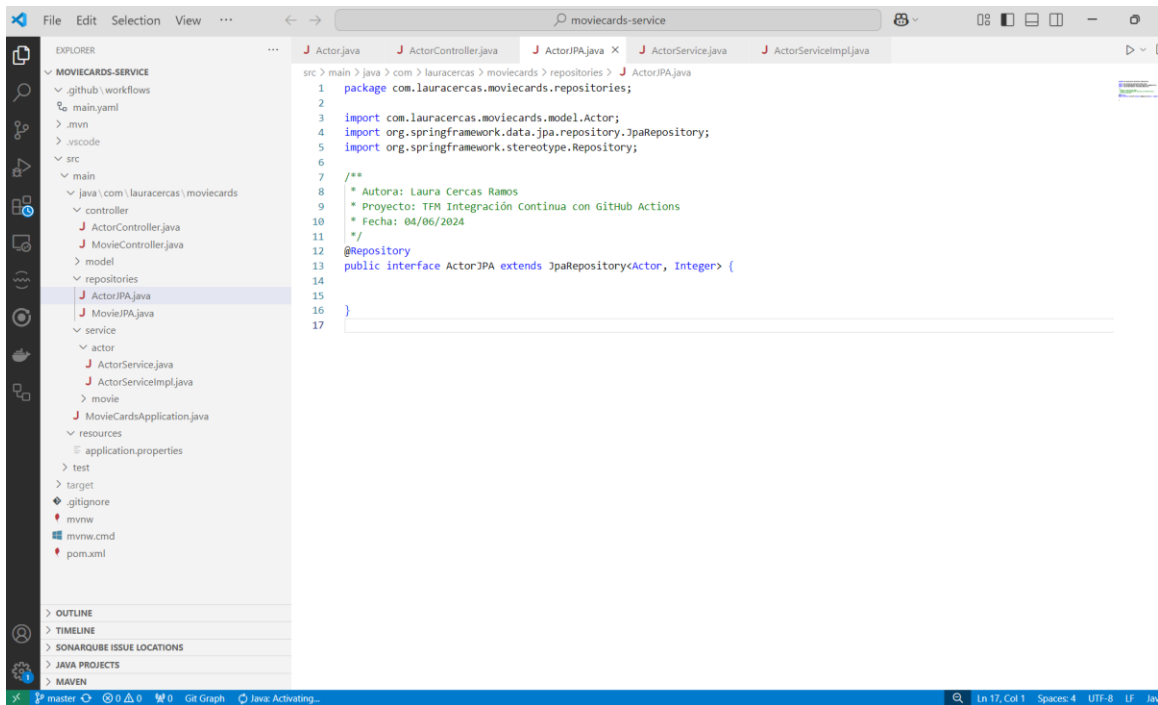
4.1.4. Despliegue en Azure de la aplicación



4.2. Nuevo repositorio moviescard-service

Se creó un nuevo repositorio para el servicio independiente que maneja la lógica de negocio y la persistencia.

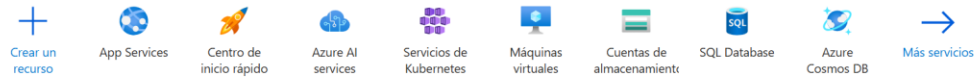
4.2.1. Código en visual studio



4.2.2. Despliegue en Azure App Services de la aplicación y del servicio

Se configuraron dos aplicaciones en Azure App Services: una para la aplicación principal y otra para el servicio.

Servicios de Azure



Recursos

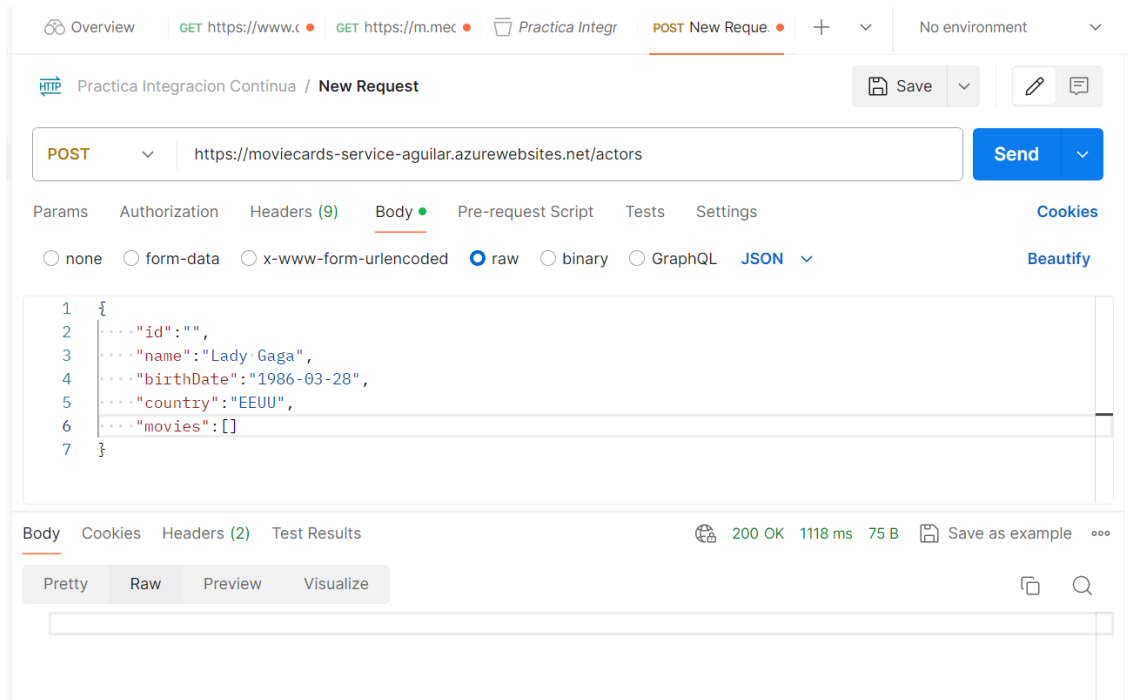
Reciente Favorito

Nombre	Tipo	Última consulta
moviecards-service-aguilari	App Service	hace unos segundos
moviecards-pre-aguilari	App Service	hace 34 minutos
moviecards-pre-aguilari_group	Grupo de recursos	hace 4 semanas
moviecards-service-aguilari_group	Grupo de recursos	hace 4 semanas
moviecards-aguilari	App Service	hace 4 semanas
moviecards-aguilari_group	Grupo de recursos	hace 2 meses

Ver todo

4.2.3. Pruebas con Postman

1. Añadir Actor



2. Consultar Actor

Practica Integracion Continua / GET-ACTORS

GET

https://moviecards-service-aguilar.azurewebsites.net/actors

Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results 200 OK 575 ms 218 B Save as example ...

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "name": "Lady Gaga",
5     "birthDate": "1986-03-28T00:00:00.000+00:00",
6     "country": "EEUU",
7     "movies": []
8   }
9 ]
```

3. Añadir Película

Practica Integracion Continua / POST-MOVIES

POST

https://moviecards-service-aguilar.azurewebsites.net/movies

Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   ... "id": "",
3   ... "title": "Joker 2",
4   ... "releaseYear": 2024,
5   ... "duration": 138,
6   ... "country": "EEUU",
7   ... "director": "Todd Phillips",
8   ... "genre": "Thriller",
9   ... "sinopsis": "Secuela de la película Joker de 2019",
10  ... "actors": []
11 }
```

Body Cookies Headers (2) Test Results 200 OK 550 ms 75 B Save as example ...

Pretty Raw Preview Visualize Text

```
1
```

4. Consultar Película

HTTP Practica Integracion Continua / GET-MOVIES

GET https://moviecards-service-aguilar.azurewebsites.net/movies

Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

Body Cookies Headers (3) Test Results 200 OK 122 ms 357 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 2,
4     "title": "Joker 2",
5     "releaseYear": 2024,
6     "duration": 138,
7     "country": "EEUU",
8     "director": "Todd Phillips",
9     "genre": "Thriller",
10    "sinopsis": null,
11    "actors": [
12      {
13        "id": 1,
14        "name": "Lady Gaga",
15        "birthDate": "1986-03-28T00:00:00.000+00:00",
16        "country": "EEUU"
17      }
18    ]
19  }
20 ]
```

4.3. Nueva versión aplicación

Realizar una nueva versión de la aplicación **moviecards**, modificando el código del repositorio **moviecards** de la práctica del tema 5, para que utilice el servicio creado en el apartado anterior.

Product Solutions Resources Open Source Enterprise Pricing Search or jump to... Sign in Sign

moviecards

View 1

Filter by keyword or by field

Todo 0
This item hasn't been started

In Progress 1
This is actively being worked on

- moviecards #18
5.6 Garantía de la calidad

Done 14
This has been completed

- moviecards #4
Añadir color al título de la página principal
- moviecards #10
Cambios en src/main
- moviecards #11
Cambios src test
- moviecards #8
Modificar el código de la aplicación en src/main
- moviecards #9
Modificar el código de las pruebas en src/test
- moviecards #21
Modificar codigo fecha fallecimiento

4.4. Nuevo campo DeadDate

Modificar el código del repositorio **moviecards-service** para añadir un nuevo atributo en la clase **Actor**, llamado **deadDate**, que contenga la fecha de fallecimiento del actor.

The image shows two screenshots of the Postman API client interface. The top screenshot shows a POST request to `https://moviecards-service-aguiar.azurewebsites.net/actors` with a JSON body containing actor details, including a new `deadDate` field. The bottom screenshot shows a GET request to the same endpoint, returning a JSON array of actors, also including the `deadDate` field.

POST Request:

```
1 {
2   ... "id": "",
3   ... "name": "Denholm Elliott",
4   ... "birthDate": "1922-05-31",
5   ... "deadDate": "1992-10-06",
6   ... "country": "Reino Unido",
7   ... "movies": []
8 }
```

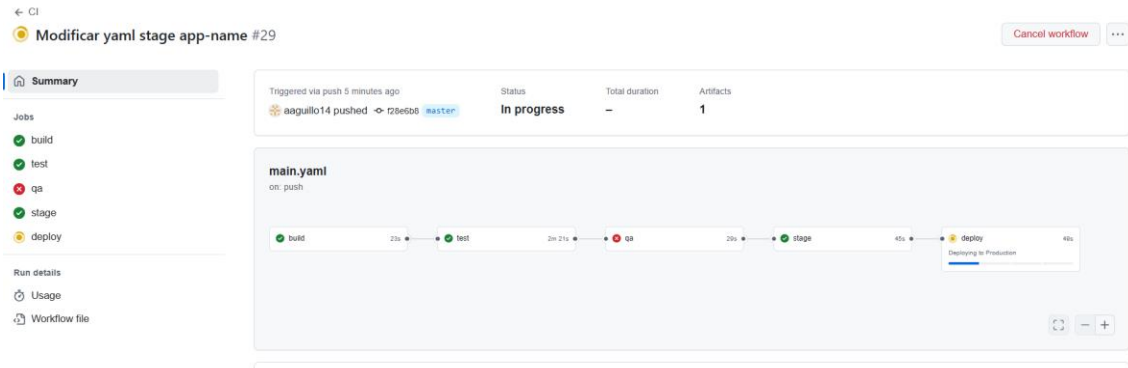
GET Response:

```
1 [
2   {
3     "id": 1,
4     "name": "Denholm Elliott",
5     "birthDate": "1922-05-31T00:00:00.000+00:00",
6     "deadDate": "1992-10-06T00:00:00.000+00:00",
7     "country": "Reino Unido",
8     "movies": []
9   }
10 ]
```

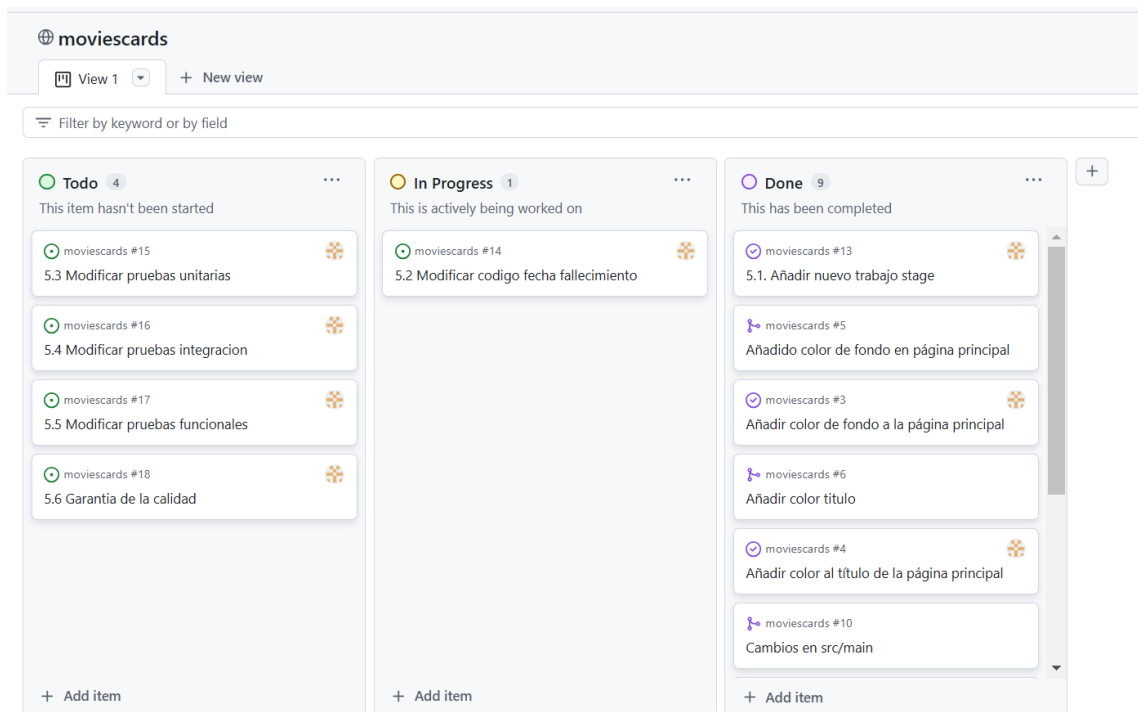
4.5. Nueva versión página web

Realizar una nueva versión de la aplicación **moviecards**, para que en la página web, para crear un nuevo actor se pida al usuario también la fecha de fallecimiento, y en la página de listado de actores aparezca una nueva columna con la fecha de fallecimiento.

- Añadir al workflow del proyecto un nuevo trabajo, entre los trabajos “qa” y “deploy”, llamado “stage” para desplegar la aplicación en un entorno de pre-producción.



- Modificar el código de la aplicación para manejar la fecha de fallecimiento de los actores.



```
C:\moviecards>git branch modificar-fecha-fallecimiento
```

```
C:\moviecards>git checkout modificar-fecha-fallecimiento  
Switched to branch 'modificar-fecha-fallecimiento'
```

Nuevo Actor

Nombre
Escriba el nombre del actor

Fecha nacimiento
dd/mm/aaaa

Fecha fallecimiento
dd/mm/aaaa

País
Escriba el país del actor

Guardar

[Volver a la Página de Inicio](#)

- Modificar las pruebas unitarias para tener en cuenta la fecha de fallecimiento de los actores.
- Modificar las pruebas de integración para tener en cuenta la fecha de fallecimiento de los actores.
- Modificar las pruebas funcionales (end to end) para tener en cuenta la fecha de fallecimiento de los actores.

← CI

✓ Cambios pruebas funcionales Dead Date #34

Re-run jobs

Summary

Jobs

- ✓ build
- ✓ test
- ✗ qa
- ✓ stage
- ✓ deploy

Run details

- Usage
- Workflow file

main.yml

on: push

build (1m) → test (2m 27s) → qa (3s) → stage (1m) → deploy (2m 28s)

Annotations

1 error

✗ qa

Process completed with exit code 1.

- Como garantía de calidad, hay que modificar el código fuente de la aplicación hasta conseguir tener menos de 5 problemas críticos, de tal forma que, si SonarQube detecta que tiene 5 o más, no debe poderse desplegar la aplicación a producción.

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Successfully updated condition.

Search for projects...

Quality Gates

Create

Sonar way

BUILT-IN

control calidad

DEFAULT

control calidad

Duplicated Lines (%)

is greater than

3.0%

Maintainability Rating

is worse than

A (Technical debt ratio is less than 5.0%)

Reliability Rating

is worse than

A (No bugs)

Security Hotspots Reviewed

is less than

100%

Security Rating

is worse than

A (No vulnerabilities)

Conditions on Overall Code

Metric

Operator

Value

Critical Issues

is greater than

5

You may click unlock to edit this quality gate. Adding extra conditions to a compliant quality gate can result in drawbacks. Are you reconsidering [Clean as You Code](#)? We strongly recommend this methodology to achieve a Clean Code status.

Unlock editing

aagullo14 / moviescards

Type to search

CodeIssues1Pull requests

ActionsProjects1WikiSecurityInsightsSettings

CI

Calidad mayor del 80% V2 #37

Re-run all jobsLatest #3

Summary

Jobs

buildtestqastagedeploy

Run detailsUsageWorkflow file

Re-run triggered last month

Status

Total duration

Artifacts

aagullo14 275855d master

Success

2m 11s

1

main.yaml

on: push

build

21s

test

2m 27s

qa

27s

stage

27s

deploy

34s

[https://moviescards-aagullo14.com/review/...](#)