

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



TEMA:
CORTO 2 - INFORME DE PRUEBAS

PRESENTADO POR:
GUILLERMO ENRIQUE MARTINEZ FIGUEROA MF17006
EDWIN ARNOLDO ARGUETA LARA AL15022
ELMER EDUARDO INTERIANO ESTRADA IE18001
VLADIMIR ENRIQUE MARTINEZ FLORES MF18030
KENIA STEPHANNIE TEPAS MAZARIEGO TM17013

CATEDRÁTICO:
ING. XENIA IVETTE PEÑATE GODOY

MATERIA:
TECNICAS DE PROGRAMACION DE INTERNET

CICLO I

ABRIL DE 2021

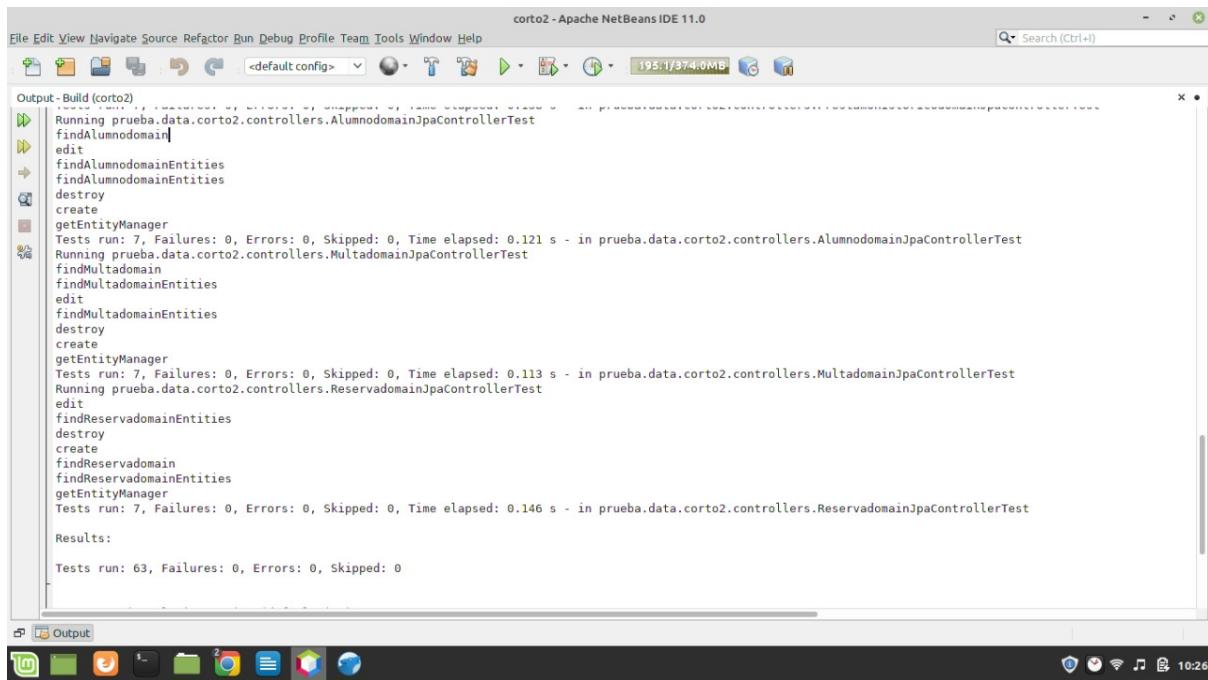
INFORME DE PRUEBAS

Se le aplicará pruebas unitarias con Mock a cada clase Controller de la base de datos que creamos anteriormente.

Los Mocks, “son objetos preprogramados con expectativas que conforman la especificación de lo que se espera que reciban las llamadas”, es decir, son objetos que se usan para probar que se realizan correctamente llamadas a otros métodos, por ejemplo, a una web API, por lo que se utilizan para verificar el comportamiento de los objetos.

Un mock es un objeto el cual intenta suplantar una dependencia de una clase. La idea es que un objeto mock nos permite crear un objeto del cual tenemos total control, de este modo, podemos definir cuál será el valor de retorno de sus funciones. Desde el punto de vista de las pruebas unitarias, los mocks son importantes porque nos ayudan a enfocar nuestras pruebas en una clase específica sin tener la molestia de lidiar con dependencias. Esto es especialmente importante cuando trabajamos con dependencias externas de nuestro software, como una base de datos, archivos de texto del computador, e incluso web services. En conclusión, Los mocks nos ayudan a poder controlar el comportamiento de las dependencias de nuestras clases. Esto es útil cuando hacemos pruebas unitarias, pues nos permite fijar el comportamiento de las dependencias de la clase que queremos probar, para así poder concentrarnos en la clase que queremos probar.

Aplicando Mock a nuestra aplicación tenemos como resultado lo siguiente:



The screenshot shows the Apache NetBeans IDE 11.0 interface with the title bar "corto2 - Apache NetBeans IDE 11.0". The main window has a toolbar at the top with various icons. Below the toolbar is a menu bar: File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. A search bar "Search (Ctrl+F)" is located at the top right. The central area is the "Output" window titled "Output: Build (corto2)". It displays the following test results:

```
Running prueba.data.corto2.controllers.AlumnodomainJpaControllerTest
findAlumnodomain
edit
findAlumnodomainEntities
findAlumnodomainEntities
destroy
create
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.121 s - in prueba.data.corto2.controllers.AlumnodomainJpaControllerTest
Running prueba.data.corto2.controllers.MultadomainJpaControllerTest
findMultadomain
findMultadomainEntities
edit
findMultadomainEntities
destroy
create
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.113 s - in prueba.data.corto2.controllers.MultadomainJpaControllerTest
Running prueba.data.corto2.controllers.ReservadomainJpaControllerTest
edit
findReservadomainEntities
destroy
create
findReservadomain
findReservadomainEntities
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.146 s - in prueba.data.corto2.controllers.ReservadomainJpaControllerTest
Results:

Tests run: 63, Failures: 0, Errors: 0, Skipped: 0
```

En esta imagen se le aplicó Test a la clase AlumnodomainJpaController y sus respectivos métodos los cuales son:

1. findAlumnodomain
2. edit
3. findAlumnodomainEntities
4. findAlumnodomainEntities
5. destroy

6. create
7. getEntityManager

También se le aplicó Test a la clase MultadomainJpaController y sus respectivos métodos, los cuales consta de:

1. findMultadomain
2. findMultadomainEntities
3. edit
4. findMultadomainEntities
5. destroy
6. create
7. getEntityManager

Y por ultimo se le aplico Test a la clase ReservadomainJpaController y sus respectivos métodos, los cuales son:

1. edit
2. destroy
3. create
4. findReservadomainEntities
5. findReservadomain
6. findReservadomainEntities
7. getEntityManager

Cabe mencionar que como se observa en la imagen todos los métodos pasaron las respectivas pruebas.

The screenshot shows the Apache NetBeans IDE 11.0 interface with the title bar "corto2 - Apache NetBeans IDE 11.0". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for file operations like Open, Save, and Print. The Output window displays the following test results:

```

Output - Build (corto2)
...
findBibliotecariodomainEntities
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.126 s - in prueba.data.corto2.controllers.BibliotecariodomainJpaControllerTest
Running prueba.data.corto2.controllers.ProfesordomainJpaControllerTest
edit
findProfesordomain
findProfesordomainEntities
findProfesordomainEntities
destroy
create
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.143 s - in prueba.data.corto2.controllers.ProfesordomainJpaControllerTest
Running prueba.data.corto2.controllers.LibrodomainJpaControllerTest
findlibrodomainEntities
edit
findlibrodomain
findlibrodomainEntities
destroy
create
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.229 s - in prueba.data.corto2.controllers.LibrodomainJpaControllerTest
Running prueba.data.corto2.controllers.PrestamohistoricodomainJpaControllerTest
edit
findPrestamohistoricodomain
findPrestamohistoricodomainEntities
findPrestamohistoricodomainEntities
destroy
create
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.138 s - in prueba.data.corto2.controllers.PrestamohistoricodomainJpaControllerTest
Running prueba.data.corto2.controllers.AlumnodomainJpaControllerTest
findAlumnodomain
edit

```

En esta imagen se le aplicó Test a la clase ProfesordomainJpaController y sus respectivos métodos los cuales son:

1. findProfesordomain
2. edit
3. findProfesordomainEntities
4. findProfesordomainEntities
5. destroy
6. create
7. getEntityManager

También se le aplicó Test a la clase LibrodomainJpaController y sus respectivos métodos los cuales son:

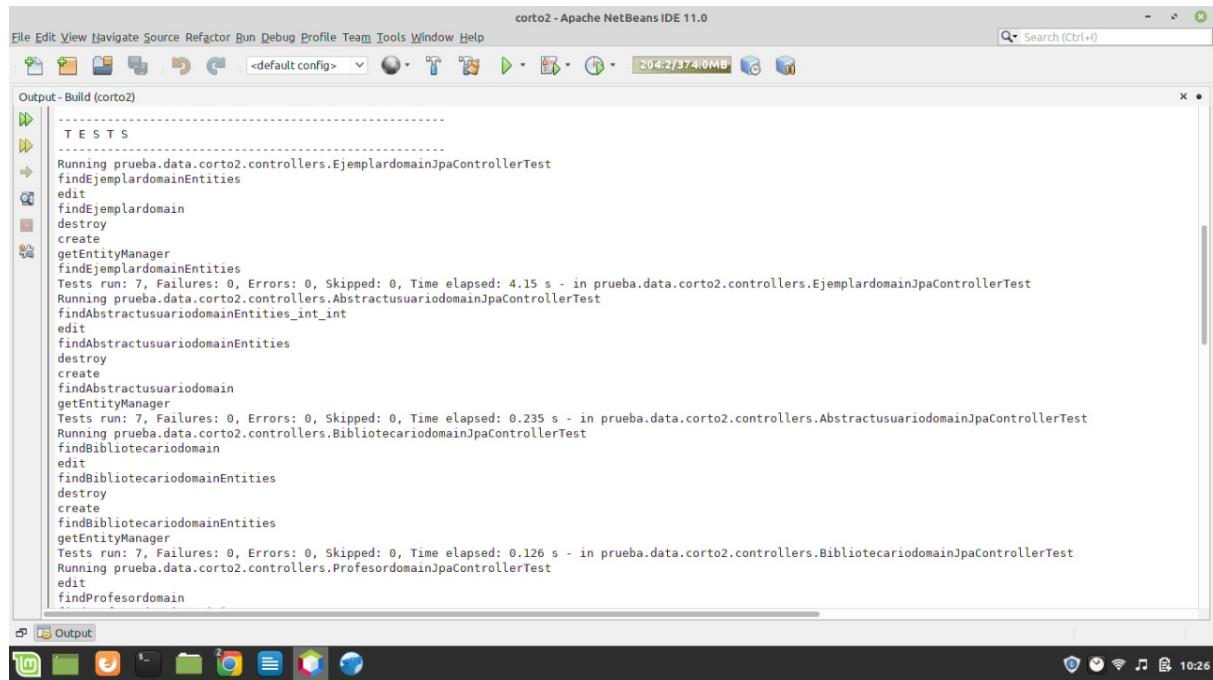
1. findLibrodomain
2. edit
3. findLibrodomainEntities
4. findLibrodomainEntities
5. destroy
6. create
7. getEntityManager

Tambien se le aplicó Test a la clase PrestamohistoricodomainJpaController la cual consta de:

1. findPrestamohistoricodomain
2. edit
3. findPrestamohistoricodomainEntities
4. findPrestamohistoricodomainEntities
5. destroy

6. create
7. getEntityManager

Cabe mencionar que como se observa en la imagen todos los métodos pasaron las respectivas pruebas.



```

corto2 - Apache NetBeans IDE 11.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 204.2/374.0MB Search (Ctrl+F)
Output - Build (corto2)
T E S T S
-----
Running prueba.data.corto2.controllers.EjemplardomainJpaControllerTest
findEjemplardomainEntities
edit
findEjemplardomain
destroy
create
getEntityManager
findEjemplardomainEntities
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.15 s - in prueba.data.corto2.controllers.EjemplardomainJpaControllerTest
Running prueba.data.corto2.controllers.AbstractusuariodomainJpaControllerTest
findAbstractusuariodomainEntities_int_int
edit
findAbstractusuariodomainEntities
destroy
create
findAbstractusuariodomain
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.235 s - in prueba.data.corto2.controllers.AbstractusuariodomainJpaControllerTest
Running prueba.data.corto2.controllers.BibliotecariodomainJpaControllerTest
findBibliotecariodomain
edit
findBibliotecariodomainEntities
destroy
create
findBibliotecariodomainEntities
getEntityManager
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.126 s - in prueba.data.corto2.controllers.BibliotecariodomainJpaControllerTest
Running prueba.data.corto2.controllers.ProfesordomainJpaControllerTest
edit
findProfesordomain

```

En esta imagen se le aplicó Test a la clase EjemplardomainJpaController y sus respectivos métodos los cuales son:

1. findEjemplardomain
2. edit
3. findEjemplardomainEntities
4. findEjemplardomainEntities
5. destroy
6. create
7. getEntityManager

También se le aplicó Test a la clase AbstractusuariodomainJpaController la cual consta de:

1. findAbstractusuariodomain
2. edit
3. findAbstractusuariodomainEntities
4. findAbstractusuariodomainEntities
5. destroy
6. create
7. getEntityManager

Por último, también se le aplicó Test a la clase BibliotecariodomainJpaController teniendo como métodos los siguientes:

1. findBibliotecariodomain
2. edit
3. findBibliotecariodomainEntities
4. findBibliotecariodomainEntities
5. destroy
6. create
7. getEntityManager

Cabe mencionar que todos los métodos pasaron sus respectivas pruebas.