

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

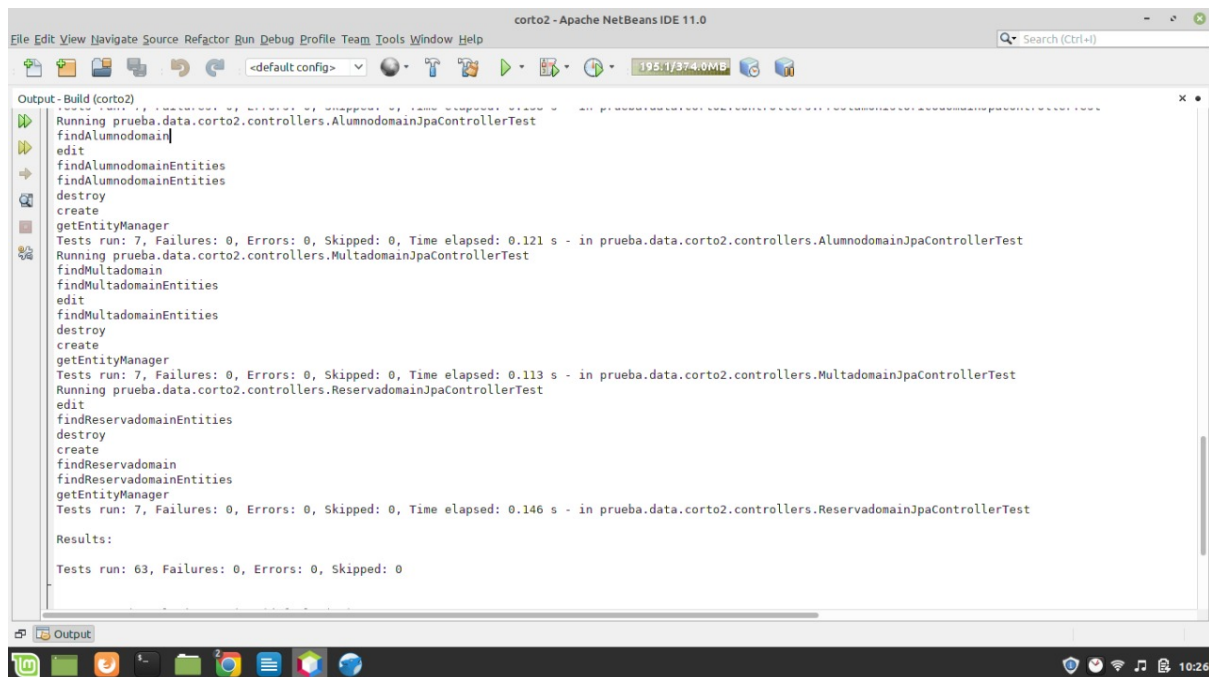
## INFORME DE PRUEBAS

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:



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

1. findAlumnomain
2. edit
3. findAlumnomainEntities
4. findAlumnomainEntities
5. destroy

6. create
7. getEntityManager

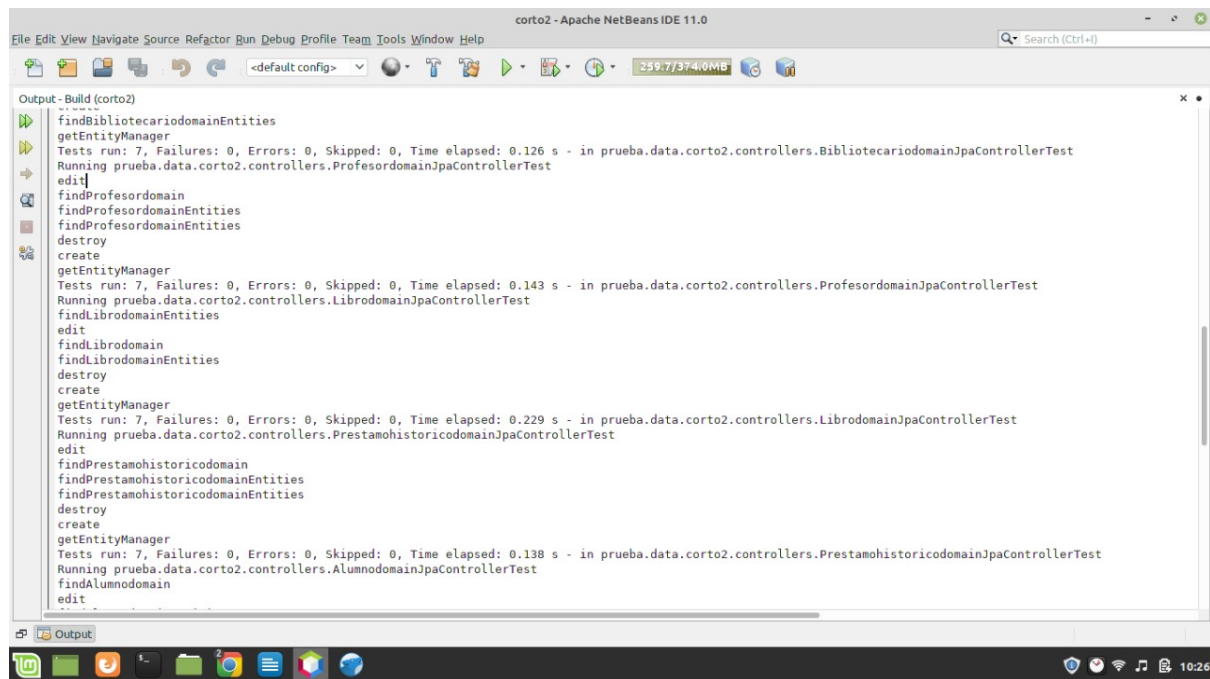
También se le aplicó Test a la clase MultadomainJpaController y sus respectivos métodos, los cuales consta de:

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.



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:

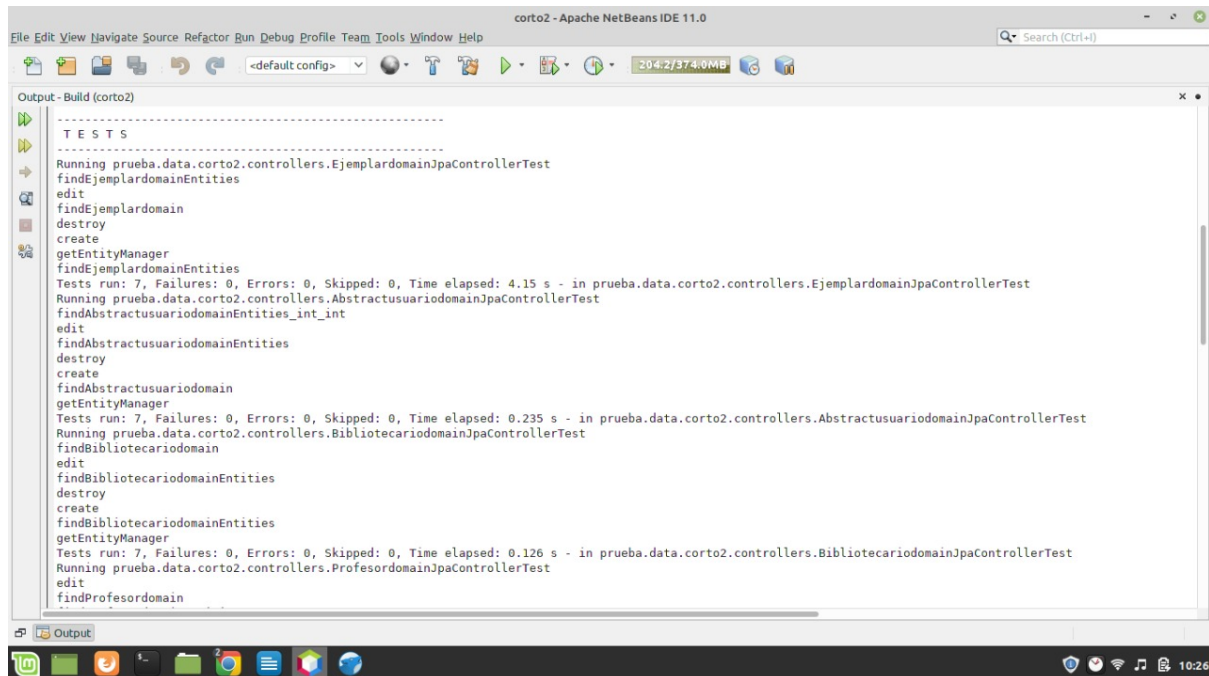
1. findLibrodomain
2. edit
3. findLibrodomainEntities
4. findLibrodomainEntities
5. destroy
6. create
7. getEntityManager

También 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.



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.