

Pruebas Unitarias

Se hace prueba de la lógica de negocio implementada en los servicios y repositorios de Actividades y Usuarios, las pruebas se realizan usando mockito

TDD Registrar Estudiante

```
1  @DataJpaTest
2  public class usuarioServicioTest {
3
4      @Mock
5      private IUsuario repositorioUsuario;
6
7      @InjectMocks
8      private usuarioServicio usuarioservicio;
9
10
11     private Usuario usuario;
12
13     @BeforeEach
14     void setUp(){
15         MockitoAnnotations.initMocks(this);
16
17         usuario = new Usuario();
18         usuario.setIdentificacion(10001);
19         usuario.setNombre("Usuario 1");
20         usuario.setApellido("Apellido 1");
21         usuario.setCorreo("dalvarezr4@ucentral.edu.co");
22         usuario.setContrasena(null);
23
24
25     }
```

```
1  @Test
2  void testGuardarUsuario() {
3      when(repositorioUsuario.save(any(Usuario.class))).thenReturn(usuario);
4      assertNotNull(repositorioUsuario.save(new Usuario()));
5      try{
6          repositorioUsuario.save(usuario);
7      }catch (Exception ex){
8          System.out.println("Error: " + ex.getMessage());
9      }
10 }
```

TDD Buscar Estudiante por ID

```
1  @Test
2  void testBuscarUsuario() {
3      when(repositorioUsuario.findById(10001)).thenReturn(java.util.Optional.of(usuario));
4      assertNotNull(repositorioUsuario.findById(10001));
5
6      try{
7          repositorioUsuario.findById(10001);
8      }catch (Exception ex){
9          System.out.println("Error: " + ex.getMessage());
10     }
11 }
```

TDD Listar todos los estudiantes

```
1  @Test
2  void testListarTodosUsuarios() {
3      assertNotNull(repositorioUsuario.findAll());
4
5      try{
6          repositorioUsuario.findAll();
7      }catch (Exception ex){
8          System.out.println("Error: " + ex.getMessage());
9      }
10 }
```

TDD Eliminar estudiante

```
1  @Test
2  void testEliminarUsuario() {
3      repositorioUsuario.deleteById(10001);
4      try{
5          repositorioUsuario.deleteById(10001);
6      }
7      catch (Exception ex){
8          System.out.println("Error: " + ex.getMessage());
9      }
10 }
```

TDD Listar todas las actividades

```
1  @DataJpaTest
2  public class actividadServicioTest {
3
4
5      @Mock
6      private IActividad repositorioActividad;;
7
8      @InjectMocks
9      private actividadServicio actividadservicio;
10
11     private Actividad actividad;
12
13
14
15     @BeforeEach
16     void setUp(){
17         MockitoAnnotations.initMocks(this);
18
19         actividad = new Actividad();
20         actividad.setCodigo(100);
21         actividad.setNombre("Actividad 1");
22         actividad.setPonderado(100);
23         actividad.setMateria(null);
24         actividad.setGrupo(null);
25         actividad.setCorte(null);
26
27     }
```

```
1  @Test
2  void testListarTodActividades() {
3
4      when(repositorioActividad.findAll()).thenReturn(Arrays.asList(actividad));
5      assertNotNull(repositorioActividad.findAll());
6
7      try{
8          repositorioActividad.findAll();
9      }catch (Exception ex){
10         System.out.println("Error: " + ex.getMessage());
11     }
12
13 }
```

TDD Guardar Actividad

```
1  @Test
2  void testGuardarActividad() {
3      when(repositorioActividad.save(any(Actividad.class))).thenReturn(actividad);
4      assertNotNull(repositorioActividad.save(new Actividad()));
5
6      try{
7          repositorioActividad.save(actividad);
8      }catch (Exception ex){
9          System.out.println("Error: " + ex.getMessage());
10     }
11 }
```

TDD Buscar Actividad

```
1  @Test
2  void testBuscarActividad() {
3      when(repositorioActividad.findById(100)).thenReturn(java.util.Optional.of(actividad));
4      assertNotNull(repositorioActividad.findById(100));
5
6      try{
7          repositorioActividad.findById(100);
8      }catch (Exception ex){
9          System.out.println("Error: " + ex.getMessage());
10     }
11 }
```

TDD Eliminar Actividad

```
1  @Test
2  void testEliminarActividad() {
3      repositorioActividad.deleteById(100);
4
5      try{
6          repositorioActividad.deleteById(100);
7      }catch (Exception ex){
8          System.out.println("Error: " + ex.getMessage());
9      }
10 }
```

Resultados de cada prueba realizada

TESTING

Filter (e.g. text, !exclude, @tag)

10/10 tests passed (100%)

- > ✓ Apache NetBeans
- ✓ Java Test 53ms
 - ✓ appnotas 53ms
 - ✓ com.ucentral.appnotas 2.0ms
 - ✓ actividadesTest 1.0ms
 - test() 1.0ms
 - ✓ AppnotasApplicationTests 1.0ms
 - contextLoads() 1.0ms
 - ✓ com.ucentral.appnotas.servicio 51ms
 - ✓ actividadServicioTest 15ms
 - testListarTodActividades() 4.0ms
 - testGuardarActividad() 4.0ms
 - testBuscarActividad() 3.0ms
 - testEliminarActividad() 4.0ms
 - ✓ usuarioServicioTest 36ms
 - testGuardarUsuario() 8.0ms
 - testBuscarUsuario() 5.0ms
 - testListarTodosUsuarios() 4.0ms
 - testEliminarUsuario() 19ms