# Unit Testing

[TestClass]

public class StackTests

{

[TestMethod]

public void IsEmpty\_NuevoStack\_RetornaVerdadero()

{

Stack stack=new Stack();

bool estaVacio = stack.IsEmpty;

Assert.IsTrue(estaVacio);

}

[TestMethod]

public void IsEmpty\_IngresoUnElemento\_RetornaFalso()

{

Stack stack=new Stack();

stack.Push(1);

bool estaVacio = stack.IsEmpty;

Assert.IsFalse(estaVacio);

}

[TestMethod]

public void IsEmpty\_IngresoYSacoUnElemento\_RetornaFalso()

{

Stack stack = new Stack();

stack.Push(1);

stack.Pop();

bool estaVacio = stack.IsEmpty;

Assert.IsTrue(estaVacio);

}

[TestMethod]

public void IsEmpty\_IngresoDosObtengoUno\_RetornaFalso()

{

Stack stack = new Stack();

stack.Push(1);

stack.Push(2);

stack.Pop();

bool isEmpty = stack.IsEmpty;

Assert.IsFalse(isEmpty);

}

[TestMethod]

public void Pop\_IngresoYObtengoUnElemento\_ElElementoEsElMismo()

{

Stack stack = new Stack();

stack.Push(1);

int element=stack.Pop();

Assert.AreEqual(1, element);

}

[TestMethod]

public void Pop\_IngresoDosElementosYObtengoUno\_ElElementoObtenidoEsIgualAlPrimero()

{

Stack stack=new Stack();

stack.Push(1);

stack.Push(2);

int element = stack.Pop();

Assert.AreEqual(2, element);

}

[TestMethod]

public void Pop\_IngresoDosElementosYObtengoDos\_ElSegundoObtenidoEsIgualAlPrimerIngresado()

{

Stack stack = new Stack();

stack.Push(1);

stack.Push(2);

stack.Pop();

int element = stack.Pop();

Assert.AreEqual(1, element);

}

# Entity Framework

private static void UpdateDepartmentToEmployee()

{

var context = new Entities();

var salesman = context.Employees

.Single(e => e.Id == 1);

var department = context.Departments.Single(e => e.Id == 1);

salesman.Department = department;

context.SaveChanges();

salesman.DepartmentId = 2;

context.SaveChanges();

}

private static void DeleteEmployee()

{

var context = new Entities();

var salesman = context.Employees

.Single(e => e.Name == "Luis Arroyo");

context.Employees.DeleteObject(salesman);

context.SaveChanges();

}

private static void UpdateEmployee()

{

var context = new Entities();

var salesman = context.Employees

.Single(e => e.Name == "Luis Arroyo");

salesman.Age = 29;

context.SaveChanges();

}

private static void AddNewEmployee()

{

var context = new Entities();

var salesman = new Salesman {

Name = "Luis Arroyo",

Age = 33,

Commission = 10.5m };

context.Employees.AddObject(salesman);

context.SaveChanges();

}

#region L2Entities

public static void EmployeeWithDepartmentLazy()

{

var context = new Entities();

var employee = (from e in context.Employees

where e.Id == 4

select e).FirstOrDefault();

var department = employee.Department;

}

public static void EmployeeWithDepartmentEager()

{

var context = new Entities();

var query = from e in context.Employees.Include("Department")

where e.Id == 4

select e;

var employee = query.FirstOrDefault();

employee = context.Employees.Include("Department")

.Where(e => e.Id == 4).SingleOrDefault();

}

public static void OnlyEmployeesWithDepartments()

{

var context = new Entities();

var query = from e in context.Employees

where e.Department != null

select e;

var employees = query.ToList();

employees = context.Employees.Where(e => e.Department != null).ToList();

}

public static void OneDepartment()

{

var context = new Entities();

var query = from d in context.Departments

where d.Id == 1

select d;

var department = query.FirstOrDefault();

department = context.Departments

.Where(d => d.Id == 1).FirstOrDefault();

}

public static void AllDepartments()

{

var context = new Entities();

var query = from d in context.Departments

select d;

var departments = query.ToList();

departments = context.Departments.ToList();

}

#endregion