UNIVERSIDAD PRIVADA DE TACNA

INGENIERIA DE SISTEMAS

TEMA:

Informe Sesión de Laboratorio Nro 02

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Jose Cuadros Quiroga

Integrantes:

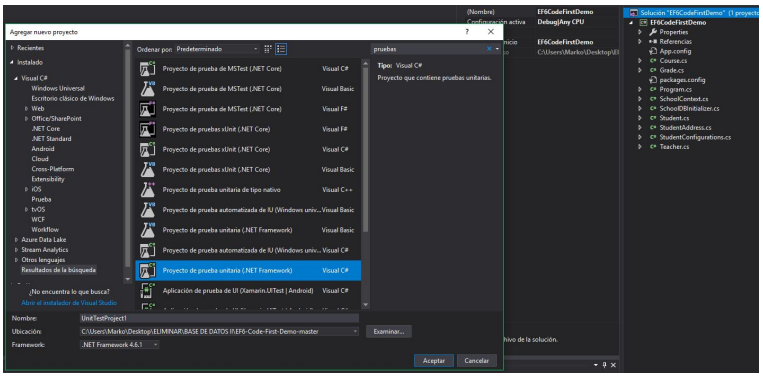
Marko Antonio RIVAS RIOS	(2016055461)
Jorge Luis MAMANI MAQUERA	(2016055236)
Andree Ludwed VELASCO SUCAPUCA	(2016055286)
Yofer Nain CATARI CABRERA	(2017059289)
Adnner Sleyder ESPERILLA RUIZ	(2015050543)
Jesus ESCALANTE ALANOCA	(2015050641)

Índice

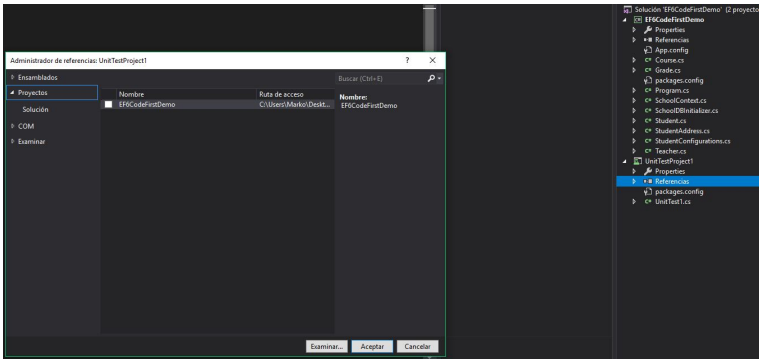
1. Ejercicio 1: Consultando Datos

Paso 1.Ingresar a Visual Studio y abrir o generar el proyecto que genera el contexto EF6-Code-First-Demo.

Paso 2. Agregar a la solución un Proyecto de pruebas unitarias .Net Framework.



Paso 3. En el proyecto de pruebas referenciar al proyecto principal al proyecto de pruebas



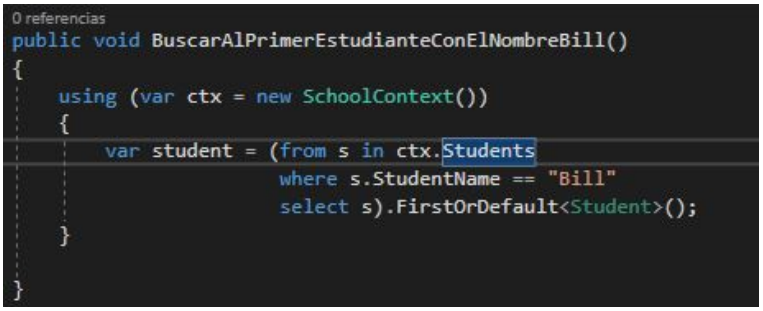
Paso 4. En la clase Unitest generada crear el método: ObtenerAlEstudianteConIDUno, puede tomar como referencia el siguiente código

```
namespace UnitTestProject1
{
    [TestClass]
    0 referencias
    public class UnitTest1
    {
        [TestMethod]
        0 referencias
        public void ObtenerAlEstudianteConIDUno()
        {
            var ctx = new SchoolContext();
            var student = ctx.Students.Find(1);
        }
    }
}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
SELECT
[Extent1].[StudentID] AS [StudentID],
[Extent1].[StudentName] AS [StudentName],
[Extent1].[GradeId] AS [GradeId]
FROM [dbo].[Student] AS [Extent1]
WHERE [Extent1].[StudentId] = @p0',N'@p0 int',@p0=1
go
```

Paso 5. Adicionar otro método: BuscarAlPrimerEstudianteConElNombreBill, puede tomar como referencia el siguiente código.

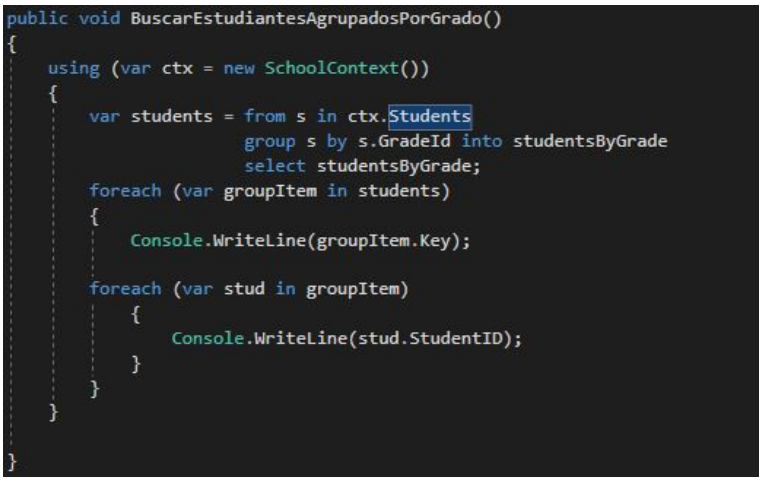


```
0 referencias
public void BuscarAlPrimerEstudianteConElNombreBill()
{
    using (var ctx = new SchoolContext())
    {
        var student = (from s in ctx.Students
                       where s.StudentName == "Bill"
                       select s).FirstOrDefault<Student>();
    }
}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
SELECT TOP (1)
[Extent1].[StudentID] AS [StudentID],
[Extent1].[StudentName] AS [StudentName],
[Extent1].[GradeId] AS [GradeId]
FROM [dbo].[Student] AS [Extent1]
WHERE 'Bill' = [Extent1].[StudentName]
```

Paso 6. Agregar otro método de prueba denominado: BuscarEstudiantesAgrupadosPorGrado, tomar como referencia el siguiente código.



```
public void BuscarEstudiantesAgrupadosPorGrado()
{
    using (var ctx = new SchoolContext())
    {
        var students = from s in ctx.Students
                       group s by s.GradeId into studentsByGrade
                       select studentsByGrade;
        foreach (var groupItem in students)
        {
            Console.WriteLine(groupItem.Key);

        foreach (var stud in groupItem)
            {
                Console.WriteLine(stud.StudentID);
            }
        }
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

Paso 7. Agregar otro método de prueba denominado: ObetenerListadoDeEstudiantesOrdenadosPorNombre, tomar como referencia el siguiente código.

```
public void ObtenerListadoDeEstudiantesOrdenadosPorNombre()
{
    using (var ctx = new SchoolContext())
    {
        var students = from s in ctx.Students
                       orderby s.StudentName ascending
                       select s;
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
SELECT
[Extent1].[StudentID] AS [StudentID],
[Extent1].[StudentName] AS [StudentName],
[Extent1].[GradeId] AS [GradeId]
FROM [dbo].[Student] AS [Extent1]
ORDER BY [Extent1].[StudentName] ASC
go
```

Paso 8. Finalmente crear el método de prueba: BuscarTodosLostudiantesConElEstandarUno, tomar como referencia el siguiente código.

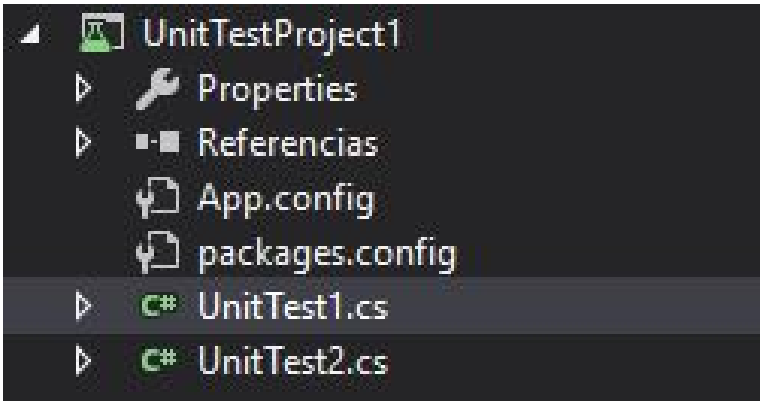
```
public void BuscarTodosLosEstudiantesConElGradoUno()
{
    using (var ctx = new SchoolContext())
    {
        var anonymousObjResult = from s in ctx.Students
                                 where s.GradeId == 1
                                 select new
                                 {
                                     Id = s.StudentID,
                                     Name = s.StudentName
                                 };
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
SELECT
[s].[StudentID] AS [Id], [s].[StudentName] AS [Name]
FROM [Student] AS [s]
WHERE [s].[ GradeId] = 1
go
```

2. Ejercicio 2: Guardando Datos

Paso 1. Agregar al proyecto de pruebas una clase de pruebas TestUnit02



Paso 2. Agregar el método InsertarEstudianteSatisfactoriamente, tomar como referencia el siguiente código

```
public void InsertarEstudianteSatisfactoriamente()
{
    using (var context = new SchoolContext())
    {
        var std = new Student()
        {
            StudentName = "Bill",

        };
        context.Students.Add(std);
        context.SaveChanges();
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
exec sp_executesql N'INSERT [dbo].[Students]([StudentName])
VALUES (@0)
SELECT [StudentId]
FROM [dbo].[Students]
WHERE @@ROWCOUNT > 0 AND [StudentId] = scope_identity()',N
''@0 nvarchar(max) ,@1 nvarchar(max) ',@0=N'Bill'
go
```

Paso 3. Agregar el método ActualizarAlPrimerEstudianteSatisfactoriamente, tomar como referencia el siguiente código

```
public void ActializarAlPrimerEstudianteSatisfactorimente()
{
    using (var context = new SchoolContext())
    {
        var std = context.Students.First<Student>();
        std.StudentName = "Steve";
        context.SaveChanges();
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
exec sp_executesql N'UPDATE [dbo].[Students]
SET [StudentName] = @0
WHERE ([StudentId] = @1)',
N'@0 nvarchar(max) ,@1 int',@0=N'Steve',@1=2
Go
```

Paso 4. Agregar el método EliminarElPrimerEstudianteSatisfactoriamente, tomar como referencia el siguiente código

```
public void EliminarElPrimerEstudianteSatisfactoriamente()
{
    using (var context = new SchoolContext())
    {
        var std = context.Students.First<Student>();
        context.Students.Remove(std);
        context.SaveChanges();
    }

}
```

La sentencia SQL generada del lado del gestor de base de datos es:

```
exec sp_executesql N'DELETE [dbo].[Students]
WHERE ([StudentId] = @0)',N'@0 int',@0=1
Go
```

Paso 5. Agregar el método: AgregarTresEstudiantesSatisfactoriamente, tomar como referencia el siguiente código:

```
public void AgregarTresEstudiantesSatisfactoriamente()
{
    IList<Student> newStudents = new List<Student>() {
    new Student() { StudentName = "Steve" },
    new Student() { StudentName = "Bill" },
    new Student() { StudentName = "James" }
};
    using (var context = new SchoolContext())
    {
        context.Students.AddRange(newStudents);
        context.SaveChanges();
    }

}
```

Paso 5. Finalmente agregar el método de pruebas: EliminarTresEstudiantesSatisfactoriamente, tomar como referencia el siguiente código:

```
public void EliminarTresEstudiantesSatisfactoriamente()
{
    IList<Student> studentsToRemove = new List<Student>() {
    new Student() { StudentID = 1, StudentName = "Steve" },
    new Student() { StudentID = 2, StudentName = "Bill" },
    new Student() { StudentID = 3, StudentName = "James" }

    };
    using (var context = new SchoolContext())
    {
    context.Students.RemoveRange(studentsToRemove);
    context.SaveChanges();
    }

}
```