

UNIVERSIDAD PRIVADA-DE-TACNA



INGENIERIA DE SISTEMAS

TITULO:

**TRABAJO FINAL DE UNIDAD**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Cuadros Quiroga

Integrantes:

Arlyn Cotrado Coaquira	(2016054466)
Yaneth Virginia Aquino Huallpa	(2017059286)
Sharon Sosa Bedoya	(2016054460)
Marlon Villegas Arando	(2015053890)

# Índice

<b>1. PROBLEMA</b>	<b>1</b>
1.1. Objetivos: . . . . .	1
1.2. Equipos, materiales, programas y recursos utilizados: . . . . .	1
<b>2. MARCO TEORICO</b>	<b>2</b>
2.1. Entity Framework . . . . .	2
2.2. Pruebas Unitarias . . . . .	3
<b>3. DESARROLLO</b>	<b>4</b>
3.1. Analisis . . . . .	4
3.2. Diseño . . . . .	4
3.3. Pruebas . . . . .	4
<b>4. REFERENCIAS</b>	<b>5</b>

# **1. PROBLEMA**

## **1.1. Objetivos:**

- Tener conocimiento acerca de Pruebas Unitarias en C Sharp

## **1.2. Equipos, materiales, programas y recursos utilizados:**

- Computadora con sistema operativo Windows 10.
- Microsoft Visual Studio 2017
- Microsoft SQL Server 2016
- Net Framework 4.6.1
- Base de datos SchoolDB

## 2. MARCO TEORICO

### 2.1. Entity Framework

- Entity Framework (EF) es un mapeador relacional de objetos (O/RM) probado y probado para .NET con muchos años de desarrollo y estabilización de características.  
Es la tecnología de acceso a datos recomendada por Microsoft para nuevas aplicaciones.
- Como O/RM, EF reduce la discrepancia de impedancia entre los mundos relacionales y orientados a objetos, permitiendo a los desarrolladores escribir aplicaciones que interactúan con datos almacenados en bases de datos relacionales utilizando objetos .NET de tipo fuerte que representan el dominio de la aplicación y eliminando la necesidad para una gran parte del código de "plumbing" de acceso a datos que normalmente necesitan escribir.
- EF implementa muchas características populares de O/RM:
  - Mapeo de clases de entidad POCO que no dependen de ningún tipo de EF
  - Seguimiento automático de cambios
  - Resolución de identidad y Unidad de Trabajo.
  - Carga ansiosa, perezosa y explícita.
  - Traducción de consultas fuertemente tipadas utilizando LINQ
  - Capacidades de mapeo enriquecidas, incluyendo soporte para:
    - Relaciones uno a uno, uno a muchos y muchos a muchos
    - Herencia (tabla por jerarquía, tabla por tipo y tabla por clase concreta)
    - Tipos complejos
    - Procedimientos almacenados
  - Un diseñador visual para crear modelos de entidad.
  - Una experiencia de "Código Primero" para crear modelos de entidad al escribir código.
  - Los modelos pueden generarse a partir de bases de datos existentes y luego editarse manualmente, o pueden crearse desde cero y luego usarse para generar nuevas bases de datos.
  - Integración con modelos de aplicaciones de .NET Framework, incluido ASP.NET, y mediante enlace de datos, con WPF y WinForms.
  - Conectividad de base de datos basada en ADO.NET y numerosos proveedores disponibles para conectarse a SQL Server, Oracle, MySQL, SQLite, PostgreSQL, DB2, etc.

## 2.2. Pruebas Unitarias

- Una prueba unitaria se utiliza para comprobar que un método concreto del código de producción funciona correctamente, probar las regresiones o realizar pruebas relacionadas (buddy) o de humo. Una prueba por orden se utiliza para ejecutar otras pruebas en un orden especificado.

### **3. DESARROLLO**

**3.1. Analisis**

**3.2. Diseño**

**3.3. Pruebas**

## 4. REFERENCIAS