Programación Web 3

UNLaM - Tecnicatura en Desarrollo Web

Trabajo Práctico de Investigación

[**1. Objetivo**](#_heading=h.30j0zll) **2**

[**2. Equipo**](#_heading=h.1fob9te) **2**

[**3. Desarrollo del Trabajo de Investigación**](#_heading=h.3znysh7) **2**

[**4. Forma de Entrega**](#_heading=h.tyjcwt) **3**

[**5. Condiciones de Aprobación**](#_heading=h.3dy6vkm) **3**

# Objetivo

Este documento describe el alcance y los requisitos para el desarrollo del trabajo de investigación de la materia Programación Web III. Este TP es condición necesaria para la aprobación de la materia.

# Equipo

El equipo para realizar el trabajo de investigación deberá ser de **4 alumnos**

# Desarrollo del Trabajo de Investigación

El trabajo de investigación será realizado a partir de un tema definido y compartido por los docentes de la materia a cada grupo en forma individual. Los temas deberán ser desarrollados en base a tecnologías dotnet.

Una vez compartido, los alumnos deberán desarrollar el trabajo de investigación que constará de las siguientes secciones:

## 

## Título

Net Core. Entity Framework en base de datos NoSQL

## Integrantes

Listado de los integrantes del equipo detallando nombre, apellido y documento.

Nombre y apellido: Reyna Rondo Sánchez DNI: 94.063.648

Nombre y apellido: Tomas DNI:

Nombre y apellido: Walter DNI:

Nombre y apellido: Federico DNI:

## Objetivo

Este trabajo de investigación se basará en comprender como se puede consumir bases de datos no SQL utilizando net Core EF. Atreves de un ejemplo de aplicación en ejecución de un sitio web desarrollado con ASP.NET MVC Core consumiendo una base de datos NoSQL MongoDb. Se explicará las diferencias entre una base de datos relacional con una no relacional.

## Situación Actual

Descripción de la situación actual de la tecnología, framework o concepto en relación al desarrollo de software utilizando tecnologías dotnet.

## Desarrollo de la Investigación

Desarrollo libre del contenido de la investigación describiendo y profundizando los conceptos claves del objetivo definido.

En esta sección se podrá incluir código de ejemplo para ejemplificar conceptos así como también links a repositorios de código propios o de terceros relacionados al objetivo.

## Historia Mongodb

Texto a desarrollar Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

## ¿Qué es mongodb?

Texto a desarrollar Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

## Implementación con .NET core y mongodb

Desarrollo libre del contenido de la investigación describiendo y profundizando los conceptos claves del objetivo definido.

En esta sección se podrá incluir código de ejemplo para ejemplificar conceptos así como también links a repositorios de código propios o de terceros relacionados al objetivo.

## Diferencias entre una base de datos no relacional y una relacional

Texto a desarrollar Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

Texto a desarrollar

## Conclusiones

La conclusión consiste en el resultado de la investigación realizada por los alumnos detallando los puntos trabajados

## Referencias

Listado de fuentes bibliográficas de internet o de libro utilizados para el desarrollo de la investigación.

* <https://docs.microsoft.com/en-us/aspnet/?view=aspnetcore-2.2#pivot=core>
* <https://docs.mongodb.com/manual/tutorial/>

# Forma de Entrega

La entrega del trabajo de investigación consta de una presentación oral por parte del equipo (máximo 20 minutos).

El contenido de la presentación consiste en la investigación realizada junto con las conclusiones obtenidas. Dependiendo del alcance del trabajo, los alumnos podrán realizar una demostración práctica a través de un ejemplo de código o la ejecución de una aplicación.

Para la presentación se deberá plasmar el contenido del trabajo dentro de una presentación que será entregada previamente a la fecha de presentación definida.

Una semana antes de la fecha definida para la exposición, la presentación deberá ser enviada/compartida a los siguientes emails:

* [juizmariano@unlam.edu.ar](mailto:juizmariano@unlam.edu.ar)
* [pnsanchez@unlam.edu.ar](mailto:pnsanchez@unlam.edu.ar)
* [mpazwasiuchnik@unlam.edu.ar](mailto:mpazwasiuchnik@unlam.edu.ar)
* agregar en copia (CC) a los restantes integrantes del grupo.

# Condiciones de Aprobación

1. El TP deberá cumplir con el desarrollo del trabajo de investigación junto a las conclusiones obtenidas.
2. Al momento de la entrega del trabajo práctico, todos los alumnos de cada equipo deberán estar presentes para la defensa del trabajo de investigación.
3. En la defensa del trabajo, se evaluará el grado de conocimiento y participación en el desarrollo del trabajo práctico de cada alumno.

Recursos Usados:

1. Visual Studio 2017 Community
2. [MongoDB](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/)
3. MongoDB.Driver (a través de Nuget)

Pasos:

1. Crear db en mongodb con una colección de documentos.
2. Crear un nuevo proyecto mediante la selección de la plantilla Web - Web Application> ASP .NET con la platilla de aplicación Web (Modelo-Vista-Controlador)
3. Para trabajar con MongoDB en la plataforma .NET debe instalar MongoDB.Driver en nuestro proyecto. Desde el proyecto clic derecho Nuget Gestor de paquetes y haga clic en Administrar paquetes Nuget para la Solución. Haga clic en Examinar y escriba el nombre MongoDB elegir MongoDB.Driver y haga clic en Instalar.
4. Crear en la carpeta de models una clase que represente al objeto a representar de la db no relacional.
   * Agregar a la clase MongoDB.Bson y Bson.Serialization.Attributes para tome la colección de la db.
   * Se anota con [BsonId] para designar esta propiedad como la clave principal del documento.
   * Se anota con [BsonRepresentation (BsonType.ObjectId)] para permitir pasar el parámetro como tipo en string lugar de una estructura ObjectId. Mongo maneja la conversión de string a ObjectId.
   * El [BsonElement] representa el nombre de la propiedad en la colección MongoDB.
5. Creamos la clase MongoDbContext que representará a nuestro contexto donde vamos a establecer las propiedades:

* ConnectionString - Obtiene la cadena de conexión
* Databasename - Obtiene el nombre de la colección
* IsSSL - Indica si SSL
* Notas - Devuelve la colección de notas

1. En el archivo appsettings.json la aplicación se establecerá conexión utilizada en nuestro proyecto y el nombre de la base de datos:

"MongoConnection": {

"ConnectionString": " **mongodb: // localhost: 27017** "

" Database": "**nombreBD**"

"IsSSL": true

},

1. Después en el archivo Startup.cs en el método ConfigureServices, obtiene la conexión y el nombre de la base de datos utilizada en la aplicación. Antes de la llamada AddMVC.

public void ConfigureServices(IServiceCollection services)

{

MongoDbContext.ConnectionString = Configuration.GetSection("MongoConnection:ConnectionString").Value;

MongoDbContext.DatabaseName = Configuration.GetSection("MongoConnection:Database").Value;

MongoDbContext.IsSSL = Convert.ToBoolean(this.Configuration.GetSection("MongoConnection:IsSSL").Value);

services.AddMvc();

}

1. En el Controller se realizaron el manejo de bd.

* **DeleteOne**: Elimina un solo elemento que coincide con los criterios de búsqueda proporcionados.
* **Find<TDocument>**: Devuelve todos los elementos de la colección que coinciden con los criterios de búsqueda proporcionados.
* **InsertOne**: Insertar el objeto proporcionado como un documento nuevo en la colección.
* **ReplaceOne**: Reemplaza el documento único que coincide con los criterios de búsqueda proporcionados con el objeto proporcionado.

Agregar una consulta con mongodb que simularía SQL