

Instituto Tecnológico de Costa Rica

Escuela de Computación

Bases de Datos

Profesor

William Mata Rodríguez

Segundo Proyecto

Trabajo de investigación
Resumen de los partidos

integrantes

Leonardo Mata Mora
Camila Víquez Alpízar

Fecha

29 de noviembre
de 2017

Contenido

Parte 1.....	3
Enunciado del Proyecto	3
Temas investigados.....	4
Instalación de mongo, carpetas log y data.....	4
Iniciar el servidor de mongo	4
Conexión con MongoDB	5
Diferencias entre SQL y MongoDB	5
Class Query.....	6
Almacenamiento de imágenes y videos	7
Documentos BSON y JSON.....	10
Tipo de dato ObjectId	11
Bson Ignore Extra Elements.....	11
Software para manejo de versiones.....	13
Git Hub.....	13
Diseño de la base de datos	14
Conclusiones del trabajo.....	15
Problemas encontrados y soluciones a los mismos.....	15
Aprendizajes obtenidos.....	17
Rubrica de evolución	18
Parte 2.....	19
Fuentes y otros objetos.....	19
Visual Estudio 2017	19
MongoDB.....	19
Sistema operativo Windows 7 o alguna versión superior	19
Robo Mongo 3T	19
Referencias Bibliográficas	20

Parte 1

Enunciado del Proyecto

Esta aplicación va a ser usada para mantener un resumen de los partidos del mundial junto con comentarios realizados por los aficionados.

El desarrollo del proyecto es en equipos de 3 estudiantes máximo, uno de ellos lo deben nombrar como coordinador. Importante: las experiencias han demostrado que los proyectos en equipos que no han sido administrados adecuadamente van a fallar, así que en cuanto noten que se presentan problemas al respecto de inmediato trátenlo primeramente con los miembros del equipo, y de no resolver lo comunican al profesor. Cualquier comunicación al profesor que vaya copiada a todos los miembros del equipo.

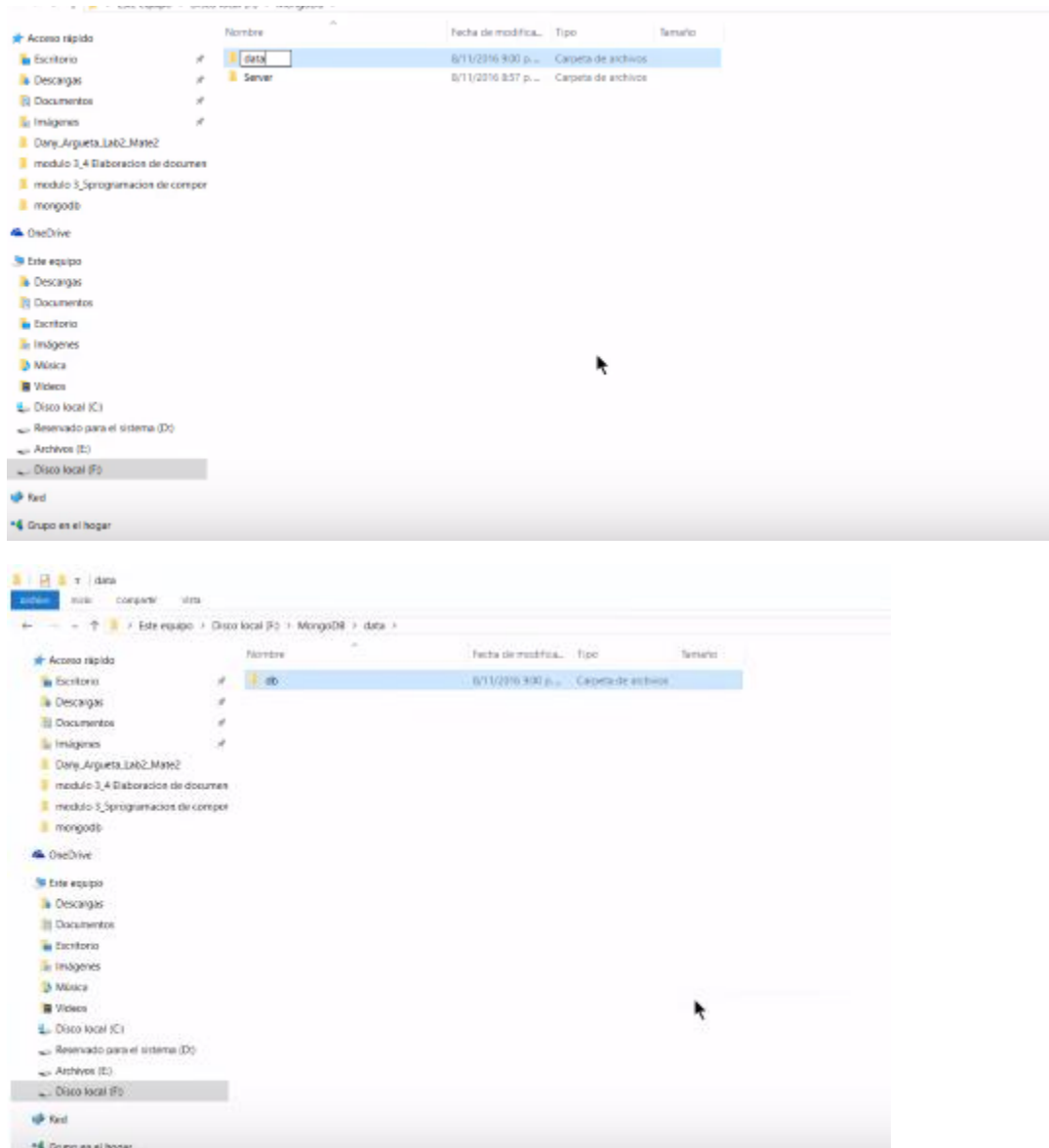
La base de datos a usar es MongoDB. Las otras herramientas de desarrollo son definidas por Ustedes garantizando que puedan cumplir con los requerimientos definidos en el proyecto.

Temas investigados

Instalación de mongo, carpetas log y data

Para la instalación de mongo se debe ingresar a la pagina de descarga de mongo y debe seleccionarse el paquete de descarga según la arquitectura y sistema operativo sobre el cual este trabajando. Una vez terminado el proceso de instalación se deben crear dos carpetas. La carpeta data dentro de esta se debe crear una carpeta que se llame db esta será en la cual se guarden todas las bases de datos que se creen en mongo.

Implementacion:



Luego debe de crear la carpeta log.

Iniciar el servidor de mongo

Una vez realizados los procesos descrito anteriormente debe acceder al cmd para iniciar la conexión con el servidor.

Implementación:

```

Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dan>mongo
MongoDB shell version: 3.2.10
connecting to: test
Server has startup warnings:
2016-11-08T21:16:28.622-0600 I CONTROL [initandlisten]
2016-11-08T21:16:28.623-0600 I CONTROL [initandlisten] ** WARNING: This 32-bit MongoDB binary is deprecated
2016-11-08T21:16:28.623-0600 I CONTROL [initandlisten]
2016-11-08T21:16:28.625-0600 I CONTROL [initandlisten]
2016-11-08T21:16:28.626-0600 I CONTROL [initandlisten] ** NOTE: This is a 32 bit MongoDB binary.
2016-11-08T21:16:28.628-0600 I CONTROL [initandlisten] ** 32 bit builds are limited to less than 2GB of data (or
less with --journal).
2016-11-08T21:16:28.637-0600 I CONTROL [initandlisten] ** Note that journaling defaults to off for 32 bit and is
currently off.
2016-11-08T21:16:28.638-0600 I CONTROL [initandlisten] ** See http://dochub.mongodb.org/core/32bit
2016-11-08T21:16:28.640-0600 I CONTROL [initandlisten]
>

```

Conexión con MongoDB

Primeramente debe crear un objeto “MongoClient” para poder interactuar con un clúster de MongoDB, por defecto la conexión se realiza a localhost 27017.

Para esto se deben instalar los drivers que permiten la conexión con Visual Studio. Una vez instalados puede hacer uso de la siguiente biblioteca y proseguir con la conexión.

```
using MongoDB.Driver.Builders;
```

Implementación:

```

29 referencias
class Mongo
{
    2 referencias
    public string connectionString { get; set; }
    2 referencias
    public string databaseName { get; set; }
    0 referencias
    public string collectionName { get; set; }
    private MongoClient mongoClient;
    private MongoServer mongoServer;
    private MongoDBDatabase dataBase;

    14 referencias
    public Mongo()
    {
        connectionString = "mongodb://localhost:27017";
        databaseName = "Campeonato";
    }
}

```

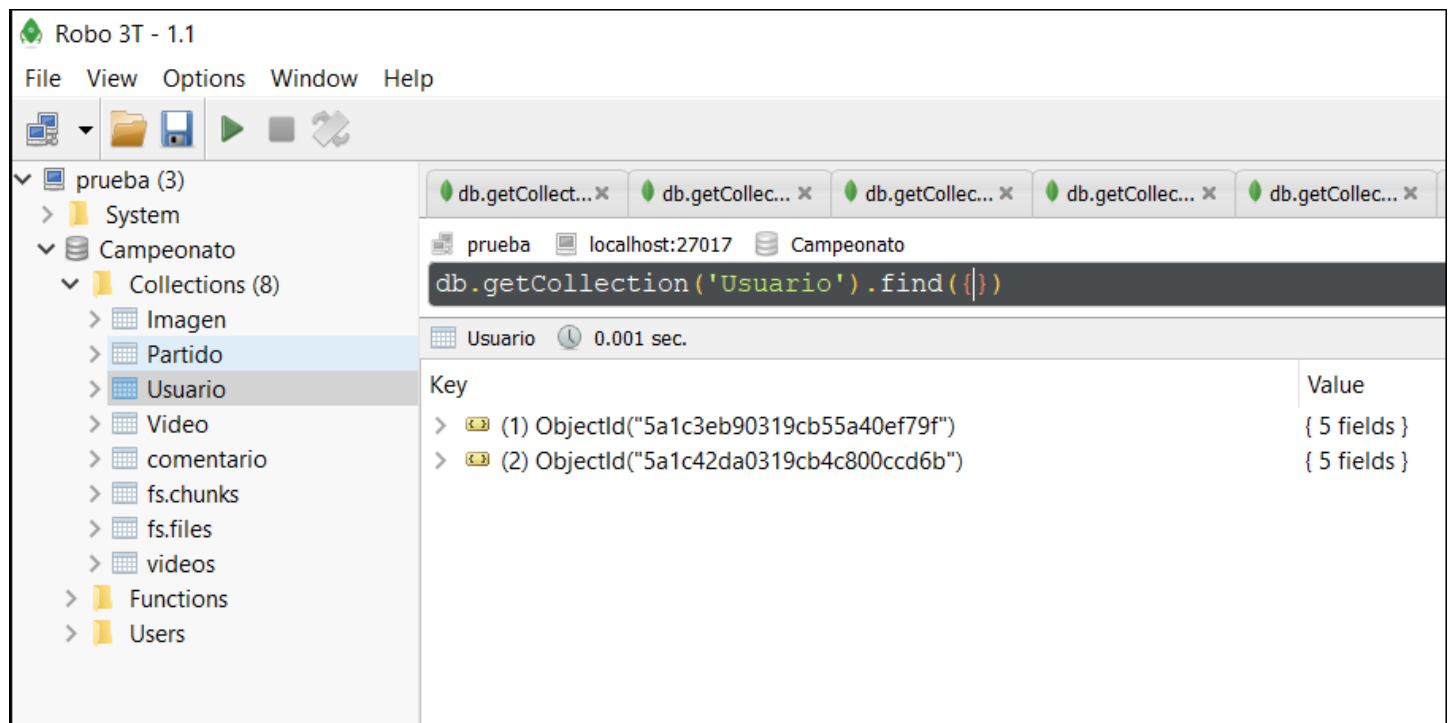
Diferencias entre SQL y MongoDB

MongoDB es no SQL por lo que se investigo sobre las diferencias que implica y a su vez es manejo de la información y la lógica que se debe implementar en una base de datos como MongoDB.

SQL	MongoDB
Database	Database
Table	Collection
Row	Document o Documento BSON
Column	Field
Index	Index
Table Joins	Documentos embebidos o enlazados
Primary Key	Primary Key
Unique	Field _id
Partición	Shard
Clave de Partición	Shard Key

Implementación:

Las colecciones fueron creadas en MongoDB mediante el uso del IDE llamado Robo 3T.



La base de datos se llama Campeonato y dentro de ella se tienen ocho colecciones imagen, partido, usuario, video, comentarios. Las colecciones fs.Chunks y fs.files las crea mongo, serán explicadas más adelante.

Class Query

MongoDB no posee la palabra reservada Query, pero para el manejo de las consultas desde el entorno de desarrollo Visual se implemento la clase Query que permite realizar consultas. Esta funcionalidad es obtenida del del driver instalado.

```
using MongoDB.Driver.Builders;
```

Implementation:

```

public Usuario getUsuario(string id) {

    var query = Query.EQ("_id", new ObjectId(id));
    MongoCollection<Usuario> collection = dataBase.GetCollection<Usuario>("Usuario");
    Usuario temp = collection.FindOne(query);
    temp.imagen = getImage(id);
    temp.imagen.Image = getImagen(temp.imagen.imageGridFS);
    return temp;
}

```

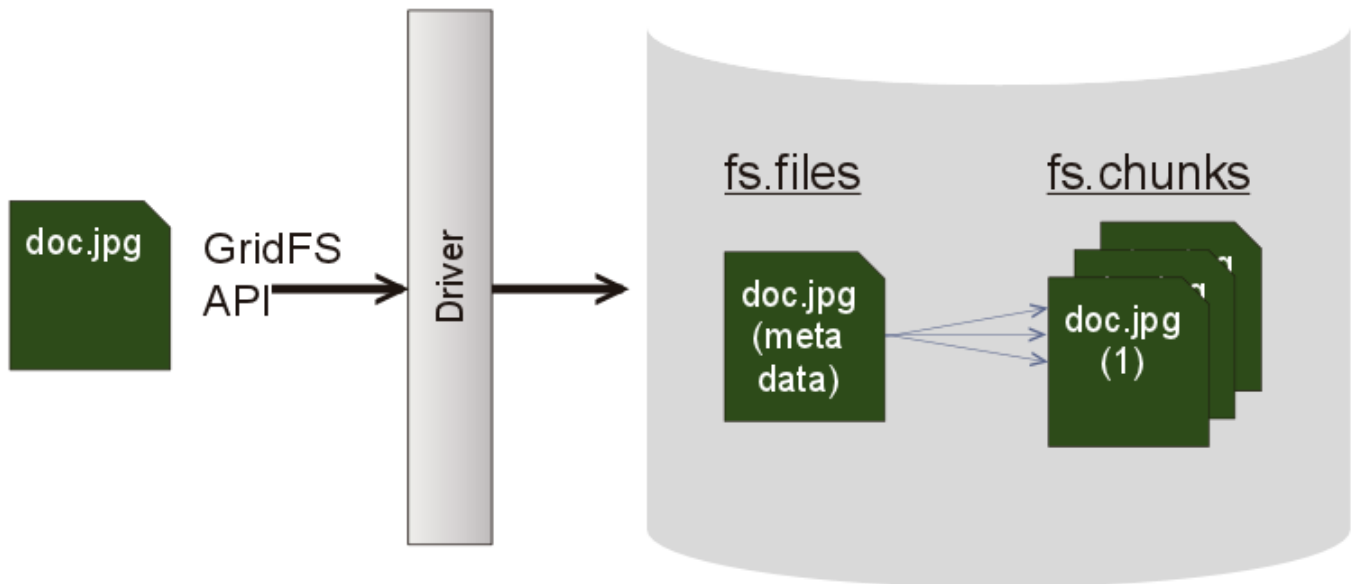
Almacenamiento de imágenes y videos

El almacenamiento de fotos y video en MongoDB sigue un flujo en el cual crea las colecciones chunks y files. Estas colecciones son creadas cuando una aplicación esta guardando datos usando GridFS, esta es necesaria para el almacenamiento de videos y fotos en MongoDB, en este proyecto.

GridFs almacena datos en estas dos colecciones:

- **chunks** stores the binary chunks. For details, see The chunks Collection.
- **files** stores the file's metadata. For details, see The files Collection.

Para comprenderlo mejor puse observar esta imagen



A nivel lógico lo que almacena cada una de estas colecciones se explica de la siguiente manera

La colección fs.Chunks se ve así:

```
{
  "_id" : <ObjectId>,
  "files_id" : <ObjectId>,
  "n" : <num>,
  "data" : <binary>
}
```

La colección fs.Files se ve así:

```
{
  "_id" : <ObjectId>,
  "length" : <num>,
  "chunkSize" : <num>,
  "uploadDate" : <timestamp>,
  "md5" : <hash>,
  "filename" : <string>,
  "contentType" : <string>,
  "aliases" : <string array>,
  "metadata" : <any>,
}
```


En el programa se implementó de la siguiente forma

Para las imágenes:

```
public string saveImagen(string rutaImagen)
{
    var fileName = rutaImagen;
    string idImagen;

    using (var fs = new FileStream(fileName, FileMode.Open))
    {
        var gridFsInfo = dataBase.GridFS.Upload(fs, fileName);
        var fileId = gridFsInfo.Id;
        idImagen = fileId.ToString();
    }

    return idImagen;
}
```

Para los videos:

```
public string saveVideo(string rutaVideo) {
    var fileName = rutaVideo;
    string idVideo;

    using (var fs = new FileStream(fileName, FileMode.Open))
    {
        var gridFsInfo = dataBase.GridFS.Upload(fs, fileName);
        var fileId = gridFsInfo.Id;
        idVideo = fileId.ToString();
    }

    return idVideo;
}
```

Recuperación del video:

```
public byte[] getVideo(string idVideo)
{
    var fileID = new ObjectId(idVideo);
    var file = dataBase.GridFS.FindOneById(fileID);

    byte[] video;

    using (var fs = file.OpenRead())
    {
        var bytes = new byte[fs.Length];
        fs.Read(bytes, 0, (int)fs.Length);
        var stream = new MemoryStream(bytes);

        video = bytes;
    }
}
```

Para la reproducción del video en la interfaz se hizo uso de la biblioteca WMPLib, se implementa de la siguiente manera `using WMPLib;`

Este es necesario para el manejo de Windows Media Player.

Documentos BSON y JSON

MongoDB es una base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Estos documentos son un formato de intercambio de datos. Consiste en una lista ordenada de elementos. Cada elemento consta de un campo nombre, un tipo y un valor. Los nombres son de tipo String y los tipos pueden ser enteros, booleanos nulos. Los tipos BSON son los mismos tipos de JSON además de añadir Date y Array de bytes.

Implementación:

Para el desarrollo de este proyecto se usaron archivos BSON. Para su implementación se crearon clases con sus respectivos getter y setter para obtener los datos de estos. Para esto se importo una biblioteca, la cual puede ver en la siguiente fracción de código

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;
using System.Drawing;
```

Tipo de dato ObjectId

Un ObjectId es un tipo de BSON de 12 bytes que tiene la siguiente estructura:

- Los primeros 4 bytes que representan los segundos desde la época de Unix
- Los siguientes 3 bytes son el identificador de la máquina
- Los siguientes 2 bytes consisten en la identificación del proceso
- Los últimos 3 bytes son un valor de contador aleatorio

MongoDB usa ObjectIds como el valor predeterminado del campo `_id` de cada documento, que se genera al crear cualquier documento. La combinación compleja de ObjectId hace que todos los campos `_id` sean únicos.

Uno puede crear su propio id pero este debe ser de 12 bytes.

Implementación:

Este `_id` que genera Mongo fue muy útil ya que asegura que cada id sea único, se implementó en las consultas y la forma para implementarlo fue instanciando el objeto `ObjectId` de la siguiente forma

```
>newObjectId = ObjectId()
```

Bson Ignore Extra Elements

Cuando crea una colección en MongoDB se crea un archivo de tipo BSON, este trae consigo datos que no son necesarios para el programa que se está desarrollando. Esto puede generar problemas, por esta razón se implementa la clase `BsonIgnoreExtraElements`. Esto se debe poner en cada clase que se implemente.

Implementación:

```

namespace investigacion
{
    [BsonIgnoreExtraElements]
    25 referencias
    class Usuario
    {
        7 referencias
        public ObjectId id { get; set; }
        4 referencias
        public string contraseña { get; set; }
        6 referencias
        public string mail { get; set; }
        10 referencias
        public Imagen imagen { get; set; }
        5 referencias
        public string rol { get; set; }
        0 referencias
        public Bitmap imag { get; set; }
        3 referencias
        public bool disponible { get; set; }
    }
}

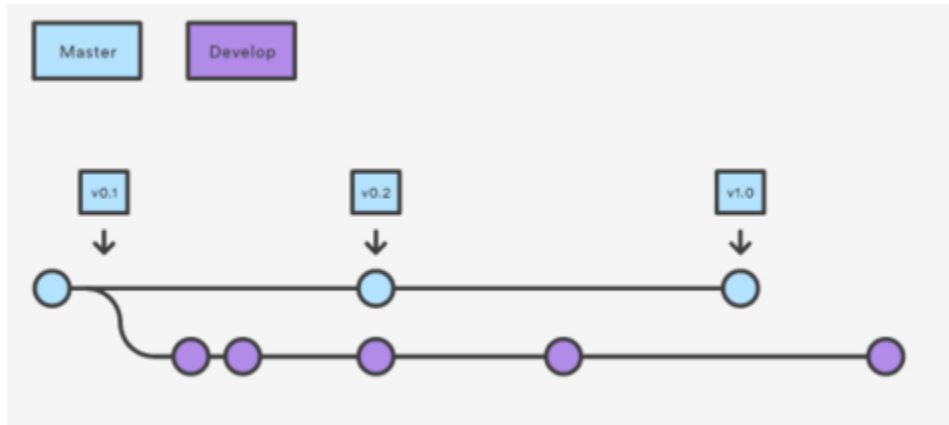
```

Software para manejo de versiones

Git Hub

Git es un sistema distribuido de control de código fuente. Está preparado para poder trabajar en equipos distribuidos.

Flujo de trabajo de git:



Para trabajar con git se debe seguir un flujo de trabajo. El cual es sencillo de seguir.

Master contiene siempre una versión estable del proyecto, y se tiene una copia de master denominada develop. Develop corresponde a la versión estable de desarrollo.

Git introduce un concepto llamado ramas, que consisten en una copia de develop, sobre la cual trabaja cada miembro del equipo. Cada vez que un miembro del equipo concluye su trabajo exitosamente o logra un avance significativo incorpora sus cambios a develop, esto se realiza mediante los comandos

`$git add .`

`$git commit -m "breve descripción sobre lo que se hizo"`

`$git push`

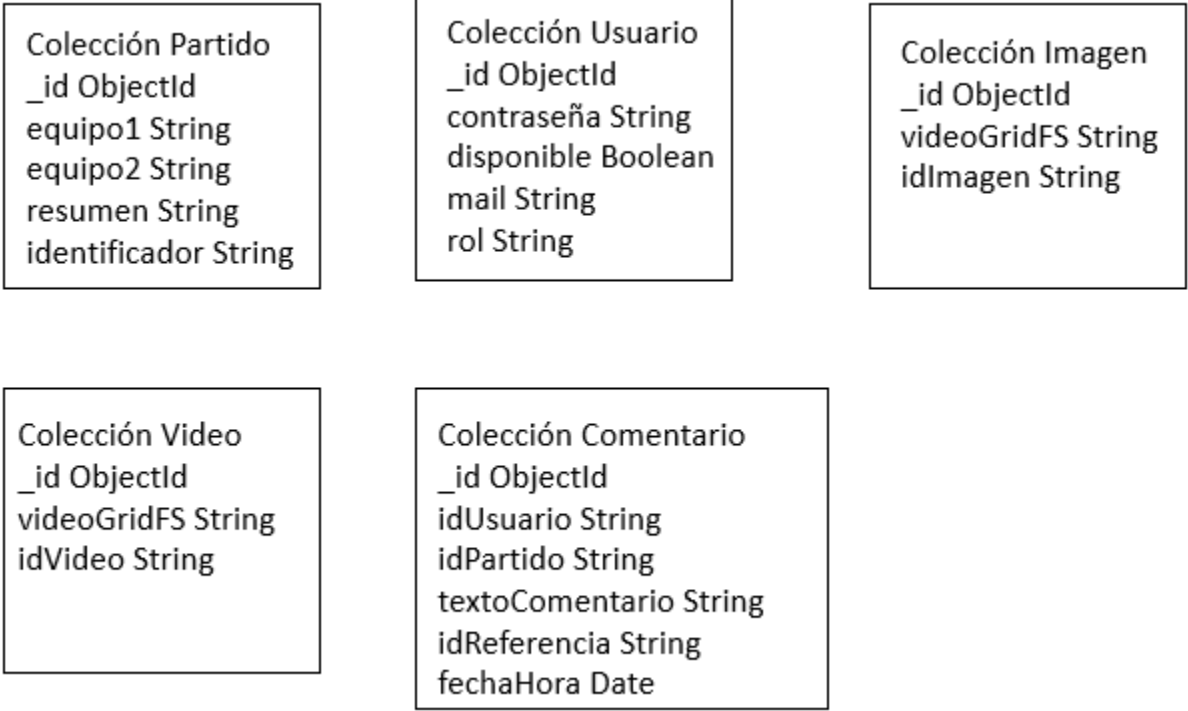
Posteriormente actualizamos nuestra rama con lo que los miembros del equipo han ido actualizando. Esto mediante `$git pull`

Es preferible que el master sea actualizado por una sola persona, después de haberle dado el visto bueno a develop

Estos cambios quedan registrados en la nube de git.

Diseño de la base de datos

El diseño planificado para el Proyecto resumen del partido, después de analizar sus requerimientos, fue el siguiente:



Conclusiones del trabajo

Problemas encontrados y soluciones a los mismos

Bson Ignore Extra Elements

Al hacer uso de las clases y posteriormente de las consultas habían problemas de consistencia cuando se utilizaban los atributos deseados, esto es por el manejo de los archivos BSON por parte de MongoDB. Cuando Mongo crea estos archivos no agrega únicamente los atributos de las conexiones, y esta información extra debe ser filtrada para que no genere dichas inconsistencias. En cada clase debe escribirse `BsonIgnoreExtraElements` entre []. De la siguiente forma:

```
]namespace investigacion
{
    [BsonIgnoreExtraElements]
    25 referencias
    class Usuario
    {
        7 referencias
        public ObjectId id { get; set; }
        4 referencias
        public string contraseña { get; set; }
        6 referencias
        public string mail { get; set; }
        10 referencias
        public Imagen imagen { get; set; }
        5 referencias
        public string rol { get; set; }
        0 referencias
        public Bitmap imag { get; set; }
        3 referencias
        public bool disponible { get; set; }
    }
}
```

ObjectId

Se presentaban inconsistencias cuando se hacían consultas a la base por medio de un atributo al cual habíamos llamado id. Después de investigar este error, el cual daba una excepción referenciada a nulo e imposibilitaba dicha consulta, se supo que un atributo no se puede llamar id aunque no sea del tipo ObjectId, ya que Mongo al tener su propio id y denominarlo _id presenta problemas. Esto lo solucionamos llamando al atributo de otra forma.

Para solucionar esto de cambio el nombre del atributo por identificador

```
[BsonIgnoreExtraElements]
23 referencias
class Partido
{
    8 referencias
    public ObjectId id { get; set; }

    3 referencias
    public string identificador { get; set; }
    5 referencias
    public string equipo1 { get; set; }
    5 referencias
    public string equipo2 { get; set; }
    12 referencias
    public List<Comentario> comentarios { get; set; }
    16 referencias
    public List<Video> videos { get; set; }
    4 referencias
    public string resumen { get; set; }
}
```

Conexión con mongo

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>F:

F:\>mongod
2016-11-08T21:15:48.928-0600 I CONTROL [main]
2016-11-08T21:15:48.930-0600 W CONTROL [main] 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2016-11-08T21:15:48.930-0600 I CONTROL [main]
2016-11-08T21:15:48.938-0600 I CONTROL [initandlisten] MongoDB starting : pid=6416 port=27017 dbpath=F:\data\db\ 32-bit
host=DESKTOP-TTQ779B
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] db version v3.2.10
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] allocator: tcmalloc
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] modules: none
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] build environment:
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] distarch: i386
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] target_arch: i386
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] options: {}
2016-11-08T21:15:48.940-0600 I STORAGE [initandlisten] exception in initAndListen: 28663 Cannot start server. The default storage engine 'wiredTiger' is not available with this build of mongod. Please specify a different storage engine explicitly, e.g. --storageEngine=mmapv1., terminating
2016-11-08T21:15:48.941-0600 I CONTROL [initandlisten] dbexit: rc: 100
```

Este error al intentar conectar con mongo surgió porque no se indicó la ruta de la carpeta data, se solucionó al indicar la ruta.

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>F:

F:\>mongod
2016-11-08T21:15:48.928-0600 I CONTROL [main]
2016-11-08T21:15:48.930-0600 W CONTROL [main] 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2016-11-08T21:15:48.930-0600 I CONTROL [main]
2016-11-08T21:15:48.938-0600 I CONTROL [initandlisten] MongoDB starting : pid=6416 port=27017 dbpath=F:\data\db\ 32-bit
host=DESKTOP-TTQ779B
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] db version v3.2.10
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] allocator: tcmalloc
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] modules: none
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] build environment:
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] distarch: i386
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] target_arch: i386
2016-11-08T21:15:48.939-0600 I CONTROL [initandlisten] options: {}
2016-11-08T21:15:48.940-0600 I STORAGE [initandlisten] exception in initAndListen: 28663 Cannot start server. The default storage engine 'wiredTiger' is not available with this build of mongod. Please specify a different storage engine explicitly, e.g. --storageEngine=mmapv1., terminating
2016-11-08T21:15:48.941-0600 I CONTROL [initandlisten] dbexit: rc: 100

F:\>mongod --storageEngine=mmapv1 --dbpath_
```


Aprendizajes obtenidos

El manejo de una base de datos no SQL deja como aprendizaje la implementación y la lógica de esto, no se puede dejar de lado la dimisión a nivel lógico que implica una base de datos no SQL como MongoDB, comprendimos como utilizar una herramienta útil para almacenamiento masivo de datos, en una de las base de datos no SQL más usada actualmente como lo es MongoDB.

Al estar trabajando durante el semestre con bases de datos relacionales este proyecto viene a expandir nuestros conocimientos y lo mas importante romper el esquema relacional que tenían como patrón en nuestra lógica personal. Sin lugar a duda este proyecto nos ha vinculado mas con la realidad actual de bases de datos no relacionales que son usadas en muchas aplicaciones que usamos día a día y que sin duda debemos conocer como realizan el manejo de sus datos.

Poder almacenar datos en archivos es muy ventajoso y antes de este proyecto no sabíamos de esta implementación en bases de datos.

El manejo de colecciones y el conocimiento de lo que son las colecciones nos hace tomar una perspectiva nueva sobre todo lo que es posible fuera del sistema relacional. Suele ser chocante el cambio entre ambos sistemas ya que las bases no relacionales violan protocolos como ACID lo cual al inicio no se comprende porque sucede, pero con la implementación de Mongo a lo largo del proyecto de investigación se va comprendiendo cada detalle y se entiende el objetivo y el porque de su existencia y no solo la existencia de bases relacionales.

Rubrica de evolución

Concepto	Puntos	Puntos obtenidos	Avance 100/%/0	Análisis de resultados
Diseño de la base de datos	5	5		
CRUD de aficionados	10	10		
CRUD del resumen	35	35		
CRUD de los comentarios	25	20		
Validaciones de datos y procesos	10	10		
OPERACIONES MONGODB: operaciones en vivo que van a desarrollar los estudiantes de forma individual según el modelo implementado por el grupo	10*		---	
Ayuda	5	5		
TOTAL	100*			
Partes desarrolladas adicionalmente				

Parte 2

Fuentes y otros objetos

Visual Estudio 2017

La aplicación fue desarrollada en el ambiente de desarrollo visual estudio 2017. Esta en una versión de pago, pero se tiene acceso gratuito mediante la alianza de Microsoft imagine y el Instituto Tecnológico de Costa Rica. En el siguiente enlace puede ver un tutorial para su correcta instalación:

<https://www.youtube.com/watch?v=o0jqKssaoW8&feature=youtu.be>

MongoDB

Para el desarrollo de este proyecto se usó la versión 3.4.10, esta no es la ultima versión, pero es muy consistente por eso se optó por usarla. La versión del servidor corresponde a la misma ya que requiere que sean la misma versión para un correcto funcionamiento. Para su instalación puede seguir como guía este video:

https://www.youtube.com/watch?v=tq0562W_3tA

Sistema operativo Windows 7 o alguna versión superior

Si esto no es posible, se puede implementar mediante el uso de una máquina virtual

Robo Mongo 3T

Para el uso de MongoDB se puede hacer desde la línea de comandos o bien se puede hacer desde una interfaz grafica. Durante el desarrollo de este programa se hizo uso de Robo 3T. Este es gratuito y fácil de usar.

Esta disponible para Windows, Mac y Linux. Es muy rápido y fácil de usar, en segundos lo conectamos a cualquier base de datos MongoDB, local o remota, y podemos hacer las operaciones más comunes. Y lo mejor de todo: es gratis y open source.

Referencias Bibliográficas

Microsoft. (s.f.). Visual Basic. Developer Network. Recuperado de: <https://msdn.microsoft.com/es-es/library/2x7h1hfk.aspx>. Consultado el 18 Nov. 2017.

Ruano, F. (2001). Google Books. Documento en línea: Analisis y desarrollo de MongoDB y Redis en Java. Recuperado de https://books.google.co.cr/books?id=9PlpBAAQBAJ&pg=PT53&lpg=PT53&dq=para+que+sirve+el+puerto+27017&source=bl&ots=grNSBfRSoE&sig=eZOL1m682_FuZ_8ZvZ66ITIPvXY&hl=es-419&sa=X&ved=0ahUKEwi8x8KdVuLXAhUHUSYKHYo5AsgQ6AEIPzAE#v=onepage&q=para%20que%20sirve%20el%20puerto%2027017&f=false. Consultado el: 25 Nov 2017.

Docs.mongodb.com. (2010). ObjectId — MongoDB Manual 3.4. Documento en línea. Recuperado de <https://docs.mongodb.com/manual/reference/method/ObjectId/> . Consultado el 18 Nov. 2017.

DB, M. (2017). GridFS — MongoDB Manual 3.4. . Documento en línea: Docs.mongodb.com. Available at: <https://docs.mongodb.com/manual/core/gridfs/#gridfs-collections>. Consultado el 18 Nov. 2017.

DB, M. (2008). MongoDB ODBC & JDBC Drivers | Simba Technologies. [online] Simba Technologies. Available at: <https://www.simba.com/drivers/mongodb-odbc-jdbc/> Consultado el 13 Nov. 2017.