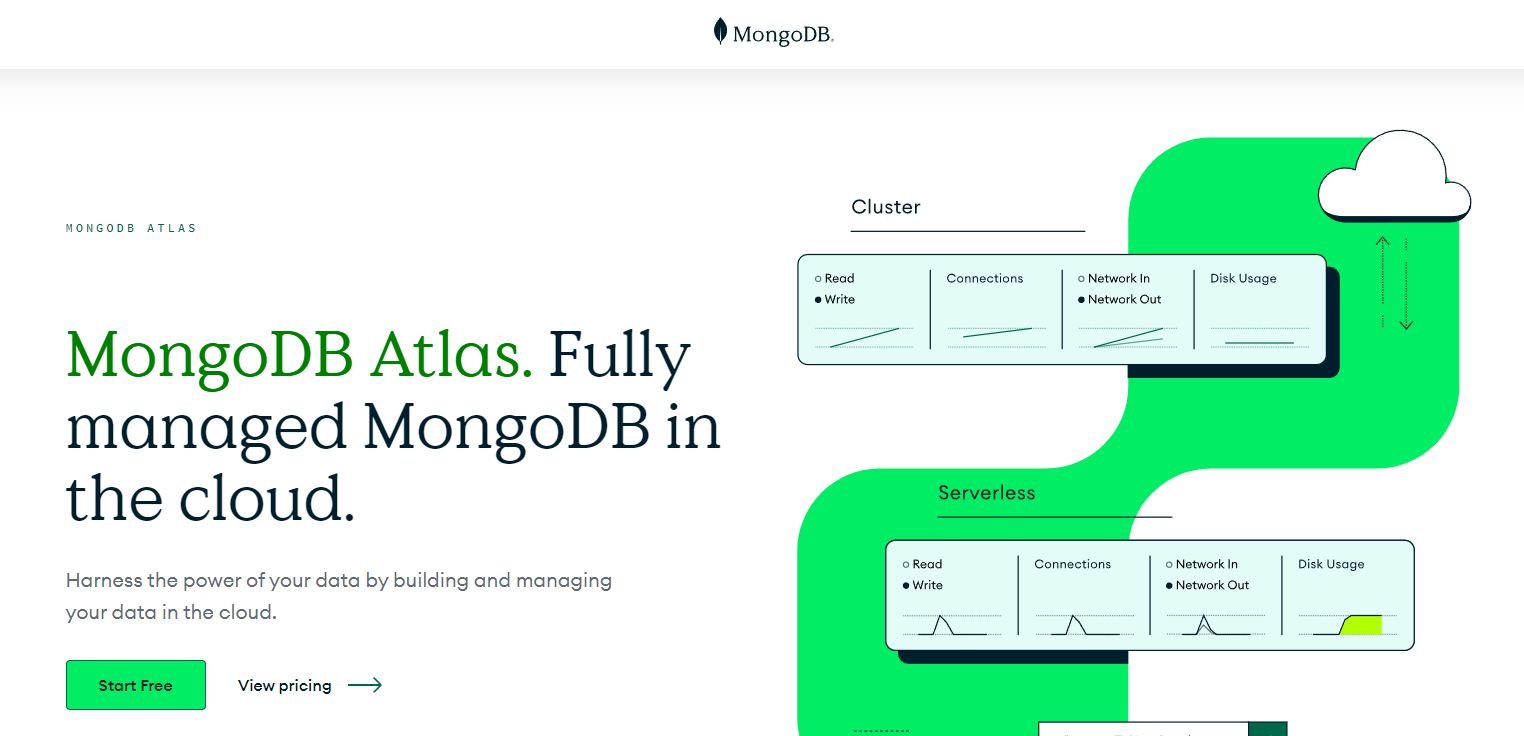
Juan Pablo Rubio Mejia 000321656

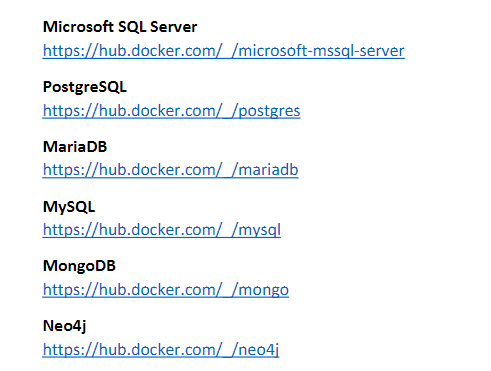
Kevin Hoyos Corrales 000374134

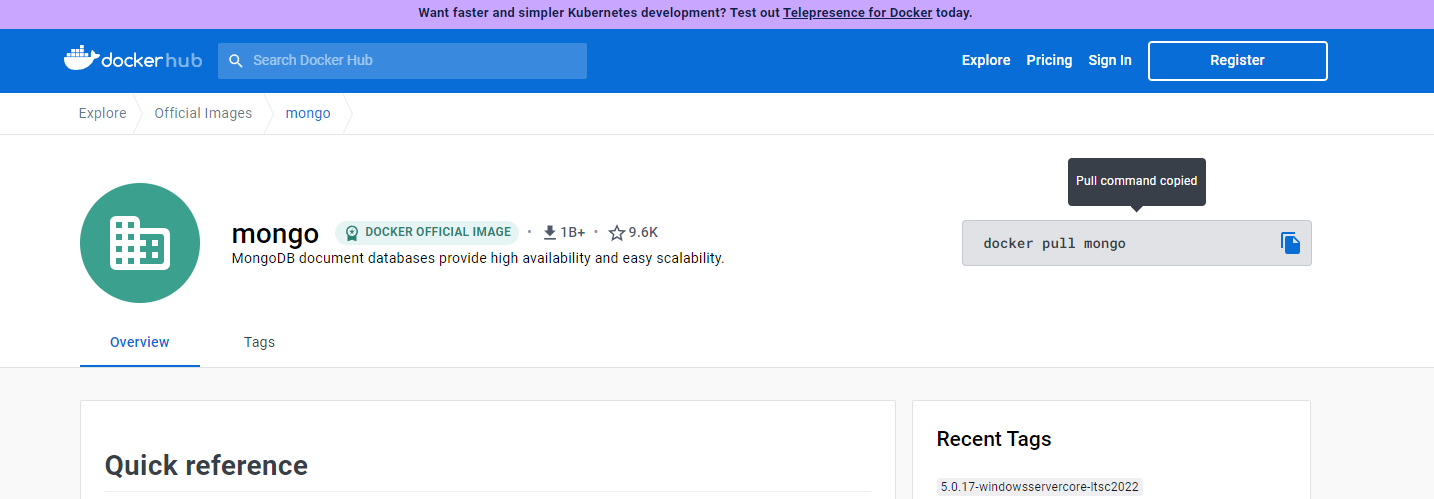
Bueno aqui estara la solucio nal examen #4 de Topicos Avanzado de Bases de Datos

Para empezar vamos a buscar nuestro motor de bases de datos NoSQL estabamos entre firebase o mongoDB pero por criterios usaremos mongoDB  
  
lo primero seria ir a su web y descargarlo



Y ahora descargamos la imagen para eso Podemos usar el document dejado al inicio de semester



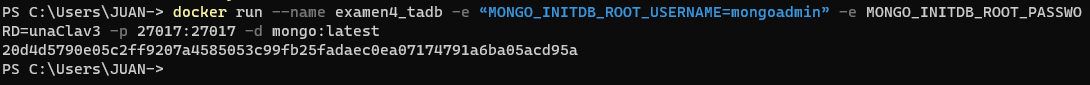


Ahí vemos la imagen y su link o comando para descargar usamos eso en powershell

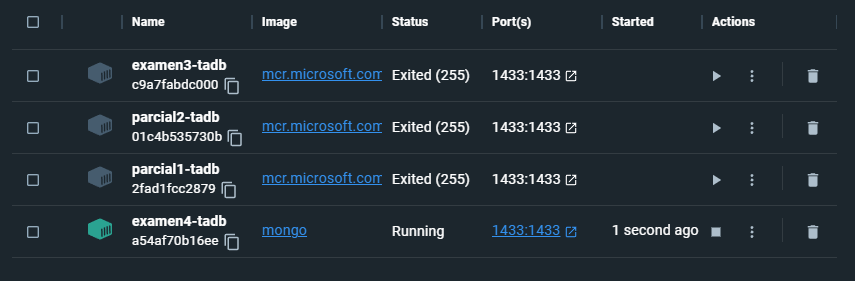
Texto

Descripción generada automáticamente

Listo ahora lo siguiente es el contenedor



Y verificamos que si este, y pues si, efectivamente si esta



También se puede ver en la app de Docker

Continuando con el examen, tenemos en cuenta que es el mismo examen 3 pero cambia que ahora es en bases nosql… entonces vamos a ver que hacer, lo principal seria implementar el CRUD en las diferentes tablas, ya lo demás vendrá por añadidura

Bueno continuando desde aquí tuvimos varios inconvenientes y problemas debido a que nunca nos quiso entrar a la base de datos salió errores de autenticación o problemas y se cierra sesión (lo cual raro ya que nunca entraba) entonces por falta de tiempo y veguenza de decirle al profesora faltando poco para entregar el examen decidimos aprender una tecnología nueva jejeje ene ste caso cambiamos a firebase (no sabemos anda de firebase pero bueno que se le va a hacer tampoco debe ser tan difícil (esperamos...))

Para empezar entramos en la pagina principal de firebase

Creamos un proyecto

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

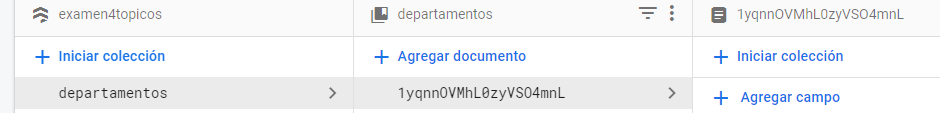
En este caso examen 4

Y nos vamos a

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y bueno aquí creamos nuestra base de datos… hasta ahora la tenemos mas o menos esto…

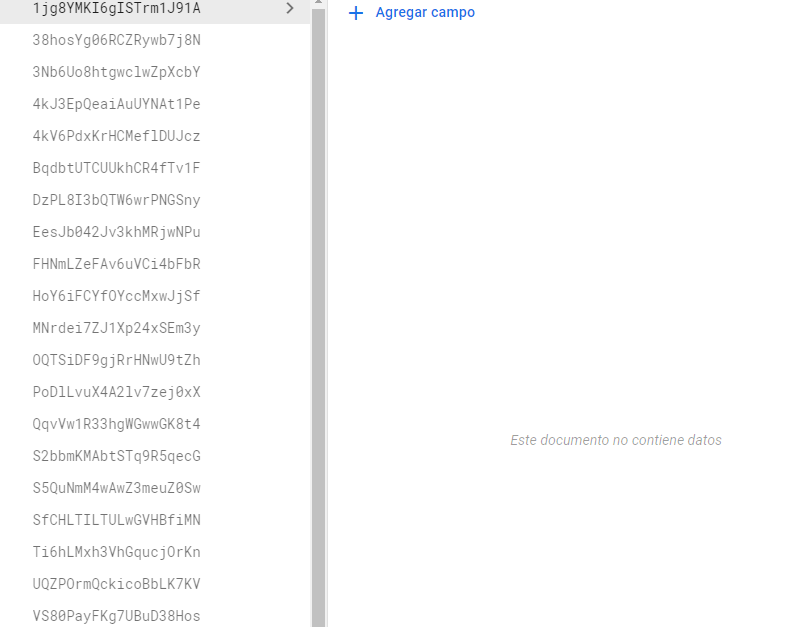


Tenemos colección, documento y campo

Entonces hasta donde tenemos entendido va de la siguiente manera… una colección es equivalente a una tabla en SQL y un documento a un registro y campo es ver que hay en el registro

Ahora según estábamos viendo e investigando… no es tan fácil ni sencillito como parece… ahora lo que sigue seria cargar nuestros archivos CSV lo cual al parecer no deja… buscando descubrimos que ahora debe haber paginas externas para poder subir documentos… lo cual también nos tocara aprender cómo hacerlo y organizar los documentos que ya teníamos para que quede diferente y aceptado por firebase… pd estos registros llevan un ID único

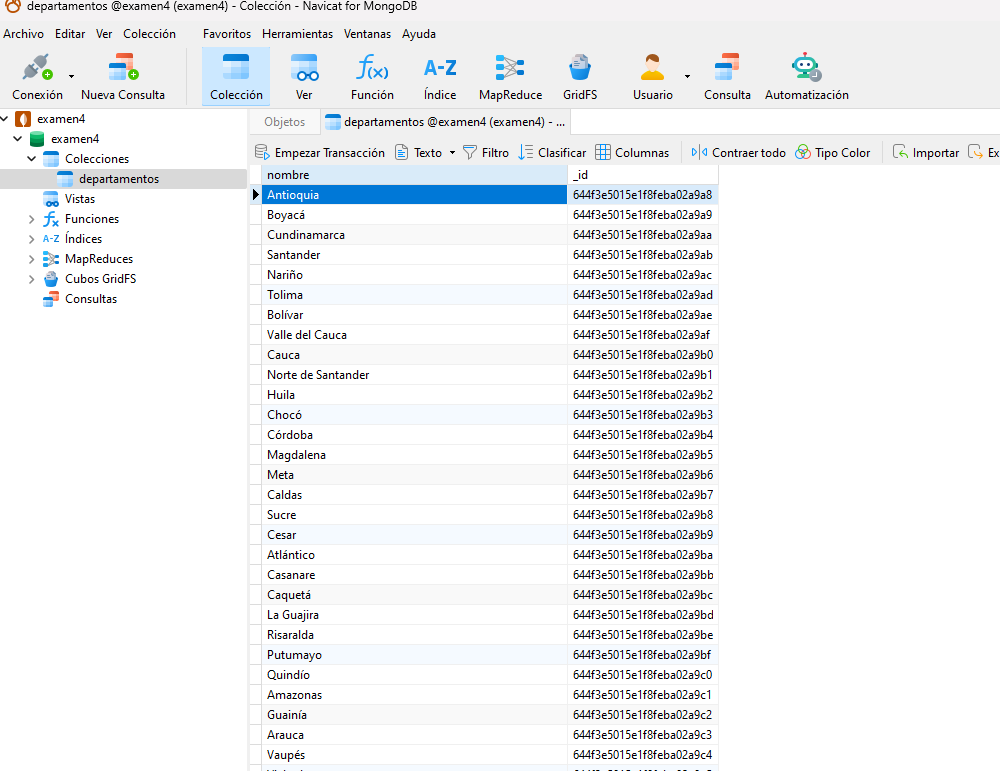
Después de mucha investigación… mucha mas aun

muchas pero muchas pero demasidas fallas e intentos… logramos conectarnos a la base de datos de mongo… }

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Terminamos usando navicat, fue el único que funciono, no sabíamos que datagrip también tenia mongo pero no nos funcionaba así que es lo que hay toco con este



Otros mil anos mirando como importar, pero progresando jajaja y saber que esto solo es carpintería solo arreglamos la base de datos, pero ahí vamos

las colecciones nos quedó de la siguiente manera

beneficiarios

{

"\_id": ObjectId("644f417215e1f8feba02a9d1"),

"id\_beneficiario": NumberLong("1"),

"nombre": "Juan",

"apellido": "Pérez",

"fecha\_nacimiento": ISODate("1985-05-14T00:00:00.000Z"),

"genero": "masculino",

"id\_municipio": NumberLong("1")

}

municipios

{

"\_id": ObjectId("64509861871678f241053da8"),

"id\_municipio": NumberLong("1"),

"nombre\_municipio": "Medellín",

"nombre\_departamento": "Antioquia"

}

departamentos

{

"\_id": ObjectId("644f3e5015e1f8feba02a9ae"),

"nombre": "Bolívar"

}

tipos\_subsidios

{

"\_id": ObjectId("644f5b3415e1f8feba02adb0"),

"id\_subsidio": NumberLong("2"),

"nombre\_subsidio": "Jóvenes en Acción",

"monto\_subsidio": NumberLong("400000")

}

asignacion\_subsidios

{

"\_id": ObjectId("644f5c0515e1f8feba02b215"),

"id\_asignacion": NumberLong("1"),

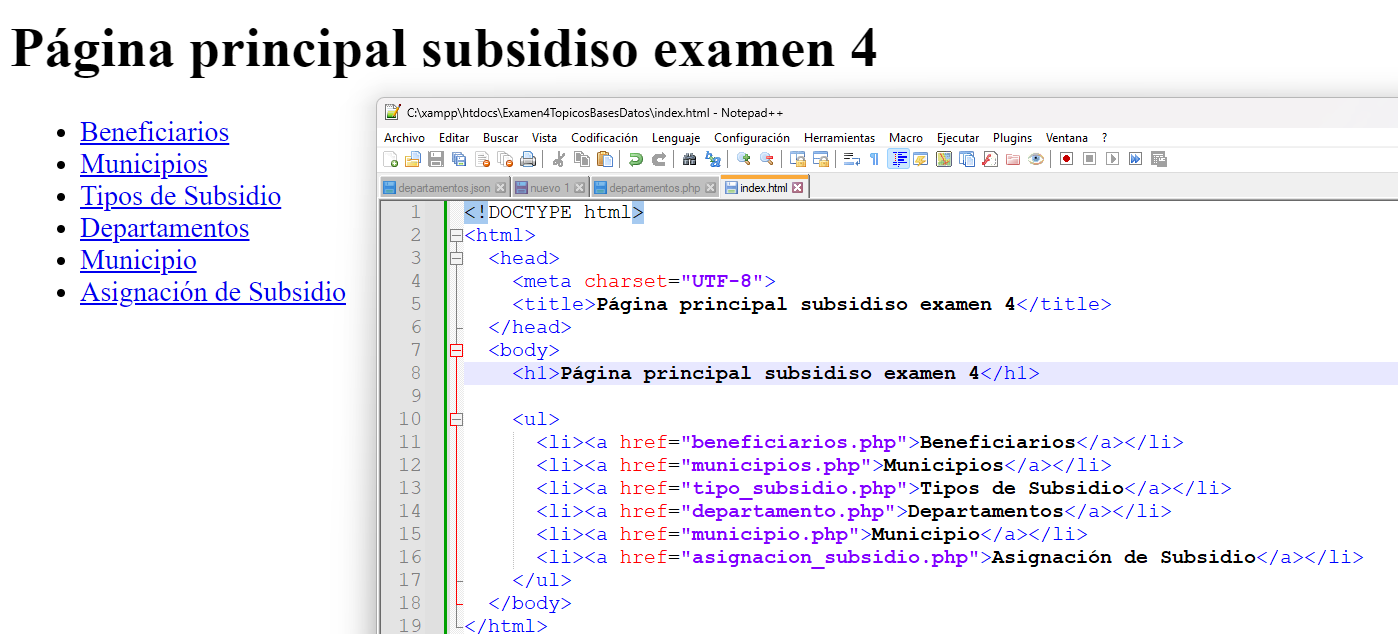
"id\_beneficiario": NumberLong("1"),

"id\_subsidio": NumberLong("2"),

"fecha\_asignacion": ISODate("2013-04-21T00:00:00.000Z")

}

Después de haber organizado e importado todo podemos usar el esqueleto de nuestro proyecto anterior



Donde tendremos un link para ir accediendo a cada uno de los parámetros y poder implementar el CRUD en la base de datos, por eso los links nos llevan a las tablas que tenemos

Por cierto, datazo que me gustaría haber sabido… este IDE tiene problemas con casi todos los tipos de archivos, excepto con xlsx.

Bueno tuvimos problemas en la implementación

Gráfico

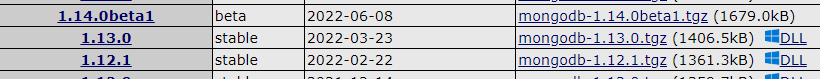
Descripción generada automáticamente con confianza media

**Fatal error**: Uncaught Error: Class "MongoDB\Driver\Manager" not found in C:\xampp\htdocs\Examen4TopicosBasesDatos\departamento.php:9 Stack trace: #0 {main} thrown in **C:\xampp\htdocs\Examen4TopicosBasesDatos\departamento.php** on line **9**

Para esto que debemos hacer… pos ya te digo

Lo primero seria ir a este link

<https://pecl.php.net/package/mongodb> buscamos la version mas reciente



En este caso la 1.13 y le damos en dll

Texto

Descripción generada automáticamente

Abajo del todo le damos en la 8.1 thread safe (ts) eso nos da una carpeta zip, lo que hacemos con ella es ir a la ruta donde tengas el xampp y en php, buscas la carpeta exe y pegas el dll, luego te vas a php.ini y configuras en las extensiones

Texto

Descripción generada automáticamente

Ahora, pues esto lo hice yo

Creas un archivo llamado test

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

También puedes solo poner localhost/php.info en el buscador pero queda mas pro así  
y miramos la info

Aplicación

Descripción generada automáticamente

Buscamos hacia abajo

Interfaz de usuario gráfica, Aplicación, Tabla

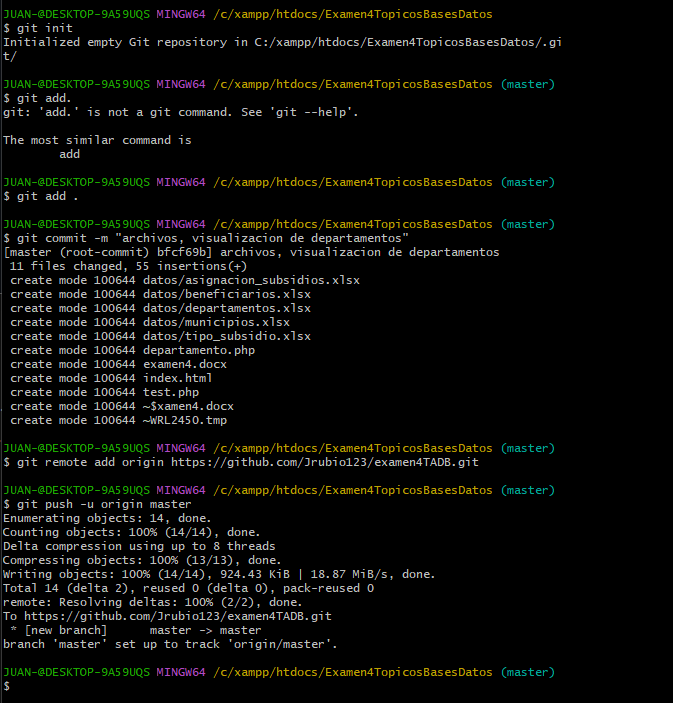
Descripción generada automáticamente con confianza media

Y nos debe aparecer esto, revisamos nuestra pagina

Interfaz de usuario gráfica, Texto, Aplicación

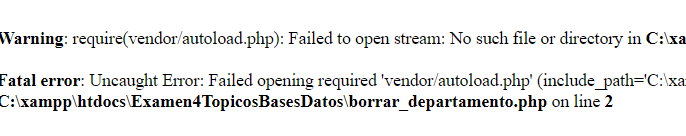
Descripción generada automáticamente

Y efectivamente ya tenemos visión



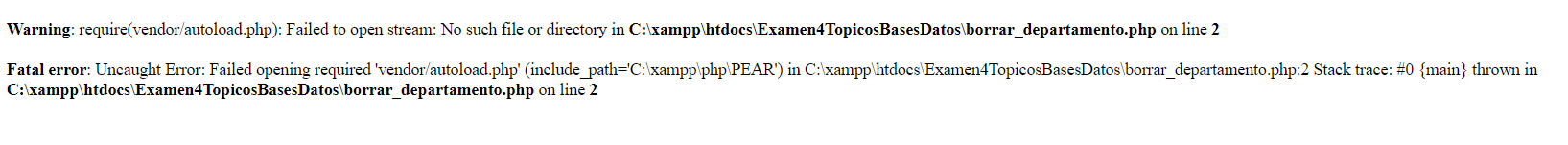
Creamos primer commit a nuestro git y continuamos

Ahora creando la parte de editar… tenemos un problema



No tenemos el vendor o autoload… lo cual se soluciona descrgando composer lo ejecutamos reiniciamos la pc…

Hay problemas con la hora de borrar algún departamento, no sabemos porque sale tanto error, entre ellos este:



**Warning**: require(vendor/autoload.php): Failed to open stream: No such file or directory in **C:\xampp\htdocs\Examen4TopicosBasesDatos\borrar\_departamento.php** on line **2**  
  
**Fatal error**: Uncaught Error: Failed opening required 'vendor/autoload.php' (include\_path='C:\xampp\php\PEAR') in C:\xampp\htdocs\Examen4TopicosBasesDatos\borrar\_departamento.php:2 Stack trace: #0 {main} thrown in **C:\xampp\htdocs\Examen4TopicosBasesDatos\borrar\_departamento.php** on line **2**

le hemos hecho de todo pero nada que nos funciona, pero continuando con esto… tenemos el botón editar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

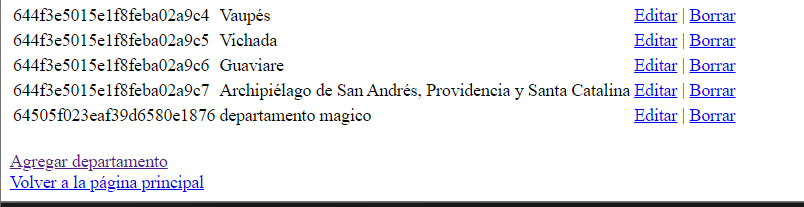
Al darle click se ve mas o menos así, igual podríamos hacer mas pero no vimos necesidad o para el ejemplo practico no, entonces ya lo tenia editado (ya que estábamos probando) ahora cambiamos el nombre

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Y como vemos se corrige correctamente o bueno edita entonces tenemos visualización edición, borrar da problemas, esperemos que vaya funcionando en la medida del caso…

Ahora seguimos con agregar



Damos click como vera ya habíamos probado agregar un departamento nuevo

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

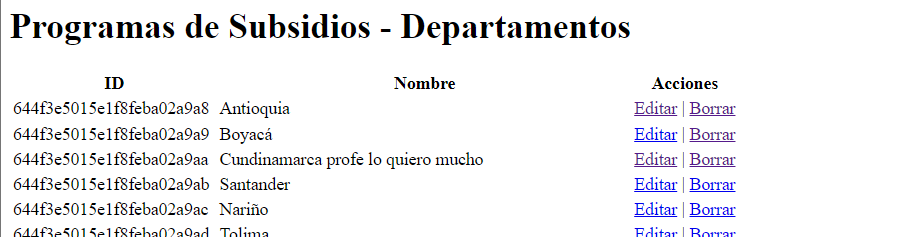
El \_id se genera automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Probamos y woala tenemos esto, ya tenemos casi todo menos elimitar o sea del CRUD nos falta DELETE iremos viendo cómo solucionarlo

Por cierto actualización, logramos descargar el vendor por lo cual el eliminar ya funciona jejeje



Para la prueba borramos el departamento mágico

|  |  |
| --- | --- |
| 64505f023eaf39d6580e1876 | departamento magico |

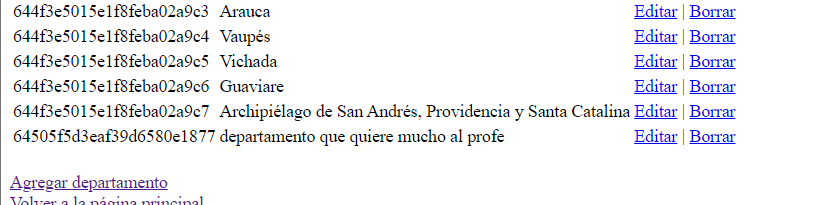
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



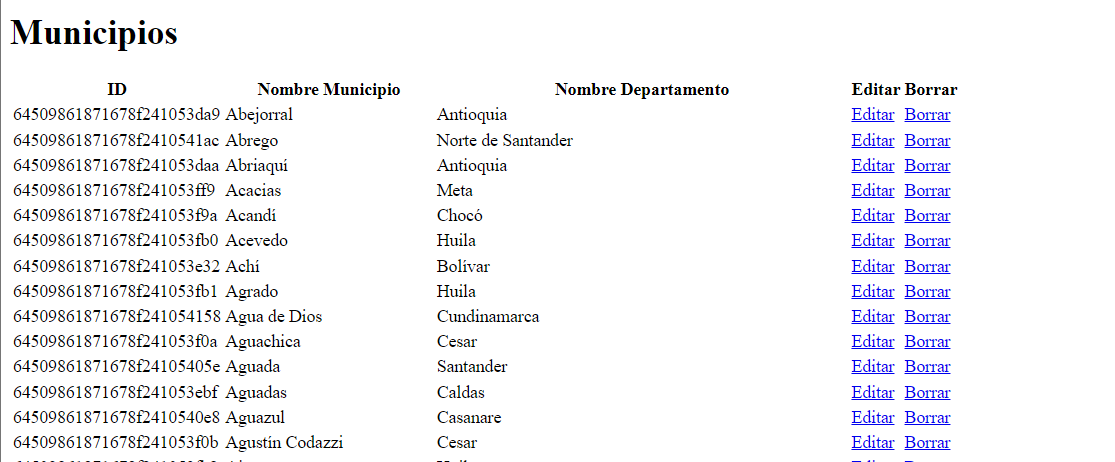
Funciono crud completo de departamento… ahora aguanta ponerle un botón o algo jejej que me devuelva

Se pondrá eventualmente



Eliminado y ya pusimos el botón

Como somos personas de backend



Son las paginas mas feas y horribles de la vida, pero eso será trabajo para los de front

Jajaajajaj

En fin implementando la misma lógica hacemos esto

Probemos el editar con Abejorral

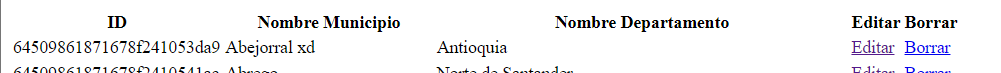
Imagen que contiene Gráfico

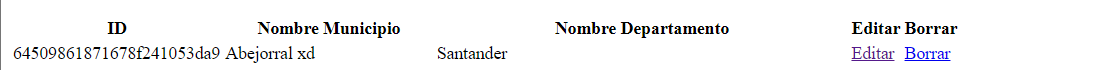
Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y si le podemos asignar distintos departamentos al municipio





Y volvemos normalidad

Gráfico

Descripción generada automáticamente con confianza baja

Seguimos con borrar

|  |  |  |
| --- | --- | --- |
| 64509861871678f241053da9 | Abejorral | Antioquia |

Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Y se borró Abejorral

Agreguemoslo de new



Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

Ahí esta

Nota: la cantidad de errores que salen a la hora de hacer las paginas es enorme, y mas que php no muestra donde te equivocas simplemente no funciona, toca tener mucho control de errores, para poder saber donde esta la equivocación o que no esta sirviendo… T\_T

Tipos de subsidios también ya realizados

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si damos en borrar

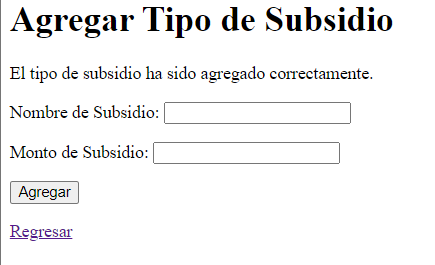
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

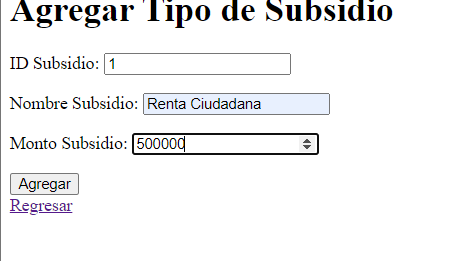
Texto

Descripción generada automáticamente

Andamos es explorando nuevos formatos, y formas de hacer las cosas

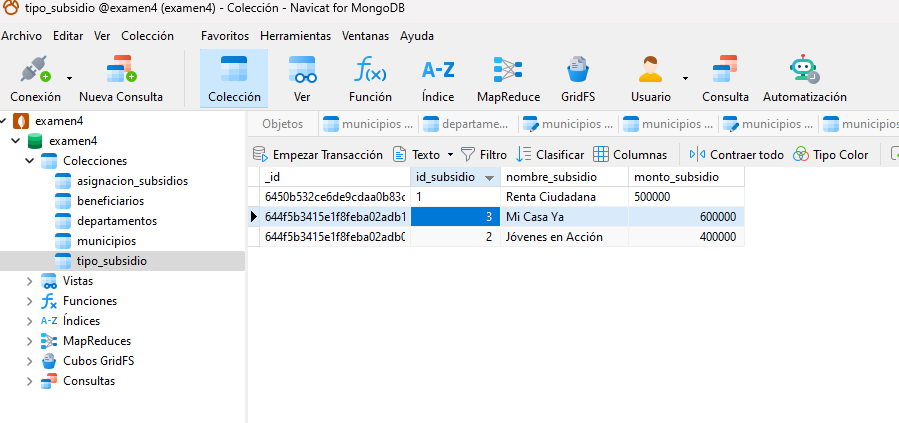


Tuvimos error de calculo

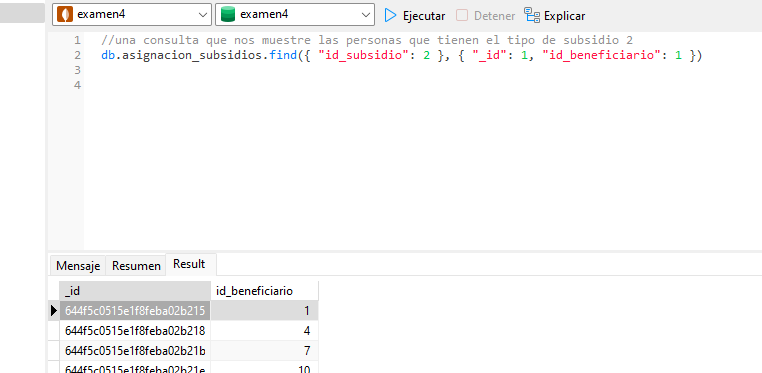


No teníamos el id y generaba un error al volver a los tipos de subsidios

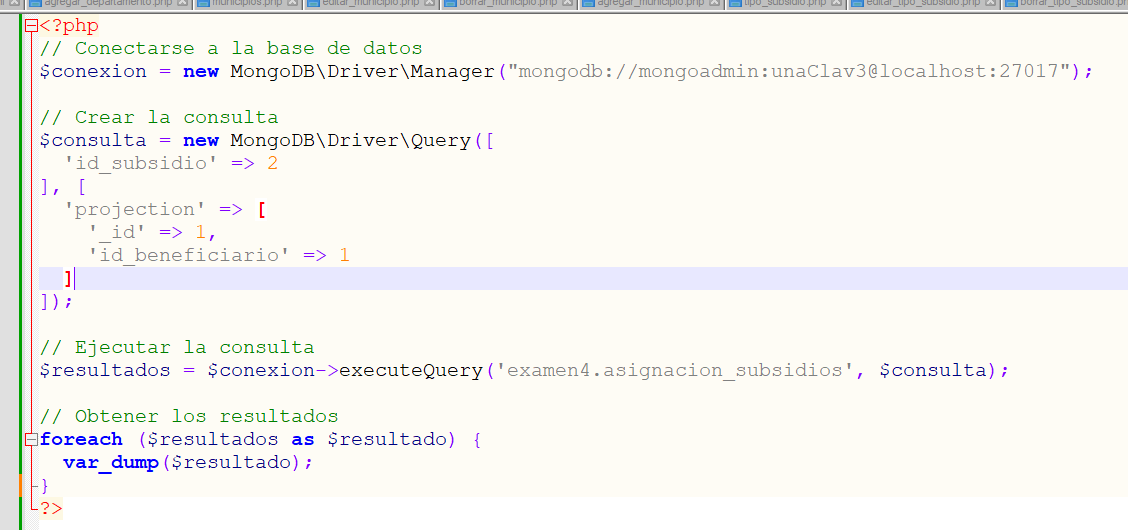
Cabe recordar que todo lo podemos ver reflejado en la base de datos



Algunas consultas en la base de datos



En código seria mas o menos así



Lo mismosolo que con php

Texto, Carta

Descripción generada automáticamentepero saca la respuesta así

**Punto final del taller: Realice una comparación de cada una de las operaciones CRUD, presentando su opinión sobre cual paradigma de base de datos es el más pertinente para este dominio de problema en particular.**

En nuestro proyecto de subsidios gubernamentales, estamos trabajando con datos estructurados, como información de los beneficiarios, los programas de ayuda y los montos asignados. MySQL es altamente eficiente para trabajar con datos estructurados y nos permite realizar consultas complejas para analizar y resumir los datos de manera eficiente (por ese lado mi compañero y yo nos inclinamos a que es mejor tener bases de datos relacionales).

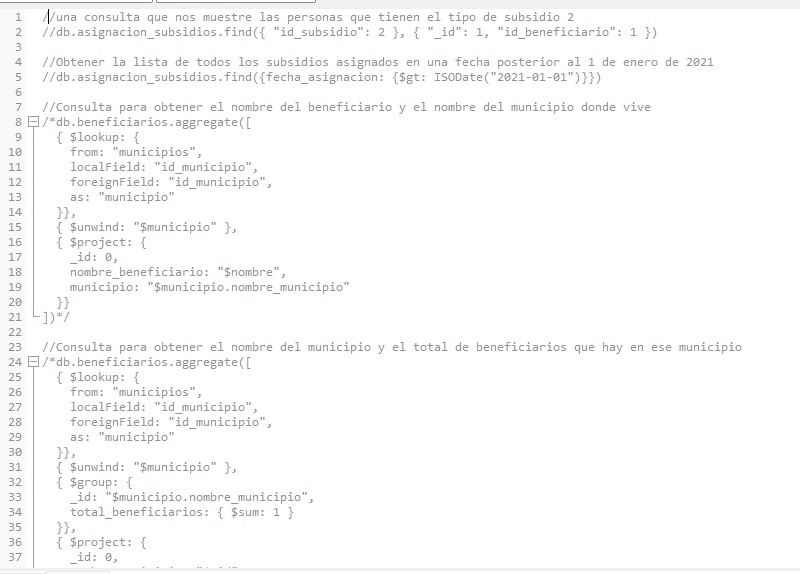
Además, MySQL es muy estable y confiable, lo que es esencial cuando se trabaja con datos sensibles del gobierno. La seguridad de los datos es de suma importancia en este proyecto y MySQL ofrece una gran cantidad de herramientas para garantizar la seguridad y la privacidad de los datos.

Otra ventaja de MySQL es su capacidad para escalar verticalmente y horizontalmente. Es decir, puedes aumentar el poder de procesamiento y almacenamiento en el mismo servidor o agregar más servidores según sea necesario, por lo cual para este proyecto en particular e imaginándolo a gran escala nos parece una opción idónea.

En conclusión, MySQL es una buena opción para este proyecto de subsidios gubernamentales en Colombia debido a su eficiencia en el manejo de datos estructurados, su estabilidad y confiabilidad, su capacidad de escalabilidad y su disponibilidad de recursos en línea. Aunque MongoDB puede ser una opción viable para proyectos con datos no estructurados o semiestructurados, en este caso, MySQL parece ser la mejor opción para manejar los datos sensibles del gobierno y garantizar la seguridad y privacidad de los datos.

Este seria nuestro análisis teniendo en cuenta nuestra poca experiencia, ya que son nuestros primeros acercamientos con las bases de datos relacionales y no relacionales, por lo cual en lo que hemos visto hasta ahora en los cursos ambos paradigmas parecen ser excelentes opciones que brindan cada uno sus ventajas, por el momento y para este problema en especifico creemos que la mejor opciones son las bases de datos relacionales.

**Consultas finales: en el momento nos encontramos realizando pruebas para validar la integridad de los datos. Consultas tipo**

****