**Proyecto QRApp**

Especificación del diseño de la base de datos

**Propiedades del Documento**

|  |  |
| --- | --- |
| **Atributos** | **Contenido** |
| Cliente: | Maria Elena Ruiz Rivera |
| Nombre Proyecto: | UNMSM-QRApp |
| Título del Documento: | Especificación del diseño de la base de datos |
| Versión del Documento: | 1.0 |
| Fecha: | 29/11/21 |
| Autor: | R Ramirez Osorio Miguel Antony |

**Histórico de los cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha de Revisión** | **Modificado por** | **Descripción** |
| 1.0 | 29/11/21 | Ramirez Miguel | Diseño de la base de datos |
|  |  |  |  |

**1. Documento de arquitectura del software**

**1.1. Objetivo del documento:**

Este documento es actualizable de acuerdo a los cambios técnicos de la base de datos que vayan apareciendo.

El objetivo del documento es especificar la base de datos y las tecnologías relacionadas a ellas que se empleará para el sistema QRApp.

**1.2. Tecnología a emplear:**

Con respecto a la base de datos que se utilizara en el proyecto QR-App utilizaremos el gestor de base de datos MongoDB, en este caso gracias MongoDB trabajaremos con colecciones y lo agruparemos en la colección Faculty. Dentro de una colección guardamos distintos datos y todo ello se guardará dentro de un objeto JSON (documentos); en conclusión, son una colección de objetos BSON.

Si tuviéramos que resumir a una la principal característica a destacar de MongoDB, sin duda esta sería la velocidad, que alcanza un balance perfecto entre rendimiento y funcionalidad gracias a su sistema de consulta de contenidos. Pero sus características principales no se limitan solo a esto, MongoDB cuenta, además, con otras que lo posicionan como el preferido de muchos desarrolladores.

Entre las características principales de MongoDB tenemos las siguientes:

* **Indexación**. El concepto de índices en MongoDB es similar al empleado en bases de datos relacionales, con la diferencia de que cualquier campo documentado puede ser indexado y añadir múltiples índices secundarios.
* **Replicación**. Del mismo modo, la replicación es un proceso básico en la gestión de bases de datos. MongoDB soporta el tipo de replicación primario-secundario. De este modo, mientras podemos realizar consultas con el primario, el secundario actúa como réplica de datos en solo lectura a modo copia de seguridad con la particularidad de que los nodos secundarios tienen la habilidad de poder elegir un nuevo primario en caso de que el primario actual deje de responder.
* **Balanceo de carga**. Resulta muy interesante cómo MongoDB puede escalar la carga de trabajo. MongoDB tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balanceo de carga o servicio de replicación de datos, de modo que podemos mantener el sistema funcionando en caso de un fallo del hardware.
* **Almacenamiento de archivos**. Aprovechando la capacidad de MongoDB para el balanceo de carga y la replicación de datos, Mongo puede ser utilizado también como un sistema de archivos. Esta funcionalidad, llamada GridFS e incluida en la distribución oficial, permite manipular archivos y contenido.
* **Ejecución de JavaScript del lado del servidor**. MongoDB tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas. Esto representa una de las diferencias más importantes con respecto a las bases de datos relacionales. Y resulta que no es necesario seguir un esquema. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.

Respecto a las ventajas, tenemos las siguientes:

* Validación de documentos.
* Motores de almacenamiento integrado.
* Menor tiempo de recuperación ante fallos.

Respecto a las desventajas, tenemos las siguientes:

* No es una solución adecuada para aplicaciones con transacciones complejas.
* No tiene un reemplazo para las soluciones de herencia.
* Aún es una tecnología joven.

**1.3. Especificación de la base de datos:**

Las colecciones y los datos a almacenar en la base de datos son las siguientes:

Dentro de la colección de **Faculty** abarca las facultades que existen en la Universidad Nacional mayor de San Marcos, aquí habrá secciones donde se podrá colocar diferentes datos que irán en cada Facultad, tenemos a los siguientes campos:

* idFaculty (este es un identificador único) el tipo de dato es ObjectId.
* history (pequeña historia de la Facultad) el tipo de dato es String.
* contact (contactos de la facultad) el tipo de dato es String.
* undergraduate (datos sobre la información de pregrado) el tipo de dato es String.
* Postgraduate (datos sobre la información de postgrado) el tipo de dato es String.
* Webpage (enlace a la página de la Facultad elegida) el tipo de dato es String.

Presentamos el siguiente diagrama referencial, para ilustrar las relaciones que existirán entre las diversas colecciones a emplear en la base de datos:

