

Instituto Tecnológico de Costa Rica

Algoritmos y estructuras de datos – II

Profesor: Isaac Ramírez

Proyecto 3:

Mi Invincible Library

Estudiantes:

Daniel Acuña Mora – 2017109993

Josué Mata Rojas – 2014081222

Contents

Análisis:..... 3

Investigación: 4

 Bases de datos no relacionales: 4

 Servidores/Clientes locales: 4

 RAID: 4

 Diseño Interfaz: 5

 Implementación: 5

 Arquitectura del proyecto: 6

Trabajo en equipo:..... 6

 Roles:..... 6

 Auto Evaluación y Coevaluación: 7

Análisis:

Como se plantea en documento de instrucciones entregado por el profesor este es un proyecto basado en el concepto de RAID – el cual se explicará más adelante-, este proyecto solicita al estudiantado que se realice una aplicación que funcione mediante una conexión cliente/servidor, para este proyecto el grupo de trabajo asumió que los datos a enviar no son grandes cantidades, el cliente en cuestión debe poseer la función de agregar imágenes y también de cambiar la información de dichas imágenes, esta información será guardada por el server en una base de datos no relacional implementada por cada grupo, el server será el que se encargue de manipular los discos virtuales donde se guardaran las imágenes separadas en archivos, y en discos distintos, el server debe ser capaz de recuperar dichos archivos basándose en la teoría de los RAID.

Investigación:

Para llegar a una conclusión de como se abordará el problema se hizo una breve investigación, la cual abarcaba temas como: bases de datos no relacionales, servidores/clientes locales, RAID, y diseño de interfaz.

Luego de la búsqueda se obtuvo la siguiente información:

Bases de datos no relacionales:

Estas bases se refieren a aquellas que son diferentes al modelo clásico de bases de datos, estas usualmente son conocidas como **NoSQL** ya que no tienen mucho o algún parecido con el modelo de SQL como modelo de consulta.

Servidores/Clientes locales:

En este apartado se buscó sobre conexiones cliente servidor mediante sockets y mediante un REST Api, a lo cual debido a la naturaleza del proyecto, se concluyo que lo más fácil de usar sería sockets, con eso se investigó sobre los protocolos que toman lugar en esta implementación, los cuales son los protocolos UDP y TCP, el más utilizado es el protocolo TCP el cual consiste en abrir un puerto en específico una sola vez, aceptando conexiones singulares cada vez, para hacer un servicio multi-cliente se puede usar TCP pero la comunicación no es tan rápida o efectiva.

RAID:

Como muy bien se especifica en el proyecto era necesaria la implementación de un RAID con discos virtuales para lo cual se investigó sobre dicho tema, resultando así que un RAID es **Redundant Array of Inexpensive Disks**, el cual es una configuración de varios discos físicos que le dan al usuario mayor rendimiento y redundancia, lo cual significa que es un sistema mas tolerante a perdida de datos, ya que dependiendo de la configuración del array, se pueden recuperar archivos perdidos.

Existen varios tipos de RAID, estos son:

1. RAID 0 : Posee stripping pero no redundancia, ofrece el mejor rendimiento, pero no tolerancia a fallos.

2. RAID 1: Consiste en la duplicación de datos entre discos y necesita 2 discos como mínimo,
3. RAID 2: Stripping entre discos, unos discos tienen chequeo de errores. (NO SE USA)
4. RAID 3: Usa stripping y dedica un disco para almacenamiento de las paridades, se puede recuperar datos.
5. RAID 4: Stripping y paridad, no tiene ventajas sobre el Raid 2.
6. RAID 5: Stripping por bloques, con paridad distribuida entre los discos, se pueden recuperar los datos. (Es el que se solicita utilizar en el proyecto)
7. RAID 6: Igual al 5, pero con segunda paridad.
8. RAID 1+0: Mayor rendimiento que el raid 1 y se combina con el raid 0.
9. RAID 0+1: Igual al anterior pero con los datos acomodados de manera distinta.

Diseño Interfaz:

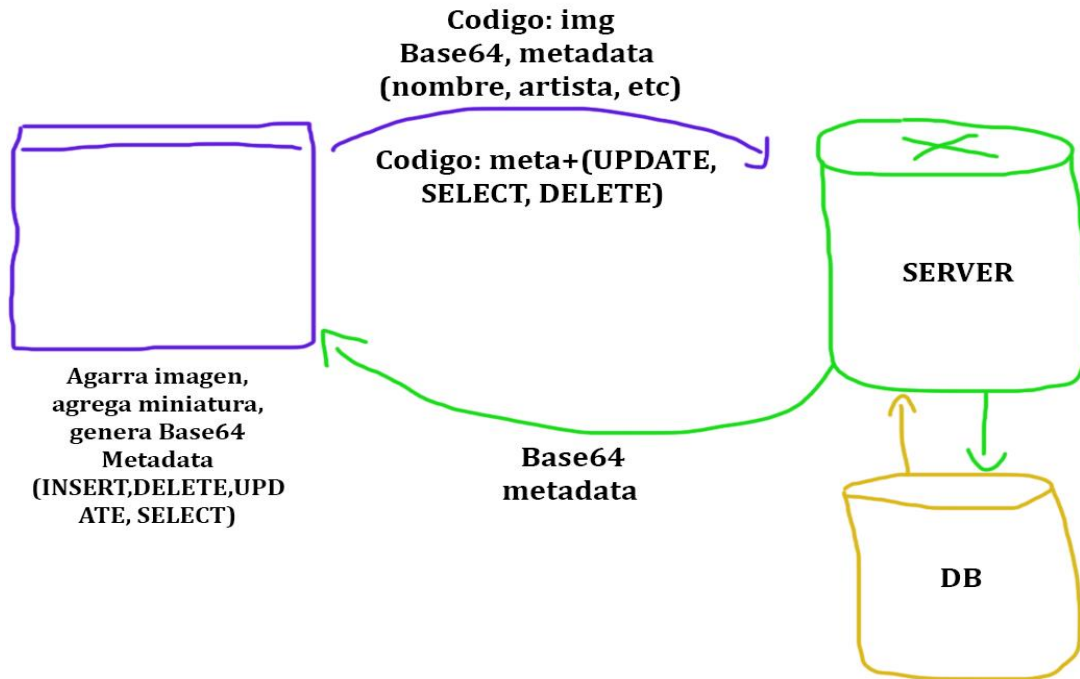
Básicamente se buscó cual sería la mejor herramienta para generar la interfaz a lo cual se llegó que se utilizaría QT Creator, y en si la investigación que se realizó fue como utilizar el IDE en este caso.

Implementación:

Luego de esta investigación se planteó la solución al problema, usando una conexión mediante sockets, el cliente envía un JSON con la información y comando que se desea realizar, el servidor entonces el servidor procede, mediante la información recibida, a descomponer la imagen en partes y a su vez crear un archivo de paridad, cuando esto está hecho se guardaría la información recibida de metadata en la base de datos implementada, la cual será un archivo JSON que, usando el nombre de la imagen como llave, guardará otro archivo json conteniendo toda la información solicitada en la especificación, la estructura de la solución tendrá como base la imagen que se mostrará más abajo.

Para resolver el problema del raid se necesitan crear discos virtuales, donde se estarán guardando las partes de cada imagen que se genere.

Arquitectura del proyecto:



Trabajo en equipo:

En este apartado se solicita brindar información acerca de los integrantes del proyecto.

Roles:

Por cantidad de votos se llegó a la conclusión que Daniel Acuña Mora sería el líder del equipo.

Josué Mata Rojas: Se encargaría de realizar la interfaz del proyecto, y el cliente del server.

Daniel Acuña Mora: Realizaría el server, el RAID y la base de datos.

Auto Evaluación y Coevaluación:

Daniel Acuña Mora		
	Auto evaluación	Coevaluación
Rendimiento	10	10

Josué Mata Rojas		
	Auto evaluación	Coevaluación
Rendimiento	10	10