

Etapa 3 – Ballenitas (G-I)

- Nathalie Alfaro Quesada, B9022I
- Jesús Porras Arguedas, C26007
- Jordan Barquero Araya, C30965

File System

Cache

Programas Representativos

Packet Tracer L2 y L3

UCR - ECCI - Proyecto Integrador Redes de Comunicación de Datos y Sistemas Operativos

```
int FileSystem::add(std::string name, std::string draw) {
    std::unique_lock<std::mutex> generalLock(knowSuperblock);
    name = name.substr(0, maxSize);
    auto it = map.find(name);
    if (it != map.end()) return 1;
    if (draw.size() > sizeDisk - blocksize) return 1;

    Superblock* sb = nullptr;

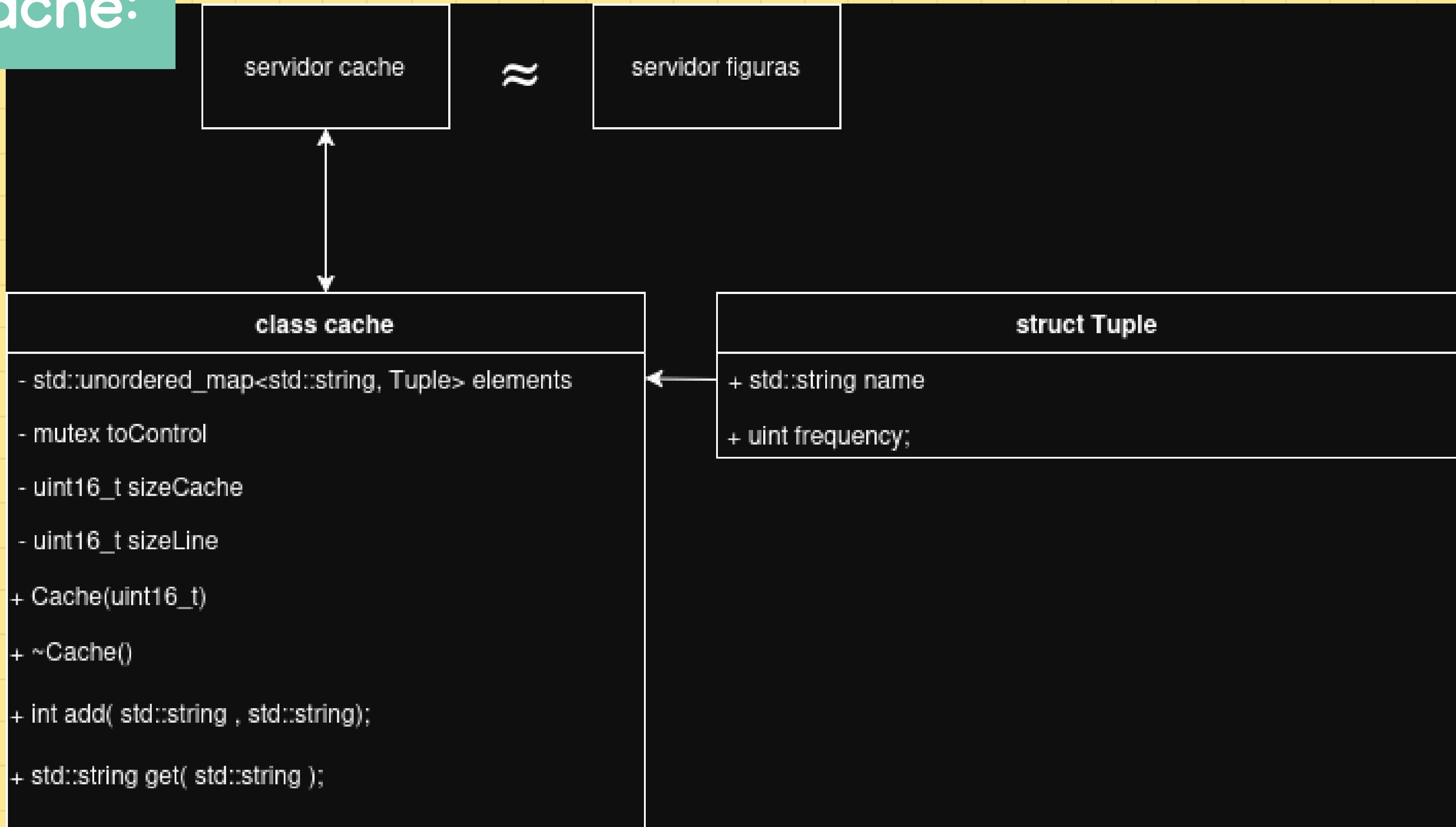
    auto that = std::max_element(
        bloques.begin(), bloques.end(),
        [] (const Superblock* a, const Superblock* b) {
            return a->freeblocks < b->freeblocks;
        }
    );

    if (that != bloques.end() && (*that)->freeblocks * blocksize >= draw.size()) {
        sb = *that;
    } else {
        sb = new Superblock(createDisk(), sizeDisk, blocksize);
        bloques.push_back(sb);
    }

    map[name] = sb;

    {
        std::lock_guard<std::mutex> guard2(sb->diskMutex);
        int i = sb->add_sb(name, draw);
        generalLock.unlock();
        return i;
    }
}
```

cache:



Interaccion servidor-cache :

```
try {
    Socket* s = new Socket('s');
    s->MakeConnection(direccionCache, 8082);
    writeProto(*s, request);

    std::string answerCache = readProto(*s);

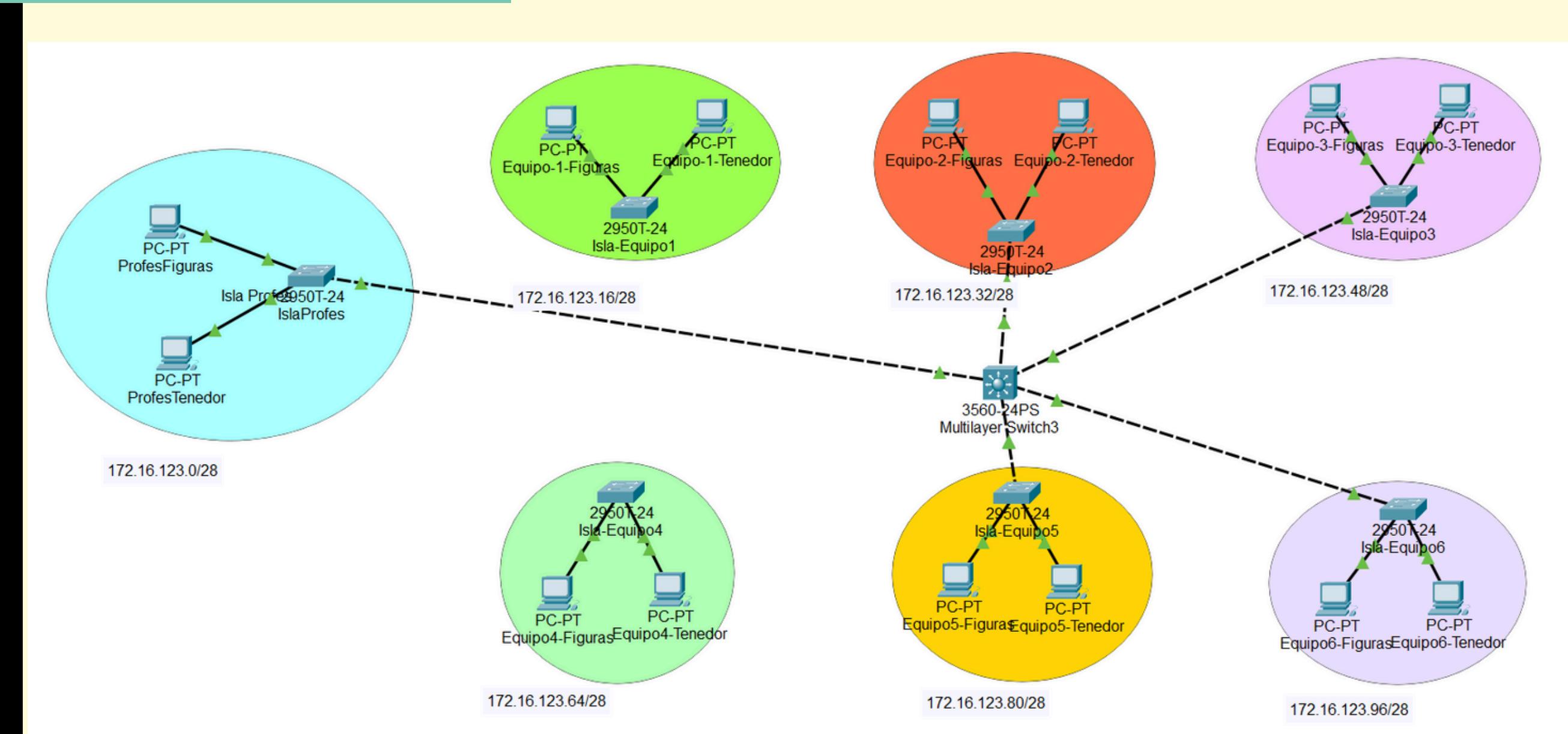
    delete s;

    if(answerCache.find("404 NOT FOUND")!= std::string::npos) {
        content = fs->get(filename);
        if (!content.empty()) {
            response = "\nBEGIN/200 OK/\ncontentLength" + std::to_string(content.size()) + "/\nEND\n" + content;

            answerCache = "ADD \nBEGIN/" + filename + "/\ncontentLength" + std::to_string(content.size()) + "/\nEND\n" + content;
            Socket* s = new Socket('s');
            s->MakeConnection(direccionCache, 8082);
            writeProto(*s, answerCache);

            std::cout << "[DEBUG] Archivo encontrado, respuesta preparada\n";
            delete s;
        } else {
            response = "\nBEGIN/404 NOT FOUND/\nEND\n";
            std::cout << "[DEBUG] Archivo no encontrado\n";
        }
    } else {
        response = answerCache;
    }
} catch(std::exception& e) {
```

Packet Tracer



Formato de Request y Response

Formato General:

COMANDO BEGIN/metadada/END
contenido

Eliminar figura:

DELETE BEGIN/cat.txt/END

Obtener figura:

GET BEGIN/cat.txt/END

Apagar/Encender:

QUIT/CONNCT BEGIN//IP/END

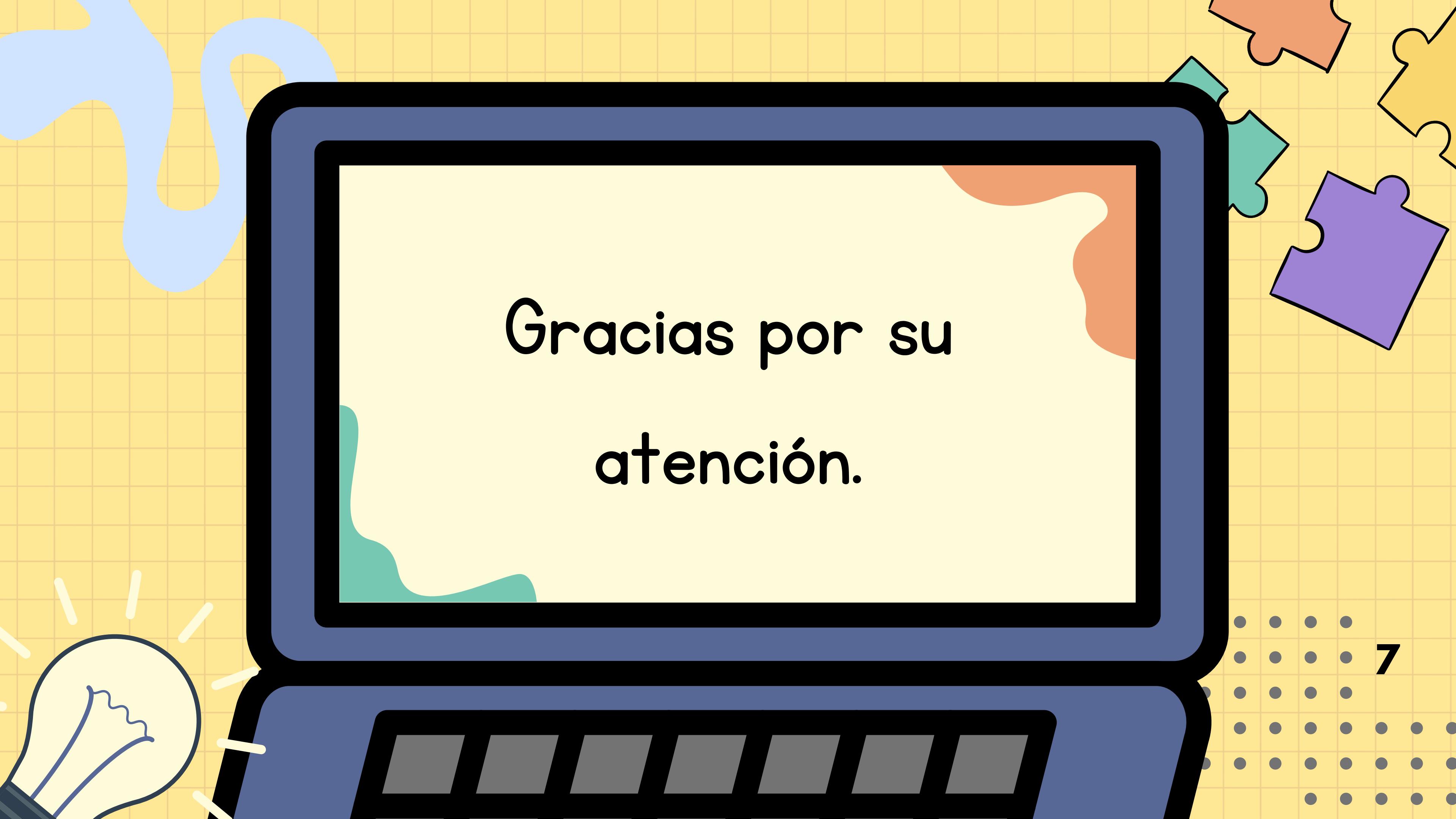
Agregar figura:

ADD BEGIN/cat.txt/contentLength100/END

<figura>

Listar figuras:

LIST BEGIN//END



**Gracias por su
atención.**