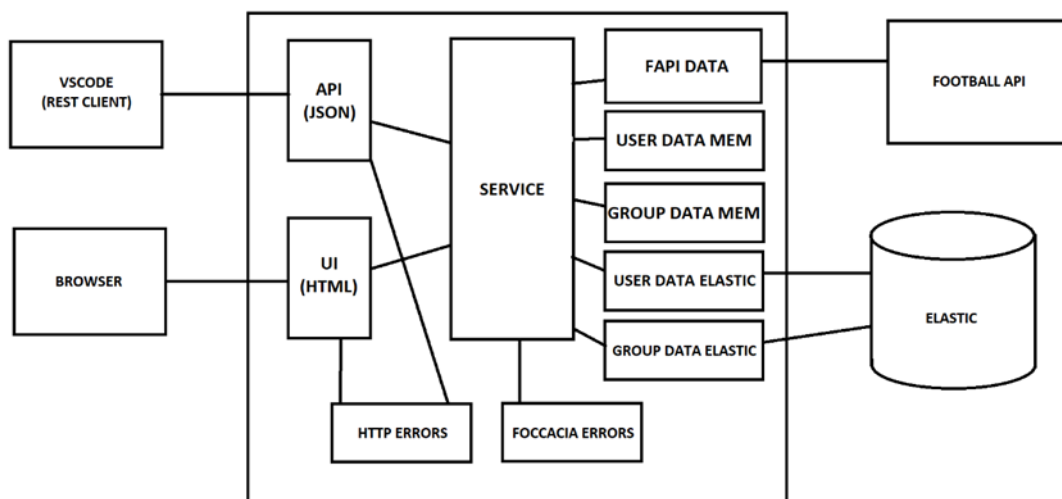


# Relatório de Programação na Internet

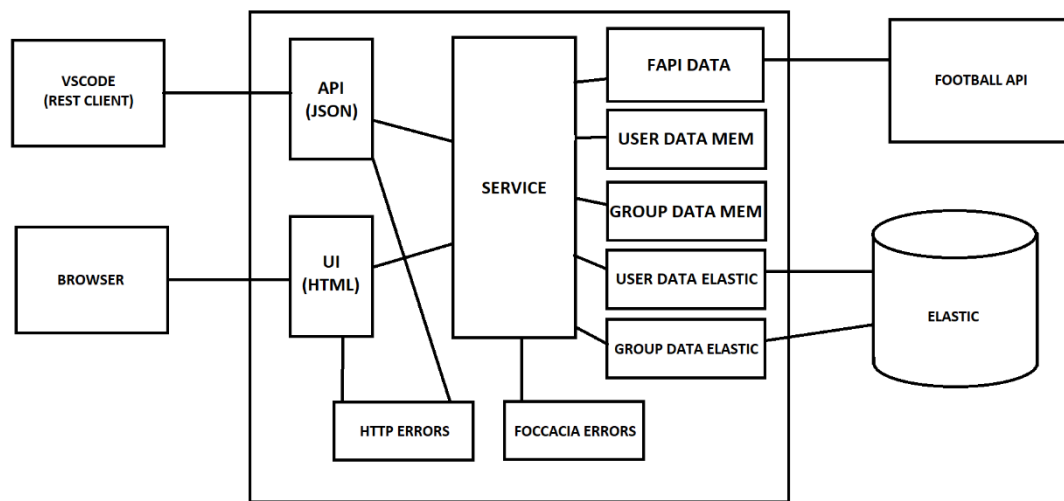
## MONOLITO



Realizado por:

Diogo Lima – 50879 Lucas Silva – 51011 Mihail Arcus – 5087

# MONOLITO



## Estrutura da aplicação

A imagem acima mostra um sistema informático que foi implementado durante este semestre. Utilizamos uma extensão do vscode para fazer pedidos à API do nosso sistema que retorna JSON e utilizamos também o browser para fazer pedidos que retorna HTML.

Tanto o módulo da API como da UI tem uma dependência que é o serviço, neste contém toda a lógica do nosso domínio e em caso de erros faz uso de um outro módulo que é o foccacia errors e propaga-os para a API e a UI, estes dois últimos módulos a partir dos erros que recebem do serviço fazem uso do módulo HTTP errors para converter o erro do domínio num erro HTTP e mostrar ao cliente.

O nosso serviço ainda faz uso do módulo fapi data que por sua vez faz uso de uma API externa (football api) que retorna para o nosso sistema os dados das equipas, ligas, etc. Para além deste módulo, ainda existem outros quatro módulos, dois deles armazenam e obtêm dados a partir da memória e outros dois que são iguais a estes últimos, mas os dados são armazenados e obtidos a partir de uma base de dados não relacional (elastic search). Um dos módulos é responsável pelos dados do utilizador e o outro pelos dados dos grupos.

## Sistema de armazenamento

Como referido anteriormente, os dados são armazenados em memória ou no elastic search, neste último caso existem dois índices que são os **groups** e **users**.

No índice groups armazenamos os grupos e este documento tem as propriedades name (string), description (string), userId (string) e teams (array de teams), cada team possui as propriedades id, season, leagueName, leagueId, name, stadium, image.

No índice users armazenamos todas as informações do utilizador e este possui as propriedades userName (string), password (string cifrada com bcrypt) e token (string).

De modo a relacionarmos os grupos com os utilizadores, optamos por armazenar em grupo o id do utilizador.

## Descrição do mapeamento entre os documentos do elastic search e os objetos da aplicação web

Este mapeamento é praticamente direto, dado que as propriedades de cada objeto do nosso domínio são salvas na base de dados, apenas o id é gerado automaticamente pela base de dados e consequentemente na resposta da base de dados temos que ir buscar o id num local diferente da resposta.

## Instruções para correr a aplicação web

É necessário não ter nenhuma aplicação no porto 9000 a correr.

É necessário ter o elastic search no porto 9200 e possuir os índices users e groups criados e sem dados salvos.

No diretório da aplicação é necessário executar o comando “npm install” para instalar todas as dependências necessárias da nossa aplicação.

Por fim, para colocar o servidor a funcionar é necessario digitar o comando “node src/foccacia-server.js”.

Para correr os testes unitários e de integração, é necessario executar o comando “npm run test”.