

Proposta de Arquitetura

Bruna do Espirito Santo Sousa
Felipe Moreira da Silva
Heitor Silva Rodrigues
Layane Grazielle Souza Dias
Pedro Ivo Santana Melo
Pedro Paulo Oliveira Lopes

Goiânia

11 de janeiro de 2023

HISTÓRICO DE REVISÕES			
Revisão	Data	Descrição	Autor
01	11/01/2023	Elaboração inicial do documento.	Grupo

Sumário

1. Introdução	2
1.1 Finalidade	2
1.2 Escopo	3
1.3 Definições, Acrônimos e Abreviações	3
1.4 Referências	3
1.5 Visão Geral	3
2. Contexto da Arquitetura	4
2.1 Funcionalidades e Restrições Arquiteturais	4
2.2 Atributos de Qualidades Prioritários	4
2.3 Tecnologias	4
3. Representação da Arquitetura	4
4. Ponto de vista dos Casos de Uso	5
4.1 Descrição	5
4.2 Visão de Casos de Uso	6

1. Introdução

1.1 Finalidade

A principal finalidade deste documento é definir os aspectos essenciais da Arquitetura de Software, sendo direcionado aos stakeholders do projeto, possuindo grande foco para os Desenvolvedores e para a Equipe de implantação.

1.2 Escopo

O “**Sistema de Apoio à Geotécnica**” tem como objetivo gerar uma modelagem de solo a partir de uma rota definida por um usuário. Para isso, utiliza das amostras colhidas e cadastradas para gerar as rotas de forma personalizada de acordo com as propriedades dos solos analisados.

1.3 Definições, Acrônimos e Abreviações

Id.: Identificador.

Software: Conjunto de documentações, guias, metodologias, processos, códigos e ferramentas para a solução de um problema.

Stakeholder: Indivíduo, grupo ou organização que possua interesse no Sistema.

Visão Arquitetural: Produto resultante da interpretação de um Stakeholder do sistema.

Arquitetura de Software: Forma como os componentes são agrupados com o objetivo de construir um software ou sistema.

Ponto de Vista Arquitetural: Produto resultante da execução de uma Visão Arquitetural.

Python: Linguagem de programação de alto nível, de propósito geral, interpretada, de sintaxe concisa e clara.

1.4 Referências

Id.	Nome do Artefato
AAS_1	Especificação de Requisitos

1.5 Visão Geral

De maneira simples, o documento visa descrever a estrutura geral do sistema a ser desenvolvido, incluindo etapas de organização lógica e física, bem como de componentes e suas relações e ainda acerca de possíveis interfaces.

As principais decisões de projeto, restrições e metodologias adotadas para o desenvolvimento do projeto estão descritas e detalhadas. Além disso, vale ressaltar que esse Projeto Arquitetural se trata de um processo contínuo, sendo assim é suscetível a possíveis melhorias futuras.

2. Contexto da Arquitetura

2.1 Funcionalidades e Restrições Arquiteturais

Id.	Tipo	Id. do Documento de Requisitos
RAS_1	Requisito Não-Funcional	RNF[002]
RAS_2	Requisito Não-Funcional	RNF[004]
RAS_3	Requisito Não-Funcional	RNF[005]
RAS_4	Regra de Negócio	RN[001]
RAS_5	Requisito Funcional	RF-01
RAS_6	Requisito Funcional	RF-05
RAS_7	Requisito Funcional	RF-07
RAS_8	Requisito Não-Funcional	RF-01
RAS_9	Requisito Não-Funcional	RNF-04
RAS_10	Requisito Não-Funcional	RNF-05

2.2 Atributos de Qualidades Prioritários

Por se tratar de um sistema que recebe constantes cadastros e pesquisas, é importante manter os aspectos de confiabilidade dentro da arquitetura escolhida, garantindo que as requisições serão devidamente processadas e que os dados serão gravados sem duplicidade e apenas por usuários autorizados.

Ainda sob o ponto de vista da qualidade, é necessário incluir na arquitetura elementos que contribuam com a manutenibilidade, de forma a garantir o máximo de independência entre as camadas e componentes, para que as futuras manutenções sejam realizadas com viabilidade.

Tendo em vista que as principais funcionalidades do sistema são permitir o cadastro das amostras e consultá-las no mapa de forma interativa e simplificada, os esforços para tornar a usabilidade mais evidente devem ser consideráveis.

2.3 Tecnologias

Linguagem de programação back-end: Python

Linguagens front-end: HTML, CSS e Javascript

Banco de dados: Postgres

Hospedagem do site: Github

3. Representação da Arquitetura

A arquitetura do software a ser desenvolvido é baseada em uma arquitetura de camadas e cliente-servidor. A camada para área de interação com o usuário consiste em uma interface web, que permite a fácil utilização do sistema. A camada de aplicação consiste em um conjunto de serviços web que processam as solicitações dos usuários e interagem com a camada de dados. A camada de dados consiste em um banco de dados relacional, fornecido pelo cliente, que armazena as informações sobre as medições no solo. As camadas se comunicam através de um conjunto de interfaces pré-definidas.

Além disso, a utilização da arquitetura cliente-servidor divide a aplicação/serviço nos componentes clientes e servidores. Os clientes são os dispositivos, nesse caso os computadores, que realizam as requisições e também fornecem aos usuários uma

interface. Por outro lado, os servidores recebem tais requisições, para realizar o seu processamento e fornecer as devidas respostas de volta aos clientes.

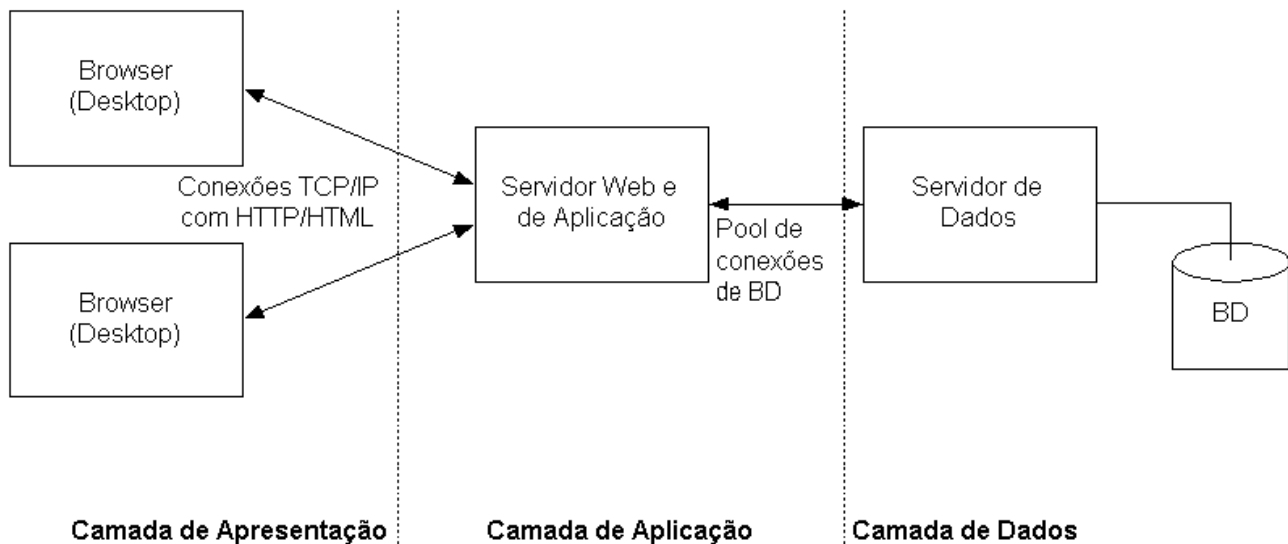


Figura 1: modelo em camadas e cliente-servidor

Com a arquitetura Cliente/Servidor, garantimos que cada instância de um cliente enviará as requisições para um servidor e aguardará a resposta. Através da decomposição em camadas independentes, atendemos também aos requisitos de manutenibilidade, por ser mais fácil executar manutenções ou substituições de servidores sem afetar o serviço. Além disso, garantimos os critérios de segurança levantados, pois as requisições serão centralizadas apenas no servidor, controlando o acesso dos recursos e permissões.

Com os protocolos de transporte e aplicações de rede dessa arquitetura, garantimos a transferência de dados de forma confiável entre cliente e servidor. Também garantimos uma maior adaptabilidade a diferentes ambientes de clientes. Devemos nos atentar à sobrecarga do servidor, que pode gerar conflito com os requisitos de performance estabelecidos.

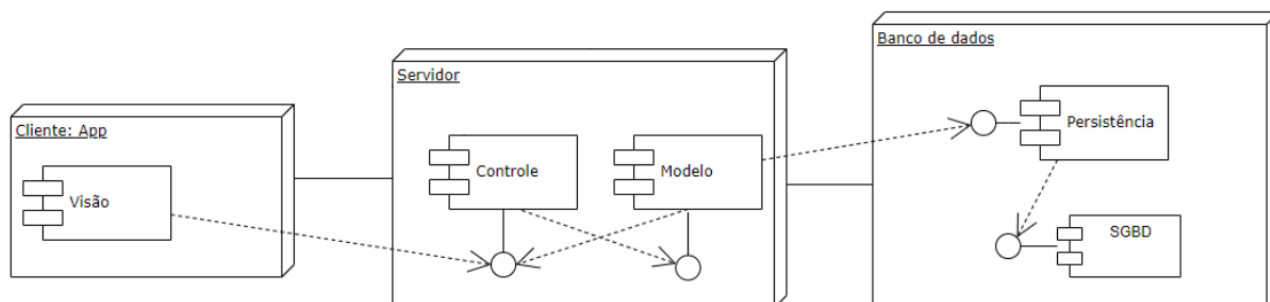


Figura 2: diagrama de componentes

4. Ponto de vista dos Casos de Uso

4.1 Descrição

Para fornecer uma base para o planejamento da arquitetura e de todos os outros artefatos que serão gerados durante o ciclo de vida do software, é gerada, na análise de requisitos, uma visão chamada visão de casos de uso. Só existe uma visão de casos de uso para cada sistema. Ela ilustra os casos de uso e cenários que englobam o comportamento, as classes e riscos técnicos significativos do ponto de vista da arquitetura. A visão de casos de uso é refinada e considerada inicialmente em cada iteração do ciclo de vida do software.

4.2 Visão de Casos de Uso

Os requisitos funcionais foram utilizados para compor os casos de uso, resumindo as principais funcionalidades do sistema nos diagramas abaixo:

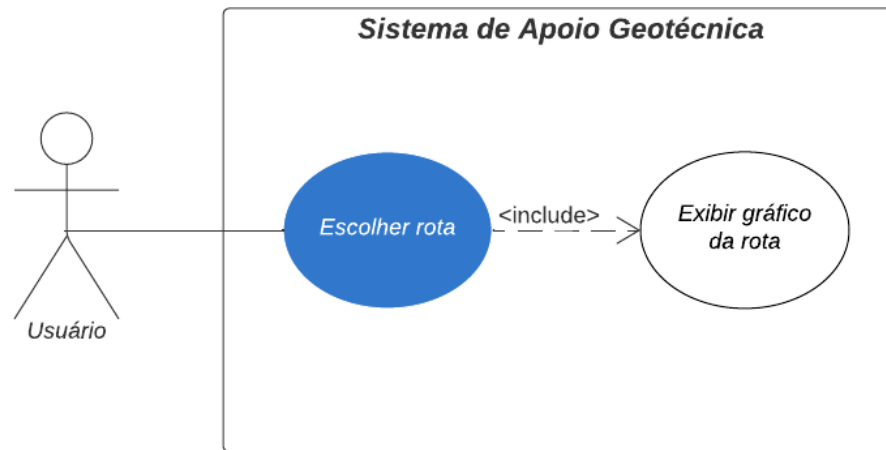


Figura 3: caso de uso da visualização gráfica

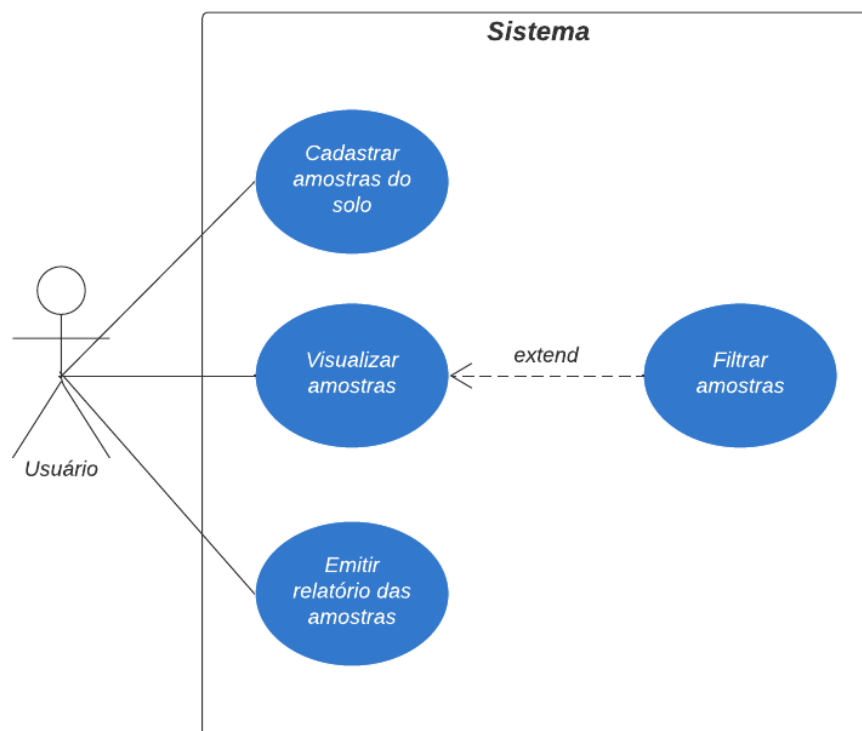


Figura 4: caso de uso das amostras e relatórios