

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído

Projeto Integrado

Relatório Técnico

Gestão de Conhecimento

Ednir Bernardino de Oliveira Araujo

Belo Horizonte
Agosto 2022.

Projeto Integrado – Arquitetura de Software Distribuído

Sumário

Projeto Integrado – Arquitetura de Software Distribuído	2
1. Introdução	3
2. Cronograma do Trabalho	5
3. Especificação Arquitetural da solução	6
3.1 Restrições Arquiteturais	6
3.2 Requisitos Funcionais	6
3.3 Requisitos Não-funcionais	7
3.4 Mecanismos Arquiteturais	7
4. Modelagem Arquitetural	8
4.1 Diagrama de Contexto	8
4.2 Diagrama de Container	9
4.3 Diagrama de Componentes	10
5. Prova de Conceito (PoC)	11
5.1 Integrações entre Componentes	12
5.2 Código da Aplicação	12
Referências	14

1. Introdução

Compartilhar conhecimento, informação ou experiências, sempre foi inerente ao ser humano durante toda sua trajetória, isso foi, é, e sempre será vital para nossa existência e evolução. Nos dias atuais esse conhecimento pode ser transmitido de inúmeras formas, uma delas que já vem sendo utilizada a séculos, é o formato de “aulas”, caracterizadas com um “tema” específico, com uma quantidade de “alunos”, um “professor”, com um período específico, recebendo o nome de “Turmas”. Esse formato é utilizado no sistema de ensino nacional, em treinamentos internos de instituições, em com comunidades para compartilhar conhecimento, entro outros. O foco do “Gestão de conhecimento” é descomplicar o registro das informações que compões a gestão das “Turmas”.

Atualmente as empresas de pequeno porte ou grupos comunitários, geralmente tem treinamentos para seus membros, e esse geralmente são divididos em turmas as quais tem o instrutor ou professor os alunos, a quantidade de horas que é necessária para transmitir o conhecimento, essas informações em sua maioria é controlada por uma lista de presença, anotações de conteúdo, e algumas observações, todas sendo anotadas em folha de papel, em um próximo passo é as informação são inseridas em uma planilha eletrônica, que fica como um banco de dados para analisar as informações. Esse processo é muito oneroso para os envolvidos, pois tem várias etapas mecânicas, como imprimir a lista de alunos, realizar o preenchimento desta lista, incluir as informações na planilha eletrônica, tratar os dados para ter um relatório, e a dificuldade em centralizar a informação.

Visando descomplicar esse fluxo desanimador e robótico de registro de informação sobre os treinamentos, e sabendo que no Brasil temos um mais de um celular por habitante.

Em relação aos smartphones, a estimativa é de 242 milhões de unidades em uso no Brasil em junho deste ano, ou seja, mais de um por habitante (Portaria FGV, 2021).

Sabendo do recurso tecnológico que a maioria tem acesso, com a necessidade de vários grupos que precisam fazer esse controle, mas só dispõem de planilhas eletrônicas e algumas folhar de papel, o Gestão de Conhecimento, vem de encontro para sanar essa dor, que é frequente em várias esferas, com o próprio smartphone, pode se fazer o processo todo, definida os temas, as turmas, apurando as presenças, acesso a relatórios, e mais, tudo de maneira fácil e rápida, deixando assim as informações disponíveis para quem interessar, um processo mais consciente, menos burocrático, e com economia de recursos e tempo dos envolvidos.

O objetivo deste trabalho é apresentar a descrição do projeto arquitetural de uma aplicação que gerencia registros, com o foco em atender os objetivos:

- Um sistema com a competência de gerenciamento de turmas, registrando as informações necessárias para controle.
- Uma arquitetura coerente ao público-alvo, visando soluções onde os dispositivos mais simples sejam capazes de utilizar.
- Uma utilização simples e eficiente nas atividades realizadas.

2. Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
12/04/2022	12/04/2022	1. Análise dos requisitos para o trabalho, Cronograma	BackLog de atividades
23/04/2022	23/04/2022	2. Definição do tema de atuação	Relação possíveis atuações
23/04/2022	23/04/2022	3. Análise da arquitetura, Restrições arquiteturais	Modelo de arquitetura
28/04/2022	28/04/2022	4. Definição de requisitos funcionais,	Relação de funcionalidades
28/04/2022	28/04/2022	5. Definição de requisitos Não-funcionais	Relação de requisitos Não-funcionais
28/04/2022	28/04/2022	6. Definição de mecanismos arquiteturais	Relação de mecanismos arquiteturais
06/08/2022	06/08/2022	7. Diagrama de Contexto	Criação do diagrama de contexto
06/08/2022	06/08/2022	8. Vídeo 1	Criação do Vídeo
14/08/2022	14/08/2022	9. Revisão das atividades 1 Etapa	Análise das atividades realizadas
14/08/2022	14/08/2022	10. Envio dos arquivos 1 Etapa	Envio dos arquivos para o GitHub
14/08/2022	14/07/2022	11. Desenvolvimento inicial da aplicação	Criação da estrutura
27/08/2022	27/08/2022	12. Diagrama de Container da aplicação	Criação do diagrama de container
27/08/2022	27/08/2022	13. Diagrama de Componentes	Criação do diagrama dos componentes
03/09/2022	03/09/2022	14. Integrações entre componentes	Definição dos componentes
10/09/2022	10/09/2022	15. Código da aplicação	Início da codificação
17/09/2022	17/09/2022	16. Revisão das atividades 2 Etapa	Análise das atividades realizadas
10/10/2022	10/10/2022	17. Envio dos arquivos 2 Etapa	Envio dos arquivos para o GitHub
15/10/2022	15/10/2022	18. Análise das abordagens arquiteturais	Analisar e relata abordagens arquiteturais
15/10/2022	15/10/2022	19. Cenários	Registrar cenário
21/10/2022	21/10/2022	20. Evidências da avaliação	Registrar análise das evidências
21/10/2022	21/10/2022	21. Resultados obtidos	Registrar análise dos resultados
29/10/2022	29/10/2022	22. Avaliação crítica dos resultados	Registrar avaliação dos resultados
12/10/2022	12/10/2022	17. Conclusão	Registrar conclusão
29/10/2022	29/10/2022	18. Vídeo de apresentação final	Criação do Vídeo
26/11/2022	26/11/2022	19. Revisão das atividades 3 Etapa	Análise das atividades realizadas
26/11/2022	26/11/2022	20. Envio dos arquivos 3 Etapa	Envio dos arquivos para o GitHub

Obs: acrescente ou retire linhas, caso seja necessário.

3. Especificação Arquitetural da solução

3.1 Restrições Arquiteturais

ID	Descrição
R01	O servidor deve ter npm versão 8.11.0
R02	O desenvolvimento do front-end deve ser desenvolvido em Vue versão 3.2.36
R03	Data Base deve ser utilizado o Realtime Database, firebase versão 9.9.2
R04	O componente para auxiliar no desenvolvido deve se o Vuex versão 4.0.2

3.2 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	Não deve ser permitido a exclusão de uma tarefa cadastrada	B	A
RF02	Toda tarefa é obrigatória a inclusão de um professor	M	A
RF03	Deve possibilitar, cadastrar várias tarefas ao mesmo professor	B	A
RF04	Deve possibilitar, cadastrar somente um professor por tarefa	M	M
RF05	Deve possibilitar, a edição de uma tarefa	A	B
RF06	Deve possibilitar, finalizar tarefa a qualquer momento	B	B
RF07	Deve possibilitar, exibir o nome o professor na tarefa	M	A
RF08	Deve possibilitar, excluir um professor	A	A
RF09	O professor excluir não pode sair de uma tarefa já registrada	M	A
RF10	O campo professor de ter mínimo de 3 caracteres obrigatórios	M	B
RF11	O campo tarefa de ter mínimo de 3 caracteres obrigatórios	M	B
RF12	O botão Início sempre deve estar ativo após a criação de uma tarefa	B	M
RF13	O botão Fim deve ficar ativo somente se o cronômetro estiver contando o tempo	M	M
RF14	O cronômetro de sempre iniciar zerado	B	A
RF15	Não deve ser permitido cadastrar professor sem nome	M	M
RF16	Deve possibilitar, incluir número no nome do professor	B	M
RF17	Deve possibilitar, incluir número no nome da tarefa	B	M
RF18	Deve ser desativado a função do botão de Fim, assim que ele for clicado	B	M
RF19	Deve ser desativado a função do botão de Início, assim que ele for clicado	B	M
RF20	Deve ser permitido cadastrar tarefa com o mesmo nome	B	B
RF21	Deve ser permitido cadastrar professor com o mesmo nome	B	B
RF22	Deve ser possível criar mais de 1000 tarefas	A	M

*B=Baixa, M=Média, A=Alta.

Obs: acrescente quantas linhas forem necessárias.

3.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve ser apresentar disponibilidade 24 X 7 X 365	A
RNF02	Deve ser acessível de dispositivos moveis	A
RNF03	Necessário ter internet para utilizar aplicação	A
RNF04	Deve existir ser realizado um BackUp periódico dos dados	M
RNF05	Deve ser compatível com os principais navegadores do mercado Chrome, Edge, Firefox, Safari, Opera.	M
RNF06	Deve existir versionamento das versões implementadas	M

Obs: acrescente quantas linhas forem necessárias.

3.4 *Mecanismos Arquiteturais*

Análise	Design	Implementação
Controle de versão	Versionamento do código fonte da aplicação	Git
Front end	Single Page Application	Vue
Persistência	Banco de dados não relaciona	Firebase Realtime Database

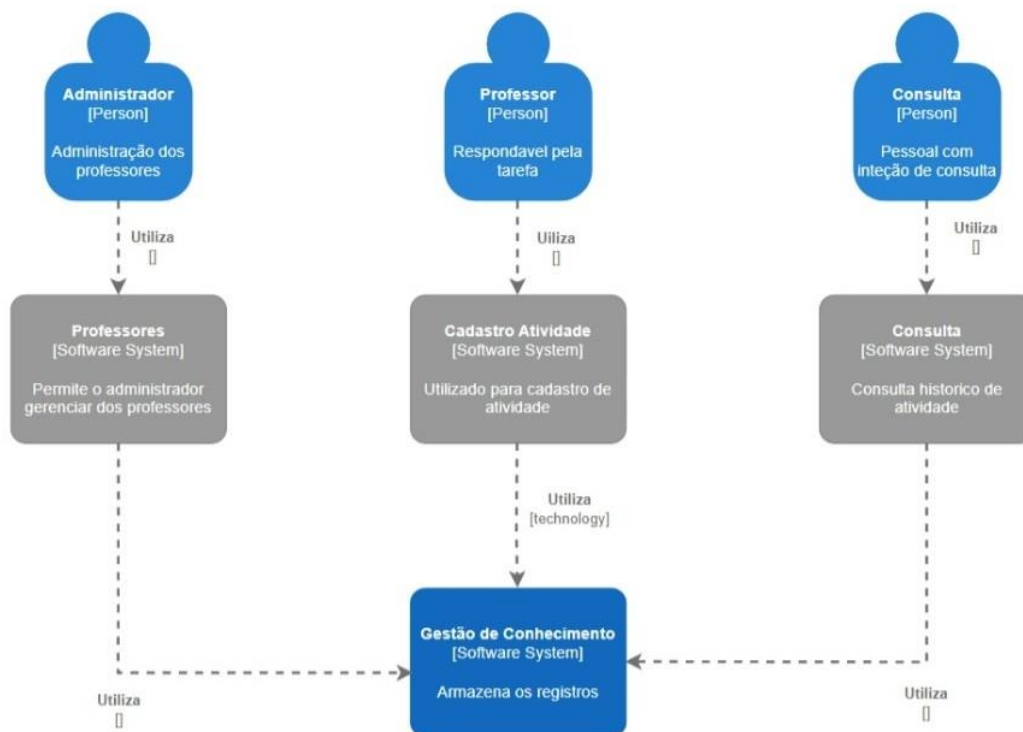
4. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da prova de conceito da aplicação que gerencia registros Gestão de Conhecimento na seção 5.

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Mais informações a respeito podem ser encontradas nos links: <https://c4model.com/> e <https://www.infoq.com/br/articles/C4-architecture-model/>. Dos quatro níveis que compõem o modelo C4 três serão apresentados aqui e somente o código será apresentado na próxima seção (5).

4.1 Diagrama de Contexto

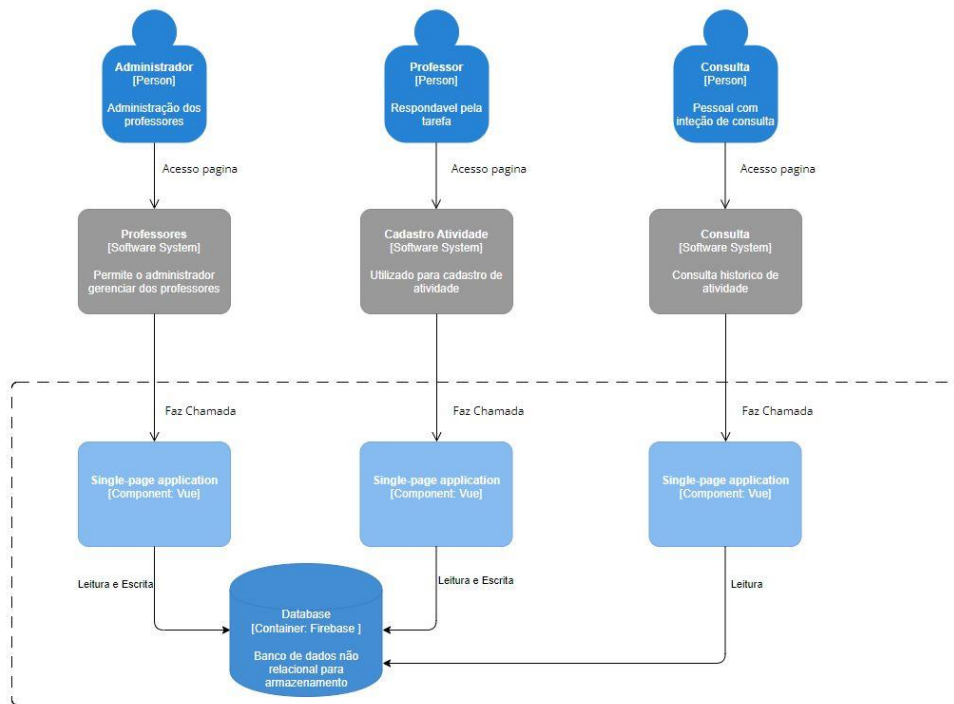
Figura 1 - Visão Geral da Solução



A Figura 1 mostra o contexto em que a aplicação Gestão de Conhecimento será inserida, demonstrando seus principais módulos e integrações.

4.2 Diagrama de Container

Figura 2 – Diagrama de container

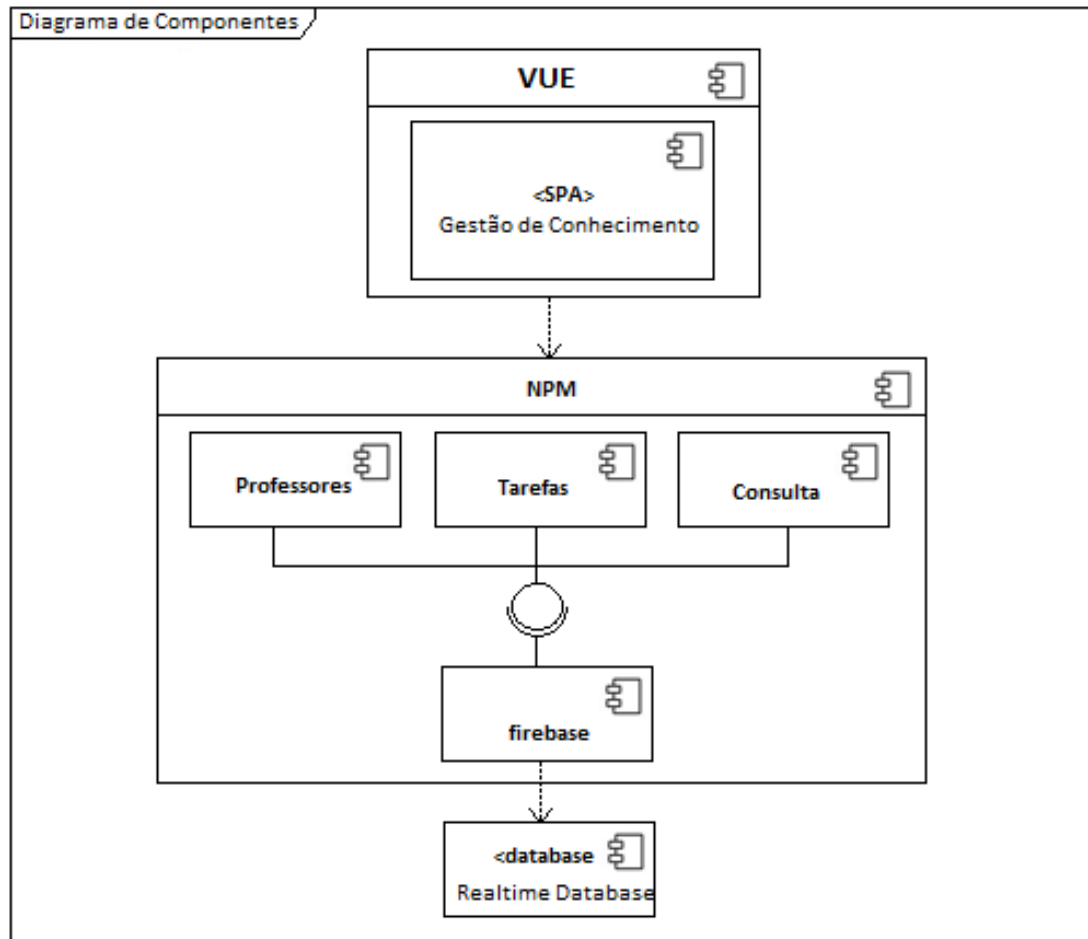


A Figura 2 diagrama de container apresenta containers da aplicação e sua distribuição, visando do banco de dados com principal responsável pelos registros e disponibilização das tarefas.

Futuras melhorias podem ser aplicadas com a criação de novas single page application, podendo utilizar validados de testes com AB com trechos de acessos.

4.3 Diagrama de Componentes

Figura 3 – Diagrama de componentes



A Figura 3 apresenta os componentes da aplicação, esse componentes pode ser detalhados:

- Vue – Vue.js é um framework JavaScript.
- Gestão de Conhecimento – SPA Aplicativo de página única (Single-page application).
- NPM – Servidor Node.JS para as transações.
- Professores – Responsável pelo registro e apresentação dos professores.
- Tarefas – Responsável pelas novas tarefas, controle do tempo de execução e apresentação das tarefas antigas.
- Consulta – Responsável por apresentar a relação de todas as tarefas já realizadas.

- Firebase – O Firebase é um conjunto de serviços de hospedagem para qualquer tipo de aplicativo.
- Realtime Database – Um banco de dados em tempo real é um sistema de banco de dados que usa processamento em tempo real para lidar com cargas de trabalho sendo um banco de dados não relacional.

5. Prova de Conceito (PoC)

Nesta prova de conceito foi utilizado como guia o baixo custo e alta disponibilidade, para isso optei por uma solução simples utilizando o VUE, linguagem apresentada em uma das matérias, para aumentar meu conhecimento o com esse framework, seguindo a mesma ideia a utilização do Node como servidor também colocando em pratica um pouco do que foi apresentado, para armazenando dos dados optei pode Firebase Realtime Database, uma ferramenta que tem alta flexibilidade e fácil entendimento e de alta disponibilidade, para publicação nesta POC, optei pela Vercel, uma plataforma com foco em hospedagem de aplicações, e que tem integração com o GitHub, assim já é possível, também apresentar algo com os primeiros passos para uma futura esteira CI/CD.

Apresentação dessa simples aplicação está com o nome de Vídeo 2, foi entregue junto com este documento.

5.1 Integrações entre Componentes

Foi desenvolvido Wireframe das telas utilizando o figma, segue link:

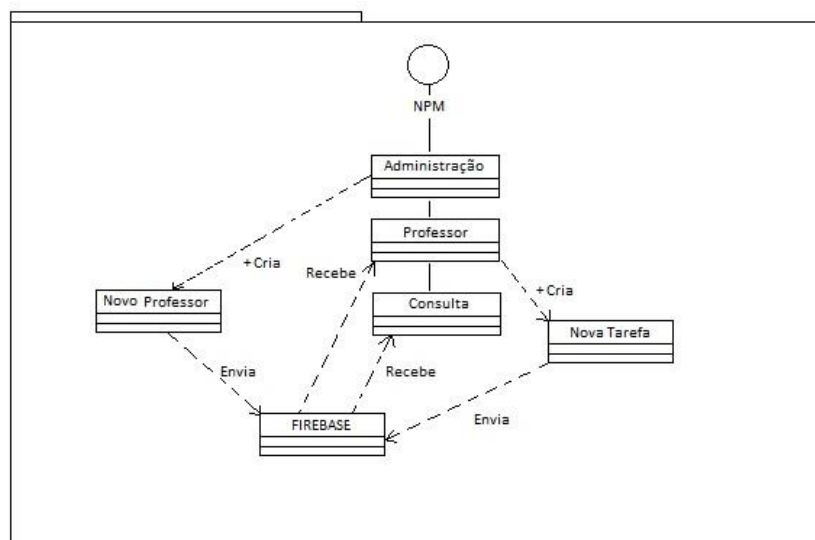
<https://www.figma.com/proto/7wpmwcBwYLuuCv0gWRkNOA/Untitled?node-id=0%3A3&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=0%3A3>
<https://www.figma.com/community/file/1162977060069350537>

Aplicação disponível no Vercel, através do link:

<https://puc-gestaodoconhecimento-2.vercel.app>

5.2 Código da Aplicação

Figura 4 – Código da aplicação



A Figura 4 apresenta a estrutura da aplicação e o relacionamento entre os componentes.

Os componentes tem como dependência principal a existência de um professor cadastrado pela administração, no projeto com todos os pontos atendidos, teríamos um administrado cadastrado previamente para gerenciar os professores e outros administrados, assim todos os professores e administradores teriam login de acesso

para realizar as atividades, mas como o intuito desta POC é apresentar algumas funcionalidades que compõem o todo do projeto, foi apresentado a funcionalidade de cadastrado o professor sem a implantação da funcionalidade de login e senha.

Esta estrutura, toma como base o administrador criando o professor na base de dados de professores, com o professor já criado o mesmo pode iniciar sua tarefa, assim que a tarefa é concluída ela é enviada a base de dados das tarefas, por fim a tarefa já criada, ela pode ser consultada por qualquer um com acesso a aplicação por meio da consulta.

Código da aplicação disponível no GitHub:

[EdnirAraujo/puc-gestao-do-conhecimento: Projeto Integrado Gestão de Conhecimento \(github.com\)](https://github.com/EdnirAraujo/puc-gestao-do-conhecimento)

Referências

Portaria FGV. Brasil tem dois dispositivos digitais por habitante, revela pesquisa da FGV., 21 de Maio de 2021. Disponível em: <https://portal.fgv.br/noticias/brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv>. Acesso em: 05 jun. 2022.