

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

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

Projeto Integrado

Relatório Técnico

Sistema de Gestão de Serviços em Canteiros de Obras

Alexandre M Izeffler

Belo Horizonte
Dezembro, 2021.

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	7
3.3 Requisitos Não-funcionais	9
3.4 Mecanismos Arquiteturais	10
4. Modelagem Arquitetural	11
4.1 Diagrama de Contexto	12
4.2 Diagrama de Container	12
4.3 Diagrama de Componentes	15
5. Prova de Conceito (PoC)	17
5.1 Integrações entre Componentes	20
5.2 Código da Aplicação	21
6. Avaliação da Arquitetura (ATAM)	24
6.1. Análise das abordagens arquiteturais	24
6.2. Cenários	25
6.3. Evidências da Avaliação	25
6.4. Resultados Obtidos	31
7. Avaliação Crítica dos Resultados	32
8. Conclusão	34
Referências	35

1. Introdução

O setor da construção civil nos últimos anos teve um crescimento surpreendente. Para se ter uma ideia, no primeiro trimestre do ano 2021 foi registrado um crescimento econômico do PIB de 2,1%, e a construção civil obteve uma representatividade no PIB nacional de 7%. Portanto, fica claro a importância deste setor na geração de empregos e renda e a sua relevância no crescimento econômico do Brasil. Mesmo no período da pandemia do Covid 19 no qual grande parte dos setores econômicos foram afetados negativamente, a Construção Civil manteve-se em destaque e com um forte crescimento, pois com o confinamento da população, novas necessidades foram descobertas em suas moradias e as pessoas passaram a valorizar mais o conforto e às suas novas necessidades “a casa como seu refúgio”.

Todo esse crescimento e potencial futuro é fonte de investimentos que são impulsionados e necessários em melhorias dos processos e adoção de novas tecnologias para ser competitivo no mercado. É justamente neste contexto que a Construtora ConstruBC, uma empresa fictícia, em seu planejamento estratégico designou parte do seu orçamento anual para a modernização dos seus canteiros de obras com o uso da tecnologia para otimização de desperdícios e melhorar a gestão da qualidade do seu produto final – a moradia do seu cliente.

A ConstruBC atua no segmento de moradias de casas em condomínios fechados para famílias das classes A e B, presente em quase todos os Estados brasileiros, possui pipeline de mais de 250 obras em andamento. São mais de 10 mil funcionários diretos e 20 mil indiretos, que juntos tem trabalhado para realizar o sonho de mais de 500 mil clientes.

Um estudo foi realizado nos canteiros de obras da ConstruBC por uma consultoria especializada em melhorias de processos e na aplicação da metodologia de design thinking. E foi identificado que a comunicação entre os engenheiros e auxiliares de engenharia com os profissionais do escritório central da companhia não tem sido fácil. Pois, todos os meses ocorrem problemas nos dados repassados para o sistema ERP central da companhia, e isso tem se tornado um parto na consolidação das medições de serviços para a realização dos pagamentos aos fornecedores.

Além dos problemas de comunicação, os usuários do sistema atual no canteiro de obras, reclamam de lentidão, usabilidade ruim e falta de mobilidade para apuração dos quantitativos dos serviços executados, sendo que muitas vezes, os dados adicionados no sistema não são totalmente confiáveis.

O resultado deste estudo forneceu informações bem positivas e animadoras, caso seja produzida uma solução centrada no usuário e que envolva mobilidade, a nova plataforma poderia reduzir em 15% os desperdícios do uso da mão-de-obra fornecidas pelos empreiteiros parceiros, diminuir em 12% o prazo da construção dos empreendimentos e por último, aumentar em 18% a qualidade do produto final.

Em resumo, os resultados dos estudos trouxeram bastante motivação para a construção de uma plataforma, esta nomeada como, plataforma de Gestão de Serviços de Canteiros de Obras (GSC Obras), sendo, portanto, o objetivo deste trabalho, a apresentação da solução arquitetural de software para o desenvolvimento desta plataforma.

Por se tratar de um produto importante e que as expectativas quanto aos resultados sejam atendidas, foram definidos 3 objetivos para esta versão inicial da plataforma GSC Obras, sendo eles:

- Ser uma solução totalmente integrada com os sistemas core da companhia, permitindo a continuidade fluida dos processos de negócios de compras, financeiro e cronograma de projetos;
- De fácil entendimento e uso, que promova mobilidade e permita o uso off-line para as funções que necessite de circulação pelo empreendimento;
- Seja segura, escalável, tolerante a falhas e robusta para suportar o crescimento da operação no decorrer dos anos.

Portanto, neste documento serão apresentados os requisitos arquiteturais, funcionais e não funcionais e as diagramações da solução para o desenvolvimento da plataforma GSC Obras que compreenda os objetivos citados acima.

2. Cronograma do Trabalho

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

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
06 / 12 / 2021	06 / 12 / 2021	1. Cronograma do Trabalho	Construção desta tabela
10 / 12 / 2021	11 / 12 / 2021	2. Contextualização do trabalho	Construção da contextualização deste projeto
06 / 01 / 2022	06 / 01 / 2022	3. Definição dos requisitos Arquiteturais	Lista dos requisitos Arquiteturais identificados
08 / 01 / 2022	09 / 01 / 2022	4. Definição dos requisitos Funcionais	Lista dos requisitos funcionais identificados
10 / 01 / 2022	10 / 01 / 2022	5. Definição dos requisitos Não-funcionais	Lista dos requisitos Não-funcionais identificados
10 / 01 / 2022	10 / 01 / 2022	6. Definição dos Mecanismos Arquiteturais	Lista dos Mecanismos Arquiteturais identificados
10 / 01 / 2022	11 / 01 / 2022	7. Construção dos Diagramas de Contextos – Modelo C4	Diagrama de contexto criado no Draw.io e documentado
12 / 01 / 2022	12 / 01 / 2022	8. Revisão da Etapa 1	Documento Etapa 1 revisado
06 / 01 / 2022	08 / 01 / 2022	9. Construção do vídeo de apresentação da Etapa 1	Vídeo criado da Etapa 1
12 / 02 / 2022	13 / 02 / 2022	10. Apresentação em PPT da Etapa 1	PPT
13 / 02 / 2022	13 / 02 / 2022	11. Publicação no repositório Github Etapa 1	Arquivos produzidos no Github disponíveis abertamente
15 / 02 / 2022	16 / 02 / 2022	12. Construção dos Diagramas de Contêineres	Diagramas de contêineres
18 / 02 / 2022	19 / 02 / 2022	13. Construção dos Diagramas de Componentes	Diagramas de componentes
20 / 01 / 2022	21 / 01 / 2022	14. Desenho dos Wireframes da POC	Protótipos de telas de baixa fidelidade
01 / 03 / 2022	14 / 04 / 2022	15. Código da aplicação	Aplicação com 3 requisitos implementados
15 / 04 / 2022	15 / 04 / 2022	16. Publicação no repositório Github Etapa 2	Arquivos produzidos no Github disponíveis abertamente
17 / 04 / 2022	18 / 04 / 2022	17. Análise das abordagens arquiteturais	Seção do documento produzido
23 / 04 / 2022	26 / 04 / 2022	18. Cenários	Seção do documento produzido
01 / 05 / 2022	07 / 05 / 2022	19. Evidências da avaliação	Seção do documento produzido
08 / 05 / 2022	09 / 05 / 2022	20. Resultados obtidos	Seção do documento produzido
15 / 05 / 2022	21 / 05 / 2022	21. Avaliação crítica dos resultados	Seção do documento produzido

23 / 05 / 2022	29 / 05 / 2022	22. Conclusão	Seção do documento produzido
01 / 06 / 2022	04 / 06 / 2022	23. Construção do vídeo de apresentação da Etapa 3	Vídeo da etapa 3 disponível
15 / 06 / 2022	15 / 06 / 2022	24. Publicação no repositório Github Etapa 3	Arquivos produzidos no Github disponíveis abertamente

3. Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitem visualizar a macroarquitetura da solução.

3.1 Restrições Arquiteturais

Os Requisitos Arquiteturais são todos os requisitos, sejam eles Funcionais ou Não-Funcionais que têm **impacto direto** sobre a Arquitetura do Sistema. Dessa forma, o Arquiteto precisa analisar os requisitos do sistema identificando algumas propriedades e então “filtrando” os Requisitos Arquiteturais. A lista a seguir apresenta os requisitos arquiteturais que foram identificados para implementação inicial da plataforma.

ID	Descrição
RA01	Deve-se usar tecnológicas Microsoft para o desenvolvimento de toda a plataforma GSC Obras.
RA02	Toda comunicação entre os sistemas core da companhia e da plataforma GSC Obras, deve ser realizada através da plataforma de Integração AIS (Azure Integration Services), preferencialmente no formato Rest/Json.
RA03	Deve ser considerado a nuvem Microsoft Azure como provedora da infraestrutura necessária para a plataforma GSC Obras.
RA04	Deve ser usado a ferramenta Azure DevOps (Boards, Git, CI e CD), ambiente da companhia, para o gerenciamento de todo o ciclo de desenvolvimento e evolução da plataforma.
RA05	Para o gerenciamento de identidades dos usuários deve ser usado a plataforma Microsoft Identity, portanto, deve ser considerado a autenticação/autorização baseada neste provedor com uso de RBAC (Controle de acesso baseado em funções) para permissões granulares.
RA06	Em muitas obras a comunicação com a Internet é precária, portanto, a solução precisa contemplar um modelo off-line/on-line.

RA07	A comunicação realizada entre os canteiros de obras e a plataforma GSC Obras da companhia deve ser pela Internet sem o uso de VPN (Rede Virtual Privada).
RA08	São usados smartphones e tablets com os sistemas operacionais Android (Google) e iOS (Apple), portanto, a solução mobile deve suportar ambas as plataformas.
RA09	Desenvolvimento mobile deve usar a plataforma Xamarin, como a equipe do projeto não possui total domínio da tecnologia, deve ser realizado uma PoC para definição dos exemplos das melhores práticas para serem seguidas.
RA10	Distribuição das versões dos aplicativos Mobile devem ser realizadas pela ferramenta de MDM (gerenciamento de dispositivo móvel) e MAM (gerenciamento de aplicativo móvel) Microsoft Intune.
RA11	Implementação do padrão Request/Replay entre as comunicações transacionais dos sistemas Backend e Mobile da plataforma GSC Obras.

3.2 *Requisitos Funcionais*

Os Requisitos Funcionais são todos aqueles que estão associados às funcionalidades que ditam **o que** sistema deve fazer. A lista a seguir apresenta os requisitos funcionais identificados para o desenvolvimento inicial da plataforma.

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O Backend da plataforma GSC Obras, deve sincronizar as Ordens de Serviços das Obras que foram geradas pela equipe de Suprimentos através do sistema ERP SAP.	A	A
RF02	O Backend da Plataforma GSC Obras, deve sincronizar o Cronograma da Obra que foi gerado pela equipe de Projetistas através do sistema Project On-line.	A	A
RF03	Os usuários devem identificar-se com as suas credenciais de acesso a rede da companhia para acesso ao sistema Mobile da plataforma GSC Obras.	M	A
RF04	Os usuários devem identificar-se com as suas credenciais de acesso a rede da companhia para acesso ao sistema Web da plataforma GSC Obras.	M	A
RF05	Os usuários devem somente visualizar e realizar qualquer ação nos sistemas Web e Mobile da plataforma GSC Obras para as Obras que possuem acesso.	M	A
RF06	O sistema Mobile da plataforma GSC Obras, deve apresentar uma lista das Ordens de Serviços planejadas cronologicamente baseando-se na semana corrente, além de permitir que seja	M	A

Sistema de Gestão de Serviços em Canteiros de Obras

	realizado uma busca por um intervalo de datas no passado/futuro.		
RF07	O Backend da plataforma GSC deve criar e liberar uma janela de medição (Abertura de período) para o mês corrente e as próximas janelas serão criadas e liberadas após o encerramento da janela corrente da medição de serviços (Fechamento de período).	M	A
RF08	O engenheiro, através do sistema Mobile da plataforma GSC Obras, deve distribuir as Ordens de Serviços aos respectivos Empreiteiros, informando a data início da execução do serviço, dentro de uma janela de medição liberada.	B	A
RF09	O engenheiro, através do sistema Mobile da plataforma GSC Obras, deve medir os serviços executados dentro da janela de medição liberada, informando a data final e quantidades da unidade de medida executadas pelos empreiteiros nas devidas Ordens de Serviços no sistema Mobile da plataforma GSC.	M	A
RF10	O Backend da plataforma GSC Obras, deve atualizar as atividades do cronograma com a data inicial da execução das Ordens de Serviços no sistema Project On-line, para que seja possível a realização do avanço físico da Obra.	A	A
RF11	O engenheiro, através do sistema Mobile da plataforma GSC Obras, deve ao finalizar as medições encerrar o período de medição.	B	A
RF12	O Backend da plataforma GSC Obras, deve enviar as medições de serviços para o sistema ERP SAP, quando for sinalizado o fechamento do período de medição de serviços, para que seja possível a geração das ordens de pagamento das medições aos Empreiteiros.	A	A
RF13	O Backend da plataforma GSC Obras, deve notificar por push (Mobile) aos usuários dos canteiros das obras quando o pagamento das medições de serviços dos Empreiteiros for realizado através do sistema ERP SAP.	M	B
RF14	O Backend da plataforma GSC Obras, deve notificar por e-mail aos usuários dos canteiros das obras quando o pagamento das medições de serviços dos Empreiteiros for realizado através do sistema ERP SAP.	B	M
RF15	O Backend da plataforma GSC Obras, deve notificar por push (Mobile) aos usuários dos canteiros de obra quando o período de medição for aberto e fechado.	M	B

Sistema de Gestão de Serviços em Canteiros de Obras

RF16	O Backend da plataforma GSC Obras, deve notificar por e-mail aos usuários dos canteiros de obra quando o período de medição for aberto e fechado.	B	M
RF17	O engenheiro, através do sistema Mobile da plataforma GSC Obras, poderá acompanhar o status do pagamento de cada empreiteiro das medições encerradas.	M	M
RF18	O engenheiro poderá a qualquer momento habilitar a opção para trabalhar off-line no sistema Mobile da plataforma GSC Obras. Este recurso deve permitir que seja realizado a medição dos serviços em campo somente. Quando o engenheiro quiser enviar as coletas realizadas e possuir conectividade com a Internet poderá enviar as medições realizadas.	A	M
RF19	O engenheiro poderá acompanhar por meio de um relatório de acompanhamento as medições de cada empreiteiro do período corrente e dos fechados, no sistema Web da plataforma GSC Obras.	M	M
RF20	Os usuários poderão acessar o seu perfil para modificar as suas preferências no sistema Mobile da plataforma GSC Obras.	B	B

*B=Baixa, M=Média, A=Alta.

3.3 Requisitos Não-funcionais

Os Requisitos Não-Funcionais estão associados às restrições de funcionalidades que **ditam como** o sistema deve fazer. A lista a seguir apresenta os requisitos funcionais identificados para o desenvolvimento inicial da plataforma.

ID	Descrição	Prioridade B/M/A
RNF01	A plataforma GSC Obras deve habilitar a autenticação e autorização baseado no modelo OAuth2 da plataforma Microsoft Identity com base no provedor Azure Active Directory.	A
RNF02	A plataforma GSC Obras deve realizar a integração com o sistema SAP ERP usando o protocolo RFC (Remote Function Call), modelo proprietário da SAP. Por meio da plataforma AIS (Azure Integration Services) deve ser usado o SAP Connector (https://docs.microsoft.com/en-us/connectors/sap/)	A
RNF03	A plataforma GSC Obras deve realizar a integração com o sistema Project On-line usando as suas APIs oficiais.	A

RNF04	O engenheiro ao habilitar o modo off-line para realizar a medição dos serviços em campo, deve ser usado o SQLite para armazenar os dados localmente no dispositivo móvel.	M
RNF05	O sistema deve permitir o funcionamento 6 x 12, sendo das 07:00 as 19:00 de segunda-feira a sábado.	A
RNF06	O sistema deve comportar com uma quantidade de 5.000 usuários e aproximadamente 300 obras em andamento no primeiro ano, podendo ter um crescimento nos próximos anos, por ano, 20% de obras.	A
RNF07	A versão Web da plataforma deve suportar os navegadores modernos, prioritariamente o Microsoft Edge.	A
RNF08	A versão Mobile Android da plataforma deve suportar os dispositivos considerados medianos na faixa de preço entre R\$ 1.200 à R\$ 1.800 com as versões mais recente do Sistema Operacional.	A
RNF09	A versão Mobile iOS da plataforma deve suportar a versão iPhone 8 ou superior. Atenção! A versão para iOS é desejável no momento, poderá no futuro ser requerido.	B
RNF10	A distribuição das aplicações e a infraestrutura (IoC – Infraestrutura como Código) devem ser automatizadas usando pipelines CI/CD do Azure DevOps.	A
RNF11	As notificações por e-mail ou push devem operar por meio de filas de mensagens.	A
RNF12	Rastreamento de uso, falhas e performance deve ser utilizado o Azure Monitor Application Insights e Workspce Analytics.	A
RNF13	As comunicações entre os sistemas Backend e Mobile da plataforma GSC Obras, para os cenários transacionais devem implementar o padrão de integração Request/Replay.	A

3.4 Mecanismos Arquiteturais

Os mecanismos arquiteturais representam conceitos técnicos fundamentais que serão padronizados por toda a solução. Eles são refinados durante o projeto em três estados, representados pelas três categorias de Mecanismos Arquiteturais:

- Mecanismo de Análise, que dá ao mecanismo um nome, uma descrição resumida e alguns atributos básicos derivados dos requisitos do projeto.
- Mecanismo de Design, que são mais concretos e assumem alguns detalhes do ambiente de implementação.

- Mecanismo de Implementação, que especifica a exata implementação de cada mecanismo.

Análise	Design	Implementação
Persistência	ORM	EntityFramework
Persistência	ORM	Sqlite
Persistência	Banco de Dados NoSQL	Azure CosmosDb Documents
Front end	MVC	Asp.Net Core MVC
Front end	Mobile	Xamarin
Front end	Navegador Web	Microsoft Edge
Back end	Serverless	Azure Functions
Back end	Serverless	Azure Logic App
Integração	iPaaS	Azure Integration Services
Log do sistema	Telemetria	Azure Monitor Application Insights
Teste de Software	Testes unitários	xUnit
Usabilidade	Notificação por Push	Azure Hub Notification
Confiabilidade	Eventos	Azure Event Grid
Confiabilidade	Service Bus	Azure Service Bus
Autenticação	Oauth2	Azure Active Directory
Autorização	Oauth2	Azure Active Directory
Gerenciamento Mobile	MDM e MAM	Microsoft Intune
Distribuição	Integração e Entrega Continua (CI/CD)	Azure DevOps

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 (PoC) da plataforma GSC Obras 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 aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>. Dos quatro nível que compõem o modelo C4 três serão apresentados aqui e somente o Código será apresentado na próxima seção (5).

4.1 Diagrama de Contexto

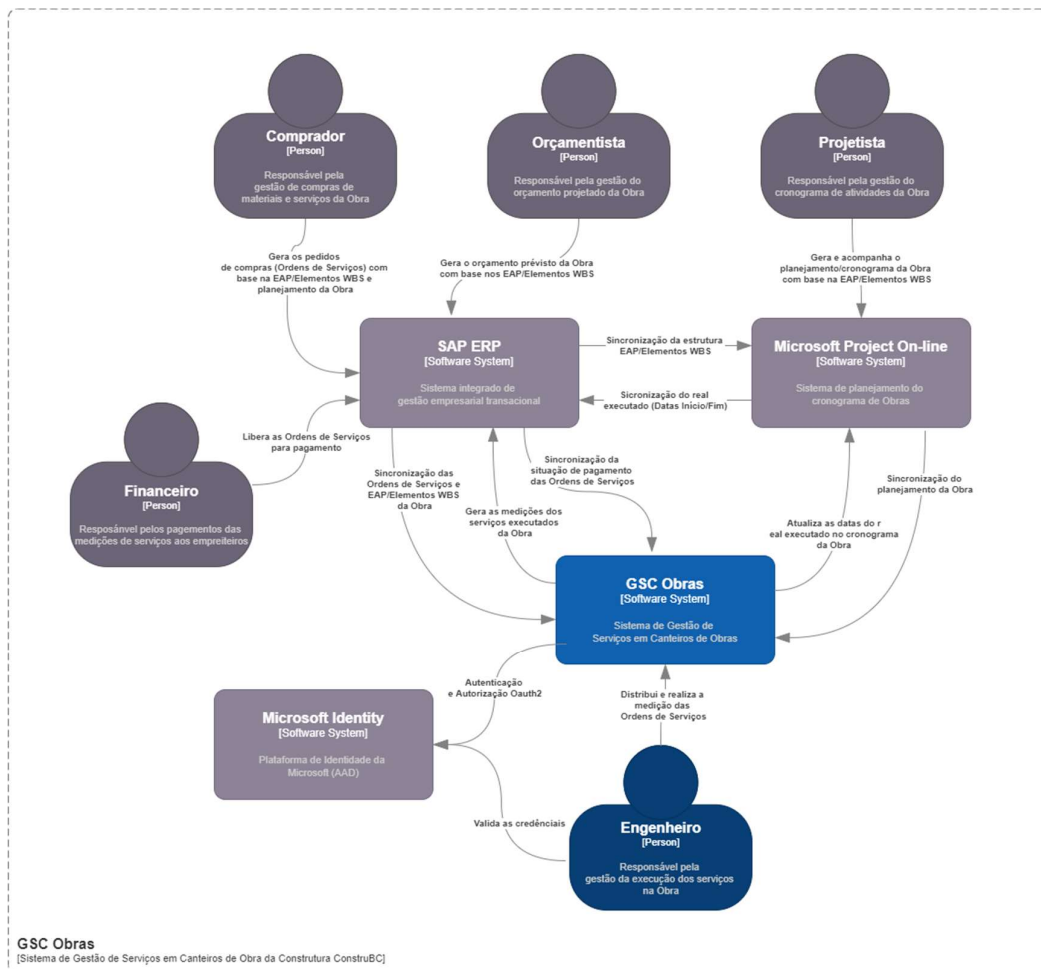


Figura 1 - Visão Geral da Solução GSC Obras.

A figura 1 mostra a especificação o diagrama geral da solução proposta, com todos seus principais sistemas e pessoas envolvidas no processo de medição de serviços. É importante compreender a dependência dos processos de Compras e Planejamento, para que seja possível a distribuição das Ordens de Serviços, a medição das Ordens de Serviços executadas e acompanhamento do pagamento aos empreiteiros referentes as janelas de medição.

4.2 Diagrama de Container

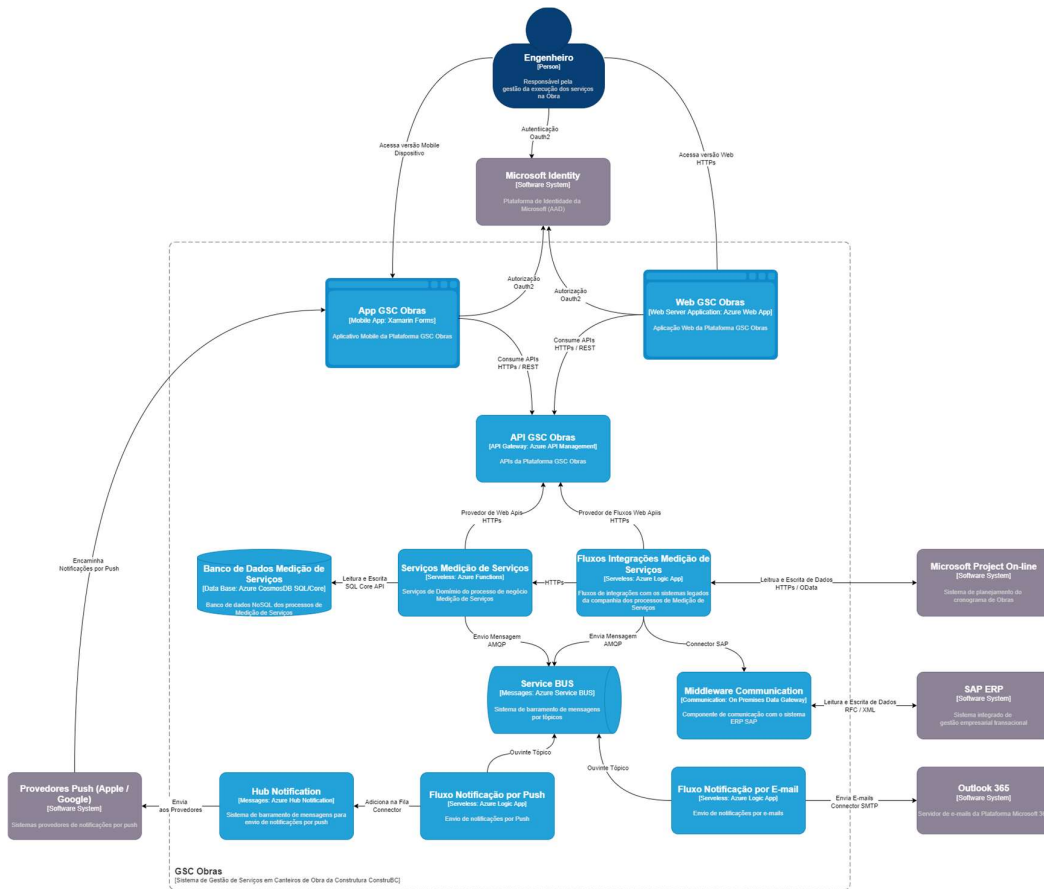


Figura 2 – Diagrama de contêineres da Plataforma GSC Obras

A figura 2 apresenta os *containers* da aplicação, na qual apresenta como os componentes (aplicativos, armazenamentos de dados, serviços Web, etc.) que compõem a plataforma e como estão distribuídos e organizados.

O principal usuário da plataforma será o engenheiro, este profissional trabalha nos canteiros de obras realizando a gestão da obra para que os projetos técnicos e cronograma da obra sejam executados de forma adequada.

A Plataforma disponibilizará uma aplicação Mobile e outra Web, que permitirá os usuários interagir com as funcionalidades para realizar as medições de serviços e acompanhamento dos pagamentos aos empreiteiros. O engenheiro usará a sua credencial do Azure Active Directory para acessar as aplicações.

Estas aplicações consumirão APIs que serão disponibilizadas através do container API Gateway, o middleware dos serviços Web dos domínios de negócios e de algumas integrações. Os serviços Web que contém as regras do domínio de negócios dos processos de medição de serviços, serão disponibilizados por meio da tecnologia Serverless Azure Functions, sendo o armazenamento para persistência dos dados o NoSQL Azure CosmosDB.

Os serviços de integrações serão realizados por meio da tecnologia Serverless Azure Logic App e serão responsáveis pelas integrações com os sistemas Project On-line e SAP.

A comunicação entre os serviços Web de domínios de negócios e de integrações serão realizados através do middleware de troca de mensagens Azure Service Bus em um modelo assíncrono. Somente nos casos de carga de dados periódicas, será usado o modelo de comunicação síncrona usando o protocolo HTTPs.

As notificações serão mediadas também pelo Azure Service Bus, e usado fluxos de trabalho desenvolvidos com Azure Logic App, na qual, permitirá um maior desacoplamento, reuso e escalabilidade para o envio das notificações por e-mail e Push.

Para visualizar a versão original do diagrama de container apresentado acima e os demais, acesse o arquivo “diagramas-pucminas-2021.drawio” no site draw.io armazenado no repositório do Github aqui: <https://github.com/aizefler/pucminas/tree/main/docs/etapa2>.

4.3 Diagrama de Componentes

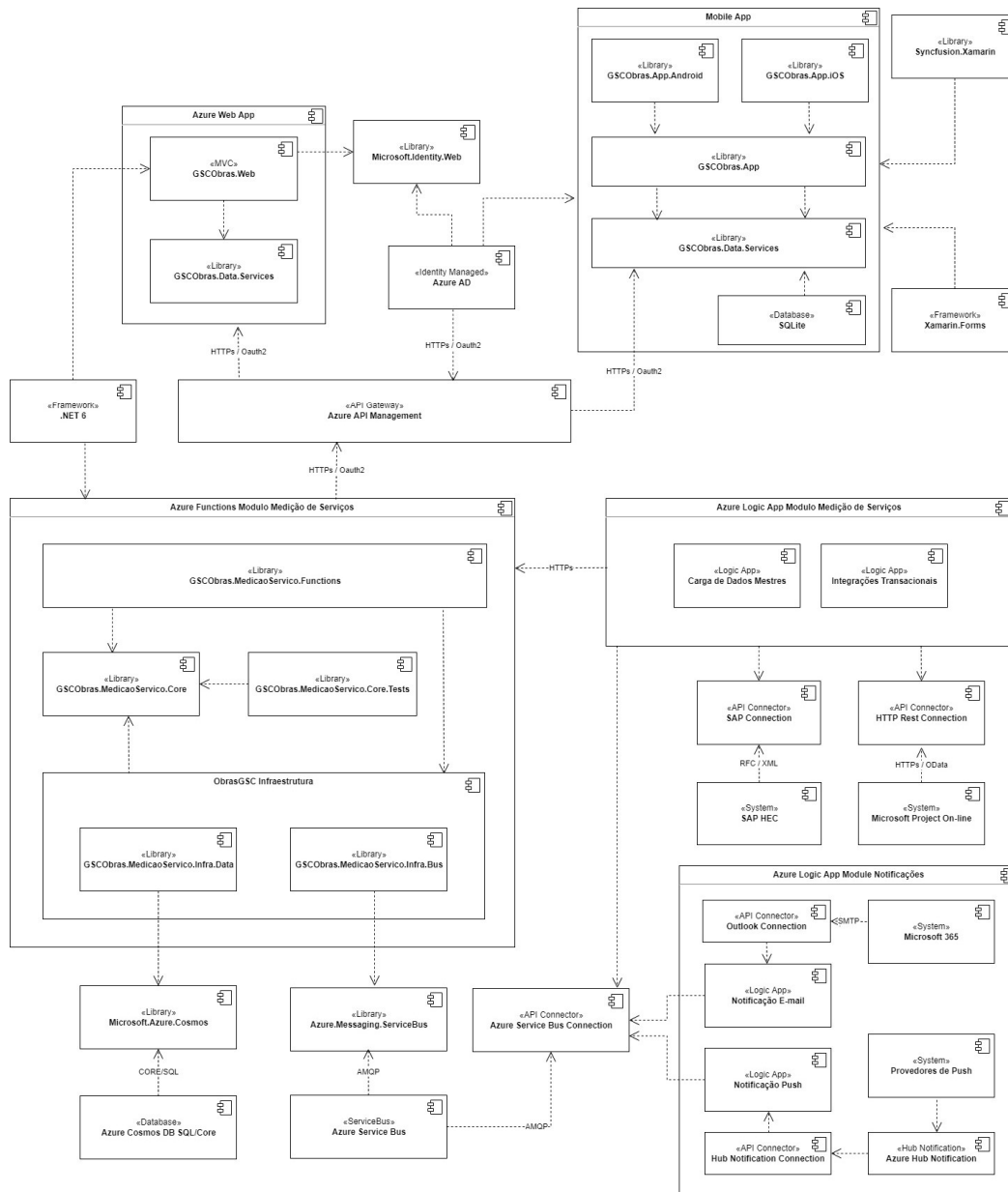


Figura 2 – Diagrama de Componentes da Plataforma GSC Obras

O diagrama de componentes tem por objetivo fornecer uma visão com maiores detalhes os componentes envolvidos na plataforma GSC Obras e para um melhor entendimento, a lista a seguir elucida cada um deles.

- **Mobile App:** Aplicativo mobile Android e iOS;
 - **GSCObras.App.Android:** Projeto de classes do framework Xamarin dedicado a plataforma Android;
 - **GSCObras.App.iOS:** Projeto de classes do framework Xamarin dedicado a plataforma iOS;
 - **GSCObras.App:** Projeto de classes do framework Xamarin compartilhado entre as plataformas Android/iOS;
 - **GSCObras.Data.Services:** Projeto de classes .Net Standard responsável pela comunicação com as APIs do Backend e acesso aos dados locais;
 - **Syncfusion.Xamarin:** Bibliotecas de classes proprietária que estende as funcionalidades de UI do framework Xamarin Forms;
 - **Xamarin.Forms:** Framework cross native que possibilita o desenvolvimento de aplicações mobile multiplataforma da Microsoft.
- **Azure Web App:** Plataforma como serviço para hospedagem da aplicação Web;
 - **GSCObras.Web:** Projeto ASP.Net MVC, Application Server, versão Web da plataforma GSC Obras;
 - **GSCObras.Data.Services:** Projeto de classes .Net Standard responsável pela comunicação com as APIs do Backend;
 - **Microsoft.Identity.Web:** Biblioteca responsável pela abstração dos processos de Autenticação e Autorização do provedor de identidades Azure Active Directory;
 - **.Net 6:** Framework multiplataforma da Microsoft;
- **Azure AD:** Provedor de identidades da Microsoft para ambientes corporativos;
- **Azure API Management:** Plataforma de APIs da Microsoft que fornece 3 componentes principais, como: API Gateway, Ferramenta de Administração e Portal do Desenvolvedor, sendo também um dos principais serviços de sua solução iPaaS (Integration Platform as a Service);
- **Azure Functions Module Medição de Serviços:** Tecnologia Serverless que fornece uma forma mais simples e descomplicada para a criação de funções com gatilhos HTTP e entre outros protocolos. Estes componentes são dedicados ao módulo para o processo de Medição de Serviços, portanto, novos módulos terão o seu conjunto de componentes totalmente desacoplados;
 - **GSCObras.MedicaoServico.Functions:** Projeto de Functions da stack plataforma .Net 6;
 - **GSCObras.MedicaoServico.Core:** Projeto de classes responsável pelas regras de negócio do módulo;
 - **GSCObras.MedicaoServico.Core.Tests:** Projeto de classes responsável pelos testes unitários do módulo;
 - **GSCObras Infraestrutura:** Organização de todos os projetos dedicados as componentes de infraestrutura;
 - **GSCObras.MedicaoServico.Infra.Data:** Projeto de classes responsável pelo acesso ao banco de dados do módulo;
 - **GSCObras.MedicaoServico.Infra.Bus:** Projeto de classes responsável pela comunicação do serviço de mensageria do módulo;
- **Microsoft.Azure.Cosmos:** Biblioteca para comunicação e operações no Banco de dados NoSQL Cosmos DB;

- **Azure Cosmos DB SQL/Core:** Banco de dados NoSQL API SQL/Core totalmente gerenciado;
- **Azure.Messaging.ServiceBus:** Biblioteca para comunicação com o Azure Service Bus;
- **Azure Service Bus:** Serviço de mensageria totalmente gerenciado;
- **Azure Service Bus Connection:** Componente para conexão com Azure Service Bus;
- **Azure Logic App Modulo Medição de Serviços:**
 - **Carga de Dados Mestres:** Logic App(s) dedicados para carga de dados mestres necessários para o modulo;
 - **Integrações Transacionais:** Logic App(s) dedicados para as integrações com os sistemas legados;
 - **SAP Connection:** Componente para conexão com o sistema SAP;
 - **SAP HEC:** Sistema ERP SAP Hana Enterprise Cloud;
 - **HTTP Rest Connection:** Componente para conexão com o protocolo HTTP/HTTPS;
 - **Microsoft Project On-line:** Sistema para gestão de projetos da Microsoft;
- **Azure Logic App Modulo Notificações:**
 - **Notificação E-mail:** Logic App para envio de e-mail ao provedor de e-mails Outlook do Microsoft 365;
 - **Outlook Connection:** Componente para conexão com Outlook;
 - **Microsoft 365:** Provedor de e-mails da Microsoft;
 - **Notificação Push:** Logic App para envio de notificação por Push;
 - **Hub Notification Connection:** Componente para conexão com Azure hub Notification;
 - **Azure Hub Notification:** Serviço de mensagens para notificação por Push multiplataforma totalmente gerenciado;

5. Prova de Conceito (PoC)

O GSC Obras por ser uma plataforma que envolve duas aplicações sendo uma mobile e outra Web, foram selecionados para o desenvolvimento da prova de conceito três requisitos funcionais, sendo eles: RF04 (tela de login Web) e RF16 (fluxo de envio de notificação por e-mail de períodos abertos e fechados) e o requisito RF19 (tela para acompanhamento das medições de serviços) da aplicação Web.

Para melhor entendimento da proposta da solução como um todo, foram criados protótipos navegáveis com a ferramenta Adobe XD do aplicativo mobile com as funções da primeira versão proposta. Os protótipos podem ser acessados aqui: <https://xd.adobe.com/view/c364a315-4797-4bea-9aa5-54abb1c3ae39-9637>.

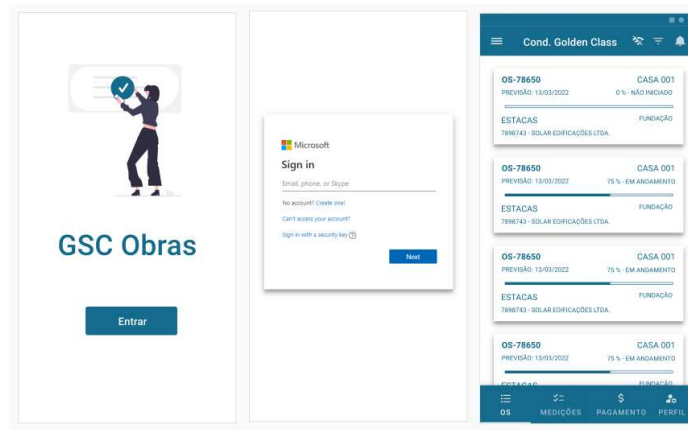


Figura 4 – Protótipos de algumas telas do aplicativo mobile

O requisito RF04 implementa a configuração do fluxo de autenticação do Azure Active Directory para aplicativos Web, conforme a documentação oficial do produto que pode ser acessada por este link: <https://learn.microsoft.com/pt-br/azure/active-directory/develop/app-sign-in-flow>.

Na aplicação Web, quando o usuário acesso a URL <https://gscobras-web.azurewebsites.net>, caso o usuário não esteja autenticado, será redirecionado para a tela da imagem a seguir para informar as suas credenciais. Para acessar a aplicação, basta informar o usuário “demo.pucminas@alexandreizefler.com.br” e senha “Pu@Mn123”.

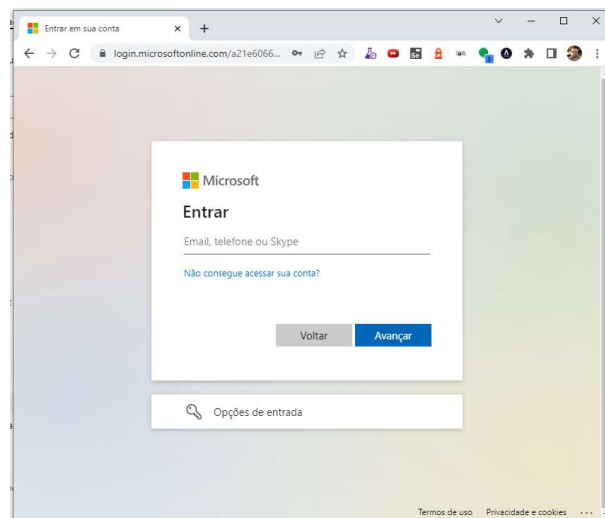


Figura 5 – Tela de login

O requisito RF16 implementa os recursos Serverless Azure Logic App e Service Bus que permite o uso de um serviço de mensagens para envio de notificações aos usuários através de Tópicos e a criação de assinantes por meio de fluxos de trabalho. Neste requisito foi criado um fluxo de trabalho com o gatilho de um assinante para envio de e-mail através do provedor Outlook, este fluxo é executado a cada 15 minutos para recuperar as mensagens no tópico “notifications”.

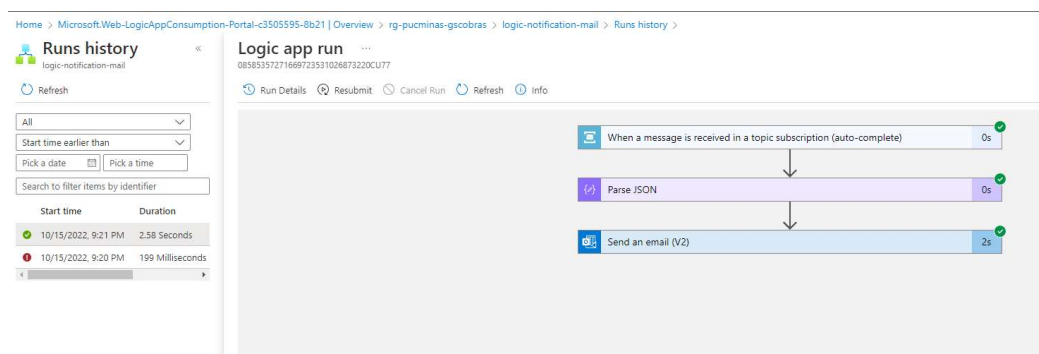


Figura 6 – Requisito RF16 fluxo de trabalho para envio de notificações por e-mail

Para testar este requisito foi necessário a implementação dos requisitos RF11 (encerramento do período de medição) e RNF11 (fluxo de notificações por e-mail com uso de tópicos do serviço mensagens do Azure Service Bus). Portanto, o teste pode ser realizado para simulação do fluxo pode ser realizado por meio da página “Teste Encerramento Medição”.

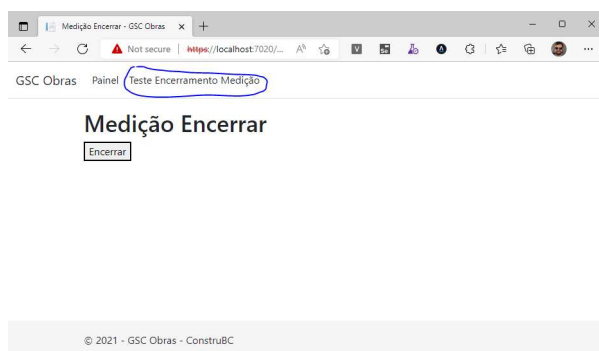


Figura 7 – Tela para teste do requisito RF16

O requisito RF19 refere-se à aplicação Web que permite o acompanhamento das medições de serviços dos fornecedores que executaram os serviços de uma determinada obra. Esta funcionalidade foi desenvolvida e está disponível para acesso aqui: <https://gscobras-web.azurewebsites.net>.

Fornecedor CNPJ	Fornecedor	Situação	Valor
Medição: 03/2022 - 3 items			
09.986.976/0001-76	Fornecedor 01	Pago	\$5.680,00
10.986.976/0001-84	Fornecedor 02	Pago	\$2.780,00
10.986.976/0001-84	Fornecedor 02	Pago	\$1.680,00
Medição: 04/2022 - 1 item			
09.986.976/0001-76	Fornecedor 01	Pago	\$1.780,00

Figura 8 – Requisito RF19 para acompanhamento das medições de serviços

5.1 Integrações entre Componentes

Nesta seção são apresentados em detalhes as integrações (protocolos, *middlewares*, padrão de troca de dados, etc) entre os três requisitos prioritários selecionados.

RF03 – Tela de Login: Neste requisito é realizado uma integração entre o aplicativo Web com o Azure Active Directory, usando o protocolo de autenticação oauth2 com o fluxo de autorização Web.

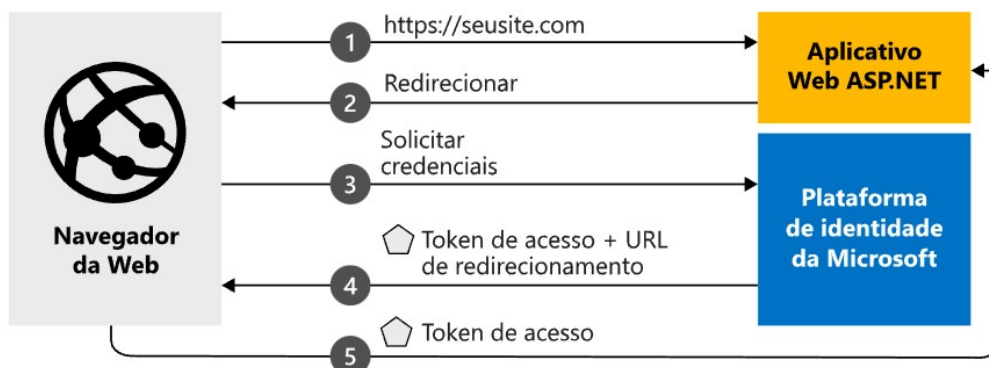


Figura 9 – Fluxo de autorização entre a aplicação Mobile e as API's

RF16 – Notificação por e-mail do período fechado/aberto: Neste requisito o aplicativo mobile com um token JWT válido acessa o aplicativo de APIs REST da plataforma para encerrar o período de medição aberto de uma obra (rota /medicaoservico/{obraId}/encerrar método POST) usando o protocolo HTTPs sem conteúdo. A API recupera o período aberto, realiza as validações, encerra período vigente e abre um novo período. O interessante neste requisito são os componentes envolvidos, como: Service Bus e Logic App, além dos API Gateway, Function e Cosmos DB.

RF19 – Tela acompanhamento das medições de serviços: Neste requisito o aplicativo Web MVC ASP.Net Core usando os frameworks frontend Bootstrap e Syncfusion disponibiliza a apresentação das medições de serviços com base em uma obra selecionada. Esta aplicação consome o aplicativo de APIs com um token válido usando o mesmo protocolo e formato de conteúdo citados no requisito acima, conforme apresentado no diagrama de componentes no item 4.3.

5.2 Código da Aplicação

Os fontes da aplicação estão disponíveis no repositório github <https://github.com/aizefler/pucminas> no diretório “src”.

A codificação da aplicação foi realizada usando a linguagem de programação Microsoft C#, framework .NET 6+ e IDE (Integrated Development Environment) Visual Studio. Para uma melhor estruturação das responsabilidades da implementação do código da aplicação foi adotado o estilo Arquitetura Limpa, visando a simplicidade, clareza das responsabilidades e a busca pelo princípio da Inversão de Dependência (Para saber mais sobre este modelo acesse este link: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>).

Clean Architecture Layers (Onion view)

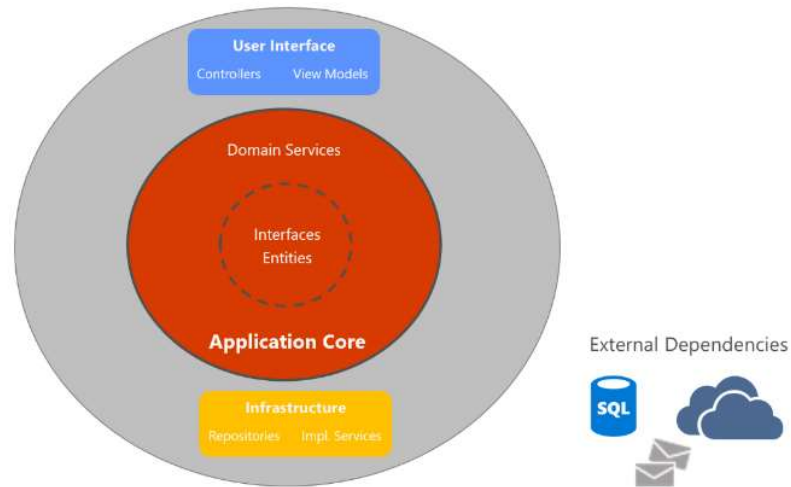


Figura 10 – Clean Architecture

A seguir a imagem da figura 8 apresenta o diagrama de classes produzido pelo Visual Studio versão Enterprise é possível compreender a associação dos projetos (Pacotes) e classes relacionadas para atingir o objetivo do requisito RF19 – Tela acompanhamento das medições de serviços.

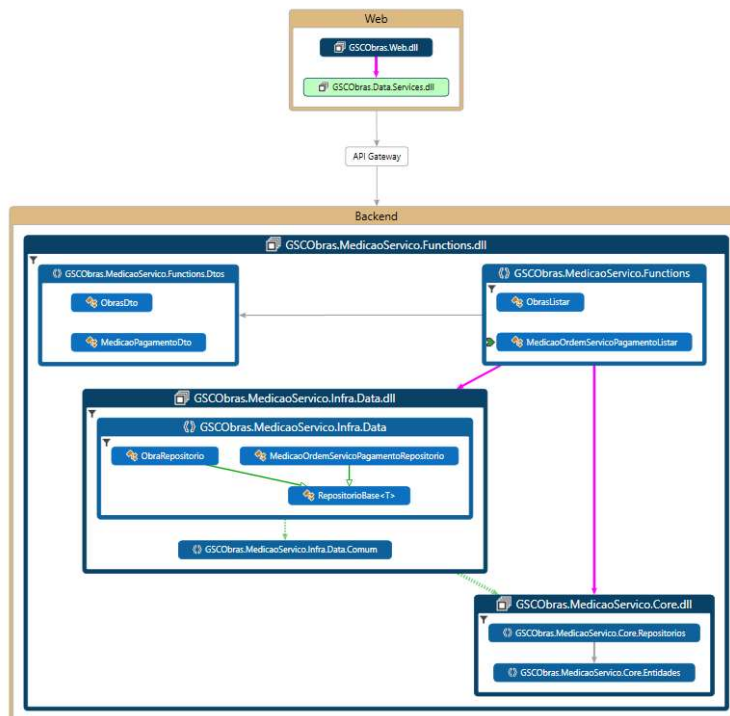


Figura 11 – Diagrama de Classes do requisito RF19

A figura acima apresenta 3 grandes componentes, sendo eles: **Web**, o frontend MVC que consome as APIs disponibilizadas pelo **API Gateway** e o **Backend**, a implementação das APIs da aplicação.

No componente Backend pode ser observado que a DLL `GSCObras.MediServico.Functions` é a camada que expõem o acesso as operações através do protocolo HTTP, sendo mediada pelo API Gateway. As operações por sua vez acessão as classes concretas através de interfaces, respeitando o princípio da Inversão de Dependência e ao modelo arquitetural Arquitetura Limpa.

Um ponto a ser notado neste cenário é o consumo de acesso aos dados, sendo realizado pelo provedor específico do banco de dados fazendo o uso de uma implementação com classes bases genéricas, que permite o reuso de código nos acessos mais simples e em casos mais específicos a classe pode ser estendida.

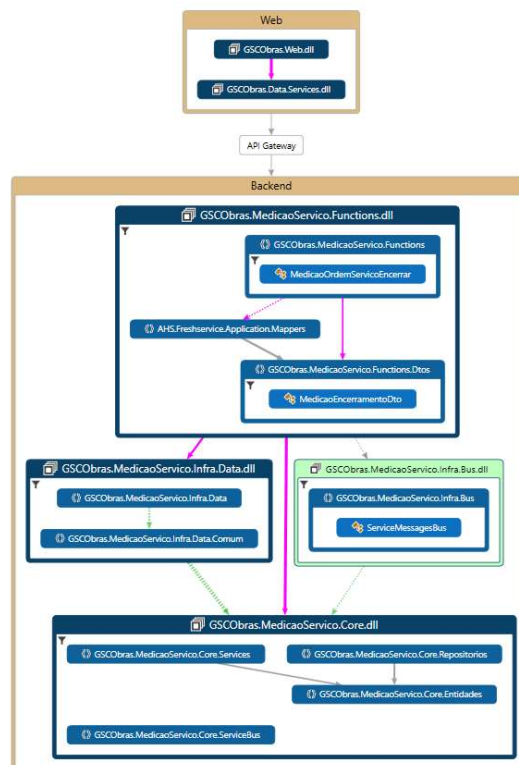


Figura 12 – Diagrama de Classes do requisito RF16

O diagrama do requisito RF16 apresenta um novo componente de infraestrutura, o Service Bus. Pode ser observado que a implementação segue a proposta da arquitetura adotada, ou seja, por meio de interfaces a implementação concreta e específica do componente é totalmente isolado das implementações das regras de negócios.

6. Avaliação da Arquitetura (ATAM)

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção visando avaliar se ela atende ao que foi solicitado pelo cliente, segundo o método ATAM.

6.1. Análise das abordagens arquiteturais

Neste sub tópico é apresentado um breve resumo das principais características da proposta arquitetural, visando a utilização do método Architecture Tradeoff Analysis Method (ATAM), no qual são utilizados cenários para fazer essa análise.

Atributos de Qualidade	Cenários	Importância	Complexidade
Mobilidade	Cenário 1: A plataforma deve permitir a utilização das funções de campo em modo on-line/off-line.	A	A
Usabilidade	Cenário 2: A plataforma deve prover boa usabilidade visando a melhor experiência do usuário.	M	A
Performance	Cenário 3: A plataforma deve oferecer boa performance durante o seu uso sem provocar lentidão na execução de suas funções.	A	A
Confiabilidade	Cenário 4: A plataforma deve permitir alta disponibilidade durante o seu horário de funcionamento, sendo das 07:00 da manhã até as 19:00 da noite, durante os cinco dias da semana.	A	A

6.2. Cenários

Os cenários utilizados na realização dos testes da aplicação foram selecionados visando as principais dores do cliente que foram identificados no processo inicial do entendimento do problema a ser resolvido.

Cenário 1. Mobilidade: Os usuários de campo ao realizar as suas tarefas de medição de serviços precisam no local avaliado conseguir adicionar os quantitativos necessários para a ordem de serviço.

Cenário 2. Usabilidade: A facilidade de uso da plataforma na versão Mobile e Web precisam ser primordiais para maior aderência no uso e a sensação de necessidade de ferramenta essencial no dia-a-dia dos usuários.

Cenário 3. Performance: As funcionalidades da plataforma Mobile e Web devem apresentar boa performance e ser capaz de responder a 1000 requisições por segundo.

Cenário 4. Confiabilidade: As funcionalidades da plataforma web devem manter uma disponibilidade mínima de 99% para os serviços disponibilizados em nuvem, garantido uma maior eficiência no horário comercial e segurança no acesso aos sistemas disponibilizados.

6.3. Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

Atributo de Qualidade:	Mobilidade
Requisito de Qualidade:	A plataforma deve permitir o uso em dispositivos moveis com o uso off-line.
Preocupação:	
O aplicativo mobile deve permitir a realização da medição de serviços com a ausência de acesso à Internet durante a movimentação pelo empreendimento.	
Cenário(s):	

Cenário 1	
Ambiente:	
Sistema em operação normal sem acesso à Internet	
Estímulo:	
O aplicativo com os dados das ordens de serviços disponibilizadas no dispositivo mobile para realização da medição de serviços em campo.	
Mecanismo:	
Criar um modelo de sincronização de dados local baseado em requisições HTTP/REST permitindo o recebimento e envio de dados para o sistema remoto.	
Medida de resposta:	
Envio e retornar os dados requisitados no formato JSON com o armazenamento de dados local com SQLite.	
Considerações sobre a arquitetura:	
Riscos:	Conflito de versões de registros no uso de diversos usuários com as mesmas ordens de serviços ao mesmo tempo.
Pontos de Sensibilidade:	Identificação da versão mais atual da ordem de serviço.
Tradeoff:	Bloqueio de registros com base em usuários ou permitir que a escolha da versão seja selecionada em caso de conflitos.

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	Usuário precisa usar o aplicativo sem a necessidade de treinamentos intensos, seja intuitivo ao processo e de fácil entendimento – seja simples.
Preocupação:	
Dificuldade no entendimento dos sistemas Web e Mobile na execução do processo de medição de serviços, impactando nas gerações das medições durante o período de fechamento da medição de serviços e pagamento aos fornecedores.	

Cenário(s):
Cenário 2
Ambiente:
Sistema em operação normal
Estímulo:
Os usuários de campo possuem pouca escolaridade e dificuldade no entendimento de sistemas complexos, portanto, entendimento da jornada do usuário no processo de ponta-a-ponta é essencial para construção de uma experiência aprimorada.
Mecanismo:
Experiência do usuário / Design intuitivo de fácil similaridade.
Medida de resposta:
<p>Fluidez na realização das medições de serviços durante o período aberto de medição e pesquisas de satisfação do produto aos usuários periodicamente para coleta da satisfação quanto a usabilidade e facilidade no uso dos sistemas Mobile e Web.</p> <p>Nesta primeira versão foi desenhado uma interface gráfica simples e bem objetivo, centralizando as ordens de serviços conforme a situação, permitindo facilmente a identificação e ganho de tempo para as medições.</p>



O acompanhamento será realizado usando uma ferramenta de APM para monitorar o uso e o tempo gasto na execução das ações, neste caso, foi escolhido o Microsoft Application Insights.



Considerações sobre a arquitetura:	
Riscos:	Entrega de um produto de baixa aderência e resistência no uso das aplicações referente ao processo de negócio.
Pontos de Sensibilidade:	Não há.
Tradeoff:	Não há.

Atributo de Qualidade:	Performance
Requisito de Qualidade:	A plataforma deve oferecer boa performance durante o seu uso sem provocar lentidão na execução de suas funções.
Preocupação:	
Lentidão nas principais funções da plataforma impactando o processo de medição de serviços e consequentemente no pagamento aos fornecedores.	
Cenário(s):	
Cenário 3	
Ambiente:	

Sistema em operação com grande carga de dados

Estímulo:

Uso concorrente com grande demanda de processamento de tarefas complexas.

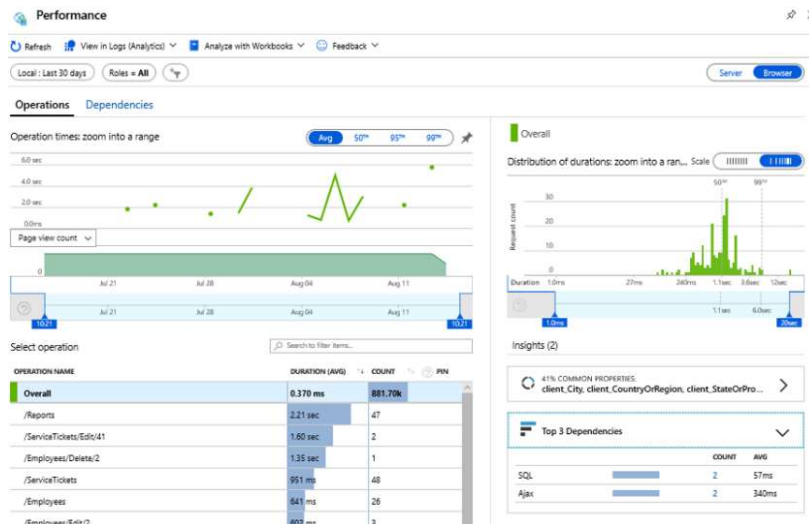
Mecanismo:

Fazer uso de ferramentas de testes de carga e performance para garantir a boa performance fora do horário comercial.

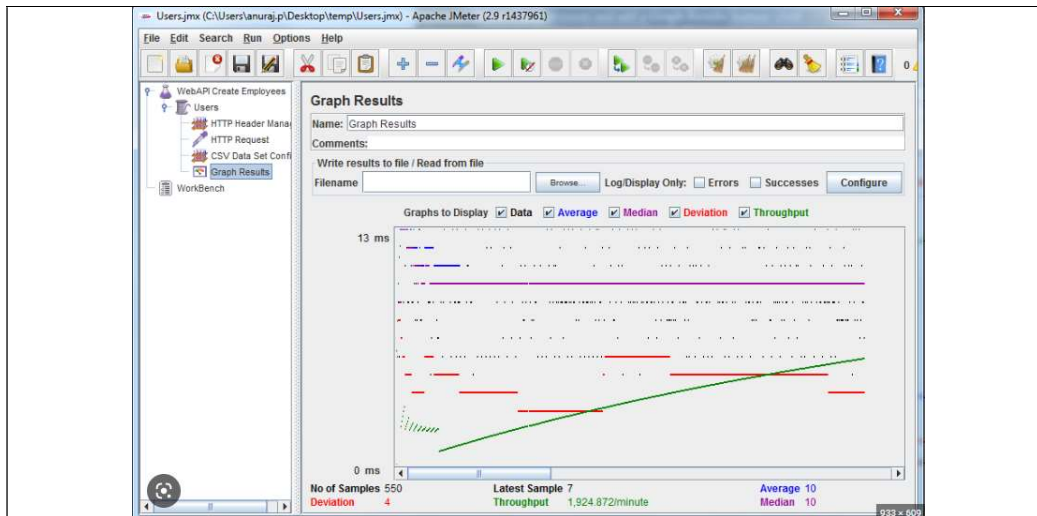
Medida de resposta:

Avaliação dos resultados da carga processada para identificação de melhorias de engenharia de sistemas e infraestrutura da plataforma.

Fazendo o uso das ferramentas JMeter e Microsoft Application Insights são possíveis a execução e a análise para o funcionamento dos sistemas.



Sistema de Gestão de Serviços em Canteiros de Obras



Considerações sobre a arquitetura:

Riscos:	Execução deste processo dentro do horário comercial.
Pontos de Sensibilidade:	Não há.
Tradeoff:	Não há.

Atributo de Qualidade:	Confiabilidade
Requisito de Qualidade:	A plataforma deve permitir alta disponibilidade durante o seu horário de funcionamento, sendo das 07:00 da manhã até as 19:00 da noite, durante os cinco dias da semana.
Preocupação:	
A plataforma deve apresentar disponibilidade de 99% para não impactar os usuários durante o processo de medição de serviços.	
Cenário(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	

Utilização dos sistemas pelos usuários.	
Mecanismo:	
Uso do sistema em diversos horários do dia e implementação do check de disponibilidade disponível pelo Microsoft Application Insights e recursos de alertas do Azure Monitor.	
Medida de resposta:	
Utilização do modelo de serviços de nuvem PaaS, com nível de serviço que forneça o mínimo de 99% de disponibilidade. Os serviços contratados foram Azure Cosmos DB, Azure Storage Account e API Gateway Consumption.	
Considerações sobre a arquitetura:	
Riscos:	Plataforma ficar indisponível e impactar no processo de medição de serviços e consequentemente o pagamento aos fornecedores.
Pontos de Sensibilidade:	Não há.
Tradeoff:	Não há.

6.4. Resultados Obtidos

Requisitos Não Funcionais	Teste	Homologação
RNF01 - Autenticação e autorização OAuth2	OK	OK
RNF02 - Integração com SAP usando o protocolo RFC	OK	N.A.
RNF03 - APIs Project On-line	N.A.	N.A.
RNF04 - Uso Off-line com SQLite	OK	OK
RNF05 - Funcionamento 6 x 12, sendo das 07:00 as 19:00 de segunda-feira a sábado	OK	OK
RNF06 - Suportar 5k de usuários e 300 obras ativas	OK	N.A.
RNF07 - Suportar navegador Microsoft Edge	OK	OK
RNF08 - Suportar OS Android em dispositivos medianos	OK	OK
RNF09 - Suportar OS iOS	N.A.	N.A.
RNF10 - Distribuição Azure DevOps CI/CD	N.A.	N.A.
RNF11 - Notificações por Push	N.A.	N.A.
RNF12 - Monitoramento com Azure Monitor	OK	N.A.
RNF13 - Padrão de integração Request/Replay	N.A.	N.A.

7. Avaliação Crítica dos Resultados

Na tabela a seguir são apresentados os principais pontos positivos e negativos da arquitetura proposta com uma postura crítica para que seja possível a compreensão das limitações arquiteturais, incluindo os prós e contras das tecnologias.

Ponto avaliado	Descrição
Usabilidade e experiência do usuário	<p><u>Ponto Positivo</u></p> <p>Com a adoção de técnicas de imersão para melhor entendimento do dia-a-dia dos usuários e as personas envolvidas foi fator decisivo na construção de funcionalidades simples e direta ao ponto, permitindo maior adoção dos usuários a nova plataforma.</p> <p><u>Ponto Negativo</u></p> <p>Por outro lado, o tempo necessário para desenvolver uma ideia é bem mais elevado do que os métodos tradicionais, na qual uma ansiedade passa a ser alimentada, gerando uma certa desconfiança na entrega do produto final.</p>
Tecnologias com foco em uma Stack padrão de desenvolvimento	<p><u>Ponto Positivo</u></p> <p>O uso de uma única Stack de desenvolvimento ajuda no ganho de maiores habilidades e facilidade na evolução do time, permitindo a construção de padrões vencedores ao longo do tempo, na qual traz uma maior agilidade na entrega de funcionalidades mais consistentes e de maior qualidade.</p> <p><u>Ponto Negativo</u></p> <p>Adaptar-se ao que a Stack oferece como proposta para solução de problemas que nem sempre é a melhor e mais produtivo.</p>

<p>Arquitetura distribuída baseada em troca de mensagens</p>	<p><u>Ponto Positivo</u></p> <p>O desenvolvimento de uma solução bem arquitetada com o uso do melhor serviço de nuvem para solucionar o problema, aliado com os princípios SOLID foi a chave para construção de uma solução reutilizável, fácil manutenção e maior qualidade.</p> <p><u>Ponto Negativo</u></p> <p>Exige um nível de senioridade do time de desenvolvimento para conseguir fazer tudo se comunicar e funcionar muito bem.</p>
<p>Infraestrutura 100% Cloud</p>	<p><u>Ponto Positivo</u></p> <p>Custos baseados na necessidade e uso, na qual permite a construção de soluções sofisticadas de alto nível com menor custo e caso seja necessária uma evolução que ofereça maior disponibilidade e escalabilidade é muito mais simples e rápido.</p> <p><u>Ponto Negativo</u></p> <p>Exige profissionais com habilidades (estes são bem mais caros e escassos) na nuvem adotada e que exatamente sabem fazer boas escolhas, levando em consideração o melhor custo/benefícios. Caso não tenha estes profissionais, a fatura pode sair bem cara!</p>
<p>Multiplataforma Mobile e Web</p>	<p><u>Ponto Positivo</u></p> <p>Reuso de código, única equipe e abrangência na adoção da plataforma permitindo aos usuários escolherem o OS de sua preferência.</p> <p><u>Ponto Negativo</u></p> <p>Exige maior esforço na construção das funcionalidades e manutenção.</p>

Gerenciador de identidades corporativa AAD e segurança	<p><u>Ponto Positivo</u></p> <p>Simplesmente fazer uso de um serviço com altíssima disponibilidade e delegar ao proprietário a entrega de um produto sempre seguro.</p> <p><u>Ponto Negativo</u></p> <p>Pouca flexibilidade em soluções customizadas.</p>
Performance	<p><u>Ponto Positivo</u></p> <p>Com o uso do modelo PaaS e escolha do plano e nível de serviço adequado foi essencial para entregar uma solução performática ao longo do tempo com as perspectivas de crescimento.</p> <p><u>Ponto Negativo</u></p> <p>Pouca interferência nas configurações de nível mais baixo para solucionar algum problema de compatibilidade entre a tecnologia de desenvolvimento com a infraestrutura (Limitações).</p>

8. Conclusão

Toda solução de desenvolvimento de software seja ela para qual for a área de negócio, sempre exigirá ações dedicadas e específicas que são da natureza da empresa, do setor e das pessoas envolvidas para a melhor aderência e adoção.

O trabalho de imersão realizado inicialmente por profissionais experientes em experiência do usuário foi decisivo para a construção das funcionalidades mais aderentes aos usuários.

Com o uso de abordagens ágeis e foco no problema/solução, foi possível priorizar as funcionalidades mais importantes que contribuíram para a concentração de energia nos itens de trabalho certo de todos os envolvidos.

O time de arquitetura totalmente alinhado com os objetivos e concentração nos recursos tecnológicos necessários para cumprimento das metas de negócios e habilidades dos times de desenvolvimento.

Portanto, os objetivos deste desafio foram satisfeitos e durante todo este período de imersão, análise, implementação e implantação geraram algumas lições aprendidas:

1. Envolvimento dos usuários da ponta na solução, pois eles serão os principais usuários;
2. Monitoramento centralizado, isso faz falta para uma análise mais rápida para tomada de decisões na identificação de problemas – deve ser implementado em breve;
3. Devido a granularidade na implementação das funções foi essencial o processo de automação de distribuição da infraestrutura (Infraestrutura como código) juntamente com os pacotes dos sistemas da plataforma usando a ferramenta de CI/CD Azure DevOps;
4. O API Gateway e gerenciador de identidades AAD foram fundamentais para garantir a segurança e a construção de APIs como ativos reutilizáveis;
5. Estruturação dos desenvolvimentos em código baseado na arquitetura limpa foi fundamental para construção de uma solução limpa e de fácil manutenção;
6. A construção de testes automatizados precisa ser aprimorada para maior confiança na evolução da plataforma e aceleração em novas publicações no ambiente produtivo.

Referências

ABRAINC. **ABRAINC Explica: A importância da Construção Civil para impulsionar a economia brasileira.** São Paulo, 28 de junho de 2021. Disponível em: <https://www.abrainc.org.br/abrainc-explica/2021/06/28/abrainc-explica-a-importancia-da-construcao-civil-para-impulsionar-a-economia-brasileira/> . Acesso em: 10 de dezembro de 2021.

MICRSOFT. **O Microsoft Intune é um provedor de MDM e de MAM para dispositivos.** 23 de dezembro de 2021. Disponível em: <https://docs.microsoft.com/pt-br/mem/intune/fundamentals/what-is-intune>. Acesso em: 01 de janeiro de 2022.

MICRSOFT. **Padrão de Request-Reply assíncrono.** 09 de dezembro de 2021. Disponível em: <https://www.abrainc.org.br/abrainc-explica/2021/06/28/abrainc->

[explica-a-importancia-da-construcao-civil-para-impulsionar-a-economia-brasileira/](#).

Acesso em: 15 de dezembro de 2021.

BROWN, Simon. **O modelo C4 de documentação para Arquitetura de Software.**

01 de agosto de 2018. Disponível em: <https://www.infoq.com/br/articles/C4-architecture-model/>. Acesso em: 01 de dezembro de 2021.

CAMPOMORI, Cleber. **Princípios SOLID: Single Responsibility Principle.** 10 de

dezembro de 2022. Disponível em: <https://www.treinaweb.com.br/blog/principios-solid-single-responsability-principle>. Acesso em: 10 de dezembro de 2022.

Microsoft. **Common web application architectures.** 11 de janeiro de 2022. Disponível

em: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>. Acesso em: 10 de dezembro de 2022.

Microsoft. **Visão geral do Application Insights.** 05 de dezembro de 2022. Disponível

em: <https://learn.microsoft.com/pt-br/azure/azure-monitor/app/app-insights-overview?tabs=net>. Acesso em: 10 de dezembro de 2022.

Microsoft. **Cloud Adoption Framework guidance.** 01 de janeiro de 2022. Disponível

em: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/>. Acesso em: 10 de dezembro de 2022.

TEIXEIRA, Fabricio. **Google Design Sprint: como funciona e como aplicar no seu projeto.** 01 de setembro de 2015. Disponível em: <https://brasil.uxdesign.cc/google-design-sprint-como-funciona-e-como-aplicar-no-seu-projeto-279107363659>.

Acesso em: 10 de dezembro de 2022.