

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

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

Projeto Integrado

Relatório Técnico

ShiftGenius – Controle e Gestão de Plantões

Rodrigo Gimenes Rodrigues

Mogi das Cruzes
Abril de 2023.

Projeto Integrado – Arquitetura de Software Distribuído

Sumário

Projeto Integrado – Arquitetura de Software Distribuído	2
1. Introdução	3
2. Especificação Arquitetural da solução	5
2.1 Restrições Arquiteturais	5
2.2 Requisitos Funcionais	5
2.3 Requisitos Não-funcionais	7
2.4 Mecanismos Arquiteturais	8
3. Modelagem Arquitetural	9
3.1 Diagrama de Contexto	11
3.2 Diagrama de Container	12
3.3 Diagrama de Componentes	14
4. Avaliação da Arquitetura (ATAM)	16
4.1 Análise das abordagens arquiteturais	16
4.2 Cenários	17
4.3 Evidências da Avaliação	17
5. Avaliação Crítica dos Resultados	23
6. Conclusão	24
Referências	25

1. Introdução

Este projeto envolve o desenvolvimento de um novo produto de Controle e Gestão de Plantões (**ShiftGenius**) para a ShiftMed e seus médicos contratados.

O setor de saúde é um setor crítico que requer processos simplificados para agendar e monitorar o pessoal médico, garantindo operações assertivas e confiáveis e a prestação de cuidados de qualidade.

A ShiftMed reconhece a importância de gerenciar eficientemente os plantões para seus médicos, e este projeto visa atender a essa necessidade, criando uma plataforma que facilite a distribuição e o controle de Plantões.

O controle de turnos é fundamental para a gestão de recursos humanos em empresas e instituições que trabalham em regime de plantão. Ele permite a distribuição equilibrada de horas trabalhadas entre os funcionários e reduz a possibilidade de erros e falhas causados por sobrecarga de trabalho e fadiga.

Segundo pesquisa da Fundação Getúlio Vargas (FGV), cerca de 60% das empresas brasileiras que trabalham em regime de plantão enfrentam problemas com o controle de turnos, o que pode levar a atrasos, falta de pessoal em momentos críticos e desmotivação dos funcionários.

Além disso, estudos mostram que a fadiga acumulada por trabalhadores em regime de plantão pode afetar negativamente a qualidade do atendimento e aumentar o risco de erros e acidentes, principalmente na área da saúde. De acordo com uma pesquisa do Conselho Federal de Medicina, a fadiga é a principal causa de erros médicos no Brasil. Para lidar com esses problemas, empresas e instituições têm adotado soluções de controle de turnos, que incluem sistemas informatizados de gestão de escalas e plataformas de comunicação para garantir o acesso rápido e eficiente às informações sobre os plantões. Segundo a mesma pesquisa da FGV, as empresas que adotam soluções de controle de turnos relatam uma redução média de 30% nas falhas relacionadas à gestão de recursos humanos.

Em suma, o controle de turnos é um elemento essencial para garantir a eficiência e a segurança no trabalho em regime de plantão. Ele permite a distribuição equilibrada de horas trabalhadas entre os funcionários, reduzindo a possibilidade de erros e falhas causados por sobrecarga de trabalho e fadiga, além de melhorar a qualidade do atendimento e reduzir os riscos de acidentes e erros.

O processo atual de gerenciamento de plantões e pagamentos na ShiftMed é demorado, ineficiente e propenso a erros.

O processo manual de agendamento e gerenciamento de plantões geralmente leva à falta de comunicação entre a ShiftMed, médicos e contratados, causando conflitos de agendamento e plantões perdidos.

A falta de uma plataforma centralizada para gerenciar todos os aspectos do agendamento de plantões e do processamento de pagamentos é um problema significativo para a ShiftMed, o que afeta a qualidade do atendimento ao paciente e a satisfação de todas as partes envolvidas.

Além disso, exige um enorme esforço de trabalho manual, causando retrabalho e impedindo que a empresa possa escalar sua capacidade de gerenciamento de plantões.

A motivação por trás deste projeto é criar um processo mais eficiente e eficaz para gerenciar plantões e pagamentos para médicos, contratados e ShiftMed.

Ao implementar uma plataforma centralizada, o objetivo é reduzir os conflitos de agendamento, aumentar a precisão do processamento de pagamentos e melhorar a qualidade do atendimento ao paciente.

A conclusão bem-sucedida deste projeto levará a uma redução significativa nas tarefas administrativas, permitindo que a ShiftMed se concentre na prestação de um melhor atendimento ao paciente e permitindo que a empresa esteja preparada para crescer e escalar seu volume de negócios em tempo relativamente mais curto, e com custo menor.

O principal objetivo deste projeto é desenvolver um produto interno para a ShiftMed que permita o gerenciamento eficiente das informações, distribuição e controle de plantões, juntamente com a capacidade de gerar relatórios e facilitar a comunicação entre todas as partes envolvidas.

Para atingir este objetivo, foram identificados os seguintes objetivos:

- Desenvolver uma plataforma que permita à ShiftMed gerenciar contratos, plantões e monitorar cuidados médicos.
- Criar uma interface amigável para que médicos e contratados monitorem seus plantões, recebam notificações e gerem relatórios sobre seu histórico de plantões.
- Integrar a plataforma com os bancos para garantir o processamento de pagamentos oportuno e preciso para os médicos.
- Desenvolver uma interface segura e confiável que permita uma integração perfeita com o Website ShiftMed existente, proporcionando uma experiência unificada para todas as partes envolvidas.
- Permitir que hospitais/planos de saúde/clínicas gerenciem o atendimento médico e a prestação de cuidados.

No geral, o escopo deste projeto inclui o desenvolvimento de um produto interno para a ShiftMed que permita o gerenciamento eficiente de informações, distribuição e controle de plantões, juntamente com a capacidade de gerar relatórios e facilitar a comunicação entre todas as partes envolvidas e fornecer processamento de pagamento seguro e integração perfeita com o Website ShiftMed.

2. 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 permitam visualizar a macroarquitetura da solução.

2.1 Restrições Arquiteturais

R1: O software deve ser desenvolvido em Salesforce
R2: O Salesforce é um provedor de PaaS, o que significa que o aplicativo deve ser desenvolvido usando os recursos da plataforma Salesforce e seguir suas diretrizes de desenvolvimento e práticas recomendadas.
R3: O aplicativo deve aproveitar a arquitetura baseada em nuvem fornecida pelo Salesforce, utilizando sua infraestrutura, serviços e recursos de escalabilidade.
R4: Se o aplicativo precisar se integrar a sistemas externos, como APIs ou bancos de dados de terceiros, a integração deverá ser projetada e implementada seguindo as práticas recomendadas de integração do Salesforce.
R5: O Salesforce tem vários limites e restrições de governança (Governor limits) em recursos como chamadas de API, armazenamento de dados e processamento. O aplicativo deve ser projetado para funcionar dentro dessas restrições e otimizar o uso de recursos.
R6: O aplicativo deve aderir ao modelo de segurança do Salesforce, implementando controles de acesso, permissões de usuário e medidas de segurança de dados para garantir a confidencialidade, integridade e disponibilidade dos dados.

2.2 Requisitos Funcionais

ID	Descrição	Dificuldade (B/M/A)*	Prioridade B/M/A
RF01	Permitir o registro e gerenciamento de contatos, incluindo médicos, coordenadores e colaboradores. Capturar informações relevantes, como nome, detalhes de contato, especialidades e funções.	M	A
RF02	Possibilitar o registro e acompanhamento de contratos com hospitais ou entidades privadas. Armazenar detalhes do contrato, termos, datas de início e término e contatos associados.	M	A

RF03	Fornecer a capacidade de registrar e gerenciar contas para hospitais e entidades privadas. Incluir detalhes da conta, informações de contato e contratos associados.	M	M
RF04	Permitir a criação e gestão de escalas de Plantões. Capturar detalhes do plantão, como data, hora, duração, local, médicos designados e contatos associados.	A	A
RF05	Gerar relatórios para gerenciamento de plantões, fornecendo insights sobre plantões, médicos designados e outros dados relevantes. Incluir opções de filtragem e classificação para relatórios personalizados.	A	M
RF06	Gerar relatórios financeiros com base em plantões e contratos, fornecendo uma visão geral de pagamentos, receitas e despesas. Incluir opções de filtragem e agregação para análise de dados.	A	M
RF07	Integrar o aplicativo com o website da ShiftMed para sincronizar dados relevantes, como horários de plantão, disponibilidade de médicos e informações de contato.	M	M
RF08	Estabelecer integração com o Banco Santander para facilitar transações financeiras, como geração de arquivos de pagamento ou conciliação de dados financeiros.	A	M
RF09	Implementar um sistema de notificação para enviar notificações automatizadas a médicos, coordenadores e colaboradores sobre atribuições de plantões, atualizações ou lembretes.	M	A
RF10	Definir e configurar funções e permissões de usuário para controle de acesso, garantindo	M	A

	níveis de autorização apropriados para ações como aprovações de plantões ou modificações de contrato.		
RF11	Definir e configurar perfis e papéis de usuários no aplicativo. Cada perfil deve ter permissões e níveis de acesso específicos com base nas responsabilidades do trabalho, garantindo visibilidade e funcionalidade de dados apropriadas para diferentes tipos de usuário.	M	A
RF12	Criar painéis personalizáveis dentro do aplicativo para fornecer aos usuários painéis personalizados e acesso rápido a informações relevantes. Os painéis podem exibir métricas importantes, relatórios ou visualizações de dados específicas com base nas preferências e requisitos do usuário.	M	M

*B=Baixa, M=Média, A=Alta.

2.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve estar disponível e acessível aos usuários 24 horas por dia, 7 dias por semana, com tempo mínimo de inatividade planejado para manutenção ou atualizações.	A
RNF02	O sistema deve lidar com um mínimo de 100 usuários simultâneos sem degradação significativa do desempenho.	A
RNF03	A interface do usuário deve ser intuitiva e fácil de navegar.	M
RNF04	O sistema deve fornecer mensagens de erro claras e concisas para ajudar os usuários a resolver problemas.	M

RNF05	O sistema deve oferecer suporte a vários idiomas para acomodar usuários de diferentes regiões.	M
RNF06	O sistema deve ter design responsivo para garantir a compatibilidade entre vários dispositivos e tamanhos de tela.	M
RNF07	A autenticação e a autorização do usuário devem ser implementadas para garantir o acesso seguro ao aplicativo.	A
RNF08	O acesso a dados confidenciais, como informações pessoais ou registros financeiros, deve ser restrito apenas a pessoal autorizado.	A
RNF09	O sistema deve ter mecanismos para impedir o acesso não autorizado, como tempos limite de sessão e requisitos de complexidade de senha.	M
RNF10	O sistema deve ser capaz de lidar com um número crescente de usuários e dados à medida que a base de usuários cresce.	M
RNF11	O sistema deve suportar fácil integração com outros sistemas ou módulos para acomodar melhorias ou adições futuras.	M
RNF12	O sistema deve aderir aos padrões da indústria e às melhores práticas de desenvolvimento e segurança de software.	A
RNF13	O sistema deve ter documentação abrangente e atualizada, incluindo manuais do usuário, guias de instalação e documentação de arquitetura do sistema.	M

2.4 Mecanismos Arquiteturais

Esta seção fornece uma visão geral dos mecanismos que compõem a arquitetura de software do aplicativo ShiftGenius. A arquitetura de software é apresentada com base em três estados: (1) análise, (2) projeto e (3) implementação. Cada estado aborda diferentes aspectos da arquitetura.

Análise	Design	Implementação
Persistência	ORM	Salesforce Object Query Language (SOQL) ou Salesforce Object Search Language (SOSL)
Front end	Salesforce Lightning Components (LWC e Aura)	Salesforce Lightning
Integração com Sistemas Legados	ETL e ESB	Mulesoft
Log do sistema	Salesforce Apex Logger	Salesforce Platform
Teste de Software	Apex testing ou Lightning Testing Service (LTS),	Salesforce Platform
Tratamento de exceções:	Blocos Try/Catch e classes customizadas	Salesforce Platform
Formato de distribuição/Deploy	Change Sets, Salesforce CLI	SFDX (Salesforce DX)

3. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta para o ShiftGenius, com o objetivo de permitir um completo entendimento da arquitetura do software, embora a implementação em si não esteja incluída no escopo deste trabalho.

A arquitetura proposta para o ShiftGenius é baseada em uma abordagem de arquitetura em camadas, que separa as diferentes responsabilidades e funcionalidades do sistema em camadas distintas. A arquitetura é projetada para ser implementada na plataforma Salesforce, aproveitando os recursos e serviços fornecidos pela plataforma.

A arquitetura em camadas proposta inclui as seguintes camadas principais:

1 - Camada de Apresentação: Esta camada é responsável pela interface do usuário e interação com os usuários finais. Ela inclui os componentes de interface do usuário do Salesforce, como Lightning Components (LWC e Aura), que são utilizados para criar as telas e elementos interativos do aplicativo.

2 - Camada de Lógica de Negócios: Esta camada contém a lógica de negócios do aplicativo. Ela processa as solicitações do usuário, executa as regras de negócios e interage com a camada de dados. Na plataforma Salesforce, a lógica de negócios é implementada usando Apex, uma linguagem de programação específica da plataforma.

3 - Camada de Acesso a Dados: Esta camada é responsável pelo acesso e gerenciamento dos dados do aplicativo. Ela inclui os objetos de dados personalizados do Salesforce, que são usados para modelar os diferentes tipos de dados necessários para o ShiftGenius. O acesso aos dados é feito usando o Salesforce Object Query Language (SOQL) ou o Salesforce Object Search Language (SOSL).

4 - Camada de Integração: Esta camada lida com a integração do ShiftGenius com outros sistemas ou serviços externos, como o site da ShiftMed ou o Banco Santander. A plataforma Mulesoft pode ser utilizada para facilitar a integração e a troca de dados entre diferentes sistemas.

5 - Camada de Persistência: Esta camada é responsável pelo armazenamento e recuperação dos dados do aplicativo. Na plataforma Salesforce, a persistência é tratada pelo próprio Salesforce, usando seu mecanismo de banco de dados subjacente.

Além dessas camadas principais, a arquitetura também considera aspectos como segurança, autenticação, autorização, tratamento de exceções e testes de software. Esses aspectos são implementados em diferentes camadas, de acordo com as melhores práticas e recursos fornecidos pela plataforma Salesforce.

Para documentar a arquitetura de software, optou-se por utilizar o modelo C4 (Contexto, Contêineres, Componentes e Código) devido à sua eficácia na representação de diferentes níveis de abstração e na comunicação de conceitos arquiteturais de forma clara e concisa. O modelo C4 oferece uma estrutura organizada para descrever a arquitetura em diferentes níveis de detalhe.

No contexto deste trabalho, serão apresentados três dos quatro níveis do modelo C4: Contexto, Contêineres e Componentes. O nível de Código será abordado na próxima seção (5), onde serão detalhados aspectos relacionados à implementação.

A utilização do modelo C4 facilita a compreensão e comunicação da arquitetura do ShiftGenius, fornecendo uma estrutura clara e consistente para representar os diferentes elementos e relacionamentos do sistema.

Mais informações a respeito podem ser encontradas aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>.

A arquitetura proposta para o ShiftGenius é baseada em uma abordagem de arquitetura em camadas, que separa as diferentes responsabilidades e funcionalidades do sistema em camadas distintas. A arquitetura é projetada para ser implementada na plataforma Salesforce, aproveitando os recursos e serviços fornecidos pela plataforma.

3.1 Diagrama de Contexto

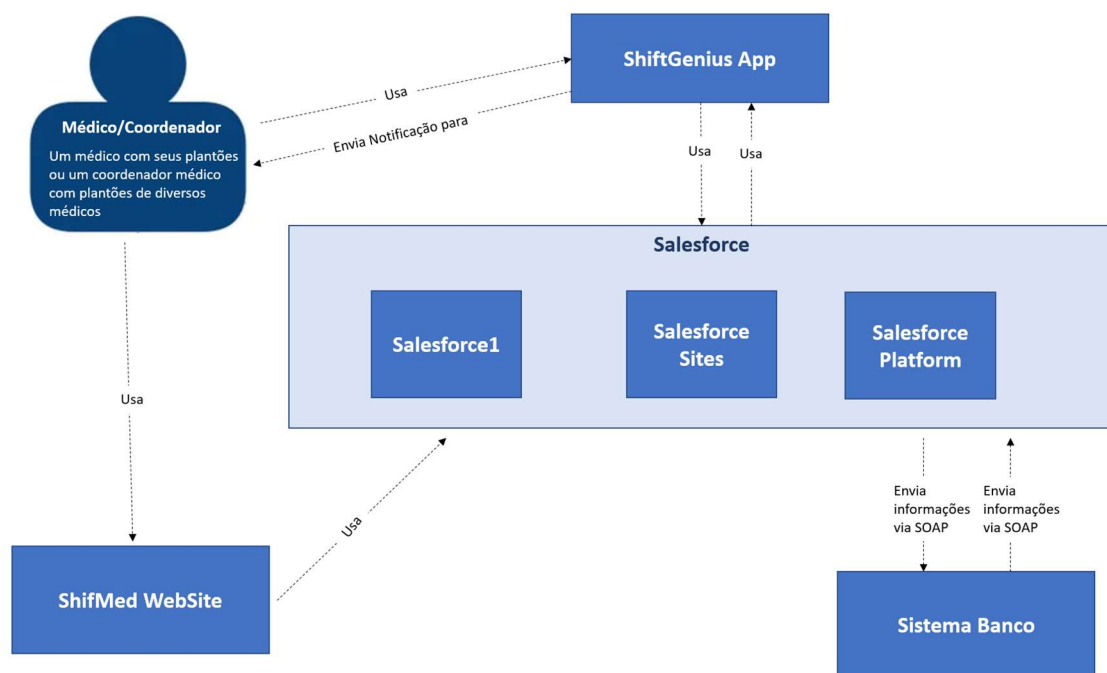


Figura 1 - Visão Geral da Solução.

A Figura 1 apresenta a especificação do diagrama geral da solução proposta, que ilustra os principais módulos e suas interfaces. Nesta arquitetura, os usuários médicos e coordenadores têm acesso ao aplicativo ShiftGenius, seja através da web, autenticando-se por meio do site ShiftMed, ou diretamente pelo aplicativo Salesforce1 em seus dispositivos móveis. O ShiftGenius é responsável pelo controle de plantões e gerenciamento das informações relacionadas.

O ShiftGenius integra-se ao sistema bancário, permitindo o envio e recebimento de informações de pagamentos. Essa integração é realizada por meio de uma comunicação SOAP, onde o ShiftGenius envia informações relevantes ao sistema bancário para processamento e recebe atualizações de status de pagamento.

Além disso, o ShiftGenius é capaz de enviar notificações aos usuários médicos e coordenadores. Essas notificações são direcionadas aos dispositivos móveis dos usuários através do aplicativo Salesforce1, mantendo-os informados sobre alterações nos plantões, atualizações importantes e outras informações relevantes.

O website do ShiftMed desempenha um papel crucial na autenticação dos usuários que acessam o aplicativo por meio da web. Os usuários são autenticados pelo website ShiftMed antes de serem redirecionados para o ShiftGenius, garantindo a segurança e o controle de acesso adequados ao aplicativo.

Em resumo, a arquitetura proposta para o ShiftGenius oferece uma solução abrangente para o controle de plantões, integração bancária e envio de notificações. Ela permite que os usuários médicos e coordenadores acessem e interajam com o aplicativo por meio de diferentes canais, garantindo uma experiência eficiente e segura.

3.2 Diagrama de Container

O Diagrama de Container ilustra a distribuição e organização dos componentes que compõem o sistema de software da aplicação.

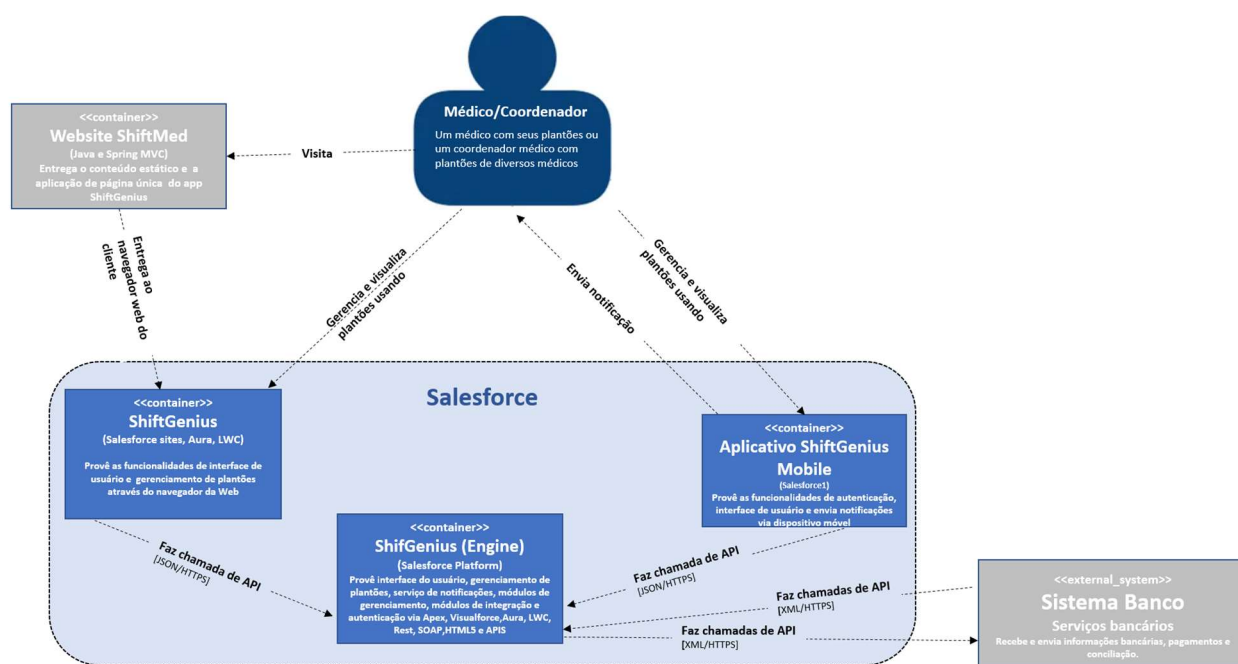


Figura 2 – Diagrama de container.

A seguir, são apresentados os principais componentes e suas interações:

1. **Salesforce Platform:** O Salesforce Platform é o componente central da aplicação e abriga os principais serviços e funcionalidades. Ele é responsável por gerenciar o armazenamento de dados, a lógica de negócio e a integração com outros sistemas. A decisão de construir a aplicação na plataforma Salesforce foi tomada, levando em consideração sua capacidade de escalabilidade, segurança e recursos integrados.

2. **Website ShiftMed:** O Website ShiftMed é um componente externo que interage com o Salesforce Platform. Ele fornece uma interface web para médicos e coordenadores de plantão acessarem e gerenciarem seus registros de plantão. Os usuários autenticam-se no Website ShiftMed para acessar as funcionalidades do sistema.

3. **ShiftGenius:** O ShiftGenius é um componente do sistema construído na plataforma Salesforce. Ele permite a gestão e visualização dos registros de plantão. Os médicos e coordenadores de plantão utilizam o ShiftGenius por meio do Website ShiftMed e do aplicativo móvel Salesforce1.

4. **Aplicativo móvel Salesforce1:** O aplicativo móvel Salesforce1 é uma interface específica para dispositivos móveis que permite aos médicos e coordenadores de plantão acessarem e gerenciarem seus registros de plantão por meio de seus smartphones ou tablets. O aplicativo envia notificações para os usuários, mantendo-os informados sobre suas escalas e outras informações relevantes.

5. **Sistema Bancário:** O sistema bancário é um componente externo integrado à aplicação. Ele é responsável por processar informações relacionadas a pagamentos, como o recebimento de informações sobre pagamentos e o envio de informações de pagamento ao ShiftMed.

No Diagrama de Container, esses componentes são representados como contêineres, cada um com sua função e interações específicas. A escolha da tecnologia Salesforce para o desenvolvimento da aplicação é refletida no diagrama, evidenciando a integração com o Salesforce Platform, o uso do Website ShiftMed como interface web e a presença do aplicativo móvel Salesforce1.

Essas decisões tecnológicas foram tomadas considerando os requisitos do sistema, a experiência do usuário e a capacidade de escalabilidade e segurança oferecida pela plataforma Salesforce. O Diagrama de Container oferece uma visão geral da arquitetura do sistema, permitindo entender como os componentes estão distribuídos e organizados.

3.3 Diagrama de Componentes

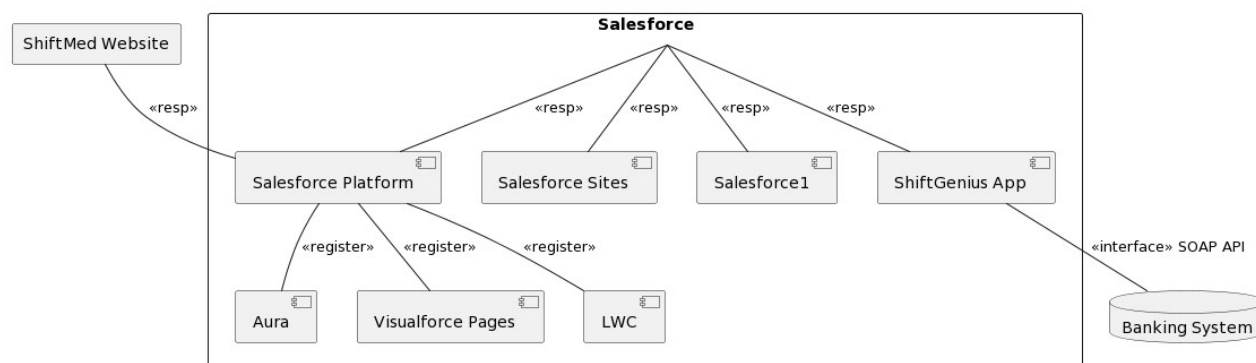


Figura 3 – Diagrama de Componentes

O diagrama de componentes do aplicativo, baseado em UML, representa a arquitetura e as interfaces entre os elementos. Utiliza os seguintes estilos/padrões arquitetônicos:

1 - Salesforce Platform: O aplicativo ShiftGenius foi criado na Salesforce Platform, que serve como base para toda a solução. Ele fornece uma plataforma robusta e escalável para desenvolver e implantar aplicativos corporativos.

2 - Arquitetura Orientada a Serviços (SOA): O aplicativo ShiftGenius se comunica com o sistema bancário por meio de uma API SOAP, que é um exemplo de arquitetura orientada a serviços. SOA permite acoplamento flexível e interoperabilidade entre diferentes sistemas.

3 - Arquitetura cliente-servidor: o aplicativo ShiftGenius, o Salesforce Sites, o Salesforce1 e o site ShiftMed representam os componentes do lado do cliente, enquanto a Salesforce Platform e o sistema bancário servem como componentes do lado do servidor. Essa arquitetura permite a separação das responsabilidades do cliente e do servidor.

4 - Padrão MVC (Model-View-Controller): LWC, Aura e Visualforce Pages são estruturas de interface do usuário fornecidas pela Salesforce Platform. Essas estruturas seguem o padrão MVC, onde os modelos lidam com o gerenciamento de dados, as exibições lidam com a apresentação da interface do usuário e os controladores lidam com as interações do usuário e a lógica de negócios.

5 – Arquitetura Multi-tier: Os componentes no diagrama podem ser categorizados em diferentes tiers ou camadas. Os componentes do lado do cliente (ShiftGenius App, Salesforce Sites, Salesforce1 e ShiftMed Website) representam a camada de apresentação, a Salesforce Platform serve como a camada de aplicativo e o Banking System representa a camada de dados. Essa separação de preocupações(concerns) facilita a escalabilidade, a facilidade de manutenção e a modularidade.

Conforme diagrama apresentado na Figura 3, as entidades participantes da solução são:

- **Componente 1 - Aplicativo ShiftGenius:** O aplicativo ShiftGenius construído na plataforma Salesforce. Permite que os usuários/coordenadores médicos gerenciem e visualizem seus plantões. Ele utiliza componentes do Salesforce, como LWC, Aura e Visualforce Pages, para a interface do usuário.
- **Componente 2 - Salesforce Sites:** Um recurso do Salesforce que permite que o site da ShiftMed seja hospedado e acessado. Ele lida com a autenticação e redireciona os usuários para o aplicativo ShiftGenius.
- **Componente 3 - Salesforce1:** O aplicativo móvel fornecido pelo Salesforce. Ele aproveita os recursos da plataforma Salesforce para fornecer acesso móvel ao aplicativo ShiftGenius e seus recursos.
- **Componente 4 - Salesforce Platform:** A plataforma subjacente que alimenta o aplicativo ShiftGenius e outros componentes do Salesforce. Ele hospeda a lógica do aplicativo, o modelo de dados e os recursos de integração.
- **Componente 5 - LWC (Lightning Web Components):** Uma estrutura de componentes web fornecida pelo Salesforce. Ele permite a criação de componentes web personalizados para a interface do usuário do aplicativo ShiftGenius.
- **Componente 6 - Aura:** Outra estrutura de componentes fornecida pelo Salesforce. Ele é usado para desenvolver componentes personalizados no aplicativo ShiftGenius.
- **Componente 7 - Visualforce Pages:** Uma tecnologia dentro do Salesforce usada para criar interfaces de usuário personalizadas. Ele é utilizado no aplicativo ShiftGenius para construir páginas e interfaces específicas.
- **Componente 8 - Site ShiftMed:** Um site separado responsável pela autenticação e redirecionamento do usuário para o aplicativo ShiftGenius. Ele interage com o componente Salesforce Sites.
- **Componente 9 - Sistema Bancário:** Sistema externo responsável pelas operações bancárias. Ele se integra ao aplicativo ShiftGenius usando a API SOAP para troca de informações.

Em termos de reutilização de componentes, o aplicativo ShiftGenius aproveita vários componentes fornecidos pelo Salesforce, como as páginas Salesforce Platform, LWC, Aura e Visualforce. Esses componentes são pré-existent e reutilizáveis no ecossistema do Salesforce. Além disso, navegadores da Web são utilizados para acessar a interface da Web do ShiftGenius App.

Os componentes proprietários incluem a própria Salesforce Platform e o aplicativo móvel Salesforce1, que são adquiridos do Salesforce.

Os componentes que precisam ser desenvolvidos especificamente para o aplicativo ShiftGenius incluem componentes personalizados criados usando LWC, Aura e Visualforce Pages, bem como o site da ShiftMed para lidar com autenticação e redirecionamento

4. 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.

4.1 Análise das abordagens arquiteturais

A solução arquitetural proposta para o aplicativo ShiftGenius visa abordar os principais atributos de qualidade, como interoperabilidade, usabilidade, manutenibilidade, escalabilidade e segurança.

O Método de Análise de Compensação de Arquitetura (ATAM) é usado para avaliar o projeto arquitetônico com base em cenários que representam esses atributos de qualidade.

A **interoperabilidade** é um atributo crucial, pois o sistema precisa se comunicar com sistemas externos que podem usar diferentes tecnologias. Isso garante a integração perfeita e a troca de dados com sistemas externos, permitindo uma interoperabilidade suave.

A **usabilidade** é dada grande importância, enfatizando a necessidade de o sistema fornecer uma interface amigável e experiência de usuário intuitiva. O objetivo é garantir que os usuários possam navegar facilmente pelo aplicativo e executar tarefas de forma eficiente.

A **manutenibilidade** é outro aspecto crítico, considerando a sustentabilidade de longo prazo do sistema. A arquitetura enfatiza a modularidade, permitindo manutenção e melhorias mais fáceis. Isso permite que a equipe de desenvolvimento faça alterações e atualizações no sistema com impacto mínimo em outros componentes.

A **escalabilidade** é um atributo de qualidade essencial, pois espera-se que o sistema lide com um número significativo de usuários e dados. A arquitetura foi projetada para acomodar os requisitos de escalabilidade, garantindo que o sistema possa ser dimensionado com eficiência para lidar com cargas de usuário maiores e volumes de dados crescentes.

A **segurança** é de suma importância, e a arquitetura aborda garantindo mecanismos de autenticação seguros e medidas de proteção de dados. O sistema é projetado para proteger informações confidenciais e fornecer um ambiente seguro para usuários e dados.

Ao considerar esses principais atributos de qualidade e conduzir uma análise com base em cenários relevantes, a solução arquitetural proposta visa fornecer uma plataforma robusta e confiável para o aplicativo ShiftGenius, permitindo integração perfeita, experiência amigável, fácil manutenção, escalabilidade e operações seguras.

Atributos de Qualidade	Cenários	Importância	Complexidade
Interoperabilidade	Cenário 1: O sistema deve se comunicar com sistemas de outras tecnologias.	A (Quantitativa)	M (Quantitativa)

Usabilidade	Cenário 2: O sistema deve prover boa usabilidade e interface intuitiva.	A (Qualitativa)	A (Qualitativa)
Manutenibilidade	Cenário 3: O sistema deve ser modularizado e ter a manutenção facilitada.	A (Qualitativa)	M (Qualitativa)
Escalabilidade	Cenário 4: O sistema deve lidar com grande número de usuários e dados	A (Quantitativa)	A (Quantitativa)
Segurança	Cenário 5: O sistema deve garantir autenticação segura e proteção aos dados	A (Qualitativa)	A (Qualitativa)

4.2 Cenários

Cenário 1 - Interoperabilidade: O aplicativo ShiftGenius deve se comunicar com êxito com sistemas externos usando a integração de API.

Cenário 2 - Usabilidade: Os usuários devem ser capazes de navegar e interagir facilmente com a interface do usuário do aplicativo ShiftGenius.

Cenário 3 - Manutenibilidade: O aplicativo ShiftGenius deve ser projetado para fácil manutenção e aprimoramentos futuros, permitindo que os desenvolvedores façam alterações sem esforço significativo ou introduzindo novos defeitos.

Cenário 4 - Escalabilidade: O aplicativo ShiftGenius deve lidar com um grande número de usuários simultâneos e plantões sem degradação de desempenho.

Cenário 5 - Segurança: O aplicativo ShiftGenius deve garantir acesso seguro, proteção de dados e aderência aos regulamentos de privacidade.

4.3 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:	Interoperabilidade
Requisito de Qualidade:	O sistema deve se comunicar com outras tecnologias.
Preocupação:	
O sistema deve ter como resposta a uma requisição uma saída de fácil leitura por outro componente.	
Cenário(s):	
Cenário 1	
Ambiente:	
Aplicativo ShiftGenius em execução.	
Estímulo:	
O usuário realiza uma transação financeira no aplicativo ShiftGenius, que requer uma interação com o sistema bancário.	
Mecanismo:	
Implementar a integração via SOAP, utilizando os protocolos e padrões adequados para a comunicação com o sistema bancário.	
Medida de resposta:	
A transação financeira é processada com sucesso, sem erros ou interrupções na comunicação com o sistema bancário.	
Considerações sobre a arquitetura:	
Riscos:	Instabilidades na rede podem resultar em conexões lentas ou perda de pacotes, afetando a interoperabilidade entre o aplicativo ShiftGenius e o sistema bancário.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	Os usuários devem ser capazes de navegar e interagir facilmente com a

	interface do usuário do aplicativo ShiftGenius..
Preocupação:	
A interface do usuário deve ser intuitiva, fornecendo uma experiência de uso agradável e eficiente.	
Cenário(s):	
Cenário 2	
Ambiente:	
Aplicativo ShiftGenius em execução.	
Estímulo:	
O usuário acessa a tela de agendamento de plantões no aplicativo ShiftGenius.	
Mecanismo:	
Projetar uma interface de usuário intuitiva e amigável, com elementos de navegação claros e fluxo de interação natural.	
Medida de resposta:	
O usuário consegue navegar facilmente pela tela de agendamento de plantões, encontrar as opções desejadas e concluir a tarefa de forma rápida e eficiente.	
Considerações sobre a arquitetura:	
Riscos:	A falta de usabilidade pode resultar em baixa adoção do aplicativo pelos usuários e insatisfação com a experiência de uso.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Atributo de Qualidade:	Manutenibilidade
Requisito de Qualidade:	O aplicativo ShiftGenius deve ser projetado para fácil manutenção e aprimoramentos futuros, permitindo que os desenvolvedores façam alterações sem esforço significativo ou introduzindo novos defeitos.
Preocupação:	

O código do aplicativo deve ser modular, bem organizado e seguir boas práticas de programação para facilitar a manutenção e a evolução futura.	
Cenário(s):	
Cenário 3	
Ambiente:	
Desenvolvimento do aplicativo ShiftGenius	
Estímulo:	
É identificada a necessidade de adicionar uma nova funcionalidade ao aplicativo ShiftGenius.	
Mecanismo:	
Utilizar uma arquitetura modular, separando as responsabilidades em componentes independentes e seguindo padrões de design e boas práticas de programação.	
Medida de resposta:	
Os desenvolvedores conseguem adicionar a nova funcionalidade ao aplicativo com facilidade, sem impactar negativamente outras partes do sistema e sem introduzir novos defeitos.	
Considerações sobre a arquitetura:	
Riscos:	A falta de manutenibilidade pode resultar em dificuldades para realizar alterações no aplicativo, aumentando o tempo e o esforço necessários para implementar novas funcionalidades ou corrigir defeitos.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Atributo de Qualidade:	Escalabilidade
Requisito de Qualidade:	O aplicativo ShiftGenius deve lidar com um grande número de usuários simultâneos e plantões sem degradação de desempenho.
Preocupação:	

O sistema deve ser capaz de dimensionar verticalmente e horizontalmente para atender à demanda crescente de usuários e plantões.	
Cenário(s):	
Cenário 4	
Ambiente:	
Período de pico de utilização do aplicativo ShiftGenius	
Estímulo:	
Um grande número de usuários simultâneos acessa o aplicativo para agendar plantões.	
Mecanismo:	
Utilizar técnicas de dimensionamento vertical e horizontal, como balanceamento de carga, replicação de servidores e otimização de consultas de banco de dados.	
Medida de resposta:	
O aplicativo ShiftGenius lida com a carga de usuários e plantões sem experimentar degradação significativa de desempenho, mantendo tempos de resposta aceitáveis.	
Considerações sobre a arquitetura:	
Riscos:	A falta de escalabilidade pode resultar em lentidão ou falhas no aplicativo durante períodos de alta demanda, levando à insatisfação dos usuários.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O aplicativo ShiftGenius deve garantir acesso seguro, proteção de dados e aderência aos regulamentos de privacidade.
Preocupação:	
As informações do usuário e os dados sensíveis devem ser protegidos contra acesso não autorizado e a transmissão de dados deve ser criptografada.	
Cenário(s):	

Cenário 5	
Ambiente:	
Comunicação de dados sensíveis no aplicativo ShiftGenius	
Estímulo:	
O usuário realiza login no aplicativo ShiftGenius e envia informações pessoais e de pagamento..	
Mecanismo:	
Implementar autenticação segura, criptografia de dados em trânsito, proteção contra ataques de injeção de código, e seguir as melhores práticas de segurança e conformidade regulatória.	
Medida de resposta:	
A transação financeira é processada com sucesso, sem erros ou interrupções na comunicação com o sistema bancário.	
Considerações sobre a arquitetura:	
Riscos:	A falta de segurança pode resultar em violação de dados e comprometimento da privacidade dos usuários, além de potenciais implicações legais e regulatórias.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

5. *Avaliação Crítica dos Resultados*

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiteturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ex. de quadro resumo:

Ponto avaliado	Descrição
XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXX

6. Conclusão

Descreva, de forma sucinta, quais foram as lições aprendidas na execução do seu projeto arquitetural. Procure apresentá-las de tal forma que fiquem configurados os *trade-offs* da arquitetura produzida, como por exemplo, Segurança X Desempenho, Granularidade X Manutenibilidade, etc.

Aqui deve ser apresentado também tudo que se aprendeu com esse projeto, de modo a servir como ajuda para outros profissionais.

Também se faz necessário evidenciar as possibilidades de melhoria do projeto, caso se deseje dar continuidade a ele. Nesse sentido, indique possíveis ajustes ou melhorias arquiteturais, que possam vir a ser realizados futuramente.

Lições aprendidas (ex.):

1. xxxxxxxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxxxxxxx

Referências

Esse trabalho não requer revisão bibliográfica e, por isso, a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências empregadas em seu trabalho, relacione-as de acordo com as normas ABNT, disponíveis em [www.pucminas.br](http://portal.pucminas.br/imagadb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf), no link: http://portal.pucminas.br/imagadb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo**. Cidade: Editora, ano.

1. "Software Architecture in Practice" by Len Bass, Paul Clements, and Rick Kazman
2. "Patterns of Enterprise Application Architecture" by Martin Fowler
3. "Domain-Driven Design: Tackling Complexity in the Heart of Software" by Eric Evans
4. Salesforce Developer Documentation (<https://developer.salesforce.com/docs/>)
5. Salesforce Trailhead (<https://trailhead.salesforce.com/>)
6. "Apex Developer Guide" (<https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/>)
7. "Lightning Component Library" (<https://developer.salesforce.com/docs/component-library/overview/components>)
8. Salesforce Blogs and Community Forums (e.g., Salesforce Developer Forums, Salesforce Developer Blogs)
9. Documentando arquiteturas na prática com C4 model - <https://www.youtube.com/watch?v=aJZPKyElP6A>

