**UNIVERSIDADE FEDERAL DE GOIÁS**

**ENGENHARIA DE SOFTWARE**

**TURMA DE ARQUITETURA DE SOFTWARE**

**Sumário:**

**1.Introdução...............................................................................................................2 2. Escopo...................................................................................................................3 2.1 Definições, Acrônimos e Abreviações................................................................... 4**

**3. Metas e Restrições da Arquitetura. ..........................................................................5**

**3.1. Metas da Arquitetura..............................................................................................5**

**3.2. Restrições da Arquitetura......................................................................................5**

**4. Visão de Casos de Uso.............................................................................................6**

**5. Visão Lógica.............................................................................................................7**

**6. Pacotes de Design Significativos do Ponto de Vista da Arquitetura…………………..7**

**7. Visão de Implantação……………………………………………………………………………8**

**8. Visão da Implementação...........................................................................................9**

**9. Tamanho e Desempenho..........................................................................................9**

**10. Qualidade...............................................................................................................9**

1. **Introdução.**

O presente documento tem como objetivo descrever o documento de arquitetura do projeto BoltCHAT Analytics, um serviço de gerenciamento de dados para data-analytics em plataformas de atendimento omnichannel. Nesta introdução, apresentamos uma visão geral do projeto, destacando sua importância, propósito e benefícios esperados.

O contexto atual dos negócios exige que as empresas ofereçam um atendimento eficiente e personalizado aos seus clientes em uma variedade de canais, como aplicativos de chat, mídias sociais, e-mails e SMS. Nesse cenário, as plataformas de atendimento omnichannel desempenham um papel fundamental na comunicação e no relacionamento com os clientes. Porém, a crescente quantidade de dados gerados por essas interações tornou-se uma oportunidade valiosa para as empresas. A capacidade de coletar, analisar e extrair insights significativos desses dados pode proporcionar uma vantagem competitiva significativa.

É nesse contexto que o projeto BoltCHAT Analytics surge como uma solução inovadora e essencial. A ideia básica por trás do gerenciamento dos dados do sistema é que eles serão armazenados em um repositório centralizado do tipo Data Warehouse (OLAP), preparado para data-analytics via SQL. Essa abordagem permitirá que as empresas tenham uma visão abrangente e unificada de todas as interações de atendimento e métricas relacionadas.

A finalidade deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas no decorrer do processo de desenvolvimento do BoltCHAT Analytics. Ao fornecer uma visão arquitetural geral do sistema, este documento auxiliará os envolvidos no projeto a compreenderem os aspectos arquiteturais essenciais que são necessários para o desenvolvimento de uma solução que atenda às necessidades dos usuários finais. Além disso, esse documento será uma fonte valiosa de informações para novos membros da equipe que precisam se familiarizar com a arquitetura e o funcionamento do sistema. Ele também servirá como um guia para a tomada de decisões futuras relacionadas ao desenvolvimento, manutenção e escalabilidade do BoltCHAT Analytics.

A seguir, exploraremos as diferentes visões arquiteturais que representarão diversos aspectos do sistema, incluindo a visão de casos de uso, a visão lógica com o padrão MVC (Model-View-Controller), a visão de implantação e outras visões relevantes. Cada visão fornecerá insights detalhados sobre a estrutura, os componentes e as interações do sistema.

Em resumo, o projeto BoltCHAT Analytics representa uma solução inovadora e essencial para o gerenciamento de dados em plataformas de atendimento omnichannel. Ao fornecer análises avançadas e informações valiosas, o sistema contribuirá para aprimorar o atendimento ao cliente, a eficiência operacional e o processo de tomada de decisões estratégicas em empresas de diversos setores. Espera-se que este documento de arquitetura seja um guia confiável para o desenvolvimento bem-sucedido do BoltCHAT Analytics e que o sistema atenda às expectativas dos usuários finais, oferecendo uma experiência excepcional e transformadora no campo do data-analytics para plataformas de atendimento omnichannel.

1. **Escopo.**

O projeto BoltCHAT Analytics é um serviço de gerenciamento de dados para data-analytics em plataformas de atendimento omnichannel. Seu objetivo é coletar e agregar dados de forma consistente, confiável e facilitada, visando fornecer informações valiosas para análise e tomada de decisões em diferentes setores e empresas que utilizam plataformas de atendimento.

O escopo do sistema abrange o desenvolvimento de uma solução que seja capaz de lidar com a coleta de dados provenientes de diversas fontes de atendimento omnichannel, incluindo aplicativos de chat, mídias sociais, e-mails, SMS, entre outros. As informações coletadas abrangem interações de usuários, históricos de atendimento, métricas de desempenho dos agentes, satisfação do cliente, e outros dados relevantes.

O sistema será projetado para armazenar os dados coletados em um repositório centralizado do tipo Data Warehouse (OLAP - Online Analytical Processing). Esse repositório será preparado para análises avançadas utilizando SQL, permitindo consultas multidimensionais e análises complexas.

Os profissionais das plataformas de atendimento terão a capacidade de configurar as métricas específicas que desejam coletar e analisar. Por meio da funcionalidade de "Configuração de Métricas", eles poderão definir quais dados devem ser coletados, as análises serão realizadas e como os resultados serão apresentados. Além disso, o sistema fornecerá uma interface intuitiva para visualização de dados, permitindo que os usuários explorem os dados coletados por meio de gráficos, tabelas e dashboards personalizados. A interface possibilitará a criação de relatórios personalizados, filtragem de dados, e análises comparativas para melhor compreensão do desempenho e resultados.

É importante destacar que o escopo do projeto também incluirá a implementação de medidas de segurança adequadas para proteger os dados coletados e armazenados no Data Warehouse. Isso garantirá que as informações estejam acessíveis apenas para usuários autorizados e que a integridade e confidencialidade dos dados sejam mantidas.

A equipe de desenvolvimento será responsável por criar a arquitetura, design, implementação e testes do sistema BoltCHAT Analytics. O sistema será desenvolvido utilizando conceitos de Programação Orientada a Objetos, frameworks como Ionic e Laravel, TypeScript, Web Container Apache, Linguagem de Programação PHP e bancos de dados Firebase e MySql.

Em resumo, o escopo do projeto abrange a criação de uma solução robusta, flexível e confiável de gerenciamento de dados para data-analytics em plataformas de atendimento omnichannel. O sistema permitirá a coleta, agregação e análise de dados em tempo real, fornecendo informações valiosas para melhorar o desempenho das operações de atendimento, a satisfação do cliente e o processo de tomada de decisões estratégicas.

**2.1 Definições, Acrônimos e Abreviações.**

Para garantir uma comunicação clara e precisa no documento de arquitetura do projeto BoltCHAT Analytics, é essencial definir e esclarecer os termos técnicos, acrônimos e abreviações utilizados ao longo do texto.

Abaixo, apresentamos uma lista detalhada dessas definições para facilitar a compreensão dos leitores:

**OLAP (Online Analytical Processing).**

O termo OLAP refere-se a um conjunto de técnicas e ferramentas utilizadas na área de banco de dados e análise de dados para permitir a análise multidimensional de informações. O OLAP possibilita a exploração de grandes volumes de dados de maneira eficiente, permitindo análises e consultas complexas para extrair insights estratégicos.

**OLTP (Online Transaction Processing).**

O termo OLTP refere-se a um tipo de sistema de banco de dados projetado para processar e gerenciar transações em tempo real. Essas transações podem incluir operações de inserção, atualização, exclusão e recuperação de dados em um ambiente de negócios em tempo real.

**MVC (Model-View-Controller).**

O padrão de arquitetura de software MVC é uma abordagem de design que divide um aplicativo em três componentes principais:

**Model (Modelo):** Responsável por conter a lógica de negócio e as regras de manipulação dos dados.

**View (Visualização):** Responsável por exibir a interface gráfica ao usuário e apresentar os dados ao usuário de forma adequada.

**Controller (Controle):** Responsável por receber as entradas do usuário, processá-las e interagir com o modelo e a visão de acordo com as ações realizadas pelo usuário.

O uso do padrão MVC ajuda a separar as preocupações em um aplicativo, tornando-o mais organizado, de fácil manutenção e escalabilidade.

**Data Warehouse.**

O Data Warehouse é um repositório centralizado de dados que contém informações coletadas de várias fontes de dados da empresa. É otimizado para permitir análises avançadas, como consultas multidimensionais e análises de tendências ao longo do tempo. O Data Warehouse fornece uma visão consolidada e histórica dos dados para auxiliar na tomada de decisões estratégicas.

**Fontes de Atendimento Omnichannel.**

As fontes de atendimento omnichannel são os diversos canais utilizados pelas empresas para interagir com os clientes. Esses canais podem incluir aplicativos de chat, mídias sociais, e-mails, SMS e outras plataformas que permitem o atendimento ao cliente. O termo "omnichannel" refere-se à capacidade de oferecer uma experiência de atendimento unificada e consistente em todos os canais.

**Data-analytics:**

Data-analytics refere-se à análise e interpretação de dados para obter insights significativos e informar a tomada de decisões. Essa análise pode incluir técnicas estatísticas, mineração de dados, aprendizado de máquina e outras abordagens para identificar padrões, tendências e oportunidades.

Essas são as principais definições, acrônimos e abreviações relevantes para o projeto BoltCHAT Analytics. Ao utilizar esses termos de forma clara e consistente ao longo do documento de arquitetura, facilitamos a compreensão dos leitores e garantimos uma comunicação eficiente entre os membros da equipe e outras partes interessadas envolvidas no projeto.

**3. Metas e Restrições da Arquitetura.**

**3.1. Metas da Arquitetura.**

As metas da arquitetura do projeto BoltCHAT Analytics são direcionadas para fornecer uma solução eficaz de gerenciamento de dados para data-analytics em plataformas de atendimento omnichannel. As principais metas incluem:

**1. Eficiência de Análise:** A arquitetura busca garantir que as análises de dados sejam realizadas de forma eficiente, permitindo que os usuários obtenham insights valiosos e tomem decisões informadas rapidamente.

**2. Escalabilidade:** O sistema deve ser projetado para lidar com grandes volumes de dados à medida que a plataforma de atendimento cresce, garantindo que a performance não seja comprometida.

**3. Segurança dos Dados:** A arquitetura deve implementar medidas robustas de segurança para proteger os dados armazenados no Data Warehouse, assegurando que somente usuários autorizados tenham acesso aos dados sensíveis.

**4. Usabilidade:** A interface do sistema deve ser intuitiva e amigável, garantindo que os usuários possam explorar os dados, criar relatórios personalizados e extrair informações relevantes com facilidade.

**3.2. Restrições da Arquitetura:**

As restrições da arquitetura do projeto BoltCHAT Analytics definem os limites e condições dentro dos quais a solução deve ser projetada e implementada. As principais restrições incluem:

**1. Padrão MVC:** O sistema será desenvolvido seguindo o padrão de arquitetura MVC (Model-View-Controller), garantindo a separação clara entre lógica de negócios, interface de usuário e controle de dados.

**2. Tecnologias Específicas**: O projeto utilizará as tecnologias Ionic e Laravel, estrutura MVC, TypeScript, Web Container Apache, Linguagem de Programação PHP e bancos de dados Firebase e MySql para a implementação do sistema.

**3. Compatibilidade:** A aplicação deve ser compatível com diferentes dispositivos, incluindo navegadores web e dispositivos móveis, permitindo que hemocentros e doadores acessem e utilizem o sistema de forma conveniente.

**4. Tamanho e Desempenho:** O sistema deve ser capaz de lidar com o tráfego e as operações tanto de hemocentros quanto de doadores. A aplicação web deve ser dimensionada para atender às necessidades de diversos usuários.

**5. Portabilidade:** A arquitetura deve permitir que o sistema seja implantado e executado em diferentes ambientes, garantindo que ele possa ser acessado e utilizado independentemente do local.

**6. Manutenibilidade:** A adoção do padrão MVC e boas práticas de desenvolvimento devem garantir que o sistema seja de fácil manutenção, permitindo atualizações e melhorias contínuas.

**7. Segurança de Dados**: Medidas de segurança robustas devem ser implementadas para proteger os dados armazenados e garantir a privacidade dos usuários.

As metas e restrições da arquitetura são fundamentais para orientar o projeto BoltCHAT Analytics, assegurando que a solução atenda às necessidades dos usuários e mantenha padrões de qualidade, desempenho e segurança.

**4. Visão de Casos de Uso.**

Nesta visão, representamos os principais casos de uso do sistema BoltCHAT Analytics.

Os principais casos de uso são:

1. **Análise de Dados:** Permite que os usuários visualizem métricas e gerem relatórios a partir dos dados armazenados no Data Warehouse (OLAP).
2. **Gerenciamento de Dados:** Permite aos usuários importar dados para o sistema e exportar dados do sistema para outras finalidades.

**5. Visão Lógica.**

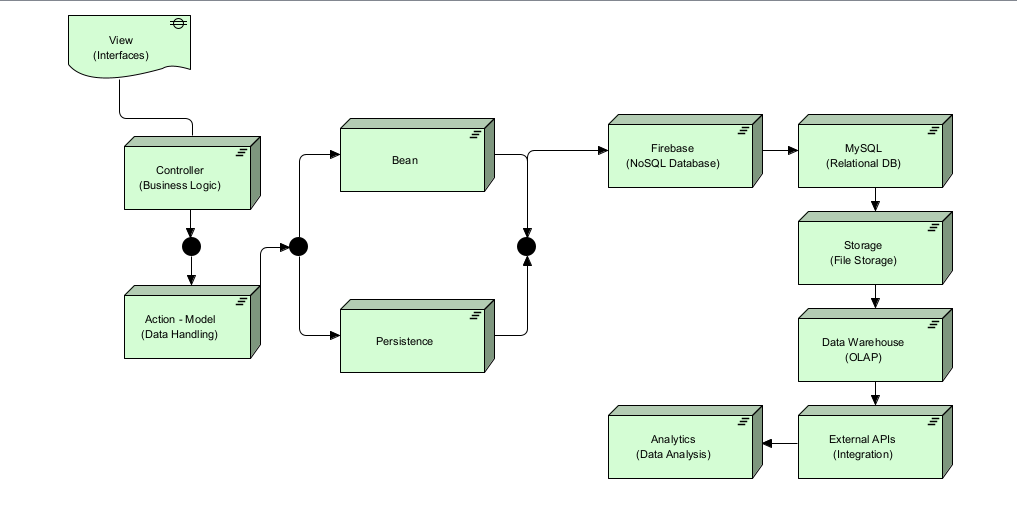
Nesta visão, representamos a estrutura da arquitetura do sistema como padrão MVC. Os principais componentes do padrão MVC são:

1. **View (Interface):** Contém as visões do projeto, ou seja, as interfaces, formulários e outros elementos com os quais o usuário interage.
2. **Controller:** Recebe as informações e requisições do pacote view e as despacha para a devida classe de controle. O controller se comunica com as classes Business Logic, que contém as regras de negócio do sistema.
3. **Action - Model:** Pacote que recebe as informações e requisições do pacote controller e as atribui às respectivas classes do pacote bean e do pacote persistence, utilizando dados de seus modelos.
4. **Bean:** Pacote que recebe e armazena as informações referentes a seus respectivos objetos.
5. **Persistence:** Pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados provenientes do pacote bean. O pacote persistence também pode enviar dados para o pacote view.

**6. Pacotes de Design Significativos do Ponto de Vista da Arquitetura.**

Esse tópico vai detalhar um pouco mais o tópico 6:

Visão do diagrama:



**Pacote Model:**

O pacote Model é onde reside a inteligência do sistema. Ele contém classes que representam os principais conceitos de negócios, como usuários, atendimentos, métricas e configurações. Essas classes não estão diretamente ligadas à interface do usuário, mas sim à lógica de negócios subjacente. Elas implementam ações como cálculos de métricas, validações de dados e regras de negócios complexas.

**Pacote View:**

O pacote View é responsável por criar uma experiência visual para os usuários. Ele contém classes que definem a interface do usuário, como telas, formulários e componentes visuais. O View não contém lógica de negócios; em vez disso, ele recebe dados do Controller e os apresenta de forma compreensível e atraente. Isso inclui exibir gráficos de desempenho, tabelas de dados e relatórios.

**Pacote Controller:**

O pacote Controller atua como um intermediário entre o Model e o View. Ele recebe interações do usuário, como cliques em botões, entradas de teclado ou seleções de menu, e decide como processar essas ações. O Controller interpreta essas ações e decide quais partes do Model precisam ser atualizadas. Ele também envia instruções para o View atualizar a interface de acordo com os resultados do processamento.

**Pacote Action:**

O pacote Action contém classes que implementam casos de uso específicos ou funcionalidades do sistema. Cada classe de ação representa uma atividade ou operação, como "Criar Atendimento", "Gerar Relatório" ou "Configurar Métricas". Essas classes encapsulam a lógica necessária para executar essas ações, interagindo com o Model para obter ou atualizar os dados relevantes.

**Pacote Bean:**

O pacote Bean armazena classes que representam objetos de dados simples, muitas vezes associados a entidades do Model. Eles têm atributos que correspondem a campos de dados, e esses objetos são frequentemente usados para transferir dados entre diferentes partes do sistema, como entre o Controller e o Action. Os Beans ajudam a estruturar e organizar os dados para facilitar a manipulação.

**Pacote Persistence:**

O pacote Persistence é responsável por lidar com a persistência dos dados no banco de dados. Ele contém classes que realizam operações de leitura e gravação no banco de dados, como criar, ler, atualizar e excluir registros. O pacote Persistence interage com o Model para obter ou salvar dados relevantes, permitindo que o sistema mantenha uma fonte confiável e consistente de informações.

Cada um desses pacotes desempenha um papel específico na estrutura do sistema, contribuindo para a modularidade, escalabilidade e manutenção do projeto BoltCHAT Analytics. Eles permitem a separação clara das diferentes preocupações do sistema, facilitando o desenvolvimento, a expansão e a evolução contínua da aplicação.

**7. Visão de Implantação.**

Na visão da implantação, representamos como o sistema será implantado em um ambiente de produção após a validação pelos stakeholders. O sistema será implantado após a validação. Nesse caso, representamos um ambiente de implantação básico, com um servidor web, onde o sistema BoltCHAT Analytics será hospedado e um servidor de banco de dados (BD), onde os dados do sistema serão armazenados.

**8. Visão da Implementação.**

Na visão da implementação, representamos as tecnologias e componentes utilizados para implementar o sistema. Neste caso, nós destacamos as tecnologias e componentes utilizados na implementação do projeto BoldCHAT Analytics:

* **Framework Ionic e Laravel:** Utilizados para desenvolver o sistema, com o padrão arquitetural MVC.
* **Linguagem de Programação PHP:** Utilizada para desenvolver a lógica do sistema.
* **Banco de Dados Firebase e MySQL:** Utilizados para armazenar os dados do sistema.
* **Web Container Apache:** Utilizado para hospedar o servidor web.

Com isso, temos uma representação das tecnologias e componentes principais utilizados na implementação do sistema.

**9. Tamanho e Desempenho.**

Nesta visão, abordamos os aspectos relacionados ao tamanho e desempenho do sistema. Neste caso, nós representamos que o sistema BoltCHAT Analytics será implantado através de um site próprio da versão web para os hemocentros, proporcionando acesso através de desktop e laptops. Além disso, haverá uma versão mobile do sistema para uso dos usuários doadores, acessível através de dispositivos móveis, como smartphones e tablets.

Essa abordagem de ter uma visão web e uma versão mobile permitirá que os usuários acessem o sistema de acordo com suas necessidades e preferências, proporcionando uma experiência consistente em diferentes plataformas.

**10. Qualidade.**

Nesta visão, descrevemos os atributos de qualidade relevantes para o sistema. Os seguintes atributos de qualidade são os destaques:

1. **Discoverabilidade de Dados:** O sistema deve oferecer recursos que permitam aos usuários descobrirem e explorarem os dados de forma eficiente e intuitiva.
2. **UI Intuitiva:** A interface do usuário deve ser projetada de forma intuitiva, facilitando a navegação e a compreensão das informações apresentadas.
3. **Flexibilidade nos formatos de dados para importação e exportação:** O sistema deve permitir a importação e exportação de dados em diversos formatos, para facilitar a integração com outras ferramentas e sistemas.

Esses atributos de qualidade ajudarão a garantir que o sistema BoltCHAT Analytics seja fácil de usar, eficiente e adequado às necessidades dos usuários.

Com todas essas visões criadas, temos uma representação abrangente da arquitetura do projeto BoltCHAT Analytics.