

Roteiro de Apresentação: Assistente Virtual para Assistência Técnica

Tempo Total: 12 minutos **Apresentadores:** Apresentador 1 (P1), Apresentador 2 (P2)

Bloco 1: Introdução e O Problema (2.0 minutos)

Apresentador: P1

Slide 1: Capa

- **Título:** Projeto de Engenharia de Software: Assistente Virtual para E-commerce de Assistência Técnica
- **Subtítulo:** Universidade Cruzeiro do Sul, Análise e Desenvolvimento de Sistemas
- **Integrantes:** Guilherme Bifani de Santana, Guilherme Soares da Silva (Grupo G&G)

(Script P1)

"Bom dia/boa tarde a todos. Meu nome é [P1] e, junto com o [P2], nós somos o Grupo G&G.

Hoje, vamos apresentar nosso projeto de Engenharia de Software: um Assistente Virtual focado em E-commerce de Assistência Técnica."

Slide 2: O Problema

- **Título:** O Desafio do Atendimento em Assistência Técnica
- **Tópicos:**
 - Alta demanda por consultas repetitivas (Status, Orçamento, Horários).
 - Sobrecarga do atendimento humano.
 - Necessidade de respostas 24/7, fora do horário comercial.
 - Processos manuais para agendamento e aprovação de orçamentos (atrito).

(Script P1)

"Para começar, qual é o problema que estamos resolvendo?

Qualquer e-commerce de assistência técnica sofre com um volume enorme de perguntas repetitivas: 'Onde está meu produto?', 'Meu orçamento ficou pronto?', 'Posso agendar uma coleta?'.

Isso sobrecarrega a equipe humana, que gasta tempo respondendo coisas que um sistema poderia fazer. Além disso, o cliente de hoje quer respostas imediatas, e a falta de automação, principalmente na aprovação de orçamentos, gera atrito e custos."

Bloco 2: A Solução e Escopo (2.0 minutos)

Apresentador: P1

Slide 3: A Solução Proposta

- **Título:** A Solução: Um Assistente Virtual (Chatbot)
- **Tópicos:**

- Respostas instantâneas e automáticas usando NLP.
- **Transacional, não apenas informativo:** vai além do FAQ.
- Integração direta com o sistema de gestão (ERP/API) da loja.
- Autonomia para o cliente (self-service).

(Script P1)

"Nossa solução é um Assistente Virtual. Mas é crucial entender que não é apenas um FAQ glorificado.

Ele é um assistente **transacional**. Ele usa Processamento de Linguagem Natural (NLP) para entender o cliente e, o mais importante, ele se conecta diretamente aos sistemas da loja para *executar ações*: ele consulta o status real de um pedido, agenda uma coleta no sistema de logística e, como vamos detalhar, ele processa a aprovação de um orçamento em tempo real."

Slide 4: Escopo do Projeto (Premissas e Restrições)

- **Título:** Nossas "Regras do Jogo" (Escopo)
- **Restrições (Limites):**
 - MVP (Produto Mínimo Viável) em **4 meses**.
 - Tecnologia obrigatória: **Java**.
 - Conformidade total com a **LGPD**.
- **Premissa (Suposição Crítica):**
 - A loja **DEVE** possuir uma API que permita o acesso ao status dos pedidos e serviços. Sem isso, o projeto é inviável.

(Script P1)

"Para tornar isso viável, definimos um escopo claro. Nossas principais restrições eram entregar um MVP em 4 meses, usando Java e garantindo total conformidade com a LGPD.

E a premissa mais importante: nosso sistema *assume* que a assistência técnica já possui uma API. Não estamos construindo o ERP da loja; estamos construindo o *cérebro* que consome os dados desse ERP para atender o cliente.

Agora, o [P2] vai detalhar exatamente como esse cérebro funciona."

Bloco 3: Funcionalidades e Arquitetura (2.5 minutos)

Apresentador: P2

Slide 5: Funcionalidades Principais (Casos de Uso)

- **Título:** O que o Assistente Virtual faz?
- **(Usar ícones para cada funcionalidade)**
 - **UC-01:** Rastrear Status do Pedido (Por CPF ou nº da OS).
 - **UC-02:** Agendar Coleta ou Entrega (Validando endereço - RN-02).
 - **UC-04:** Aprovar ou Rejeitar Orçamentos (Notificação ativa - RN-01).

- **UC-05:** Transferir para Atendimento Humano (Fluxo de "escape" - RN-05).

(Script P2)

"Obrigado, [P1]. Vamos focar no valor que o assistente entrega. Nós modelamos 6 casos de uso principais, e eu quero destacar quatro deles:

1. **Rastrear Pedido (UC-01):** A função básica. O cliente informa o CPF ou a OS, e o bot consulta o status real.
2. **Agendar Coleta (UC-02):** Aqui o sistema fica inteligente. O cliente pede uma coleta, e o bot valida *automaticamente* se o endereço está na área de cobertura de 15km, que é a nossa Regra de Negócio RN-02.
3. **Aprovar Orçamento (UC-04):** Esse é um grande diferencial, que vou detalhar mais à frente. O bot *notifica ativamente* o cliente sobre o valor, e o cliente aprova ou rejeita na hora.
4. **Transferência Humana (UC-05):** E se tudo falhar? Se o bot não entender 3 vezes (RN-05) ou se o cliente pedir, o sistema transfere a conversa, com todo o histórico, para um operador humano."

Slide 6: A Arquitetura (Diagrama de Classes)

- **Título:** A Estrutura do Sistema
- **(Mostrar o Diagrama de Classes do documento)**
- **Destacar (circular ou usar setas):** `Cliente` , `SessaoAtendimento` , `AssistenteVirtual` , `OrdemServico` , `Agendamento` , `RegraNegocio` .

(Script P2)

"Para suportar essas funções, esta é a nossa arquitetura de classes. O núcleo é: o `Cliente` inicia uma `SessaoAtendimento` , gerenciada pelo `AssistenteVirtual` .

O mais importante é como ele se conecta às entidades de negócio: a `OrdemServico` (para rastrear status) e o `Agendamento` (para coletas).

Notem a classe `RegraNegocio` separada. Isso é crucial, pois permite que o sistema consulte as regras (como o raio de 15km ou o prazo de 7 dias para cancelamento) de forma dinâmica, sem 'chumbar' isso no código."

Bloco 4: Modelagem Comportamental (2.5 minutos)

Apresentador: P2

Slide 7: O Comportamento: Zoom no Processo de Orçamento (UC-04)

- **Título:** Como o Sistema se Comporta? Foco no Orçamento
- **(Mostrar os 3 diagramas lado a lado ou em sequência rápida)**
 1. **Diagrama de Estados (OrdemServico)**
 2. **Diagrama de Atividades (Aprovação de Orçamento)**
 3. **Diagrama de Sequência (CU-03: Aprovar Orçamento)**

(Script P2)

"Agora, vamos aprofundar em como o sistema realmente se comporta, usando nosso processo mais crítico: a Aprovação de Orçamento.

1. **Primeiro, o Diagrama de Estados:** Este diagrama mostra o ciclo de vida de um pedido. O processo do bot começa quando a OS entra no estado '**Aguardando Aprovação**'. Esse estado é o gatilho para toda a ação.
2. **Segundo, o Diagrama de Atividades:** Ele mostra o *fluxo* de negócio. Quando o orçamento fica pronto, o sistema notifica o cliente. O cliente tem três caminhos: Aprovar (muda o estado para 'Em Serviço'), Rejeitar (muda para 'Cancelada') ou não fazer nada. E aqui entra a Regra de Negócio RN-01: se não houver resposta em 7 dias, o sistema automaticamente cancela o pedido.
3. **Terceiro, o Diagrama de Sequência:** Ele mostra *como* os objetos de software interagem. O 'Laboratorio' atualiza a 'OrdemServico'. O 'GerenciadorOS' (nosso controlador) percebe essa mudança e usa o 'Chat' para notificar o 'Cliente'. Quando o cliente responde 'aprovo', o Gerenciador comanda a 'OrdemServico' para mudar seu estado para 'Em_Servico' e notifica o cliente da confirmação.

"Essa modelagem tripla garante que o fluxo de negócio, as regras e a interação técnica estejam perfeitamente alinhados."

Bloco 5: Nossos Principais Aprendizados (2.0 minutos)

Apresentador: P1

Slide 8: O que Aprendemos com o Projeto

- **Título:** O que Aprendemos com o Projeto
- **Tópicos:**
 - A diferença crucial entre Requisitos (RFs) e Regras de Negócio (RNs).
 - Modelagem como ferramenta de validação (UC -> 3 Diagramas).
 - O papel das Premissas (viabilidade) e Restrições (foco).
 - A tradução de um "Problema de Negócio" em uma "Especificação Técnica".

(Script P1)

"Antes de concluir, gostaríamos de compartilhar o que esse projeto nos ensinou na prática sobre a matéria de Engenharia de Software.

Primeiro, aprendemos a diferenciar claramente **Requisitos Funcionais** de **Regras de Negócio**. Uma coisa é o requisito 'Aprovar Orçamento'. Outra, bem diferente, é a regra de negócio RN-01, que diz 'cancelar automaticamente após 7 dias'. O sistema precisa implementar ambos, e agora entendemos essa separação.

O segundo aprendizado, e talvez o mais importante, foi que **modelagem é validação**. Como o [P2] mostrou, para um único Caso de Uso, o UC-04, nós precisamos de *três diagramas*: o de **Estados** (para saber o status 'Aguardando Aprovação'), o de **Atividades** (para mapear o fluxo de 'sim', 'não' ou 'timeout de 7 dias') e o de **Sequência** (para ver quais objetos de software trocam mensagens). Só assim garantimos que o fluxo funciona.

Por fim, entendemos o poder das **Premissas e Restrições**. Uma premissa como a 'P-01' (a loja ter uma API) foi o que definiu se o nosso projeto era *viável* ou não. Esse projeto nos levou do começo ao fim do processo de engenharia: de um problema de negócio vago a uma especificação técnica completa e modelada."

Bloco 6: Conclusão (1.0 minuto)

Apresentador: P1

Slide 9: Impacto e Conclusão

- **Título:** O que o Projeto Entrega?
- **Tópicos:**
 - Redução de custo operacional no atendimento.
 - Agilidade (aprovação de orçamentos em minutos, não dias).
 - Melhoria na satisfação do cliente (Serviço 24/7).
 - Um sistema robusto, baseado em regras e pronto para escalar.

(Script P1)

"Em resumo, nosso projeto não é apenas um chatbot. É uma ferramenta de automação de processos de negócio.

Ao modelar corretamente os requisitos, regras e comportamentos—especialmente em fluxos críticos como o de orçamento—criamos um sistema que não só reduz custos operacionais, mas que agiliza o serviço (transformando a aprovação de orçamentos de dias para minutos) e, o mais importante, melhora a experiência do cliente final com um atendimento 24/7."

Slide 10: Obrigado

- **Título:** Obrigado
- **Subtítulo:** Perguntas?
- **Integrantes:** Grupo G&G

(Script P1)

"Obrigado pelo seu tempo. Estamos abertos para perguntas."