

1. Estrutura Geral do XML

O XML é como uma receita de bolo que descreve exatamente como o agente de uma clínica deve se comportar ao interagir com os pacientes via WhatsApp. Ele está organizado em diferentes seções, cada uma com uma função específica, da mesma forma que uma receita é dividida em ingredientes, modo de preparo, e dicas.

2. Elemento `<Agent>`

O elemento principal `<Agent>` é como o nome da receita, que encapsula tudo o que está dentro. Dentro deste elemento, temos outras partes que definem a descrição do agente, o estilo de comunicação, as funções disponíveis, e outras configurações.

3. Elemento `<Description>`

Dentro de `<Agent>`, temos o `<Description>`, que é como o prefácio da receita, explicando o propósito geral do agente. Aqui, diz que o agente é um "Agente de clínica para WhatsApp que realiza agendamentos, consulta resultados de exames e tira dúvidas."

4. Elemento `<Language>`

O `<Language>` especifica o idioma e o tom que o agente deve usar. É como se a receita dissesse: "Este bolo deve ser servido em pratos elegantes e com um toque informal." Aqui, o idioma é "Português Brasileiro, informal e humanizado."

5. Elemento `<CommunicationStyle>`

O `<CommunicationStyle>` define o estilo de comunicação do agente. Dentro dele:

- `<Style>` especifica que o estilo é "Informal".
- `<Guide>` é como uma dica na receita: "Use os exemplos como inspiração, mas não mencione a validação na conversa."

6. Elemento `<FieldsConfigurator>`

Este elemento é como as instruções especiais na receita, que dizem como certos ingredientes devem ser manuseados. Aqui, define que certas ferramentas (funções internas do sistema) devem ser chamadas imediatamente para validar dados antes mesmo de responder.

7. Elemento <Validations>

Este é o conjunto de regras que o agente deve seguir para validar as informações. É como as regras de ouro de uma receita: "Sempre verifique se a massa está cozida antes de tirar do forno". Aqui, a validação deve considerar todo o histórico de mensagens, não apenas a última.

8. Elemento <Rules>

A seção <Rules> adicionada é como as instruções de segurança de uma receita de cozinha que dizem "Não deixe o forno ligado sem supervisão" ou "Verifique se a temperatura é adequada antes de assar". Estas regras garantem que o agente siga diretrizes claras para evitar fornecer informações erradas ou não confiáveis:

1. **Primeira Regra:** O agente deve sempre se basear em dados concretos e evitar especulações. Isso impede que o agente "invente" informações.
2. **Segunda Regra:** Se o agente não souber algo, ele deve ser transparente e sugerir contato com a clínica para obter informações seguras.
3. **Terceira Regra:** O agente só deve fornecer informações médicas que estejam claramente respaldadas por dados específicos do paciente ou fontes confiáveis.
4. **Quarta Regra:** O agente deve oferecer alternativas, como encaminhamento para suporte humano, caso não tenha a informação necessária.

9. Elemento <Functions>

Este é onde a "mágica" acontece, similar às instruções passo a passo do modo de preparo da receita. Cada <Function> dentro de <Functions> define uma tarefa específica que o agente pode realizar, como agendar consultas ou consultar exames.

9.1. Função <Function> - Agendamento

- <Name> : Define o nome da função, aqui é "Agendamento".
- <Fields> : Estes são os ingredientes necessários para essa função. Cada <Field> dentro de <Fields> é como um ingrediente com instruções específicas sobre como deve ser solicitado e validado:
 - nome : O agente pergunta o nome completo e valida se o nome tem mais de uma palavra.
 - email : Solicita o email e corrige erros óbvios de digitação.
 - medico : Pergunta com qual médico o paciente quer marcar.
 - data_hora : Pergunta a data e hora preferidas e valida se a data é no futuro.

- `<ResponseTemplate>` : É como a apresentação final do prato, onde o agente monta a resposta usando as informações coletadas, por exemplo: "Tudo certo! Seu agendamento com {medico} está marcado para {data_hora}."

9.2. Função `<Function>` - Consulta de Exames

Esta função permite consultar resultados de exames:

- `<Fields>` : Pede o CPF para identificar o paciente.
- `<ResponseTemplate>` : Retorna os resultados dos exames realizados.

9.3. Função `<Function>` - Tira Dúvidas

Esta função responde a perguntas dos pacientes:

- `<Fields>` : Pede ao paciente para descrever sua dúvida.
- `<ResponseTemplate>` : Usa a resposta do sistema para informar o paciente sobre a dúvida levantada.

Dicas Extras:

- **Tags e Atributos:** As tags como `<Field>` podem ter atributos como `name`, `type`, e `prompt`, que funcionam como descrições adicionais. Pense neles como pequenas notas na receita que especificam detalhes, como "Adicione 200g de açúcar refinado".
- **Validações e Ferramentas:** Algumas tags como `<Field>` têm validações e ferramentas específicas associadas. Isso é como uma receita pedindo para bater a massa por 10 minutos (validação) e usar uma batedeira (ferramenta).