**Levantamento de Requisitos - Sistema de Controle de Expiração de Veículos - AutoAlert**

**1. Introdução**

O objetivo deste sistema é fornecer uma solução automatizada para o controle de vencimentos de documentos de veículos, oferecendo um serviço completo para empresas, que inclui o cadastro de veículos, notificação de vencimentos e geração de relatórios. O sistema será desenvolvido em Python, utilizando banco de dados MySQL e integração com o WhatsApp para envio de notificações.

**2. Requisitos Funcionais**

**2.1 Cadastro de Empresas**

* **CNPJ da empresa**
* **Nome da empresa**
* **E-mail de contato**
* **Telefone de contato**
* **Telefone para receber notificações do sistema**

**2.2 Cadastro de Veículos**

* **Placa do veículo**
* **Tipo de documento**
* **Data de vencimento do documento**

**2.3 Notificação de Vencimentos**

* O sistema deve enviar notificações via WhatsApp para os responsáveis, de acordo com a seguinte regra:
  + 30, 15, 10, 5, 3, 1, 0 e quando vencido.
* A notificação deve incluir:
  + **Placa do veículo**
  + **Tipo de documento**
  + **Número de dias restantes para o vencimento**

**2.4 Relatórios**

* Relatórios de vencimentos de documentos de veículos devem ser gerados e enviados automaticamente:
  + **Relatório mensal** para cada empresa, com todos os veículos e seus vencimentos.
  + **Relatório de vencimentos** para o administrador, de todas as empresas cadastradas e vencimentos de documentos por empresa.
* Relatórios poderão ser solicitados a qualquer momento por meio de um **chatbot no WhatsApp**.

**2.5 Backup**

* O sistema deverá realizar um **backup mensal** de todos os dados cadastrados, garantindo a integridade das informações.

**2.6 Níveis de Acesso**

* **Nível Master**: Acesso completo, incluindo gestão de empresas, veículos e relatórios.
* **Nível Administrador**: Acesso restrito, podendo apenas gerenciar veículos e consultar relatórios.

**2.7 Interface Web**

* O sistema terá uma **interface web** para login e gestão de todas as funcionalidades:
  + Cadastro de empresas e veículos
  + Gerenciamento de relatórios
  + Configurações do sistema

**2.8 Validações**

* O sistema deve garantir que todos os campos sejam preenchidos corretamente antes de realizar o salvamento dos dados no banco de dados.
  + Verificação de formatos de e-mail, telefone, data de vencimento e CNPJ.

**2.9 Logs de Atividades**

* O sistema deve manter **logs de atividades** para monitoramento, incluindo:
  + Ações executadas no sistema (cadastros, edições, exclusões).
  + Logs de falhas no sistema (problemas de envio de notificações, falhas de banco de dados, etc.).

**3. Requisitos Não Funcionais**

**3.1 Desempenho**

* O sistema deve ser rápido e eficiente, mesmo com grandes volumes de dados.

**3.2 Escalabilidade**

* O sistema deve ser capaz de suportar a adição de várias empresas e veículos sem comprometer o desempenho.

**3.3 Segurança**

* O sistema deve garantir a **segurança dos dados** sensíveis, incluindo:
  + Criptografia de senhas.
  + Proteção de dados bancários e informações pessoais.
  + Autenticação segura para login de usuários.

**4. Tecnologias Utilizadas**

* **Linguagem**: Python
* **Banco de Dados**: MySQL
* **Interface Web**: Flask (framework para Python)
* **Integração WhatsApp**: Biblioteca que possibilite o envio de mensagens via WhatsApp (ex: Twilio, WhatsApp API)
* **Biblioteca de Conexão com o Banco de Dados**: mysql.connector

**5. Fluxo de Uso**

**5.1 Cadastro de Empresas**

1. O administrador envia os dados da empresa via e-mail.
2. O sistema realiza o cadastro da empresa automaticamente.

**5.2 Cadastro de Veículos**

1. A empresa envia os dados dos veículos para o e-mail específico.
2. O administrador insere as informações no sistema, associando a empresa aos veículos.

**5.3 Envio de Notificações**

1. O sistema verifica diariamente os vencimentos dos documentos dos veículos.
2. Quando o vencimento está próximo (30, 15, 10, 5, 3, 1, 0 ou vencido), uma notificação é enviada via WhatsApp para o telefone cadastrado da empresa.

**5.4 Geração de Relatórios**

1. O sistema gera relatórios de vencimento mensal automaticamente para as empresas.
2. O administrador também pode gerar relatórios gerais de todos os veículos e vencimentos a qualquer momento.

**5.5 Chatbot de Solicitação**

* O cliente pode solicitar relatórios ou informações via chatbot no WhatsApp a qualquer momento.

**6. Automação**

O sistema deve ser **totalmente automatizado**, incluindo:

* Envio de notificações diárias de vencimento.
* Geração e envio de relatórios mensais para as empresas.
* Solicitação de relatórios via chatbot.
* Backup mensal dos dados.

**7. Considerações Finais**

Este sistema visa simplificar o controle de vencimento de documentos de veículos para empresas, automatizando todas as etapas, desde o cadastro de dados até o envio de notificações e relatórios. A automação, a integração com o WhatsApp e a interface web serão as principais características que garantirão a eficiência e a experiência do usuário.

**1. Camadas do Sistema**

Dividiremos o sistema em camadas para facilitar a organização, escalabilidade e manutenção:

* **Camada de Apresentação (Front-end)**:
  + Usaremos a biblioteca **Flet** para criar uma interface web amigável.
  + A interface permitirá o login, gerenciamento de empresas, veículos, e visualização de relatórios.
* **Camada de Aplicação (Back-end)**:
  + Utilizaremos **Flask** para gerenciar rotas e APIs.
  + A lógica de negócio, como notificações, relatórios, e validações, ficará aqui.
* **Camada de Dados (Banco de Dados)**:
  + O banco de dados será **MySQL**.
  + Estruturaremos tabelas para empresas, veículos, documentos, notificações, logs, entre outros.

**2. Estrutura do Banco de Dados (DER)**

Aqui está uma ideia inicial da estrutura de tabelas:

1. **Empresa**:
   * ID (chave primária)
   * Nome
   * CNPJ
   * Email
   * Telefone de contato
   * Telefone para notificações
2. **Veículo**:
   * ID (chave primária)
   * Placa
   * Tipo
   * Empresa\_ID (chave estrangeira para Empresa)
3. **Documento**:
   * ID (chave primária)
   * Nome (e.g., IPVA, Licenciamento)
   * Data de Validade
   * Veículo\_ID (chave estrangeira para Veículo)
4. **Notificação**:
   * ID (chave primária)
   * Data de envio
   * Status (Enviado/Pendente/Falha)
   * Documento\_ID (chave estrangeira para Documento)
5. **Usuário**:
   * ID (chave primária)
   * Nome
   * Email
   * Senha (hash)
   * Nível de acesso (Master/Admin)
6. **Log de Atividades**:
   * ID (chave primária)
   * Ação (e.g., Cadastro de veículo, Exclusão de documento)
   * Data e Hora
   * Usuário\_ID (chave estrangeira para Usuário)

**3. Fluxo de Funcionalidades**

* **Cadastro de Empresas**:
  + Admin master registra dados das empresas.
* **Cadastro de Veículos e Documentos**:
  + Empresa envia dados por e-mail, equipe insere no sistema.
* **Notificações Automáticas**:
  + Serviço que roda em segundo plano verifica diariamente os documentos a vencer e envia notificações via WhatsApp.
* **Relatórios Mensais e Sob Demanda**:
  + Envio automático de relatórios mensais.
  + Chatbot no WhatsApp para solicitar relatórios em tempo real.
* **Logs e Backup**:
  + Sistema armazena logs de alterações e realiza backups mensais.

**4. Ferramentas e Tecnologias**

* **Front-end**: Flet (Python)
* **Back-end**: Flask (Python)
* **Banco de Dados**: MySQL
* **Automação e Integração WhatsApp**: API (e.g., Twilio ou biblioteca como WhatsApp Web Automation)
* **Segurança**: Hashing de senhas (e.g., bcrypt), HTTPS para a interface, validação de inputs.