

Projeto: Trading Bot na Nuvem Azure

Objetivo do Projeto

Este projeto visa criar um **Trading Bot** que utilize a API da Binance para realizar operações de compra e venda de criptomoedas, baseado em sinais de trading, como **Engolfo** e **Inside Bar**. O sistema deve ser capaz de registrar cada transação realizada e gerar relatórios de **ganho e perda** para monitorar a performance da conta. Além disso, o bot deve ser configurado para usar **ordens com Stop Loss e Take Profit** para proteger os investimentos.

Grupo no máximo 3 alunos

Tecnologias Utilizadas

1. **API da Binance:** Para realizar as operações de compra e venda de criptomoedas.
 2. **Azure SQL Database:** Para armazenar e gerenciar os registros das transações e dos relatórios.
 3. **Azure App Service:** Para hospedar e rodar o bot na nuvem.
 4. **Azure Monitor:** Para monitorar o desempenho e os erros do bot na nuvem.
-

Requisitos do Projeto

1. Funcionalidades Principais

- **Consulta de Preços de Moedas:** O bot deve ser capaz de consultar os preços atuais de diferentes criptomoedas na Binance, permitindo que ele decida quando e qual moeda comprar ou vender.
- **Execução de Ordens de Compra e Venda:** O bot deve permitir a compra e venda de criptomoedas usando ordens de mercado ou limitadas. As ordens de compra e venda devem ser acionadas com base em sinais de trading, como:
 - **Engolfo:** Um padrão de velas em que uma vela maior engolfa uma vela menor, indicando uma possível reversão de tendência.
 - **Inside Bar:** Um padrão onde a vela mais recente está dentro da faixa da vela anterior, sugerindo uma continuação ou indecisão do mercado.
- **Ordem com Stop Loss e Take Profit:** O bot deve ser capaz de configurar **Stop Loss** (para minimizar perdas) e **Take Profit** (para garantir lucros ao atingir um preço alvo).
- **Gerenciamento de Relatórios de Ganhos e Perdas:** O bot deve registrar todas as transações realizadas, calculando o ganho ou perda de cada operação. O sistema deve gerar um relatório completo com todos os trades e os resultados de cada um.

2. Banco de Dados

- **Armazenamento de Transações:** O bot deve gravar as informações das transações realizadas (como moeda, quantidade, preço de compra, preço de venda, ganho/perda) em um banco de dados SQL.
- **Armazenamento de Relatórios:** Deve haver uma tabela ou estrutura para armazenar os relatórios de performance, com o total de ganho ou perda, para monitorar a evolução do bot ao longo do tempo.

3. Hospedagem e Escalabilidade

- **Azure App Service:** O bot deve ser hospedado na nuvem usando o Azure App Service, que permitirá escalabilidade e manutenção simplificada do sistema. O código deve ser configurado para rodar continuamente, realizando operações de compra e venda conforme os sinais de trading forem identificados.

4. Monitoramento e Logs

- **Azure Monitor:** O bot deve ser monitorado para garantir que as operações estão sendo realizadas sem erros. Azure Monitor deve ser configurado para enviar alertas em caso de falhas no sistema, como falhas de conexão com a Binance, falhas no banco de dados ou problemas de execução das ordens.

Regras do Projeto

1. Execução de Operações

- As operações de compra e venda devem ser feitas automaticamente, com base em sinais de trading predefinidos (como Engolfo e Inside Bar). O bot não pode realizar operações arbitrárias sem que haja um sinal válido.
- As ordens de compra ou venda devem ser configuradas com **Stop Loss** e **Take Profit**. O preço de Stop Loss deve ser configurado para limitar as perdas, e o Take Profit deve ser configurado para realizar a venda automaticamente quando um determinado preço de lucro for atingido.

2. Relatórios de Ganhos e Perdas

- Cada transação realizada deve ser registrada em um banco de dados, incluindo informações sobre a moeda, a quantidade comprada/vendida, os preços de compra e venda, e o total de ganho ou perda.

- O sistema deve gerar relatórios periódicos de ganhos e perdas para o usuário, que deve incluir:
 - Total de ganhos e perdas.
 - Desempenho histórico das transações.
 - Detalhes de cada transação (quando foi realizada, por qual preço, etc.).

3. Implementação na Nuvem (Azure)

- O código do bot deve ser hospedado no **Azure App Service**, que garantirá que o bot esteja em execução constante.
- O bot deve se conectar ao **Azure SQL Database** para armazenar as transações e os relatórios gerados.

4. Monitoramento e Alertas

- O bot deve ser configurado com **Azure Monitor** para garantir que todos os erros e exceções sejam registrados. Alertas devem ser configurados para notificar os desenvolvedores em caso de falha, como problemas na conexão com a API da Binance, problemas com o banco de dados ou qualquer erro crítico que possa impactar a operação do bot.

Etapas para Desenvolvimento

1. **Configuração do Ambiente de Desenvolvimento:**
 - Criar uma conta no Azure e configurar o **Azure App Service**.
 - Criar um banco de dados SQL no **Azure SQL Database**.
 - Configurar a conexão com a API da Binance para consulta de preços e execução de ordens.
2. **Desenvolvimento do Bot:**
 - Implementar a lógica de **sinais de trading** (Engolfo, Inside Bar).
 - Implementar a lógica de **compra e venda** de criptomoedas com **Stop Loss** e **Take Profit**.
 - Implementar o armazenamento de dados (transações e relatórios) no **Azure SQL Database**.
3. **Teste e Validação:**
 - Realizar testes no ambiente local e na nuvem para garantir que o bot está funcionando conforme esperado.
 - Validar que as ordens estão sendo executadas corretamente e que o cálculo de ganhos e perdas está correto.
4. **Deploy para o Azure:**
 - Fazer o deploy do código no **Azure App Service**.
 - Configurar **Azure Monitor** para rastrear erros e métricas de desempenho.
5. **Documentação e Apresentação:**

- Documentar o projeto, explicando as funcionalidades implementadas, como configurar o bot, como analisar os relatórios de desempenho e como configurar o monitoramento no Azure.
 - Apresentar o projeto para a turma, demonstrando o funcionamento do bot, os relatórios gerados e a operação na nuvem.
-

Critérios de Avaliação

- **Funcionalidade:** O bot deve realizar as operações corretamente, com base nos sinais de trading e com ordens de Stop Loss e Take Profit configuradas.
 - **Usabilidade:** O sistema deve ser fácil de configurar e usar. O aluno deve ser capaz de entender e manipular os parâmetros de operação.
 - **Desempenho:** O bot deve operar de maneira eficiente, sem falhas frequentes ou perda de dados.
 - **Escalabilidade:** O sistema deve ser capaz de rodar de forma escalável e sem interrupções no ambiente de nuvem (Azure).
 - **Documentação:** A documentação deve ser clara, explicando como o sistema funciona, como as operações são realizadas e como os relatórios podem ser acessados.
-