**Monitoramento de Execuções**

Este projeto é um sistema de **Monitoramento de Execuções** que acompanha e gerencia o processamento de execuções de fluxos em um banco de dados. A aplicação permite que o usuário visualize o status de diferentes execuções, com a possibilidade de realizar ações manuais, como **remover da fila**, **retornar para fila** e **reiniciar o fluxo** de execuções.

**Funcionalidades Principais**

1. **Exibição das Execuções**:

As execuções são listadas em uma tabela, mostrando detalhes como o ID da execução, nome da procedure associada, data de início, data de fim, status atual, duração e observações.

O sistema suporta **paginação**, exibindo as últimas 20 execuções mais recentes no topo, garantindo que o usuário tenha acesso rápido às execuções mais relevantes.

1. **Ações Manuais**:

**Remover da Fila**: Permite que execuções que estão na fila de processamento sejam removidas manualmente. Esta ação evita que o processamento continue, mas mantém um histórico da remoção no campo de observação da execução.

**Retornar para Fila**: Execuções que foram removidas da fila podem ser manualmente retornadas para o processamento. O robô então pega essas execuções novamente para processá-las. A ação é registrada no histórico.

**Reiniciar Fluxo**: Para execuções que estão em andamento ou com status de erro, o sistema oferece a opção de reiniciar o fluxo manualmente, também registrando essa ação no histórico.

1. **Histórico de Ações**:

O campo de **Observação** mantém um histórico das ações realizadas, como "Retirado da fila manualmente", "Retornado para fila manualmente", ou "Fluxo reiniciado manualmente". Isso garante que o usuário de negócios possa acompanhar todas as interações realizadas com cada execução.

1. **Atualizações em Tempo Real**:

A aplicação utiliza **SignalR** para atualizações em tempo real, permitindo que as mudanças nas execuções sejam refletidas diretamente na tela do usuário.

Uma contagem regressiva de **60 segundos** é exibida no canto superior direito da página, indicando quando a página será recarregada automaticamente, garantindo que as informações estejam sempre atualizadas.

1. **Log de Ações**:

O sistema mantém um log detalhado de todas as ações realizadas, facilitando auditorias e verificações sobre o que aconteceu com cada execução.

**Tecnologias Utilizadas**

* **ASP.NET Core MVC**: Para construção da aplicação web.
* **C#**: Linguagem principal do backend.
* **SQL Server**: Banco de dados utilizado para armazenar e gerenciar as execuções.
* **SignalR**: Para atualizações em tempo real no frontend.
* **HTML/CSS/JavaScript**: Utilizados na camada de visualização para exibir os dados e permitir a interação do usuário.
* **Bootstrap**: Para estilização e responsividade da interface.

**Como Funciona**

1. O sistema consulta o banco de dados **CadastroDB** para listar as execuções presentes na tabela TBy9\_LogExecucao.
2. O usuário pode interagir com a lista, realizando ações manuais para remover, retornar ou reiniciar execuções.
3. A cada ação realizada, o campo de observações é atualizado para manter o histórico, e um log é gerado para registrar o evento.
4. O sistema utiliza paginação para melhorar a experiência do usuário, exibindo as últimas 20 execuções e organizando as execuções em diferentes páginas.
5. Atualizações em tempo real são feitas para garantir que o usuário sempre tenha as informações mais recentes sem precisar recarregar a página manualmente.

Explicação sobre a procedure:

**Antes:**

A procedure originalmente:

1. **Criava uma nova execução sempre**:

Toda vez que o robô era executado, ela criava uma nova entrada no log com o status de *"Executando"*.

Um tempo de processamento (simulado com WAITFOR DELAY '00:00:05') era incluído para simular a execução de um processo.

Depois disso, atualizava o log, finalizando a execução com o status *"Concluída"* e a data de término.

1. **Não considerava execuções em fila**:

A procedure não verificava se já havia execuções pendentes ou em fila.

Apenas criava novas execuções independentemente do que já existia.

1. **Nenhuma observação automática**:

As execuções não recebiam um histórico ou observação sobre sua origem ou processamento.

**Agora:**

Após as alterações, a procedure foi ajustada para:

1. **Processar execuções pendentes ("Em Fila")**:

**Antes de criar uma nova execução**, a procedure verifica se há alguma execução com o status *"Em Fila"*.

Se houver uma execução em fila, ela:

* + - Atualiza o status para *"Executando"*.
    - Adiciona uma observação na coluna Observacao indicando que foi *"Retomado pelo robô"*.
    - Após o tempo de simulação (com WAITFOR DELAY), a procedure atualiza a execução como *"Concluída"* e adiciona mais uma observação indicando que foi *"Concluída pelo robô"*.

1. **Manter o histórico de observações**:

A coluna Observacao agora mantém um histórico de ações, adicionando as observações sem sobrescrever o conteúdo anterior. Isso inclui:

* + - Execuções retomadas pelo robô.
    - Execuções concluídas pelo robô.

1. **Criar novas execuções somente quando não houver pendentes**:

Se não houver nenhuma execução com o status *"Em Fila"*, a procedure cria uma nova execução no log.

Essa nova execução já vem com uma observação inicial: *"APROV\_AUTOMATICA"*.

Depois do processamento simulado, o log é atualizado com o status *"Concluída"* e a observação *"Concluída automaticamente"*.

**Benefícios das mudanças:**

* **Melhor gerenciamento de pendências**: Agora, a procedure não cria novas execuções sem antes verificar se há pendentes, otimizando o processo e garantindo que execuções em fila sejam priorizadas.
* **Histórico claro**: As observações foram ajustadas para incluir informações relevantes, diferenciando execuções manuais de automáticas e rastreando o histórico de ações do robô.
* **Processos automáticos claramente identificados**: As execuções que foram processadas automaticamente são claramente marcadas no campo de observação com *"APROV\_AUTOMATICA"*.

Essas mudanças garantem que o sistema mantenha o controle sobre execuções pendentes e documente claramente o que foi feito de forma manual e automática.