

8. Aula 08 – Aplicações do Tipo – Dashboard, Blank e Gráficos

Objetivos

Nesta aula, iremos criar um conjunto de aplicações que serão usados para exibir outras aplicações, onde nosso sistema terá uma interface amigável para o usuário final.

8.1. Nova Aplicação – Aplicações do Tipo Aba – Agendamento

Este grupo irá utilizar 3 (três) aplicações no mesmo fluxo: Agendamento – *grid_agendamentos*, Histórico do Agendamento – *grid_historico_agendamento* e Detalhes do Agendamento – *grid_detalhe_agendamento*.

Nome da Aplicação: tabs_agendamento

1. Criar 3 (três) abas na aplicação.
2. Para cada aba deverá montar a ligação com as seguintes aplicações: Agendamento – *grid_agendamentos*, Histórico do Agendamento – *grid_historico_agendamento* e Detalhes do Agendamento – *grid_detalhe_agendamento* e associar a variável *var_appointment_id* e *var_cur_status*.
3. Agora precisamos fazer a ligação através dos campos (criar o campo) com as seguintes ligações: *grid_agendamento_admin*, *grid_agendamento_clientes* e *grid_agendamento_tecnico*.

8.2. Nova Aplicação – DashBoard do Administrador.

Nós nos iremos criar 4 (quatro) aplicações que irão mostrar algumas informações importantes sobre os agendamentos para o usuário do tipo administrador. Em seguida, iremos criar um painel para mostrar os aplicativos ao mesmo tempo no menu do usuário tipo administrador.

8.2.1. Nova Aplicação – grid_agendamentos_adm_dashboard

Esta aplicação mostrar informações para cada membro da equipe cadastrado no sistema.

1. Incluir o comando SQL no momento da criação da aplicação (código anexo).
2. Remover todos os botões da barra de ferramentas e remover o cabeçalho.

8.2.2. Nova Aplicação – chart_agendamentos

Gráficos com o total dos agendamentos totalizando por mês.

1. Criar uma aplicação do tipo gráfico baseado na tabela agendamentos.
2. Campos Seleccionados: *data_inicial* (YYYYMM) e Métricas: *Quantidade de Registros* e *valor_total* (Trimestre).
3. Em configurações mudar o parâmetro do gráfico para *Desktop Responsivo*.
4. Remover o cabeçalho e todos os botões.

8.2.3. Nova Aplicação – chart_agendamentos_avg_avaliacao

Gráfico com a média das avaliações dos agendamentos.

1. Utilizar a tabela agendamentos.
2. Incluir uma clausula WHERE no SQL da aplicação: *WHERE status_atual = 5 and avaliacao > 0*.
3. Dimensões: *data_inicial* (YYYYMM) e *id_tecnico* e Métricas: *Quantidade* de registros e *avaliacao* (AVG).
4. Incluir o lookup de consulta para o campo *id_tecnicos*.
5. Em configurações mudar o parâmetro do gráfico para *Desktop Responsivo*.
5. Remover o cabeçalho e todos os botões.

8.2.4. Nova Aplicação – chart_agendamentos_vendas

Gráfico com os 3 maiores clientes - Valores.

6. Utilizar a tabela *agendamentos*.
7. Dimensões: *id_cliente* e Métricas: *valor_total* (SUM).
8. Incluir o lookup de consulta para o campo *id_cliente*.
9. Em configurações mudar o parâmetro do gráfico para *Desktop Responsivo*.
10. Em ordenação a *Ordenação Inicial* deverá ser descendente para o campo *valor_total_sum*.
11. Em gráfico escolher o seguinte tipo de gráfico *Barra » Coluna 2D*.
12. Em Limit escolher o campo = *valor_total* , sumarização = *Soma*, Tipo de Limit = *Topo* e Quantidade = 3.
13. Remover o cabeçalho e todos os botões.

8.2.5. Nova Aplicação – dashboard_admin

Agora iremos criar um painel (Dashboard) para mostrar as 4 (quatro) aplicações criadas anteriormente, mais 3 (três) que serão criadas na aplicação de dashboard.

1. Deverá ser criada uma aplicação do tipo dashboard com 7 (sete) colunas.
2. 1º Widget criar será do tipo *Índice*, onde terá os seguintes parâmetros; Tabela – *agendamentos*, campo de métrica – *valor_total*, Função da Métrica – *Soma*, Exibir como – *Diferença*, Campo de Período – *data_inicial*, Função de Período – *Ano e Trimestre* e Exibir Valores de – *Ambos os períodos*.
3. 2º Widget criar será do tipo *Índice*, onde terá os seguintes parâmetros; Tabela – *agendamentos*, campo de métrica – *valor_total*, Função da Métrica – *Média*, Exibir como – *Diferença*, Campo de Período – *data_inicial*, Função de Período – *Ano e Trimestre* e Exibir Valores de – *Ambos os períodos*.

4. 3º Widget criar será do tipo *Índice*, onde terá os seguintes parâmetros; Tabela – *agendamentos*, campo de métrica – *id_agendamento*, Função da Métrica – *Contagem*, Exibir como – *Valor*, Campo de Período – *data_inicial*, Função de Período – *Ano* e Exibir Valores de – *Ambos os períodos*.
5. 4º Widget criar será do tipo *Ligação*, e chamar a aplicação *grid_agendamentos_adm_dashboard*.
6. 5º Widget criar será do tipo *Ligação*, e chamar a aplicação *chart_agendamentos_vendas*.
7. 6º Widget criar será do tipo *Ligação*, e chamar a aplicação *chart_agendamentos*.
8. 7º Widget criar será do tipo *Ligação*, e chamar a aplicação *chart_agendamentos_avg_avaliacao*.

8.3. Nova Aplicação – Aplicação do Tipo Calendário para o Administrador

Primeiro iremos utilizar uma biblioteca externa JS para nos ajudar a criar a aplicação de calendário.

- Ferramentas > Bibliotecas Externas, iremos criar uma nova biblioteca chamada *lib_appointments*, que iremos salvar no projeto.
- Iremos criar uma nova pasta chamada *js* e dentro desta pasta iremos criar um novo arquivo chamado '*js_appointments.js*', depois iremos mudar o parâmetro da biblioteca externa para Usar Biblioteca e salvar. (Código anexo).
- Criar uma nova aplicação do tipo Blank chamada *meu_calendario* e usar o código incluso na pasta (código anexo).
- Criar uma nova aplicação do tipo Blank chamada "*json*" e utilizar o código incluso na pasta (código anexo).
- Gerar o código fonte e chamar aplicação *meu_calendario*.