

3. Aula 03 – Aplicações do Administrador

Objetivos
Nesta lição, iremos criar aplicações de suporte para o administrador. A primeira aplicação será um relatório completo com todos os detalhes dos agendamentos, outra aplicação onde o administrador poderá selecionar os serviços, a aplicação irá calcular a duração e o valor total do serviço e outra aplicação aonde o administrador poderá ter uma visão gerencial de todos os agendamentos existentes.

3.1. Nova Aplicação – Relatório de Agendamento – Administrador

Tabela: agendamento

Nome da Aplicação: grid_agendamento_adm

1. Cria a aplicação.
2. Adicionar o comando no SQL: (código anexo)
3. Selecionar os seguintes campos: *data_inicial*, *hora_inicial*, *id_tecnicos*, *id_clientes*, *valor_total* e *status_atual*.
4. Os seguintes campos deverão utilizar *Lookup Consulta*: *id_tecnicos*, *id_clientes* e *status_atual*.
5. Quebra Estática para o campo: *id_tecnico*.
6. Quebra Dinâmica para os campos: *data_inicial* e *id_clientes*.
7. Filtro Dinâmico para os campos: *valor_total*, *id_clientes* e *id_tecnico*.
8. Incluir as aplicações no menu abaixo do item *Administrador*.

3.2. Nova Aplicação – Formulário de Agendamento – Admin

Aplicação será utilizada para agendar os atendimentos dos técnicos. Ao selecionar uma data e hora específica, ela irá calcular o valor do serviço de forma automática e, em seguida, irá enviar um e-mail para todos os envolvidos (Clientes e Técnicos).

Tabela: agendamentos

Nome da Aplicação: form_agendamentos

1. Retirar os campos: *Id_agendamento*, *recorrencia*, *período*, *avaliacao*, *data_final* e *hora_final*.
2. Criar um bloco: O título não deve aparecer (Exceto o bloco de serviço, ele deve começar fechada e com a exibição do label ativado), os labels dos campos deverão estar acima.
 - a. Bloco Inicial com 3 colunas; *id_tecnico*, *id_cliente* e *status_atual*;
 - b. Data e Hora (2º bloco) com 2 colunas: *data_inicial* e *hora_inicial*;
 - c. Valores (3º bloco) com 4 colunas: *valor*, *valores_adicionais*, *desconto* e *valor_total*;
 - d. Serviço (4º bloco) com 1 coluna, detalhe dos serviços sem label no bloco.
3. Altere os campos *id_tecnicos* e *id_cliente* para tipo *número auto_complete* e faça a ligação com suas respectivas bases de dados.
4. Altere o campo *status_atual* para tipo *Radio* e faça a ligação com suas respectivas bases de dados.
5. Para os campos *hora_inicial* e *data_inicial* remova a exibição do formato.
6. Altere as configurações dos campos *valor* e *valor_total* para tipo *label*.
7. Criar um método PHP (*m_php_upd_price*) para atualização do campo *valor_total* (código anexo).

8. Criar *evento onChange ajax* para os campos *valores_adicionais* e *desconto* para que quando forem alterados chame o método criado acima (código anexo).
9. Criar um novo método PHP (*m_php_fill_values*) (código anexo).
10. Chamar o método PHP (*m_php_fill_values*) criado acima nos *eventos ajax onChange* para os seguintes campos: *data_inicial* e *hora_inicial* (código em anexo).
11. Crie a aplicação detalhe:
 - a. Nova Aplicação – Formulário de Requisição de Agendamento Detalhe - *grid view editável*;
 - b. **Tabela:** *agendamento_detalhe*
 - c. **Nome da Aplicação:** *form_detalhes_agendamento*
12. Mostrar somente o campo *id_tipos_agendamento* e deverá ser do tipo *Select* para que possa mostrar o nome do serviço.
13. Criar 2 (dois) campos virtuais do tipo *Label*, são eles *duracao* e *preco* e alterar as configurações dos campos para tipo *label*.
14. Criar um novo método PHP (*m_php_get_price*) (código anexo).
15. Chamar o método *m_php_get_price* no *evento ajax OnChange* para o campo *id_tipos_agendamento* e no evento *onLoadRecord*.
16. Criar um método PHP (*m_php_upd_master*) (código anexo).
17. Chamar o método PHP nos seguintes eventos *onAfterInsert*, *onAfterUpdate* e *onAfterDelete events*.
18. Gerar o Código Fonte.
19. Retornar ao formulário *form_agendamentos*.
20. Criar uma ligação do tipo *Mestre/Detalhe* abaixo do bloco serviços.

21. Criar uma ligação do tipo aplicação com a aplicação *grid_agendamentos_admin* e em propriedades do formulário habilitar somente o *botão de update* e *botão de edição do registro na consulta*.
22. Criar um método PHP (*m_php_send_email*) (código anexo).
23. Chamar o método PHP nos seguintes eventos *onAfterInsert*, *onAfterUpdate*.
24. Testar a Aplicação.

Aplicação de Menu -> Incluir as aplicações no menu abaixo do item 'Administrador'.

3.3. Nova Aplicação – Sumário de Agendamento – Administrador

Tabela: agendamento

Nome da Aplicação: grid_sumario_agendamento_admin

1. Cria a aplicação do tipo consulta (grid).
2. Adicionar o comando no SQL: WHERE (código anexo)
3. Criar uma Quebra Estática como os seguintes campos; *data_inicial*(por Ano e Trimestre), e *id_tecnico* em *configurações de lookup* e faça a ligação com suas respectivas bases de dados. O campo *valor_total* deverá aparecer das seguintes formas; *Soma* e *Média*.
4. A aplicação deverá inicial pela opção de *Resumo*.
5. Na opção de *resumo* definir os seguintes campos como filtro; *valor_total*.
6. Incluir as aplicações no menu abaixo do item *Administrador*.

3.4. Exercício

Importante na aula de hoje é replicar cada passo deste documento.