ProjetoMedCal - Fluxo de cadastro de paciente e agendamento médico

Caso de uso

* Informatizar o fluxo de atendimentos de um médico, criando um cadastro simples de pacientes CRUD (Nome Completo, Data de Nascimento, Endereço, Sexo, Telefone e E-mail);
* Criar um cadastro básico de agenda de consultas (CRUD) para estes pacientes, contendo os seguintes atributos: (Data de Agendamento, Horário, Especialidade (Criar um tabela auxiliar com uma lista simples: Clínico Geral, Cardiologia ou Neurologista) e Status (Executado, Pendente).

Características:

* Criar as entidades em uma tabela em banco de dados relacional (preferencialmente postgres). Deverá utilizar Foreign Key na tabela de médicos para integridade.;
* Utilizar um modelo MVC (Model View Controller) para manipular os dados.
* A camada de controle deverá seguir o padrão REST e receber e devolver dados em formato JSON.
* utilizar um formulário básico em React para consumir essa API e manipular os dados no banco de dados.

1 - Regras de negócio Banco de dados:

* Cadastro de pacientes:
  + Código do paciente - Pk - Not Null - Auto\_increment
  + Nome completo - Varchar - Not Null -
  + Data de nascimento - date - not null
  + Endereço - VARCHAR(150) - notNull
  + Numero - VARCHAR(20)
  + Bairro - VARCHAR(50)
  + Cep - Int(8), varchar(10)
  + Cidade - VARCHAR(50) - not null
  + Estado - varchar not null
  + Sexo - varchar - not null
  + Telefone - BIGINT(14)
  + E-mail - VARCHAR(100)
* Cadastro de Convênio
  + Código - Pk - Not Null - Auto\_increment
  + Descrição - Varchar(50)
* Cadastro de médico
  + Código - Pk - Not Null - Auto\_increment
  + Nome - Varchar(80)
  + Especialidade - Varchar (80) - Chave estrangeira
  + Crm - Varchar(20)
* Especialidade:
  + Código - Pk - Not Null - Auto\_increment
  + Descrição - VARCHAR(30)
* Status
  + Código - PK
  + Descrição
* Agendamento:
  + Código - Pk - Not Null - Auto\_increment
  + Data -
  + Horário
  + Paciente
  + Médico
  + Especialidade
  + Convênio
  + Status

2 - Regras de negócio backend:

* Requisição tipo post para cadastro de clientes
  + Validar
    - Data do tipo date
    - Horário number
    - Paciente String, não vazio e maior que 3 caracteres
    - Médico não vazio
    - Especialidade não vazio
    - Convênio não vazio
    - Retorna um json com os dados do agendamento
* Requisição tipo post para cadastro do médico
  + Validar
    - Nome > 3 não vazio, string
    - Retorna json com cadastro do médico
* Requisição tipo post para cadastro do convenio
  + Validar
    - Descrição > 3 não vazio, string
    - Retorna json com cadastro de convenio
* Requisição tipo post para agendamento de paciente
  + Validar
    - Data tipo date, não vazio, number
    - Horário não vazio
    - Nome não vazio
    - Médico não vazio
    - Convênio não vazio
* Requisição do tipo put para alteração de agendamento (Não pode permitir apagar todos os dados)
* Requisição do tipo get para consulta de agendamento
* Requisição do tipo get para consulta de médico

Iniciando o Backend da aplicação

* Dados solicitado para agendamento de paciente:
  + Rota post:
    - id do agendamento - appointment\_Id - (Gerado automaticamente);
    - data do agendamento - scheduleData;
    - Nome do paciente - patientName;
    - Nome do médico - doctorName
    - Especialidade do médico - doctorSpecialty
    - Status do agendamento - scheduleStatus
      * Estrutura da rota post
        + Arquivo contendo a classe de agendamentos instanciada