

ROTEIRO DE TESTES – Consultas Medicas Agendadas API

CONTROLLERS:

UsuarioController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar usuários através do método *Listar*, do controlador *UsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os usuários na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar usuários através do método *Listar*, do controlador *UsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os usuários na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar usuários médicos através do método *ListarMedico*, do controlador *UsuarioController*.

Caso de Teste:

Descrição: Testar o método *ListarMedico* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os usuários médicos na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar usuários pacientes através do método *ListarPaciente*, do controlador *UsuarioController*.

Caso de Teste:

Descrição: Testar o método *ListarPaciente* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os usuários pacientes na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar usuários através do método *Listar*, do controlador *UsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os usuários na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

PacienteController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar pacientes através do método *Listar*, do controlador *PacienteController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os pacientes na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar pacientes através do método *Listar*, do controlador *PacienteController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os pacientes na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Inserir pacientes através do método *Cadastrar*, do controlador *PacienteController*.

Caso de Teste:

Descrição: Testar o método *Cadastrar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Inserir os pacientes na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar pacientes através do método *Listar*, do controlador *PacienteController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os pacientes na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

MedicoController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar médicos através do método *Listar*, do controlador *MedicoController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os médicos na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar médicos através do método *Listar*, do controlador *MedicoController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os médicos na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Inserir médicos através do método *Cadastrar*, do controlador *MedicoController*.

Caso de Teste:

Descrição: Testar o método *Cadastrar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Inserir os médicos na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar médicos através do método *Listar*, do controlador *MedicoController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os médicos na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

TipoUsuarioController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar os tipos de usuários através do método *Listar*, do controlador *TipoUsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os tipos de usuários na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar os tipos de usuários através do método *Listar*, do controlador *TipoUsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os tipos de usuários na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar os tipos de usuários através do método *Listar*, do controlador *TipoUsuarioController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todos os tipos de usuários na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

EspecialidadeController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar especialidades através do método *Listar*, do controlador *EspecialidadeController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todas as especialidades na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar especialidades através do método *Listar*, do controlador *EspecialidadeController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todas as especialidades na base de dados fake.

Resultado esperado: Um retorno como *Status Code: 200*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Inserir especialidades através do método *Cadastrar*, do controlador *EspecialidadeController*.

Caso de Teste:

Descrição: Testar o método *Cadastrar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Inserir as especialidades na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar especialidades através do método *Listar*, do controlador *EspecialidadeController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todas as especialidades na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

ConsultaController:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar consultas através do método *Listar*, do controlador *ConsultaController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar/consultar todas as consultas na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar consultas através do método *Listar*, do controlador *ConsultaController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar todas as consultas na base de dados fake.

Resultado esperado: Um retorno como *Status Code*: 200.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Inserir consultas através do método *Cadastrar*, do controlador *ConsultaController*.

Caso de Teste:

Descrição: Testar o método *Cadastrar* para ver se está funcionando corretamente.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Inserir as consultas na base de dados fake.

Resultado esperado: Um retorno como *OkObjectResult*.

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Buscar consultas através do método *Listar*, do controlador *ConsultaController*.

Caso de Teste:

Descrição: Testar o método *Listar* para ver se está funcionando corretamente e verificar se ele retorna um objeto nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Buscar todas as consultas na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

MODELS:

Usuario:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar usuários através do objeto criado, da classe *Usuario*.

Caso de Teste:

Descrição: Testar o objeto *Usuario* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todos os usuários na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

Paciente:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar pacientes através do objeto criado, da classe *Paciente*.

Caso de Teste:

Descrição: Testar o objeto *Paciente* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todos os pacientes na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

Medico:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar médicos através do objeto criado, da classe *Medico*.

Caso de Teste:

Descrição: Testar o objeto *Medico* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todos os médicos na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

TipoUsuario:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar os tipos de usuários através do objeto criado, da classe *TipoUsuario*.

Caso de Teste:

Descrição: Testar o objeto *TipoUsuario* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todos os tipos de usuários na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

Especialidade:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar especialidades através do objeto criado, da classe *Especialidade*.

Caso de Teste:

Descrição: Testar o objeto *Especialidade* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todas as especialidades na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.

Consulta:

Localização: Tela da IDE através do Gerenciador de Testes. E tela do *Swagger* da API.

Objeto de Teste: Verificar consultas através do objeto criado, da classe *Consultas*.

Caso de Teste:

Descrição: Testar o objeto *Consultas* para ver se está funcionando corretamente e verificar se ele é nulo.

Pré-condição: API – Consultas Medicas completa.

Procedimento: Verificar todas as consultas na base de dados fake.

Resultado esperado: Um retorno como *NotNull*.