

1. Mundo comum

Uma nova clínica médica chamada SP Medical Group, empresa de pequeno porte que atua no ramo da saúde, foi criada pelo médico Fernando Strada em 2020 na região da Paulista em São Paulo. Fernando tem uma equipe de médicos que atuam em diversas áreas (pediatria, odontologia, gastroenterologia etc.).

Sua empresa, por ser nova, iniciou a administração dos registros de forma simples, utilizando softwares de planilhas eletrônicas e, com o sucesso da clínica, sua gestão passou a se tornar complicada devido à alta demanda dos pacientes.

2. Chamado para a aventura

Fernando Strada lhe contratou para desenvolver um sistema web/mobile integrado onde seja possível realizar a gestão da clínica de forma automatizada e ter acesso fácil aos dados sobre as informações de seus pacientes.

Fernando levantou os seguintes requisitos:

Sistema Web

Perfis de usuário:

1. **Administrador:** Para o colaborador da área administrativa da clínica;
2. **Médico:** Colaboradores que atuam na área da saúde;
3. **Paciente:** Clientes da clínica;

Funcionalidades:

1. O **administrador** poderá cadastrar qualquer tipo de usuário (administrador, paciente ou médico);
2. O **administrador** poderá agendar uma consulta, onde será informado o paciente, data e hora do agendamento e qual médico irá atender a consulta (o médico possuirá sua determinada especialidade);
3. O **administrador** poderá cancelar o agendamento;
4. O **administrador** deverá informar os dados da clínica (como endereço, horário de funcionamento, CNPJ, nome fantasia e razão social);
5. O **médico** poderá ver os agendamentos (consultas) associados a ele;

6. O **médico** poderá incluir a descrição da consulta que estará vinculada ao paciente (prontuário);
7. O **paciente** poderá visualizar suas próprias consultas;

Sistema Mobile

Perfis de usuário:

1. **Médico:** Colaboradores que atuam na área da saúde;
2. **Paciente:** Clientes da clínica;

Funcionalidades:

1. O **paciente** poderá visualizar suas próprias consultas;
2. O **médico** poderá ver as consultas (agendamentos) associados a ele;

3. SPRINT 1 – Banco de dados

A clínica SP Medical Group contratou você para desenvolver sua solução em software. De acordo com a cultura de desenvolvimento de softwares criada no seu aprendizado, o primeiro passo para a construção do sistema é desenvolver a modelagem do banco de dados que será utilizado em seus modelos conceitual, lógico e físico.

Para ajudar na construção da modelagem, Fernando Strada disponibilizou as planilhas onde guardava os dados administrativos da clínica, com informações referentes a: médicos cadastrados, pacientes e consultas. Você utilizará estes documentos como base para a construção do banco e dados.

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade

Banco de dados

1. Identificar as características de banco de dados relacional e não-relacional

Definiu corretamente a importância do banco de dados

Definiu corretamente o que é banco de dados relacional

Identificou as diferenças entre bancos de dados relacionais e não-relacionais

2. Configurar o ambiente para utilização de banco de dados relacional

Conectou-se corretamente ao banco de dados local

Liberou o acesso remoto para acesso externo

4. Elaborar diagramas de modelagem do banco de dados de acordo com a arquitetura definida

Exportou o diagrama de modelo lógico de acordo com a situação problema

Exportou o diagrama de modelo conceitual de acordo com a situação problema

Exportou o diagrama de modelo físico de acordo com a situação problema

Obs.: brModelo / draw.io

7. Documentar a estrutura do banco de dados por meio de dicionário de dados

Criou o arquivo de documentação para entrega ao cliente conforme solicitado na situação problema

Incluiu as modelagens no arquivo de documentação conforme solicitado no item 4.

Detalhou corretamente as descrições das modelagens

3. Utilizar tipos de dados para definição dos atributos do banco de dados

Utilizou os tipos de dados corretos para os campos das tabelas (usuário, paciente e especialidade)

Utilizou o default para deixar um valor padrão para o valor na tabela de consultas

5. Utilizar relacionamentos entre as tabelas do banco de dados

Relacionou as tabelas de acordo com a situação problema (vinculou a tabela de consultas com a de pacientes e médicos, vinculou a tabela de médicos com a tabela de especialidades, vinculou a tabela de médicos com a tabela de clínicas)

Definiu as chaves primárias de todas as tabelas de acordo com a situação problema

9. Utilizar linguagem de definição de dados (DDL)

Criou todas as tabelas de acordo com a situação problema (especialidades, médicos, pacientes, consultas e clínicas)

10. Utilizar linguagem de manipulação

Inseriu os dados na tabela de usuários de acordo com as colunas criadas no item 9 e especificadas na situação problema

Atualizou os registros que não possuem data de nascimento conforme especificado pelo cliente

6. Normalizar a estrutura do banco de dados

Utilizou a primeira, a segunda e a terceira formas normais de banco de dados

Utilizou a quarta e a quinta formas normais de banco de dados

13. Executar importação e exportação da base de dados

Importou os dados para o banco de dados criado de acordo com a planilha recebida

Os dados importados estão de acordo com os dados da planilha recebida

Exportou a base de dados para um arquivo .sql

Exportou a base de dados para um arquivo .csv

11. Utilizar funções nativas do banco de dados

Mostrou a quantidade de usuários após realizar a importação do banco de dados

Converteu a data de nascimento do usuário para o formato (mm-dd-yyyy) na exibição

Calculou a idade do usuário a partir da data de nascimento

12. Aplicar programação em banco de dados utilizando functions, stored procedures, triggers e eventos

Criou um evento para que a idade do usuário seja calculada todos os dias

Criou uma função para retornar a quantidade de médicos de uma determinada especialidade

Criou uma função para que retorne a idade do usuário a partir de uma determinada stored procedure

Capacidades sociais, organizativas e metodológicas

1. Demonstrar atenção aos detalhes

Realizou os exercícios propostos de acordo com as instruções

2. Demonstrar capacidade de organização

Utilizou a ferramenta Trello para organizar e controlar as atividades a serem realizadas no projeto, de acordo com a metodologia ágil

3. Demonstrar raciocínio lógico na organização das informações

Resolver problemas inesperados, que serão propostos pelos instrutores

4. Demonstrar visão sistêmica

Identificar e demonstrar por meio do projeto como o banco de dados se conecta com todo o resto do sistema

5. Seguir método de trabalho

Aplicar metodologia ágil nas atividades individuais e de projeto utilizando a ferramenta Trello.

4. SPRINT 2 – Back-End (API)

De acordo com os padrões modernos no desenvolvimento de projetos de software atuais, você decidiu que o software solicitado pela SP Medical Group deverá ser criado em plataforma API (Application Programming Interface).

Você deverá desenvolver a API com as seguintes funcionalidades:

Perfis de usuário:

1. **Administrador:** Para o colaborador da área administrativa da clínica;
2. **Médico:** Colaboradores que atuam na área da saúde;
3. **Paciente:** Clientes da clínica;

Funcionalidades:

1. O **administrador** poderá cadastrar qualquer tipo de usuário (administrador, paciente ou médico);
2. O **administrador** poderá agendar uma consulta, onde será informado o paciente, data do agendamento e qual médico irá atender a consulta (o médico possuirá sua determinada especialidade);
3. O **administrador** poderá cancelar o agendamento;
4. O **administrador** deverá informar os dados da clínica (como endereço, horário de funcionamento, CNPJ, nome fantasia e razão social);
5. O **médico** poderá ver os agendamentos (consultas) associados a ele;
6. O **médico** poderá incluir a descrição da consulta que estará vinculada ao paciente (prontuário);
7. O **paciente** poderá visualizar suas próprias consultas;

Persistência de dados

Deve ser realizada no banco de dados criado anteriormente na fase de modelagem do sistema.

Autenticação

Deve utilizar o modelo JWT (JSON Web Token) para realizar a autenticação no sistema.

Autorização

Os endpoints devem ter autorização baseada nas funcionalidades descritas acima. Onde a funcionalidade deve ser somente feita pelo ator estabelecido nela.

Exemplo: O **administrador** poderá **cancelar o agendamento**.

Administrador: Ator

Cancelar o agendamento: Funcionalidade do endpoint.

Documentação

Deve ser criada a documentação de todos os endpoints disponíveis na aplicação.

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade.

Web Back-End

1. Identificar as características de programação back-end em ambiente web

Identificou as características dos verbos HTTP (GET, POST, PUT, DELETE)

Identificou as diferenças entre um aplicativo WebAPI e um WebMVC

2. Preparar o ambiente necessário ao desenvolvimento back-end para a plataforma web

Utilizou o Visual Studio para a criação de um novo projeto WebAPI

Utilizou o prompt de comando para criar um novo projeto WebAPI

4. Utilizar design patterns no desenvolvimento da aplicação web

Criou uma solução com a separação correta de pastas (domínio, interface, repositório e controle) de acordo com a situação problema e seguindo as boas práticas

Utilizou os verbos corretos na criação dos endpoints

Retornou os dados após cadastro de um novo registro (especialidades, pacientes)

10. Desenvolver sistemas web de acordo com as regras de negócio estabelecidas

Não permitiu que uma consulta fosse inserida sem um médico vinculado

Não permitiu que uma consulta fosse inserida sem data de consulta

5. Definir os frameworks a serem utilizados no desenvolvimento da aplicação web

Os dados são salvos na base de dados utilizando um ORM

6. Utilizar interações com base de dados para desenvolvimento de sistemas web

Conectou-se a base de dados local utilizando o SQLServer cliente

Realizou as operações de leitura, inserção, exclusão e atualização (CRUD – Create, Read, Update e Delete) com o banco de dados

9. Desenvolver API (web services) para integração de dados entre plataformas

Criou os endpoints de acesso conforme situação problema

Restringiu o acesso apenas aos usuários autenticados e autorizados conforme situação problema

3. Definir os elementos de entrada, processamento e saída para a programação da aplicação web

Persistiu os dados enviados pela aplicação cliente conforme situação problema

Validou todos os campos com menor tamanho do que o permitido pela aplicação

Validou todos os campos requeridos enviados pela aplicação cliente

Validou a data de nascimento do paciente para que não seja maior do que a data atual

7. Transferir arquivos entre cliente e servidor por meio de aplicação web

Recebeu da aplicação cliente um arquivo do tipo imagem e salvou o caminho no banco de dados

Restringiu a aplicação para que salve apenas arquivos com o tipo .png e .jpg

11. Publicar a aplicação web

Disponibilizou em ambiente online a aplicação back-end funcional

Utilizou corretamente os arquivos de configuração em aplicação para ambientes de produção e desenvolvimento

Endpoints:

1. Login;
2. Lista de médicos;
3. Lista de pacientes;
4. Cadastro de consulta;
5. Lista de consultas (todas / somente de um paciente / somente de um médico);
6. Documentação;

Capacidades sociais, organizativas e metodológicas

1. Demonstrar atenção aos detalhes

Realizar os exercícios propostos de acordo com as instruções

2. Demonstrar capacidade de comunicação com profissionais de diferentes áreas e especialidades

Realizar entrevista com o cliente do projeto para levantar requisitos

3. Demonstrar capacidade de organização

Utilizar a ferramenta Trello para organizar e controlar as atividades a serem realizadas no projeto, de acordo com a metodologia ágil

4. Demonstrar raciocínio lógico na organização das informações

Resolver problemas inesperados, que serão propostos pelos instrutores.

5. Demonstrar visão sistêmica

Identificar e demonstrar por meio do projeto como o back-end se conecta com todo o resto do sistema

6. Seguir método de trabalho

Aplicar metodologia ágil nas atividades individuais e de projeto utilizando a ferramenta Trello

7. Trabalhar em equipe

Participar ativamente das decisões e desenvolvimento das atividades em equipe

5. SPRINT 3 – UI / UX

A SP Medical Group é uma empresa que preza pela qualidade de seus serviços assim como a interface do usuário. E, hoje em dia, sabemos o quanto a interface com o usuário é importante para que a experiência seja satisfatória em qualquer sistema.

Você deverá desenvolver as seguintes atividades:

- Layout de baixa fidelidade;
- Layout de alta fidelidade;
- Teste de usabilidade;

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade.

Web Front-End

5. Diferenciar os aspectos de aplicabilidade entre as experiências do usuário (UX) e a interface do usuário (UI)

Criou os layouts de alta fidelidade para web de acordo com a situação problema

Criou os layouts de baixa fidelidade web de acordo com a situação problema

Realizou os testes de usabilidade da aplicação web com o cliente da aplicação de acordo com a situação problema

Interfaces para dispositivos móveis

6. Projetar interfaces para dispositivos móveis

Criou os layouts de alta fidelidade para dispositivos móveis de acordo com a situação problema

Criou os layouts de baixa fidelidade para dispositivos móveis de acordo com a situação problema

Realizou os testes de usabilidade da aplicação mobile com o cliente da aplicação de acordo com a situação problema

4. Interpretar os requisitos do sistema, tendo em vista a elaboração dos componentes em ambiente mobile

Criou os componentes de lista conforme situação problema

Criou os componentes de entrada de dados conforme situação problema

Capacidades sociais, organizativas e metodológicas

1. Demonstrar atenção aos detalhes

Realizar os exercícios propostos de acordo com as instruções

2. Demonstrar capacidade de comunicação com profissionais de diferentes áreas e especialidades

Realizar entrevista com o cliente do projeto para levantar requisitos

3. Demonstrar capacidade de organização

Utilizar a ferramenta Trello para organizar e controlar as atividades a serem realizadas no projeto, de acordo com a metodologia ágil

4. Demonstrar raciocínio lógico na organização das informações

Resolver problemas inesperados, que serão propostos pelos instrutores.

5. Demonstrar visão sistêmica

Identificar e demonstrar por meio do projeto como o front-end se conecta com todo o resto do sistema

6. Seguir método de trabalho

Aplicar metodologia ágil nas atividades individuais e de projeto utilizando a ferramenta Trello

7. Trabalhar em equipe

Participar ativamente das decisões e desenvolvimento das atividades em equipe

6. SPRINT 4 – Front-End

Após ter desenvolvido toda a interface – API do seu sistema e seus dados estarem sendo armazenados no banco, você agora necessita desenvolver a interface que o usuário fará interação.

Você utilizará a biblioteca ReactJs para desenvolver a parte do front-end e realizar a integração com a API desenvolvida anteriormente.

Perfis de usuário:

1. **Administrador:** Para o colaborador da área administrativa da clínica;
2. **Médico:** Colaboradores que atuam na área da saúde;
3. **Paciente:** Clientes da clínica;

Funcionalidades:

1. O **administrador** poderá cadastrar qualquer tipo de usuário (administrador, paciente ou médico);
2. O **administrador** poderá agendar uma consulta, onde será informado o paciente, data do agendamento e qual médico irá atender a consulta (o médico possuirá sua determinada especialidade);
3. O **administrador** poderá cancelar o agendamento;
4. O **administrador** deverá informar os dados da clínica (como endereço, horário de funcionamento, CNPJ, nome fantasia e razão social);
5. O **médico** poderá ver os agendamentos (consultas) associados a ele;
6. O **médico** poderá incluir a descrição da consulta que estará vinculada ao paciente (prontuário);
7. O **paciente** poderá visualizar suas próprias consultas;

Autenticação

Somente usuários autenticados deverão acessar as páginas restritas de sua aplicação.

Autorização

Somente usuários autorizados poderão visualizar as suas próprias informações.

Exemplo: Um **paciente** poderá **visualizar** somente informações referentes a ele mesmo.

Telas

Administrador / Médico / Paciente

1. Login;

Administrador

1. Cadastrar nova consulta;
2. Listagem de consultas;

Médico

1. Listagem de consultas;
2. Inserir uma nova descrição na consulta;

Paciente

1. Listagem de consultas;

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade.

Web Front-End

1. Adequar a interface web para diferentes dispositivos de acesso

Utilizou media queries para deixar o site responsivo conforme situação problema (Mobile, iPad e Web – 600, 768, 1366, 1920)

Utilizou JavaScript para adequar o site a diferentes tipos de acesso

2. Desenvolver interfaces web interativas com linguagem de programação

Validou os campos de data de nascimento do paciente (não pode ser maior que a data atual)

Validou os campos requeridos na aplicação cliente

Inseriu novos registros na tabela de pacientes a partir dos dados inseridos pelo usuário

4. Desenvolver interfaces web consumindo API

Enviou os dados da aplicação cliente para o servidor

Listou as informações do servidor na aplicação cliente

Realizou a atualização dos registros em tela, após enviar uma atualização para o servidor

3. Desenvolver interfaces web utilizando frameworks

Utilizou algum framework front-end na construção do layout

Disponibilizou a aplicação em ambiente online conforme situação problema

Utilizou uma dependência externa para facilitar o desenvolvimento do software (conexão com a API)

Capacidades sociais, organizativas e metodológicas

1. Demonstrar atenção aos detalhes

Realizar os exercícios propostos de acordo com as instruções

2. Demonstrar capacidade de comunicação com profissionais de diferentes áreas e especialidades

Realizar entrevista com o cliente do projeto para levantar requisitos

3. Demonstrar capacidade de organização

Utilizar a ferramenta Trello para organizar e controlar as atividades a serem realizadas no projeto, de acordo com a metodologia ágil

4. Demonstrar raciocínio lógico na organização das informações

Resolver problemas inesperados, que serão propostos pelos instrutores.

5. Demonstrar visão sistêmica

Identificar e demonstrar por meio do projeto como o front-end se conecta com todo o resto do sistema

6. Seguir método de trabalho

Aplicar metodologia ágil nas atividades individuais e de projeto utilizando a ferramenta Trello

7. Trabalhar em equipe

Participar ativamente das decisões e desenvolvimento das atividades em equipe

7. SPRINT 5 – Mobile

Após um tempo de uso de sua aplicação web, Fernando percebeu que para melhorar a usabilidade de sua plataforma, é necessário desenvolver uma aplicação mobile focando no paciente e em quais informações ele precisa visualizar.

Você deverá desenvolver um aplicativo mobile integrando com a API que foi desenvolvida anteriormente (Sprint 2) e deverá conter as seguintes funcionalidades:

Perfis de usuário:

1. **Médico:** Colaboradores que atuam na área da saúde;
2. **Paciente:** Clientes da clínica;

Funcionalidades:

1. O **paciente** poderá visualizar suas próprias consultas;
2. O **médico** poderá ver os agendamentos associados a ele;

Persistência de dados

Os dados referentes ao usuário logado na aplicação mobile deverão ser salvos localmente.

Telas

Médico / Paciente

1. Login

Médico

1. Visualizar as suas consultas

Paciente

1. Visualizar as suas consultas

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade.

Interfaces para dispositivos móveis

1. Identificar as características de programação de dispositivos móveis

Identificou as diferenças entre aplicativos híbridos e nativos

Identificou as vantagens entre aplicativos híbridos e nativos

3. Preparar o ambiente necessário ao desenvolvimento do sistema para a plataforma mobile

Instalou o node em ambiente local para desenvolvimento

Instalou o React Native

7. Implementar o código respeitando as características da linguagem na plataforma mobile

Utilizou o React Native para programar o aplicativo mobile utilizando os componentes corretos de acordo com a situação problema

2. Utilizar os elementos da programação orientada a objetos em interfaces para dispositivos móveis

Utilizou polimorfismo para criação das telas

Utilizou interface para criação das telas

Utilizou abstração para criação das telas

5. Definir os elementos de entrada, processamento e saída para a codificação das funcionalidades mobile

Permitiu que somente usuários autenticados acessem o menu de navegação

Mostrou uma mensagem para o usuário após realizar login inválido

Capacidades sociais, organizativas e metodológicas

1. Demonstrar atenção aos detalhes

Realizar os exercícios propostos de acordo com as instruções

2. Demonstrar capacidade de comunicação com profissionais de diferentes áreas e especialidades

Realizar entrevista com o cliente do projeto para levantar requisitos

3. Demonstrar capacidade de organização

Utilizar a ferramenta Trello para organizar e controlar as atividades a serem realizadas no projeto, de acordo com a metodologia ágil

4. Demonstrar raciocínio lógico na organização das informações

Resolver problemas inesperados, que serão propostos pelos instrutores.

5. Demonstrar visão sistêmica

Identificar e demonstrar por meio do projeto como a aplicação mobile se conecta com todo o resto do sistema

6. Seguir método de trabalho

Aplicar metodologia ágil nas atividades individuais e de projeto utilizando a ferramenta Trello

7. Trabalhar em equipe

Participar ativamente das decisões e desenvolvimento das atividades em equipe

8. SPRINT 6 – NoSQL

Com o software finalizado, a SP Medical Group deseja aplicar funcionalidades de Data Analytics em seu sistema. A empresa deseja começar com um sistema de localização de todos os seus clientes. Deverá ser criada uma parte separada do sistema para realizar o cadastro de localizações e qual tipo de atendimento médico está sendo feito.

- Latitude e Longitude do endereço do paciente - obrigatória;
- A doença identificada pelo médico (a descrição da consulta) - opcional;
- Idade do paciente - opcional;
- Especialidade do médico - opcional;

Todos os itens destacados acima poderão ser inseridos como texto sem nenhuma validação. A empresa quer fazer um levantamento sobre quais regiões estão sendo atendidas por mais especialidades de médicos.

Poderá ser utilizado o MongoDB.

Capacidades e critérios

Os critérios estão em vermelho abaixo de cada capacidade.

Banco de dados

1. Identificar as características de banco de dados relacional e não-relacional

Identificou as diferenças entre banco de dados relacional e não-relacional

Documentou no arquivo especificado na situação problema a estrutura utilizada no projeto de big data

8. Configurar usuário e permissões de acesso ao banco de dados

Criou um usuário para realizar todas as operações na base de dados criada

Criou um usuário para realizar somente leitura na base de dados criada

Web Back-End

8. Estabelecer envio de notificações entre cliente e servidor por meio de aplicação web

Enviou uma mensagem na aplicação web após um novo cadastro de endereço / especialidade ter sido feito

9. Desenvolver API (web services) para integração de dados entre plataformas

Criou um endpoint em que o administrador cadastre novos registros

Web Front-End

3. Desenvolver interfaces web utilizando frameworks

Criou uma interface gráfica para realizar o cadastro de novos registros