

**GESTÃO DE UM CENTRO DE SAÚDE-PRIMEIRA
FASE
PROGRAMAÇÃO ORIENTADA AOS OBJETOS**

Eva Alexandra Pereira Gomes 27484

Orientador

nome do orientador

Ernesto Casanova

novembro, 2024

ÍNDICE

ÍNDICE.....	III
INTRODUÇÃO.....	1
1. OBJETIVO DO SISTEMA.....	3
2. ESTRUTURA DO PROJETO	3
3. FUNCIONAMENTO DO SISTEMA.....	5
4. FUNCIONALIDADES PRINCIPAIS QUE SE PRETENDE APLICAREM NO FUTURO.....	6
4.1. PROCESSO DE INTEGRAÇÃO	6
5. CONSIDERAÇÕES FINAIS	7
CONCLUSÕES.....	8

INTRODUÇÃO

Este relatório descreve o projeto da criação de um programa em C# para a gestão de uma Clínica de Saúde, que visa facilitar o agendamento de consultas, a prescrição de medicamentos e a realização de exames. O sistema foi desenvolvido utilizando a linguagem de programação C# e tem como objetivo melhorar a eficiência no atendimento a pacientes, bem como a gestão de médicos e salas.

1. OBJETIVO DO SISTEMA

O sistema de Gestão de Clínica de Saúde desenvolvido em C# tem como objetivo facilitar a administração de uma clínica de saúde, incluindo a gestão de staff (médicos e outros), pacientes, medicação e exames prescritos pelos médicos, consultas e salas. O sistema permite que os pacientes registem as suas informações, façam marcações de consultas e que os médicos registem diagnósticos, medicações e exames realizados. Pretende-se que numa segunda fase da entrega do trabalho que todos os dados sejam armazenados numa base de dados para garantir a persistência e a acessibilidade da informação.

2. ESTRUTURA DO PROJETO

O projeto é composto por várias classes que representam diferentes entidades e funcionalidades do sistema. As principais classes incluem:

2.1. Classe Appointment (Consulta)

Esta classe é responsável por representar uma consulta. Contém as seguintes propriedades:

- ID: Identificador único da consulta.
- Date: Data da consulta.
- Doctor: Médico responsável pela consulta.
- Patient: Paciente que estará presente na consulta.
- Room: Sala onde a consulta será realizada.
- Medications: Lista de medicamentos prescritos.
- Exams: Lista de exames solicitados.

Métodos Principais:

- Schedule(): Verifica a disponibilidade da sala e agenda a consulta.
- CalculateTotalCost(): Calcula o custo total da consulta, incluindo preço da consulta, medicamentos e exames.

2.2. Classe Patient

Esta classe herda da classe Person e é responsável por representar um paciente. Contém as seguintes propriedades:

- Name: Nome do paciente.
- Address: Morada do paciente.
- BirthDate: Data de nascimento do paciente.
- Email: Endereço de email do paciente.
- Telephone: Número de telefone do paciente.
- NIF: Número de identificação fiscal do paciente.

2.3. Classe Doctor (Médico)

Esta classe herda de Staff e é responsável por representar um médico na clínica. Contém as seguintes propriedades:

- SpecialtyID: Identificador da especialidade do médico.
- ConsultationFee: Valor cobrado pela consulta.

Métodos principais:

- PrescribeMedication(Appointment appointment, Medication medication): Prescreve um medicamento ao paciente.
- RequestExam(Appointment appointment, Exam exam): Solicita um exame para o paciente.

2.4. Classe Medication

Esta classe é responsável por representar um medicamento. Contém as seguintes propriedades:

- ID: Identificador único do medicamento.
- MedicationName: Nome do medicamento.
- Dosage: Dosagem do medicamento.
- Cost: Custo do medicamento.

2.5. Classe Exam

Esta classe é responsável por representar um exame. Contém as seguintes propriedades:

- ID: Identificador único do exame.
- ExamName: Nome do exame.
- Cost: Custo do exame.

2.6. Classe Room

Esta classe é responsável por representar uma sala de consulta. Contém as seguintes propriedades:

- ID: Identificador único da sala.
- RoomNumber: Número da sala.
- Availability: Estado de disponibilidade da sala.

Métodos principais:

- CheckAvailability(): Verifica se a sala está disponível.
- MarkUnavailable(): Marca a sala como indisponível.
- MarkAvailable(): Marca a sala como disponível.

2.7. Classe **Specialty**

Esta classe é responsável por representar uma especialidade médica. Contém as seguintes propriedades:

- ID: Identificador único da especialidade.
- SpecialtyName: Nome da especialidade.

2.8. Classe **Staff**

Esta classe é responsável por representar um funcionário da clínica e serve de classe base para médicos.

Contém as seguintes propriedades:

- Name: Nome do funcionário.
- Address: Morada do funcionário.
- BirthDate: Data de nascimento do funcionário.
- Email: Endereço de email do funcionário.
- PhoneNumber: Número de telefone do funcionário.
- NIF: Número de identificação fiscal do funcionário.
- Role: Função do funcionário na clínica.

2.9. Classe **ClinicManagement**

Esta classe é responsável por gerir as operações da clínica. Contém as seguintes funcionalidades:

- AddDoctor(): Adiciona um novo médico à clínica.
- AddPatient(): Adiciona um novo paciente à clínica.
- AddRoom(): Adiciona uma nova sala à clínica.
- ScheduleAppointment(): Agenda uma nova consulta.

2.10. Classe **DatabaseHelper**

Esta classe é responsável por interagir com a base de dados, permitindo a inserção e recuperação de dados.

Contém as seguintes funcionalidades:

- SavePerson(): Insere uma nova pessoa na base de dados.
- SaveStaff(): Insere um novo funcionário na base de dados.
- SaveSpecialty(): Insere uma nova especialidade na base de dados.
- SaveDoctor(): Insere um novo médico na base de dados.

3. FUNCIONAMENTO DO SISTEMA

O sistema entregue nesta primeira fase começa por criar uma instância da classe **ClinicManagement**. Médicos, pacientes e salas são adicionados à clínica. As consultas são agendadas verificando a disponibilidade das salas. O sistema também permite a prescrição de medicamentos e a solicitação de exames durante as consultas.

Exemplos de Uso

1. **Criação de Especialidades:** É criada uma especialidade, como "Cardiologia".
2. **Adição de Médicos:** Médicos são adicionados à clínica com suas especialidades e taxas de consulta.
3. **Adição de Pacientes:** Pacientes são registrados no sistema.
4. **Adição de Salas:** Salas são criadas e adicionadas à clínica.
5. **Agendamento de Consultas:** Consultas são agendadas, verificando a disponibilidade das salas.

4. FUNCIONALIDADES PRINCIPAIS QUE SE PRETENDE APLICAREM NO FUTURO

- **Registo de Pacientes:** Os pacientes podem criar um perfil na clínica, preenchendo informações pessoais, como nome, endereço, número de telefone, NIF, email.
- **Agendamento de Consultas:** Os pacientes podem marcar consultas com médicos específicos, escolhendo uma especialidade.
- **Registo de Diagnóstico e Prescrição:** Durante a consulta, o médico pode registrar o diagnóstico do paciente, prescrever medicamentos e solicitar exames, caso necessário.
- **Gestão de Exames e Medicação:** O sistema permite a associação de medicações e exames às consultas, sendo possível registrar detalhes de dosagem e frequência dos medicamentos prescritos.
- **Gestão de Staff e Salas:** A clínica pode gerir o pessoal (registando as suas informações pessoais) e a alocação de salas, garantindo a disponibilidade de salas para as consultas marcadas.

4.1. PROCESSO DE INTEGRAÇÃO

O processo de integração do banco de dados no sistema ocorre de maneira sequencial, passando pelas seguintes etapas:

1. **Inserção de Especialidades:** Quando uma nova especialidade médica é criada, ela é inserida na tabela Specialty. O ID gerado é utilizado para associar médicos à especialidade correspondente.
2. **Cadastro de Médicos e Pacientes:** Médicos e pacientes são registados como "pessoas" na tabela Person. Após isso, os médicos são inseridos na tabela Staff com o seu cargo, e os pacientes são registados na tabela Patient.
3. **Agendamento de Consultas:** As consultas são agendadas e armazenadas na tabela Appointment, onde são associadas a médicos, pacientes e salas específicas.
4. **Prescrição de Medicamentos e Exames:** Durante as consultas, os médicos podem prescrever medicamentos ou solicitar exames, que são registados nas tabelas Medication e Exam. Esses dados são vinculados às consultas apropriadas.

5. CONSIDERAÇÕES FINAIS

O sistema de gestão de consultas da Clínica de Saúde foi desenvolvido com o intuito de melhorar a eficiência e organização do atendimento a pacientes. Através da utilização de classes bem definidas e métodos para gerir as operações, o projeto proporciona uma base sólida para futuras expansões e melhorias. A futura integração com uma base de dados permite a persistência de informações, garantindo que dados cruciais sejam armazenados e acessíveis.

Este projeto pode ser uma excelente base para o desenvolvimento de um sistema mais complexo e abrangente, que atenda a todas as necessidades de uma clínica moderna.

CONCLUSÕES

O sistema de Gestão de Clínica de Saúde em C# é o início de uma solução para a administração eficiente de uma clínica, permitindo uma gestão completa de pacientes, staff, consultas e recursos prescritos pelos médicos. A implementação de herança facilita a reutilização de código e a futura integração com a base de dados assegura a persistência e acessibilidade das informações essenciais.

