



Aula6 (17/10/2020) Python 2 e Banco de Dados 1

Introdução a Python

□ <https://docs.google.com/presentation/d/19co-ni85kk1MoaN Dw4rTDUf7ryTMJHZWPEecMAwfRrM/edit?usp=sharing>



Powered by

Google Colaboratory

□ <https://colab.research.google.com/drive/1QrfflFuT4J-aCv04Xhujc-o0BYZCdGGG#scrollTo=SzGfxrLeMY84>



Estruturas condicionais com Python

Uma Estrutura de Condição, como o próprio nome já diz, verifica a condição dos argumentos passados e, executa um comando caso a condição seja verdadeira. Frequentemente,

👉 <https://liniribeiro.github.io/2020-10-16-estrutura-condicional-python/>

ESTRUTURAS CONDICIONAIS COM PYTHON

Instalando Python

Para unificar todos os conteúdos em apenas uma página, trouxe um passo a passo da instalação do Python, separado por sistema operacional: O primeiro passo é acessar o site do

👉 <https://liniribeiro.github.io/2020-10-14-instaling-python/>



<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/52e9cc9-bf75-469f-b9e0-3447c7b2d046/python2.pdf>

Parte 2 - a partir da pagina 42

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dff6151d-318d-478d-9726-052727d80bbd/Banco_de_Dados1.pdf

Completo

Links pra leitura e um cursinho da devmedia

Links para aula de banco de dados

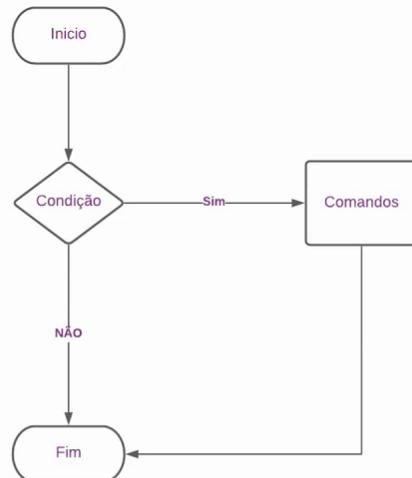
<https://en.wikipedia.org/wiki/Database>
<https://www.devmedia.com.br/ahistoria-dos-banco-de-dados/1678>
<https://www.devmedia.com.br/curso/cursomodelagem-de-bancos-de-dados-relacionais/409>

Estruturas condicionais

Exemplo de um **if simples**:

Emitir um alerta de multa quando a velocidade de um carro estiver acima do permitido (70 km/h)

```
velocidade = float(input("Informe a velocidade: "))
if(velocidade > 70):
    print("você será multado")
print("fim")
```



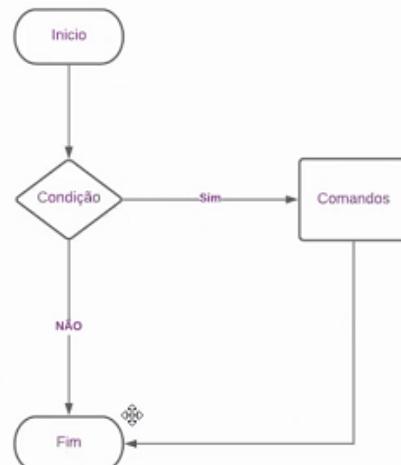
{:l

Estruturas condicionais

Com as estruturas condicionais, podemos fazer com que algumas linhas do nosso algoritmo não sejam executadas.

Sintaxe de um **if simples**:

```
se condicao:
    comando1
    comando2
```



{:l



Estruturas condicionais

Exemplos de um **if simples** em situações compostas:

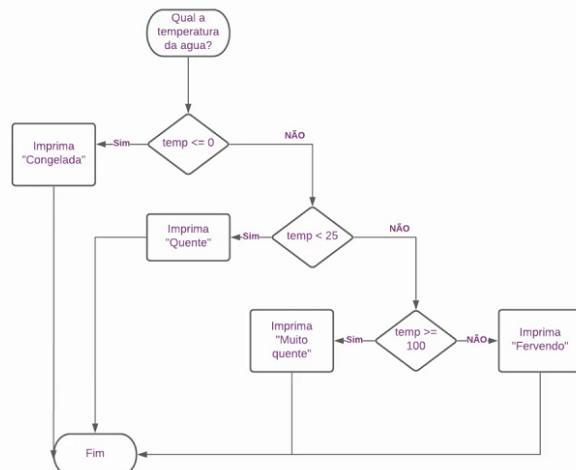
```
# Descubra a situação da massa corporal
massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do recomendado")
if (imc >= 18 and imc < 26):
    print ("Recomendado")
if (imc >=26):
    print("Acima do recomendado")
print("Fim!! PS: você é linda em qualquer imc!")
```



Estruturas condicionais

Sintaxe de um **if composto**:

```
# Sintaxe de um if composto
se condição
    comando1
    comando2
senao
    comando3
```



Estruturas condicionais

Exemplo de um if composto (que tem o else):

```
# Imprimir o maior de dois números
n1 = float(input('Informe o primeiro numero:'))
n2 = float(input('Informe o segundo número:'))
if(n1 > n2):
    print(n1)
else:
    print(n2)
```



Encadeamento de comandos condicionais

```
# Descubra a massa corporal, utilizando encadeamento de comandos
# Encadeamento de comandos seria, colocar um if dentro de um outro if

massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do peso!")
else:
    if(imc >= 18 and imc < 26):
        print ("Peso normal!")
    else:
        print("Acima do peso!")
print (imc)
print ("Fim")
```



Encadeamento de comandos condicionais, com “elif”

↳

```
# Descubra a massa corporal, utilizando encadeamento de comandos
# utilizando o elif

massa = float(input("Informe o peso:"))
altura = float(input("Informe altura:"))
imc = massa / (altura*altura)
if (imc < 18):
    print("Abaixo do peso!")
elif (imc >= 18 and imc < 26):
    print ("Peso normal!")
else:
    print("Acima do peso!")
print (imc)
print ("Fim")
```



Indentação e blocos de código

- No Python, tudo que vem depois de “:” é um bloco de código;
- É como se existisse uma hierarquia dentro do código;
- Se os blocos não forem indentados, irá ocorrer erro no console.

x = 0
while x < 20:
 print(x)
 x += 2

Tudo dentro do while é um bloco

↑
Indentação é o espaçamento em relação ao primeiro nível hierárquico

↳

```
# Exemplo de código não indentado:  
y = 20  
if y % 2 == 0:  
    print('y é par')  
else:  
    print('y é ímpar')  
  
# O console retorna:  
#     print('y é par')  
#  
# IndentationError: expected an indented block
```

↑
No Python, os Blocos precisam ser indentados



Exercícios em Python [6]

Dez pessoas assinalaram que iriam participar do encontro do Maria vai com as Devs hoje. Faça um programa que leia o número de pessoas presentes e escreva “Sucesso!” se todas as pessoas compareceram, e “Alguém faltou :/” caso contrário. Se houverem mais de dez pessoas imprima “Temos um intruso entre nós :v”.



Exercícios em Python [6]

```
marias = int(input("Digite o numero de mulheres presentes:"))
if marias == 10:
    print("Sucesso")
elif marias > 10:
    print("Temos um intruso entre nós :v")
else:
    print("Alguém faltou :/")

Digite o numero de mulheres presentes:2
Alguém faltou :/
```



Untitled3.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Salvando...

- Código + Texto

```
nota_prova = float(input("Digite a nota da prova:"))
nota_trabalho = float(input("Digite a nota do trabalho:"))
if nota_prova >= 5 and nota_trabalho > 6:
    print("aprovado")
else:
    print("reprovado")
```

... Digite a nota da prova:

Exercícios em python [8]

```
vagas = float(input("Número de vagas disponíveis:"))
ja_participaram = float(input("Número de pessoas que já participaram:"))
interessados = float(input("Número de pessoas interessadas:"))

if interessados == vagas:
    #há vagas o suficiente para todos os interessados
    print("Todos irão")
elif interessados - ja_participaram <= vagas:
    #De acordo com a regra de que, quem nunca participou tem prioridade, todos que nunca foram irão.
    print("Todos que nunca foram irão")
else:
    # Inicia que há mais interessados que nunca foram para a Python Brasil do que vagas.
    print("Haverá sorteio")

Número de vagas disponíveis:10
Número de pessoas que já participaram:2
Número de pessoas interessadas:10
Todos irão
```

{:l}

Untitled3.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

+ Código + Texto

```
[2] if nota_prova >= 7 and nota_trabalho > 0:
[2]     print("aprovado")
[2] else:
[2]     print("reprovado")
```

Digite a nota da prova:2
Digite a nota do trabalho:10
reprovado

```
vagas = int(input("Digite número de vagas disponíveis:"))
ja_participaram = int(input("Digite número de pessoas que já participaram:"))
interessadas = int(input("Digite número de pessoas interessadas:"))

if interessadas == vagas:
    print("todos irão")
elif (interessadas - ja_participaram) <= vagas:
    print("Todos que nunca foram, irão")
else:
    print("Haverá sorteio")
```

Digite número de vagas disponíveis:10
Digite número de pessoas que já participaram:5
Digite número de pessoas interessadas:20
Haverá sorteio

+ Código + Texto

```
temperatura = int(input("Temperatura:"))
umidade = int(input("Umidade:"))

if temperatura < 10 or temperatura > 25:
    print("Nada acontece, feijoada")
else:
    if temperatura < 15:
        if umidade < 40:
            print("Blusa de frio e regata")
        else:
            print("Blusa de frio, regata e guarda-chuva")
    elif temperatura < 20:
        if umidade < 40:
            print("Blusa de frio e regata")
        else:
            print("Blusa de frio, regata e guarda-chuva")
    else:
        if umidade < 40:
            print("Regata")
        else:
            print("Regata e guarda-chuva")
```

```
Temperatura:14
Umidade:50
Blusa de frio, regata e guarda-chuva
```

```
# Fala que quero começar do 1 e ir até a posição 3
for x in range(1, 3): print(x)
```

```
# Até qual posição eu quero ir
for x in range(3): print(x)
```

```
# Fala que quero começar do 1 e ir até a posição 10, incrementando de 2 em 2.
for x in range(1, 10, 2): print(x)
```

```

# Crie um loop que, para sair dele a pessoa deve digitar a palavra sair.
while True:
    palavra = input("Digite a palavra secreta para sair:")
    if palavra == "sair":
        print("Fim")
        break

    if len(palavra) <= 2:
        print("Palavra muito pequena!")
        continue
    print("Nao executou porque tem o continue")

    print("Digite \"sair\" para sair!")

reta para sair:

```

```

# enquanto a soma for menor que 20, peça um numero

soma = 0
while soma <=20:
    soma += int(input("Digite um número:"))

print(f"A soma total foi de {soma}")

```

```

Digite um número:6
Digite um número:10
Digite um número:3
Digite um número:4
A soma total foi de 23

```

Conceitos Básicos

Dado

Fato do mundo real registrado.

Informação

Fato útil que pode ser extraído direta ou indiretamente a partir de dados.

Banco de Dados

“É uma coleção de dados inter-relacionados e persistentes, representando informações sobre um domínio específico.”

Henry F. Korth



Sistemas de Arquivos Computacional

Similar ao convencional, utiliza estrutura de pastas e arquivos para armazenar as informações.

Os dados eram mantidos em um arquivo de forma sequencial.

Exemplo: um arquivo armazenava os dados de clientes e outro os dados de vendas.

Problemas

- Acesso;
- Tolerância à falhas;
- Segurança;
- Limitações.



Modelo Hierárquico

- Década de 1960;
- Avanço na área de armazenamento de dados, podiam ser relacionados;
- Estrutura em forma de árvore, possibilitando ligações entre registros;
- Cada registro podia ter vários filhos mas apenas um pai;
- Pra chegar em um determinado registro, era necessário passar por todos os anteriores.

{:l



Modelo de Rede

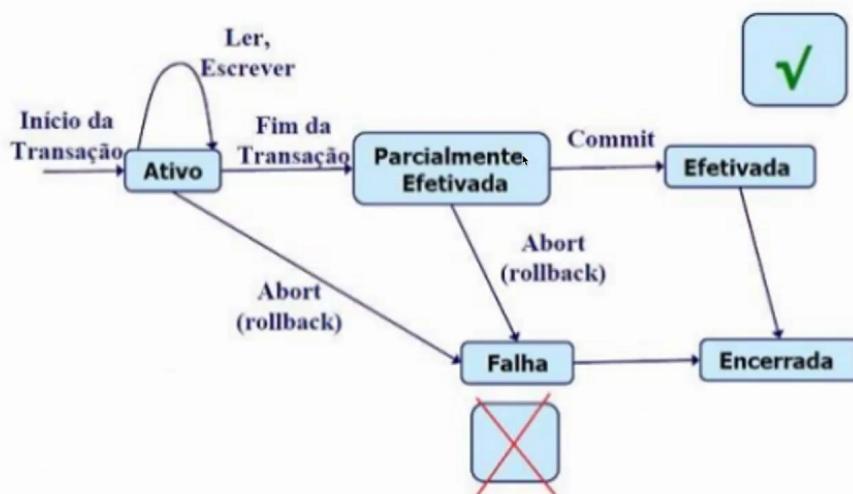
- Tentativa de melhorar o modelo hierárquico;
- Estrutura em forma de árvore que permitia um registro ter vários pais diferentes;
- Atingiu buscas mais rápidas e complexas;
- Porém o problema persistiu: dificuldade em trabalhar com volume grande de dados, além de difícil manutenção na hora de alterar as estruturas..

{:l



Modelo Relacional

- Criado com base na teoria matemática de conjuntos por Edgar Frank Codd;
- Estruturado em tabelas (relação) que podem relacionar-se entre si;
- Permite consultas que não foram previstas por quem projetou a base de dados;
- Pode ser utilizado por várias aplicações diferentes;
- Transações;
- Padrão ACID.



- **ACID**

- A: Atomicidade
 - Unidade lógica atômica (tudo ou nada)
- C: Consistência (Preservação)
 - Ao final de uma transação o banco continua consistente
- I: Isolamento
 - A execução de uma transação não deve sofrer interferência de outras transações concorrentes
- D: Durabilidade (Persistência)
 - Após o ponto de confirmação, as alterações devem persistir no banco de dado



Modelo Não Relacional

- Final dos anos 2000;
- Suporta grandes volumes de dados;
- Estrutura não definida, não limita os campos;
- Escalabilidade;
- Tipos de armazenamento mais utilizados:
 - Banco de documentos (JSON);
 - Chave-Valor.



```
[{
  "_id": ObjectId("5e6261a1df9bcf90c29726d4"),
  "nome": "Henrique Marques Fernandes",
  "idade": 29
},
{
  "_id": ObjectId("5e6261a1df9bcf90c29726d3"),
  "nome": "Terry Crews",
  "idade": 65,
  "pais": "USA"
}]
```

JSON

Chave-valor

- 1 { "id": 1, "nome": "Terry Crews", "idade": 65, "pais": "USA" }
- 2 Henrique Marques Fernandes



A screenshot of the Postman application. On the left, the sidebar shows a collection named 'v1/' containing various API requests like 'POST v1/internal', 'POST ino-agreements', 'POST ino-bankslip', etc. In the center, a specific POST request for 'ino-agreements' is selected. The request details show the URL as `((no_host))/ino/agreements/v1`. The 'Body' tab is active, showing a JSON response with one item. The response body is:

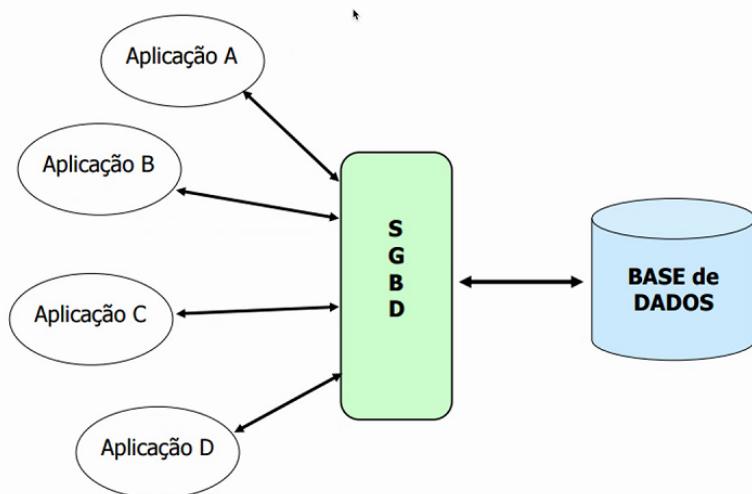
```
1 [
2   {
3     "id": "af61604c-a8d7-4688-81a9-a20d5f2a1cd4",
4     "createdAt": "2020-10-13",
5     "total": 249.0,
6     "ioTotal": null,
7     "ceTotal": null,
8     "installmentCount": 3,
9     "currentInstallment": {
10       "dueDate": "2020-10-10",
11       "installment": 1,
12       "value": 83.0,
13       "status": {
14         "key": "open",
15         "name": "Em aberto"
16       },
17       "paymentId": null,
18       "payday": null,
19       "paymentAvailable": true
20     }
21 ]
```

SGBD

- Sistema de Gerenciamento de Banco de Dados



A tecnologia de Bases de Dados



SGBD - Relacional

ORACLE®



IBM DB2®



Principais objetos de um SGBD

- Esquemas;
- Tabelas;
- Visões;
- Índices;
- Usuários e Papeis;
- Procedimentos armazenados (Stored Procedures, Functions e Triggers).



Estrutura

Tabelas: entidade ou relação

É um conjunto de dados com um número determinado de colunas e um número infinito de linhas.

Colunas: atributos

As colunas são os campos, ou seja, os nomes que representam os dados armazenados.

Linhas: tuplas ou registros

É o registro que contém os dados de cada campo da tabela.



Projeto de Banco de Dados

- Modelo Conceitual
- Modelo Lógico
- Modelo Físico



Modelo Conceitual

- Diagrama da visão geral do negócio;
- Independente da tecnologia usada para o banco de dados;
- Facilidade de entendimento entre usuários e desenvolvedores;
- Possui as entidades e relacionamentos.

{:l

Modelo Lógico

- Diagrama mais completo;
- Possui entidades associativas no lugar de relacionamentos;
- Define as chaves primárias e estrangeiras;
- Normalização até a 3^a forma normal;
- Segue o padrão de nomenclatura a ser utilizado no banco.



Modelo Físico

- Depende da tecnologia escolhida (SGBD);
- Possui tabelas e colunas físicas;
- Utiliza SQL, a linguagem padrão para manipular a estrutura de dados em bancos relacionais.



CONCEITUAL

Definição

- Trabalhamos com requisitos e regras de negócio do domínio do problema;
- Não nos preocupamos com aspectos físicos do banco ou tecnologia;
- O domínio deve ser bem delimitado pois estabelece a fronteira do que deve estar no banco de dados e o que não deve.



Aplicação em um Exemplo

Problema

Atualmente os alunos são cadastrados em fichas de papel guardadas em pastas. Isso acarreta em dificuldade no resgate de informações e fragilidade das mesmas. O ideal seria um banco de dados organizado onde as fichas dos alunos possam ser identificadas com facilidade através do número de matrícula, assim como outras informações, como professores, disciplinas e notas dos alunos, podendo gerar relatórios diversos como o boletim, que hoje é feito no aplicativo Word.

Objetivo

Armazenar os dados dos alunos de forma automatizada, possibilitando consultas ágeis que retornem as informações de forma rápida e prática.



Requisitos

O banco de dados deve refletir as seguintes informações

- Cada aluno deve ter matricula, nome, endereço, telefone e email;
- Cada disciplina deve ter código, nome e carga horária;
- Cada matricula deve ter a matrícula do aluno, código da disciplina e data de matricula;
- Cada aluno deve se matricular em pelo menos uma disciplina e no máximo em 3 disciplinas.



Identificar Entidades

- Cursos;
- Aluno;
- Professores;
- Disciplinas;
- Turmas.



Definição dos atributos

- Cursos (codigo, nome, duracao);
- Alunos (matricula, nome, endereco, cep, telefone, email);
- Professores (matricula, nome, endereco, cep, telefone, email);
- Disciplinas (codigo, descricao, carga_horaria);
- Turmas (codigo, descricao).



Esboço

Diagrama de conjuntos

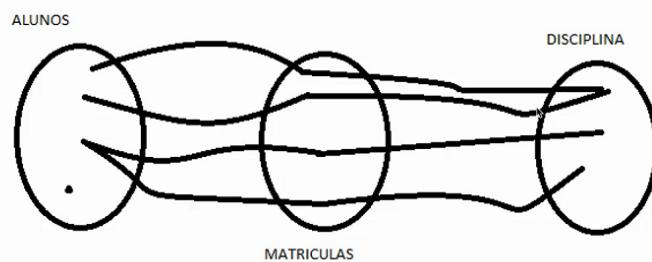


Diagrama de Entidade e Relacionamento



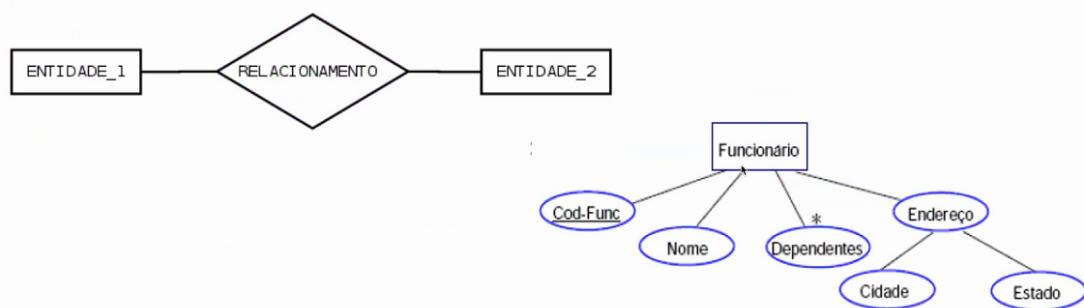
ENTIDADE - RETANGULOS ALUNO E DISCIPLINA

LINHA - RELACIONAMENTO

RESULTADO - DESCREVE A LIGACAO DAS ENTIDADES

NUMEROS - DEFINICAO DO RELACIONAMENTO

Notação Peter Chan



Cardinalidade

1:1 – Um pra Um



1:N – Um pra Muitos
N:1 – Muitos pra Um



N:N – Muitos pra Muitos



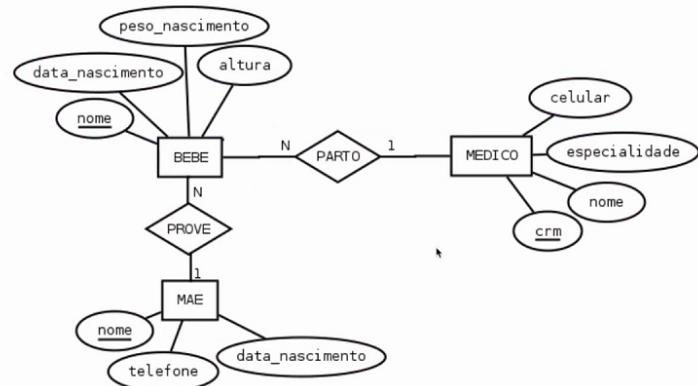
{:l

Exercício 01 - DER

Um berçário deseja informatizar suas operações. Quando um bebê nasce, algumas informações são armazenadas sobre ele, tais como: nome, data de nascimento, peso, altura, a mãe deste bebê e o médico que fez seu parto. Quanto às mães, o berçário também deseja manter um controle, guardando informações como: nome, endereço, telefone e data de nascimento. E dos médicos, é importante saber: CRM, nome, telefone e especialidade. Desenho o DER com base na notação de Peter Chan.

{:l

Gabarito 01 - DER



8

{:l

Exercício 02 - DER

Um mercado está começando a ter prejuízo nas finanças e não sabe onde está o problema. O dono precisa salvar de forma organizada os produtos que ele tem no estoque, os produtos vendidos para seus clientes, e as perdas de produtos.

8

{:l

Exercício 02 - DER

Em uma clínica deseja-se manter os dados das consultas referentes aos pacientes e médicos envolvidos. Os pacientes têm nome, cpf, data de nascimento, endereço (o qual é formado por logradouro, número, bairro, cidade e uf) e diversos telefones. Os médicos que atuam na clínica possuem nome, especialidade, CRM e diversos telefones. As consultas são agendadas para determinada data e hora. Cada médico estipula o seu valor de consulta próprio. Um paciente pode consultar-se com mais de um médico, desde que em consultas diferentes. Em cada consulta o medico receita medicamentos.



{:l