**DCC060 – Trabalho Prático**

**GRUPO: Lohan Rodrigues N. Ferreira, Lucas Carvalho**

**1. INTRODUÇÃO**   
Desenvolvimento conceitual e lógico de um sistema (banco de dados) para clínicas nutricionistas onde podem ser encontrados todos os clientes sendo atendidos, nutricionistas trabalhando em clinicas, dietas e tratamentos sendo realizados assim como receitas e produtos para auxilio dessas dietas e seus fornecedores.  
  
**2. REQUISITOS DO PROJETO**   
O sistema deve ser capaz de armazenar informações sobre as várias clínicas participantes (guardando código, nome da clínica e endereço) além de todas as nutricionistas que trabalham em cada uma delas (contendo CPF, nome e sua especialização) e seus dependentes(CPF e nome).

O sistema deve permitir o cadastro de pacientes (CPF, nome, endereço , telefones e planos de saúde que possui) e permitir que os mesmos sejam capazes de marcar consultas ( armazenando a data) com as nutricionistas, e se desejar iniciar um tratamento .

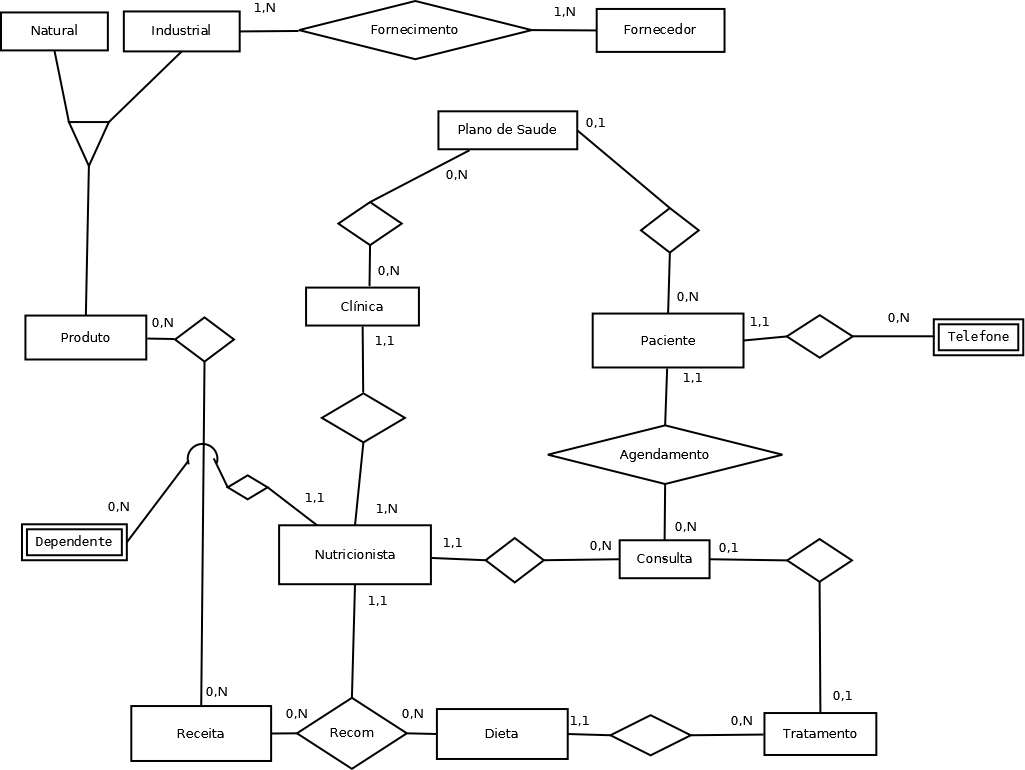
O sistema deve permitir o cadastro de dietas que serão seguidas pelos pacientes em tratamento bem como permitir que nutricionistas recomendem receitas adicionais para uma dieta em específico.

O sistema deve possuir uma lista de planos de saúde aceitos pelas diversas clínicas permitindo descontos ou coberturas nos tratamentos.

As receitas recomendadas por nutricionistas devem ter seus produtos ( ou ingredientes) armazenados bem como possíveis fornecedores no caso de produtos industrializados.

|  |
| --- |
| **Regras de Negócio** |
|  |

**3. MODELAGEM RELACIONAL**



**3.2 Modelagem lógica**

Paciente(CPF,Nome,Endereço,DataNascimento,Cod\_Plan)

Cod\_Plan referencia plano de saude

Telefone(CPF,Numero)

Cada CPF deve possuir no máximo 3 números cadastrados

CPF referencia Paciente

Clínica(Cod,Nome,Endereço)

Plano de Saude (Cod,Nome,ValorDesconto)

PlanosAceitos(Cod\_Plan,Cod\_Clin)

Cod\_Plan referencia plano de saúde

Cod\_Clin referencia clinica

Nutricionista(CPF,Cod,Nome,Especializacao)

Cod referencia Clínica

Dependente(CpfN,CPF,nome)

CpfN referencia Nutricionista

Consulta(CpfP,CpfN,Data)

CpfP referencia Paciente

CpfN referencia Nutricionista

Dieta(Cod,Definições)

Tratamento(,CpfP,CpfN ,Cod\_Dieta,Instrucoes,Data\_Inicio)

Cod\_Dieta referencia Dieta, e este campo não pode ser Null.

CpfP,CpfN referenciam Consulta

Receita(Cod,CpfN,Preparo,Nome)

CpfN referencia Nutricionista

Recomendação(Cod\_Dieta,CpfN,Cod\_R)

Cod\_Dieta referência Dieta

CpfN referencia Nutricionista

Cod\_R referencia receita

Cada dieta possui no máximo 4 receitas recomendadas.

Produto(Cod, Descrição,Faixa de Preço)

ReceitaProduto(Cod\_P,Cod\_R)

Cod\_P referencia Produto

Cod\_R referencia Receita

Natural(Cod,Nome)

Industrial(Cod,PrazoVencimento,Nome)

Fornecedor(CNPJ,Nome\_Emp)

Fornecimento(Cod\_P,CNPJ)

Cod\_P referencia Produto

CNPJ referencia Fornecedor

**4 CONSULTAS EM ÁLGEBRA RELACIONAL**(devem cobrir todas as sintaxes e casos-exemplo vistos em sala de aula; necessariamente incluir junção, diferença e divisão; explicitar a pergunta que define a consulta e a sequência de operações em Álgebra Relacional necessária para conseguir a resposta à pergunta) 

|  |
| --- |
| **Consulta 1: “Quais nutricionistas trabalham na clínica ‘X’?”** |
| ΠNome(Nutricionista ‘Join’Cod (σNome=’X’Clínica)) |
|  |

⚫ ⚫ ⚫

|  |
| --- |
| **Consulta 2: “Planos de Saúde aceito por todas as clínicas”** |
| Π Nome(PlanodeSaude ‘Join’cod=cod\_plan (PlanosAceitos / (πcod\_clin PlanosAceitos) |

|  |
| --- |
| **Consulta 3: “Planos de Saúde aceitos pela clinica ‘X’?”** |
| Π Nome(PlanodeSaude ‘Join’cod=cod\_plan (PlanosAceitos ‘Join’ cod\_clin = cod(ΠCod(σnome=’x’Clínica)))) |

|  |
| --- |
| **Consulta 4: “Nome dos pacientes que consultaram com a nutricionista ‘A’?”** |
| Π Nome (Paciente ‘Join’Cpf(Π CpfP(Consulta ‘Join’CpfN=Cpf (ΠCpf(σnome=’A’Nutricionista))))) |

|  |
| --- |
| **Consulta 5: “Planos de Saude aceitos somente pela Clinica ‘X’?”** |
| ρ(Planos,((Π Cod\_plan (PlanosAceitos ‘Join’ cod\_clin = cod(ΠCod(σnome=’x’Clínica)))) - Π Cod\_plan(PlanosAceitos ‘Join’cod\_plan = cod\_plan^cod\_clin <> cod\_clin(PlanosAceitos ‘Join’ cod\_clin = cod(ΠCod(σnome=’x’Clínica)))))  Π Nome(PlanodeSaude ‘Join’cod=cod\_plan Planos) |

|  |
| --- |
| **Consulta 6: “Nome dos pacientes em tratamento com a nutricionista ‘A’?”** |
| Π Nome (Paciente ‘Join’Cpf(Π CpfP(Tratamento ‘Join’CpfN=Cpf (ΠCpf(σnome=’A’Nutricionista))))) |

|  |
| --- |
| **Consulta 7: “Receitas recomendadas pela nutricionista ‘A’?”** |
| Π Nome,Preparo (Receita ‘Join’Cod(Π Cod\_R(Recomendação ‘Join’CpfN=Cpf (ΠCpf(σnome=’A’Nutricionista))))) |
|  |

|  |
| --- |
| **Consulta 8: “Produtos usados em exatamente uma receita?”** |
| (Π Cod\_P Produto) – (Π Cod\_P(ProdutoReceita ‘Join’ Cod\_R<>Cod\_R ^ Cod\_P = Cod\_P ProdutoReceita)) |

**5 MODELO FÍSICO**

Todos os itens abaixo devem ser descritos no relatório e inseridos em um arquivo create\_schema.sql e enviado em conjunto com o relatório.

**5.1 Tabelas**

(DDL de criação de tabelas para o esquema relacional, consistente com a modelagem lógica com todas as restrições de integridade presentes)

**5.2 Índices**(definição e criação de índices secundários úteis, utilizados nas consultas) 

**5.3 Gatilhos e Funções**(pelo menos duas funções acionadas por meio de gatilhos)   
  
**6. CARGA E VERIFICAÇÃO**

Todos os itens abaixo devem ser descritos no relatório e inseridos em um arquivo dump.sql e enviado em conjunto com o relatório.

**7.1 Carga de Dados**(INSERTs para carga dos dados) 

**7.2 Verificação**(INSERTs, UPDATEs e DELETEs que demonstram o correto comportamento dos controles de integridade criados)

**7. CONSULTAS EM SQL**

(devem cobrir todas as sintaxes e casos-exemplo vistos em sala de aula; necessariamente incluir junção, de quatro formas distintas, junção externa, diferença e divisão; explicitar a pergunta que define a consulta e o comando em SQL necessário para conseguir a resposta à pergunta)

|  |
| --- |
| **Consulta 1: “pergunta a ser respondida”** |
| (resposta em SQL) |

⚫ ⚫ ⚫

|  |
| --- |
| **Consulta n: “pergunta a ser respondida”** |
| (resposta em SQL) |

**8. TELAS DA APLICAÇÃO**

(mostrar algumas telas da aplicação e relacioná-las com os requisitos de negócio da seção 2. O sistema não precisará ser completo. Basta implementar algumas telas de inserção e resultados de consultas que sejam interessantes para a aplicação que está desenvolvendo, de forma a demonstrar que é capaz de criar uma aplicação que se comunique com um SGBD relacional. O código do sistema deve ser enviado em conjunto com o relatório.)