

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS | FATEC**

**ASSISTÊNCIA SOCIAL**

**FELIPE HENRIQUE DE OLIVEIRA CÂNDIDO**

**MARIELI BURI DA SILVA**

**OURINHOS/2024**

## Scripts das Estruturas de Alterações

### 1- Contar as pessoas

```
UPDATE num_pessoas np
SET np.`num_pessoasT` = (SELECT COUNT(*) FROM pessoasn),
    np.`data_inclusao` = CURDATE();
INSERT INTO num_pessoas (`num_pessoasT`, `data_inclusao`) VALUES ((SELECT
COUNT(*) FROM pessoasn), CURDATE());
```

### 2- Contar as famílias

```
UPDATE num_familias nf
SET nf.`qnt_familias` = (SELECT COUNT(*) FROM familiasn),
    nf.`data_inclusao` = CURDATE();
INSERT INTO num_familias (`qnt_familias`, `data_inclusao`) VALUES ((SELECT COUNT(*)
FROM familiasn), CURDATE());
```

### 5- Alfabetizações

```
-- Primeiro, calcule os valores
SET @num_alfabetizado = (
    SELECT
        SUM(CASE WHEN cod_sabe_ler_escrever_memb = 1 THEN 1 ELSE 0 END)
    FROM alfabetizacoes
);

SET @num_analfabetizado = (
    SELECT
        SUM(CASE WHEN cod_sabe_ler_escrever_memb = 2 THEN 1 ELSE 0 END)
    FROM alfabetizacoes
);
INSERT INTO cont_alfabetizados (num_alfabetizado, num_analfabetizado,
data_inclusaoA)
VALUES (@num_alfabetizado, @num_analfabetizado, CURDATE())
ON DUPLICATE KEY UPDATE
    num_alfabetizado = VALUES(num_alfabetizado),
    num_analfabetizado = VALUES(num_analfabetizado),
    data_inclusaoA = VALUES(data_inclusaoA);
```

### 8- Script para contagem de sexo

```
UPDATE sexo s
JOIN (
    SELECT cod_sexo_pessoa, COUNT(*) AS quantidade
    FROM pessoasn
    GROUP BY cod_sexo_pessoa
) p ON s.cod_sexo = p.cod_sexo_pessoa
```

```
SET s.qntsexo = p.quantidade,  
    s.data_inclusao = CURDATE();
```

### 9. Pessoas com Deficiência

```
CREATE TABLE faixas_deficiencia (  
    faixa_id INT PRIMARY KEY AUTO_INCREMENT,  
    faixa_nome VARCHAR(255)  
);
```

```
INSERT INTO faixas_deficiencia (faixa_nome)  
VALUES  
    ('Cegueira'),  
    ('Baixa Visão'),  
    ('Surdez Profunda'),  
    ('Surdez Leve'),  
    ('Deficiência Física'),  
    ('Deficiência Mental'),  
    ('Síndrome de Down'),  
    ('Transtorno Mental');
```

```
CREATE TABLE contagem_deficiencia (  
    faixa_id INT,  
    qtd_pessoas INT,  
    data_registro DATE,  
    FOREIGN KEY (faixa_id) REFERENCES faixas_deficiencia(faixa_id)  
);
```

```
ALTER TABLE condicoes_fisicas  
    CHANGE `ind_def_cegueira_memb` ind_def_cegueira_memb VARCHAR(255), --  
Cegueira  
    CHANGE `ind_def_baixa_visao_memb` ind_def_baixa_visao_memb  
VARCHAR(255), -- Baixa Visão  
    CHANGE `ind_def_surdez_profunda_memb` ind_def_surdez_profunda_memb  
VARCHAR(255), -- Surdez Profunda  
    CHANGE `ind_def_surdez_leve_memb` ind_def_surdez_leve_memb  
VARCHAR(255), -- Surdez Leve  
    CHANGE `ind_def_fisica_memb` ind_def_fisica_memb VARCHAR(255), --  
Deficiência Física  
    CHANGE `ind_def_mental_memb` ind_def_mental_memb VARCHAR(255), --  
Deficiência Mental  
    CHANGE `ind_def_sindrome_down_memb` ind_def_sindrome_down_memb  
VARCHAR(255), -- Síndrome de Down  
    CHANGE `ind_def_transtorno_mental_memb` ind_def_transtorno_mental_memb  
VARCHAR(255); -- Transtorno Mental
```

```
INSERT INTO contagem_deficiencia (faixa_id, qtd_pessoas, data_registro)  
SELECT  
    1 AS faixa_id, -- Cegueira
```

```

COUNT(*) AS qtd_pessoas,
CURDATE() AS data_registro
FROM
condicoes_fisicas
WHERE ind_def_cegueira_memb = 1
UNION ALL
SELECT
2 AS faixa_id, -- Baixa Visão
COUNT(*) AS qtd_pessoas,
CURDATE()
FROM
condicoes_fisicas
WHERE ind_def_baixa_visao_memb = 1
UNION ALL
SELECT
3 AS faixa_id, -- Surdez Profunda
COUNT(*) AS qtd_pessoas,
CURDATE()
FROM
condicoes_fisicas
WHERE ind_def_surdez_profunda_memb = 1
UNION ALL
SELECT
4 AS faixa_id, -- Surdez Leve
COUNT(*) AS qtd_pessoas,
CURDATE()
FROM
condicoes_fisicas
WHERE ind_def_surdez_leve_memb = 1
UNION ALL
SELECT
5 AS faixa_id, -- Deficiência Física
COUNT(*) AS qtd_pessoas,
CURDATE()
FROM
condicoes_fisicas
WHERE ind_def_fisica_memb = 1
UNION ALL
SELECT
6 AS faixa_id, -- Deficiência Mental
COUNT(*) AS qtd_pessoas,
CURDATE()
FROM
condicoes_fisicas
WHERE ind_def_mental_memb = 1
UNION ALL
SELECT

```

```

    7 AS faixa_id, -- Síndrome de Down
    COUNT(*) AS qtd_pessoas,
    CURDATE()
FROM
    condicoes_fisicas
WHERE ind_def_sindrome_down_memb = 1
UNION ALL
SELECT
    8 AS faixa_id, -- Transtorno Mental
    COUNT(*) AS qtd_pessoas,
    CURDATE()
FROM
    condicoes_fisicas
WHERE ind_def_transtorno_mental_memb = 1;

```

## 10. Grau de Instrução

```

CREATE TABLE faixas_escolaridade (
    faixa_id INT PRIMARY KEY AUTO_INCREMENT,
    descricao VARCHAR(255),
    cod_curso INT
);

```

```

INSERT INTO faixas_escolaridade (descricao, cod_curso)
VALUES
('Creche', 1),
('Pre-escola (exceto CA)', 2),
('Classe de Alfabetizacao - CA', 3),
('Ensino Fundamental 1ª a 4ª series, Elementar (Primario)', 4),
('Ensino Fundamental 5ª a 8ª series, Medio 1º ciclo (Ginasial)', 5),
('Ensino Fundamental (duracao 9 anos)', 6),
('Ensino Fundamental Especial', 7),
('Ensino Medio, 2º grau, Medio 2º ciclo', 8),
('Ensino Medio Especial', 9),
('Ensino Fundamental EJA - series iniciais', 10),
('Ensino Fundamental EJA - series finais', 11),
('Ensino Medio EJA', 12),
('Superior, Aperfeicoamento, Especializacao, Mestrado, Doutorado', 13),
('Alfabetizacao para Adultos (Mobral, etc.)', 14),
('Nenhum', 15);

```

```

CREATE TABLE contagem_escolaridade (
    faixa_id INT,
    qtd_pessoas INT,
    data_registro DATE,

```

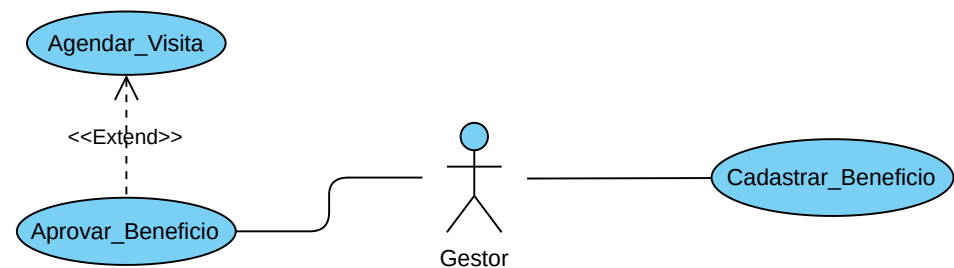
```
FOREIGN KEY (faixa_id) REFERENCES faixas_escolaridade(faixa_id)
);
```

```
INSERT INTO contagem_escolaridade (faixa_id, qtd_pessoas, data_registro)
SELECT
    f.faixa_id,
    COUNT(alf.cod_curso_frequentou_pessoa_memb) AS qtd_pessoas,
    CURDATE() AS data_registro
FROM
    alfabetizacoes AS alf
JOIN
    faixas_escolaridade AS f
ON
    alf.cod_curso_frequentou_pessoa_memb = f.cod_curso
GROUP BY
    f.faixa_id;
```

## **12 - Raça**

```
INSERT INTO raca (branco, preto, amarelo, pardo, indigena, dta_inclusao)
SELECT
    SUM(CASE WHEN p.cod_raca_cor_pessoa = 1 THEN 1 ELSE 0 END) AS branco,
    SUM(CASE WHEN p.cod_raca_cor_pessoa = 2 THEN 1 ELSE 0 END) AS preto,
    SUM(CASE WHEN p.cod_raca_cor_pessoa = 3 THEN 1 ELSE 0 END) AS amarelo,
    SUM(CASE WHEN p.cod_raca_cor_pessoa = 4 THEN 1 ELSE 0 END) AS pardo,
    SUM(CASE WHEN p.cod_raca_cor_pessoa = 5 THEN 1 ELSE 0 END) AS indigena,
    CURDATE() AS dta_inclusao
FROM
    pessoasn p;
```

**Gestor**



Agendar\_visita

- 1- Ver perfil da pessoa
- 2- Solicitar visita
- 3- Agendar visita ao sistema
- 4- Comunicar pessoa da data e horario
- 5- Caso confirmado: confirmar visita ao sistema se não vai para alternativo.

Alternativo

- 5.1- Volta para passo 3

Cadastrar\_Benefício

- 1- Fazer login ao sistema
- 2- Clicar em cadastrar novo benefício
- 2.1 - Nomear benefício
- 3 - Salvar benefício
- 4 - Emite mensagem de sucesso

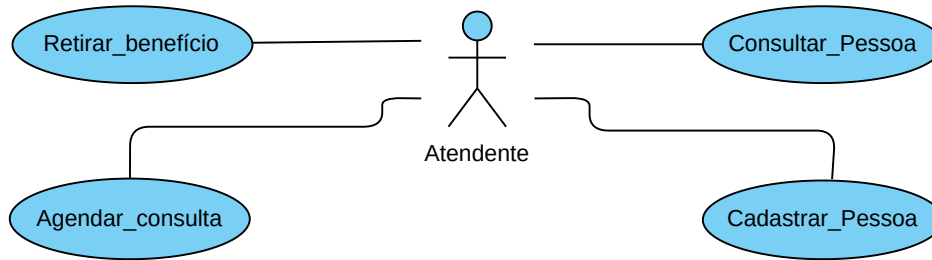
Aprovar\_Beneficio

- 1- Fazer Login
- 2 - Ver benefícios em espera
- 2.1 - Avaliar benefício solicitado
- 2.2 - Se reprovado vai p/ alternativo
- 3 - Aprovar benefício
- 4 - Emite mensagem de sucesso

Alternativo

- 2.3- Msg. "Benefício não aprovado";

# Atendente



## Consultar\_Pessoa

- 1- Insere CPF
- 1.1 Verifica se cadastrado
- 1.2 Se não cadastrado vai p/ Alternativo
- 1- Exibe dados
- Alternativo
- 1.3 Pessoa Não possui cadastro
- 1.4 Msg. não cadastrado

## Cadastrar\_Pessoa

- 1- Insere CPF
- 1.1 Verifica se cadastrado
- 1.2 Se cadastrado vai p/ alternativo
- 2- Insere dados
- 3- Salvar dados
- 4- Emite msg.
- Alternativo
- 1.3 Pessoa já tem cadastro
- 1.4 Msg. já cadastrado

## Retirar\_benefício

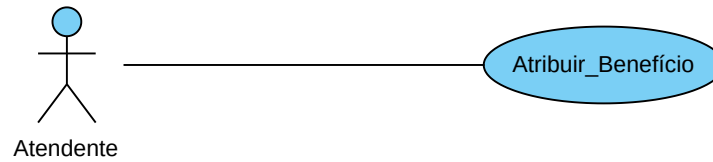
- 1- Insere CPF
- 1.1 Verifica se aprovado
- 1.2 Se não aprovado vai p/ Alternativo
- 2- Registra entrega do benefício
- 3- Atualizar estoque
- Alternativo
- 1.3 Benefício não aprovado
- 1.4 Msg. Beneficio não aprovado

## Agendar\_Consulta

- 1- Insere CPF
- 1.1 Verifica se cadastrado
- 1.2 Se não cadastrado vai p/ alternativo
- 2- Insere dados
- 2- Agenda consulta com psicóloga
- 3- Emite msg.
- Alternativo
- 1.3 Pessoa não tem cadastro
- 1.4 Msg. pessoa não cadastrada



# Atendente



## Atribuir\_Beneficio

1. Atendente solicita atribuir benefício;
2. Sistema solicita CPF;
3. Atendente informa CPF;
4. Sistema verifica se a pessoa já está cadastrada;
5. Sistema obtém benefícios disponíveis;
6. Sistema exibe os benefícios para a requisição da pessoa;
7. Atendente informa o benefício desejado;
8. O sistema obtém dados do benefício;
9. o sistema altera status ('entregue');
10. O sistema solicita confirmação;
11. O Atendente confirma a solicitação;
12. O sistema atribui o benefício a pessoa;
13. o sistema emite a mensagem 'benefício solicitado'.

### Alternativos

Caso 1: A pessoa não possui cadastro.

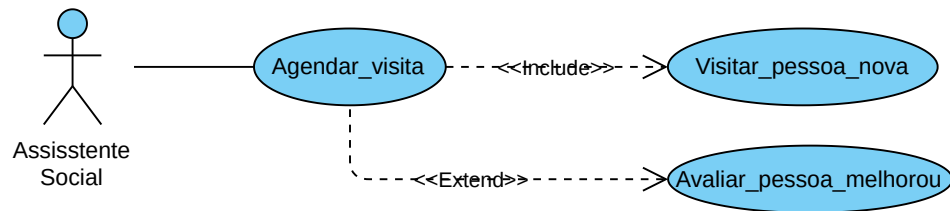
4.1. O sistema inicia o use case 'Cadastrar\_Pessoa'

4.2. O sistema volta para o passo 3.

Caso 2: Benefício não disponível.

8.1. Emite msg: Benefício não disponível no momento.

# Assistente Social



## Avaliar\_pessoa\_melhorou

- 1- Agendar visita a casa de beneficiado
- 2- Visitar casa
- 3- Avaliar caso
- 4- Com base na visita, definir se o benefício continua ou não
- 5- Adicionar ao sistema a aprovação ou reprovação
- 6- Msg. "Enviado com sucesso"

## Visitar\_pessoa\_nova

- 1- Anotar dados importantes
- 2- Avaliar caso
- 3- Criar laudo
- 4- Enviar laudo ao sistema
- 5- Msg. "Laudo anexado com sucesso"

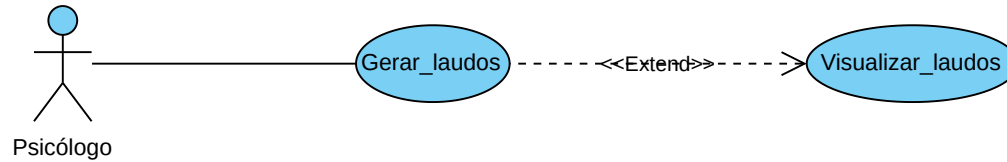
## Agendar\_visita

- 1- Ver perfil da pessoa
- 2- Solicitar visita
- 3- Agendar visita ao sistema
- 4- Comunicar pessoa da data e horario
- 5- Caso confirmado: confirmar visita ao sistema se não vai para alternativo.

### Alternativo

- 5.1- Volta para passo 3

# Psicólogo



## Gerar\_laudos

- 1- Insere CPF
  - 1.1- Verifica se cadastrado
  - 1.2- Se não cadastrado vai p/ alternativo
- 2- Anexa laudo
- 3- Envia laudo
- 4- Msg. Enviado com sucesso
- Alternativo
  - 1.1 Pessoa não possui cadastro
  - 1.2 Msg. Não possui cadastro

## Visualizar\_laudos

- 1- Insere CPF
  - 1.1- Verifica se cadastrado
  - 1.2- Se não cadastrado vai p/ alternativo
- 2- Abre laudo da pessoa
- 3- Visualiza laudo
- Alternativo
  - 1.1 Pessoa não possui cadastro
  - 1.2 Msg. Não possui cadastro

Diagrama de Sequência

Marcel Burt and Felipe Henrique

Descrição do caso de uso: Cadastrar\_Beneficio

Curso Normal

1. Gestor faz login ao sistema;
2. Solicitar cadastrar novo benefício;
3. Clicar em cadastrar novo benefício;
4. Nomear benefício;
5. Salvar benefício;
6. Emite msg: "Benefício cadastrado com Sucesso".

Cursos Alternativos

- Caso 1: Benefício já cadastrado.  
2.1.1. Emite msg: "Benefício já cadastrado".

Descrição do caso de uso: Aprovar\_Beneficio

Curso Normal

1. Gestor faz login ao sistema;
2. Ver benefícios em espera;
3. Avaliar benefício solicitado;
4. Informa ao sistema se aprovado ou reprovado;
- 4 - Emite mensagem de sucesso;

Descrição do caso de uso: Agendar\_Visita

Curso Normal

1. Gestor faz login ao sistema;
2. Ver perfil da pessoa;
3. Solicitar visita;
- 4- Agendar visita ao sistema
- 4- Comunicar a pessoa da data e horário

Descrição do caso de uso: Consultar\_Pessoa

Curso Normal

1. Atendente solicita consulta de pessoa;
  2. Sistema solicita CPF da pessoa;
  3. Atendente insere CPF da pessoa;
  4. Sistema verifica se cadastrado;
  5. Sistema obtém informações disponíveis;
  6. Sistema exibe informações da pessoa cadastrada;
- Cursos Alternativos  
Caso 1: A pessoa não possui cadastro.  
4.1. O sistema inicia o use case 'Cadastrar\_Pessoa'  
4.2. O sistema volta para o passo 5.

Descrição do caso de uso: Gerar\_Laudo

Curso Normal

1. Psicólogo insere o CPF da pessoa;
2. O sistema verifica se ela está cadastrada;
- 2- Psicólogo anexa laudo;
- 3- Envia laudo ao sistema;
- 4- Msg. Enviado com sucesso;

Cursos Alternativos

- Caso 1: Sistema verifica se ela está cadastrada;  
1.1 Pessoa não possui cadastro  
1.2 Msg. Não possui cadastro

Descrição do caso de uso: Agendar\_Consulta

Curso Normal

1. Atendente insere CPF da pessoa no sistema
2. O sistema verifica se ela está cadastrada;
- 3- Insere dados da pessoa no sistema
- 2- Agenda consulta com psicóloga
- 3- Emite msg. com os dados da consulta

Cursos Alternativos

- Caso 1: Sistema verifica se ela está cadastrada;  
1.1 Pessoa não possui cadastro  
1.2 Msg. Não possui cadastro

Descrição do caso de uso: Cadastrar\_Pessoa

Curso Normal

1. O atendente solicita cadastro;
2. O sistema solicita dados da pessoa;
3. O atendente fornece os dados;
4. O sistema verifica se esta pessoa já está cadastrada;
5. O sistema faz o cadastro
6. O sistema emite a msg 'Cadastro efetuado com sucesso'.

Descrição do caso de uso: Retirar\_beneficio

Curso Normal

1. Atendente solicita retirada de benefício;
2. Sistema solicita CPF;
3. O atendente informa o CPF;
4. Sistema verifica se cadastrado;
5. O atendente consulta a situação do benefício;
6. Sistema obtém dados do benefício;
7. Atendente declara retirada do benefício;
8. Sistema registra retirada do benefício;
9. Sistema atualiza status do benefício;

Cursos Alternativos

- Caso 1: A pessoa não possui cadastro.  
2.1. O sistema inicia o use case 'Cadastrar\_Pessoa'  
2.2. O sistema volta para o passo 3.  
Caso 2: Benefício não aprovado.  
4.1. Emite msg: Benefício não aprovado.

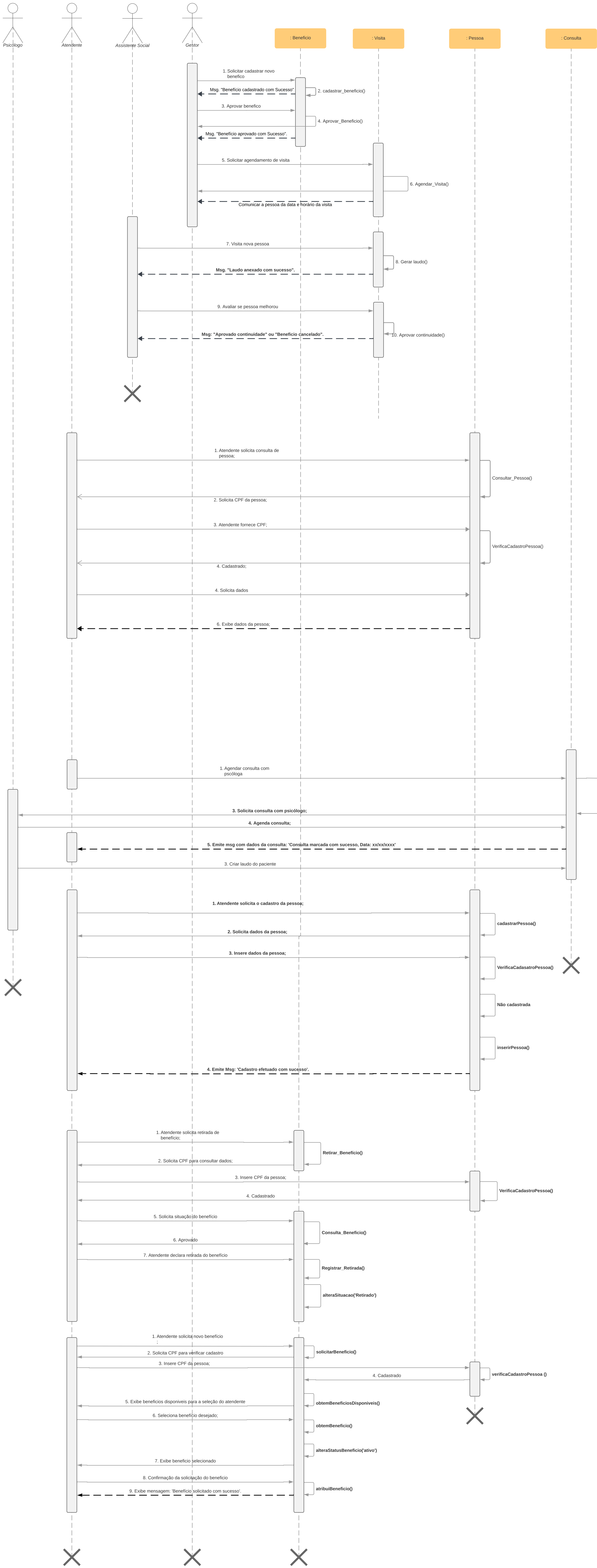
Descrição do caso de uso: Solicitar\_Beneficio

Curso Normal

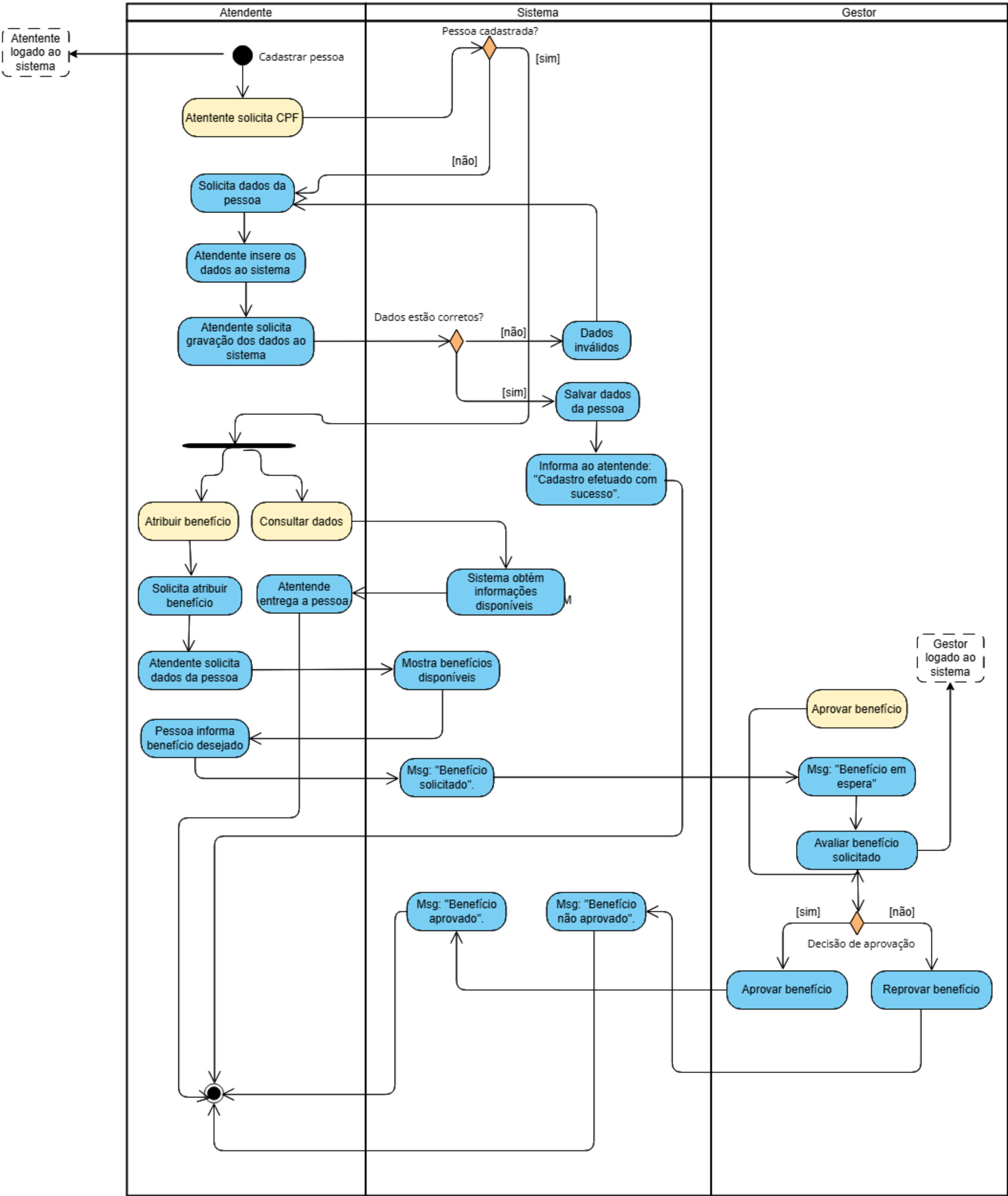
1. Atendente solicita atribuir benefício;
2. Sistema solicita CPF;
3. Atendente informa CPF;
4. Sistema verifica se a pessoa já está cadastrada;
5. Sistema obtém benefícios disponíveis;
6. Sistema exibe os benefícios para a requisição da pessoa;
7. Atendente informa o benefício desejado;
8. O sistema obtém dados do produto;
9. O sistema altera o status ('ativo');
10. O sistema solicita confirmação;
11. O Atendente confirma a solicitação;
12. O sistema atribui o benefício a pessoa;
13. o sistema emite a mensagem 'benefício solicitado'.

Cursos Alternativos

- Caso 1: A pessoa não possui cadastro.  
4.1. O sistema inicia o use case 'Cadastrar\_Pessoa'  
4.2. O sistema volta para o passo 3.  
Caso 2: Benefício não disponível.  
5.1. Emite msg: Benefício não disponível no momento.







Cadastrar Pessoa

**Fluxo Principal**  
**Curso Normal**  
1. O atendente solicita cadastro.  
2. O atendente insere NIS da pessoa.  
3. O sistema verifica se esta pessoa já está cadastrada. [A1]  
4. O sistema solicita dados da pessoa.  
5. O atendente fornece os dados.  
6. O atendente solicita gravação dos dados. [A2]  
7. O sistema realiza o cadastro dos dados da pessoa.  
8. O sistema emite a msg 'Cadastro efetuado com sucesso'.

**Fluxos Alternativos**  
A1. Pessoa já cadastrada  
O sistema informa que a pessoa já está cadastrado.  
Volta para o passo 1 do Fluxo Principal.  
A2. Dados inválidos  
Se algum dado da pessoa estiver em desacordo com as regras de validação e restrições, o sistema informa a situação ao gestor.  
Volta para o passo 4 do Fluxo Principal.

**Pós-condições**  
Os dados da pessoa são incluídos no sistema conforme a solicitação do atendente.

**Pré-condições**  
Atendente está logado no sistema.

**Restrições e Validações**  
Nenhum campo poderá ser deixado em branco.  
Datas de nascimento devem ser de antes do ano atual -18.

Atribuir Benefício

**Fluxo Principal**  
O atendente solicita atribuir um benefício. [A1]  
O sistema solicita dadps da pessoa. [A2]  
O sistema obtém os benefícios disponíveis.  
Se houver benefícios disponíveis, continua para o próximo passo.  
Se não houver benefícios disponíveis, o sistema emite a mensagem "Benefício não disponível no momento." (Caso 1).  
O sistema exibe os benefícios disponíveis para a requisição da pessoa.  
O atendente informa o benefício desejado. [A4]  
O sistema obtém os dados do benefício selecionado.  
O sistema altera o status do benefício para "ativo".  
O sistema solicita confirmação da atribuição do benefício.  
O atendente confirma a solicitação.  
O sistema atribui o benefício à pessoa.  
O sistema emite a mensagem "Benefício solicitado".

**Fluxos Alternativos**  
**Caso 1: Benefício não disponível**  
5.1. O sistema emite a mensagem "Benefício não disponível no momento".

**Pós-condições**  
O status do benefício é atualizado para "ativo", caso a atribuição tenha sido confirmada.

**Restrições e Validações**  
O atendente deve fornecer um CPF válido para prosseguir.  
Todos os benefícios atribuídos devem estar disponíveis no sistema.

Aprovar Beneficio

**Fluxo principal**  
1. Gestor está logado ao sistema;  
2. Verifica benefícios em espera;  
3. Avalia benefício solicitado[A1];  
4. Informa ao sistema se aprovado ou reprovado;  
4 - Emite mensagem de sucesso;

**Fluxos Alternativos**  
Caso 1: 4.1 Caso benefício aprovado sistema exibe mensagem "Benefício Aprovado".  
4.2 Caso benefício reprovado sistema exibe mensagem "Benefício Reprovado".

Consultar dados

**Fluxo principal**  
1. Atendente solicita consultar dados;  
2. Sistema obtém informações disponíveis;  
3. Sistema retorna dados da pessoa;  
4. Atendente disponibiliza dados;



