



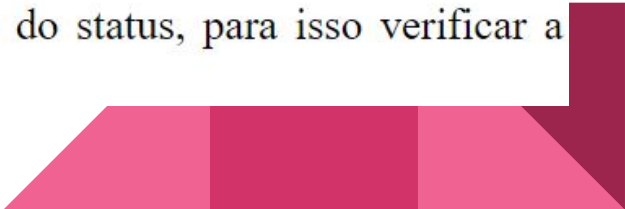
DIM0541- Banco de Dados

Motivação

O problema de	Gerenciar bens patrimoniais
Afeta	Bolsistas, servidores, alunos, professores
Cujo impacto é	Localizar, cadastrar e ter controle de entrada e saída dos bens patrimoniais de forma fácil
Uma boa solução seria	Um sistema simples e prático que permitisse o cadastro, a localização e acautelamento desses bens.



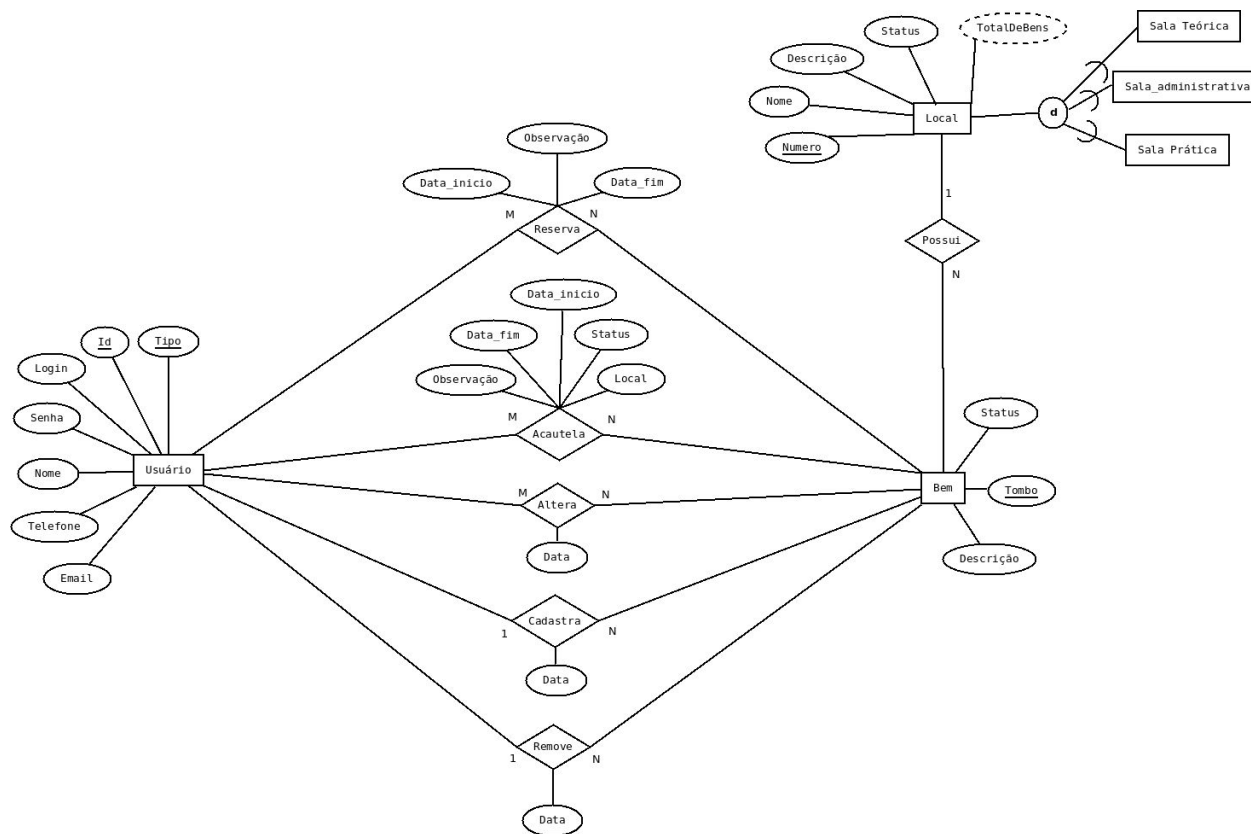
Funcionalidades

- **Cadastro de Usuário:** cada usuário será cadastrado no sistema e terá um tipo que lhe atribui determinada permissão de acesso ao sistema (para isso verificar a permissão do usuário);
 - **Alteração de Usuário:** mudança/atualização dos dados cadastrais do usuário;
 - **Listagem de Usuário:** lista dos usuários no sistema;
 - **Remoção de Usuário:** remover usuário do sistema (para isso verificar a permissão do usuário);
 - **Cadastro de Objeto:** cadastro dos dados do objeto no sistema;
 - **Alteração de Objeto:** mudança/atualização dos dados do objeto (status, localização, etc.);
 - **Listagem de Objeto:** lista de objetos cadastrados no sistema;
 - **Remoção de Objeto:** remover objeto do sistema (dependendo do status, para isso verificar a permissão do usuário);
- 

Funcionalidades

- **Cadastro de Local:** cadastrar um local no sistema, para que esse possa receber bens patrimoniais;
- **Alteração de Local:** alterar dados da localização, bem como os bens do local;
- **Listagem de Local:** listagem dos locais cadastrados no sistema;
- **Remoção de Local:** remoção de local cadastrado no sistema (para isso verificar a permissão do usuário);
- **Cadastro de Cautela:** cadastrar uma cautela para caso de empréstimo de bens;
- **Alteração de Cautela:** alterar algum dado na cautela para empréstimo de bem (renovação da cautela);
- **Listagem de Cautela:** listar cautelas cadastradas no sistema;
- **Remoção de Cautela:** após a entrega e conferência dos bens, remover a cautela do sistema (inativar);

Modelo ER



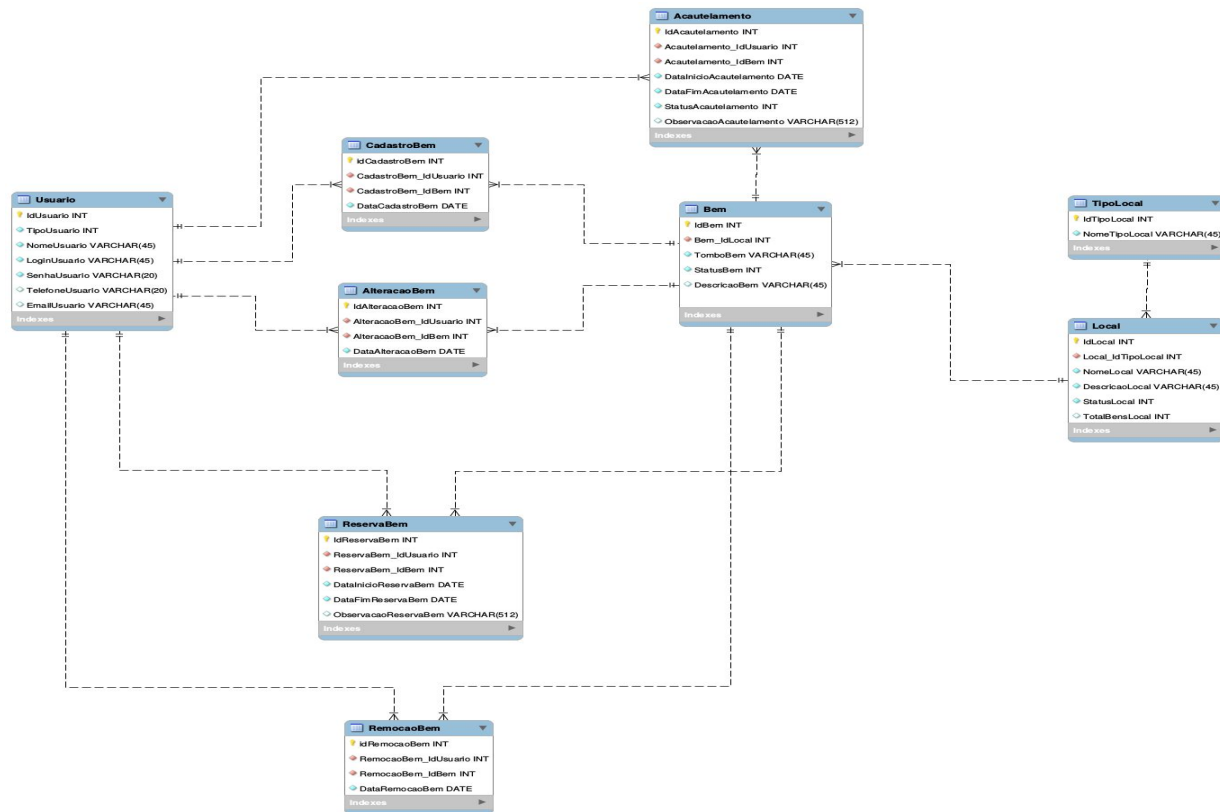
Modelo ER - Desafios

- Modelagem dos tipos de Local
- Definição das operações por tipo de usuário
- Divisão dos relacionamentos

Para chegar nesse modelo que tem três entidades e seis relacionamentos precisamos de 6 aulas, pois os modelos iniciais além de serem mais complexos, não conseguiam abranger de forma correta o escopo do problema.



Modelo Relacional



Modelo Relacional - Desafios

- Adaptar as diferentes especializações de Local
- Adaptar os relacionamentos entre Usuário e Bem
- Definição do Domínio dos atributos



Dados do Modelo Relacional

Modelo com sete tabelas:

- Usuário (Entidade);
- Bem (Entidade);
- Local (Entidade);
- Acautelamento (Relacionamento);
- Tipo de Local (Definição de tipo de entidade);
- Alteração do Bem (Relacionamento entre Bem e Usuário);
- Reserva Bem (Relacionamento).



Normalização do Modelo Relacional

```
schema([acautelamento_idusuario, acautelamento_tombobem, datafimacautelamento, datainicioacautelamento, idacautelamento, observacao, statusacautelamento]).
fds([
  [[idacautelamento], [datafimacautelamento]],           /* idacautelamento -> datafimacautelamento */
  [[idacautelamento], [datainicioacautelamento]],        /* idacautelamento -> datainicioacautelamento */
  [[idacautelamento], [acautelamento_idusuario]],        /* idacautelamento -> acautelamento_idusuario */
  [[idacautelamento], [observacao]],                      /* idacautelamento -> observacao */
  [[idacautelamento], [statusacautelamento]],            /* idacautelamento -> statusacautelamento */
  [[idacautelamento], [acautelamento_tombobem]]          /* idacautelamento -> acautelamento_tombobem */
]).
verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['Acautelamento.pl'].
true.
```

```
?- verificaBCNF.
true.
```

Normalização do Modelo Relacional

```
schema([alteracaoBem_idusuario, alteracaoBem_tombobem, dataalteracaoBem, idalteracaoBem]).
fds([
    [[idalteracaoBem], [alteracaoBem_idusuario]], /* idalteracaoBem -> idusuario */
    [[idalteracaoBem], [alteracaoBem_tombobem]], /* idalteracaoBem -> tombobem */
    [[idalteracaoBem], [dataalteracaoBem]]      /* idalteracaoBem -> data */
]).

verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['AlteracaoBem.pl'].
```

```
true.
```

```
?- verificaBCNF.
```

```
true .
```



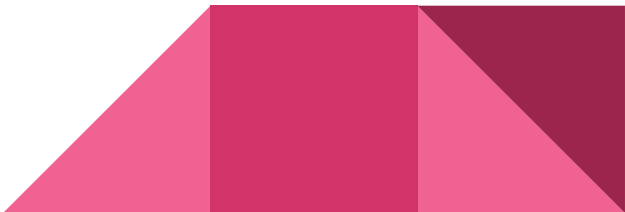
Normalização do Modelo Relacional

```
schema([bem_idlocal, descricaobem, idbem, statusbem, tombobem]).  
fds([  
    [[idbem], [statusbem]],           /* idbem -> statusbem */  
    [[idbem], [tombobem]],           /* idbem -> tombobem */  
    [[idbem], [descricaobem]],       /* idbem -> descricaobem */  
    [[idbem], [bem_idlocal]]         /* idbem -> bem_idlocal */  
]).
```

```
verifica3NF :- schema(R), fds(F), is3NF(R,F).  
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['Bem.pl'].  
true.
```

```
?- verificaBCNF.  
true.
```



Normalização do Modelo Relacional

```
schema([alteracaobem_idusuario, alteracaobem_tombobem, datacadastroalteracaobem, idcadastrobem]).
fds([
  [[idcadastrobem], [alteracaobem_idusuario]],           /* idcadastrobem -> alteracaobem_idusuario */
  [[idcadastrobem], [alteracaobem_tombobem]],           /* idcadastrobem -> alteracaobem_tombobem */
  [[idcadastrobem], [datacadastroalteracaobem]]         /* idcadastrobem -> datacadastroalteracaobem */
]).

verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['CadastroBem.pl'].
true.
```

```
?- verificaBCNF.
true.
```

Normalização do Modelo Relacional

```
schema([descricaolocal, idlocal, nomelocal, statuslocal, tipolocal_idtipolocal, totalbenslocal]).
fds([
    [[idlocal], [descricaolocal]],          /* idlocal -> descricaolocal */
    [[idlocal], [nomelocal]],               /* idlocal -> nomelocal */
    [[idlocal], [statuslocal]],             /* idlocal -> statuslocal */
    [[idlocal], [tipolocal_idtipolocal]],   /* idlocal -> tipolocal_idtipolocal */
    [[idlocal], [totalbenslocal]]           /* idlocal -> totalbenslocal */
]).

verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['Local.pl'].
true.
```

```
?- verificaBCNF.
true.
```

Normalização do Modelo Relacional

```
schema([dataremocaoremocaobem, idremocaobem, remocaobem_idusuario, remocaobem_tombobem]).
fds([
    [[idremocaobem], [remocaobem_idusuario]],           /* idremocaobem -> remocaobem_idusuario */
    [[idremocaobem], [remocaobem_tombobem]],           /* idremocaobem -> remocaobem_tombobem */
    [[idremocaobem], [dataremocaoremocaobem]]          /* idremocaobem -> dataremocaoremocaobem */
]).

verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['RemocaoBem.pl'].
```

```
true.
```

```
?- verificaBCNF.
```

```
true.
```

Normalização do Modelo Relacional

```
schema([datafimreservabem, datainicioreservabem, idreservabem, observacaoreservabem, reservabem_tombobem, reservabem_usuario_id]).
fds([
    [[idreservabem], [reservabem_tombobem]],          /* idreservabem -> reservabem_tombobem */
    [[idreservabem], [datafimreservabem]],           /* idreservabem -> datafimreservabem */
    [[idreservabem], [datainicioreservabem]],        /* idreservabem -> datainicioreservabem */
    [[idreservabem], [observacaoreservabem]],        /* idreservabem -> observacaoreservabem */
    [[idreservabem], [reservabem_usuario_id]]        /* idreservabem -> reservabem_usuario_id */
]).
```

```
verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['ReservaBem.pl'].
true.
```

```
?- verificaBCNF.
true .
```

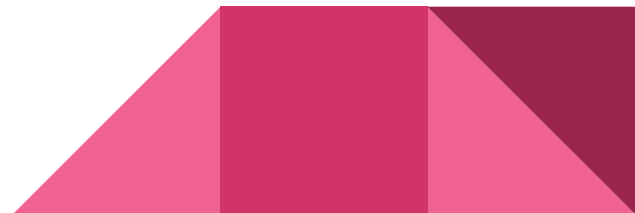

Normalização do Modelo Relacional

```
schema([idtipolocal, nometipolocal]).  
fds([  
    [[idtipolocal], [nometipolocal]] /* idtipolocal -> nometipolocal */  
]).
```

```
verifica3NF :- schema(R), fds(F), is3NF(R,F).  
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['TipoLocal.pl'].  
true.
```

```
?- verificaBCNF.  
true .
```



Normalização do Modelo Relacional

```
schema([emailusuario, idusuario, loginusuario, nomeusuario, senhausuario, telefoneusuario, tipousuario]).
fds([
    [[idusuario, tipousuario], [emailusuario]],      /* idusuario, tipousuario -> emailusuario */
    [[idusuario, tipousuario], [loginusuario]],      /* idusuario, tipousuario -> loginusuario */
    [[idusuario, tipousuario], [nomeusuario]],       /* idusuario, tipousuario -> nomeusuario */
    [[idusuario, tipousuario], [senhausuario]],      /* idusuario, tipousuario -> senhausuario */
    [[idusuario, tipousuario], [telefoneusuario]],   /* idusuario, tipousuario -> telefoneusuario */
    [[loginusuario], [emailusuario]],                /* loginusuario -> emailusuario */
    [[loginusuario], [nomeusuario]],                 /* loginusuario -> nomeusuario */
    [[loginusuario], [senhausuario]],                /* loginusuario -> senhausuario */
    [[loginusuario], [telefoneusuario]],             /* loginusuario -> telefoneusuario */
    [[loginusuario], [idusuario, tipousuario]]       /* loginusuario -> idusuario, tipousuario */
]).
```

```
verifica3NF :- schema(R), fds(F), is3NF(R,F).
verificaBCNF :- schema(R), fds(F), isBCNF(R,F).
```

```
?- ['Usuario.pl'].
true.
```

```
?- verificaBCNF.
true.
```