



ENGENHARIA DE SOFTWARE

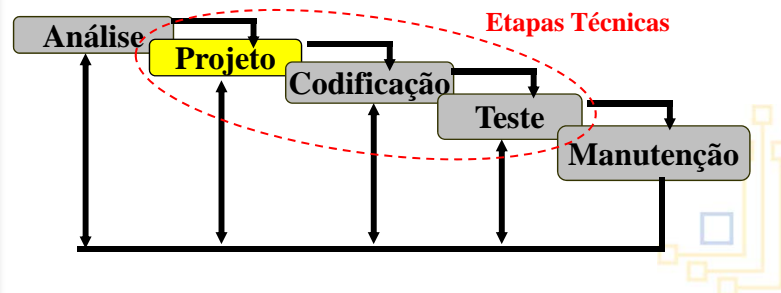
UNIDADE 4 – Fase de Projeto (Modelagem de Dados)
(Aula 6 – Dicionário de Dados e Especificação de Processos)

Prof. Ivan Nunes da Silva

1. Introdução

1.1 Revisão (1)

- Como vimos na aula passada, a fase de projeto é a primeira dentre as três etapas técnicas do desenvolvimento do software (Projeto, Codificação, Teste).



1. Introdução

1.1 Revisão (2)

- A abordagem de projeto orientado ao Fluxo de Dados é baseada nos seguintes agentes:
 - Diagrama de Fluxo de Dados (DFD).
 - Dicionário de Dados (DD).
 - Especificação de Processos (EP).
 - Diagramas Entidade Relacionamento (DER).

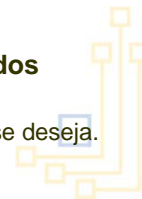


3

1. Introdução

1.1 Revisão (3)

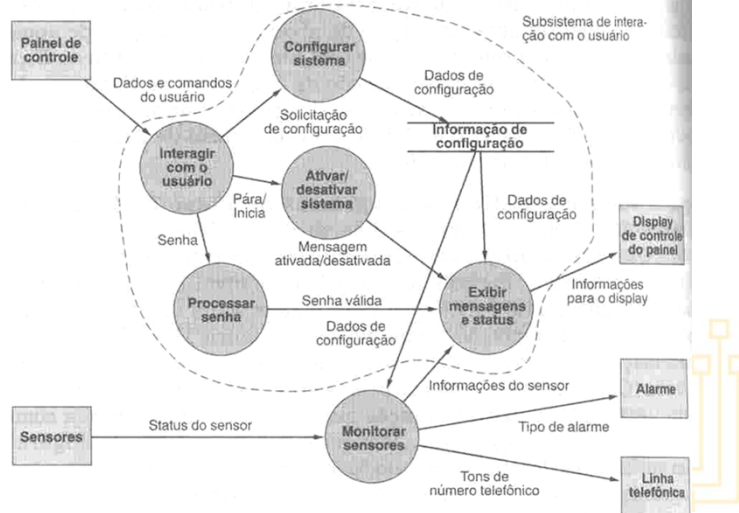
- **O Diagrama de Fluxo de Dados (DFD):**
 - **É um modelo que permite representar o sistema como uma rede de processos:**
 - Salienta as funções que precisam ser implementadas e os fluxos de dados manipulados por estas funções.
 - **Técnica de análise gráfica que procura representar o fluxo de informação:**
 - Ressalta as sucessivas transformações que sofrem ao longo dos diferentes processos do sistema.
 - **Ferramenta gráfica que transcreve a lógica dos procedimentos do sistema em estudo:**
 - Descreve o sistema com o nível de detalhe que se deseja.



4

1. Introdução

1.1 Revisão (4) // Software “SafeHome”



5

1. Introdução

1.1 Revisão (5) // Software “Comércio de Livros”



6

2. Dicionário de Dados

2.1 Definições

- É a segunda ferramenta importante para o projeto de software.
- É uma listagem única e organizada de todos os elementos de dados pertinentes ao sistema.
- Possui definições precisas e rigorosas para que o usuário e o engenheiro de software possam conhecer as estruturas de entradas e saídas, envolvendo:
 - Depósitos de Dados.
 - Fluxo de Dados.
 - Entidades e Relacionamentos com Atributos.



7

2. Dicionário de Dados

2.2 Dicionário de Dados Deve Descrever:

- O **significado** dos fluxos e depósitos mostrados nos diagramas de fluxo de dados.
- A **composição** de pacotes agregados que se movimentam pelos fluxos.
 - Ex: Endereço_Cliente (Rua, Número, Cidade, Etc).
- A composição dos **registros** de dados que constitui o depósito de dados.
- **Tipos** e **unidades** dos dados (variáveis) que compõem os fluxos e depósitos de dados.
- Os detalhes dos **relacionamentos** entre os depósitos de dados realçados no Diagrama de Entidades-Relacionamentos.



8

2. Dicionário de Dados

2.3 Organização e Simbologia:

- Na definição dos dados são utilizados os seguintes símbolos:

= → é composto de

+ → e

() → opcional (pode estar presente ou ausente)

[] → escolha uma das alternativas

| → separador de alternativas em []

→ identificador (chave) de um arquivo

* * → delimitador de comentário

n{ }m → delimitador de quantidade
n ⇒ quantidade mínima
m ⇒ quantidade máxima



9

2. Dicionário de Dados

2.4 Exemplo (1):

- Seja um depósito de dados (arquivo), denominado "Alunos", contendo registros sobre os mesmos.
- Cada registro é composto por: nome do aluno, sexo, número USP, endereço e telefone.
- Definir o item do Dicionário de Dados que corresponderá a este arquivo.

ALUNOS = #Nro_USP + Nome + Sexo + Endereço

Nro_USP = *Número de Identificação do Aluno*

Tipo: Numérico (7 posições)

Nome = *Nome do Aluno*

Tipo: Alfabético (30 posições)

Sexo = [M | F]

Endereço = *Endereço Residencial do Aluno*

Tipo: Alfanumérico (60 posições)

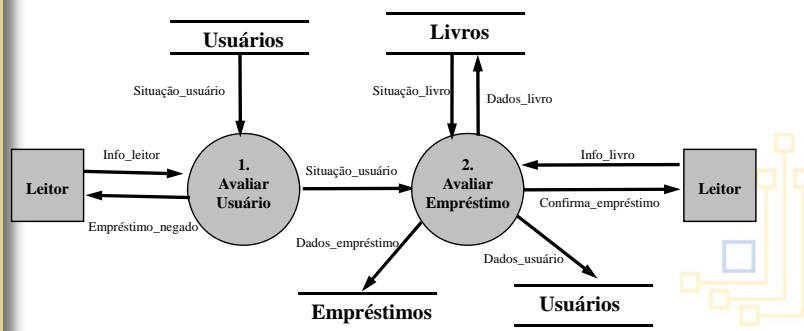


10

2. Dicionário de Dados

2.4 Exemplo (2):

- O diagrama seguinte define parte de um DFD do software “BIBLIOTECA”.
- Elabore o dicionário de dados dos Depósitos de Dados e de seus Fluxos de Dados.



11

2. Dicionário de Dados

Dicionário de Dados {Exemplo (2)}:

- O arquivo **USUÁRIOS** deve conter os seguintes parâmetros:
 - Tipo de usuário: “Professor”, “Aluno” ou “Funcionário”.
 - Número USP do usuário.
 - Nome do usuário.
 - Campo de status indicando se usuário pode retirar o livro.
 - Quantidade de livros retirados (máximo de 8 livros).

USUÁRIOS = #Nro_USP + Nome + Tipo + Status + Qdade

Nro_USP = *Tipo: Numérico (7 posições)*

Tipo = [Professor | Aluno | Funcionário]

Nome = *Tipo: Alfabético (30 posições)*

Status = *Campo que indica se o usuário pode retirar livro*
[OK | Não OK]

Qdade = *Quantidade de livros retirados*

Tipo: Numérico (1 posição); Máximo de 8

12

2. Dicionário de Dados

Dicionário de Dados {Exemplo (2)}:

- O arquivo **LIVROS** deve conter os seguintes parâmetros:
 - Código do livro.
 - Título do livro.
 - Campo indicando se o livro está disponível.
 - Local (prateleira) em que se encontra o livro (Opcional).

LIVROS = #Código + Título + Disponibilidade + (Local)

Cód_Livro = *Tipo: Numérico (8 posições)*

Título = *Tipo: Alfanumérico (30 posições)*

Disponibilidade = *Indica se o livro está disponível ou não*
[Sim | Não]

Local = *Indica a prateleira em que se encontra o livro*
Tipo: Alfanumérico (10 posições)

13



2. Dicionário de Dados

Dicionário de Dados {Exemplo (2)}:

- O arquivo **EMPRÉSTIMOS** deve conter os seguintes parâmetros:
 - Identificação do Empréstimo
 - Número USP do usuário
 - Código do livro
 - Data de saída do livro.
 - Data de devolução do livro.
 - Cada empréstimo pode conter no máximo dois livros.

EMPRÉSTIMOS = #Identificador + Nro_USP + 1{Reg_Livros}2

Identificação = *Tipo: Numérico (8 posições)*

Nro_USP = *Tipo: Numérico (7 posições)*

Reg_Livros = Cod_livro + Data_Saída + Data_Devolução

Cod_Livro = *Numérico (8 posições)*

Data_Saída = *Formato data: dd/mm/aa*

Data_Devolução = *Formato data: dd/mm/aa*

14



2. Dicionário de Dados

Dicionário de Dados {Exemplo (2)}:

● Descrição dos FLUXOS DE DADOS (1):

Info_leitor = *Dados de leitor*

Nro_USP = *Tipo: Numérico (7 posições)*

Empréstimo_negado = *Rejeição do empréstimo*

“Usuário em débito ou já retirou 8 livros”

Situação_usuario = *Descreve se usuário pode retirar livro*

Status = [OK | Não OK]

Info_livro = *Informação de livro a ser emprestado*

Cod_livro = *Tipo: Alfabético (30 posições)*

Situação_livro = *Informa se livro está disponível*

Disponibilidade = [Sim | Não]

15



2. Dicionário de Dados

Dicionário de Dados {Exemplo (2)}:

● Descrição dos FLUXOS DE DADOS (2):

Dados_livro = *Dados para atualização de disponibilidade*

Disponibilidade = [Sim | Não]

Dados_empréstimo = *Dados para inclusão de empréstimo*

Identificação = *Tipo: Numérico (8 posições)*

Nro_USP = *Tipo: Numérico (7 posições)*

Cod_Livro = *Numérico (8 posições)*

Data_Saída = *Formato data: dd/mm/aa*

Data_Devolução = *Formato data: dd/mm/aa*

Dados_usuario = *Dados para atualização de quantidade*

Qdade = *Tipo: Numérico (1 posição)*

Confirma_empréstimo = *Faz confirmação do empréstimo*

[“Emite recibo” | “Comunica indisponibilidade”]

16



3. Especificação de Processos

3.1 Definições

- É a terceira ferramenta importante para o projeto de software.
- Todos os processos devem ser especificados.
- Encarrega de definir o que deve ser feito dentro de um processo para transformar entradas em saídas.
- Deve ser detalhado o suficiente para ser transformada em um procedimento computacional.
- Pode ser confeccionada de várias formas:
 - Linguagem Estruturada (Pseudocódigo).
 - Tabelas de Decisão.
 - Fluxograma
 - Condições pré/pós, etc.



17

3. Especificação de Processos

3.2 Linguagem Estruturada (1)

- Descrição detalhada dos passos que definem a lógica associada com um procedimento (processo).
- Independe da linguagem de programação que será adotada na codificação.
- Estruturas de Comparação:
 - **Estrutura de Comparação (1a. Forma)**
Se <condição>
Então <Grupo_instrução_1>
Fim_Se
 - **Estrutura de Comparação (2a. Forma)**
Se <condição>
Então <Grupo_instrução_1>
Senão <Grupo_instrução_2>
Fim_Se



18

3. Especificação de Processos

3.2 Linguagem Estruturada (2)

- **Estruturas de Comparação:**

- Estrutura de Comparação (3a. Forma)

Escolha <opção>

1: <grupo_comandos_1>

2: <grupo_comandos_2>

(...)

n: <grupo_comandos_n>

Fim_Escolha

- **Estruturas de Entrada/Saída:**

- Estrutura de Entrada

Obtenha(<lista_variáveis>) ou Leia(<lista_variáveis>)

- Estrutura de Saída

Escreva(<lista_variáveis>) ou Imprima(<lista_variáveis>)

19



3. Especificação de Processos

3.2 Linguagem Estruturada (3)

- **Estruturas de Repetição:**

- Estrutura de Repetição (Enquanto)

Enquanto <condição> faça

<grupo_comandos>

Fim_Enquanto

- Estrutura de Repetição (Repita)

Repita

<grupo_comandos>

Até <condição>

- Estrutura de Repetição (Para)

Para <Var> =<Valor_Inicial> até <Valor_Final> Faça

<grupo_comandos>

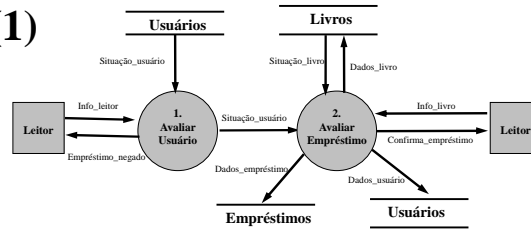
Fim_Para

20



3. Especificação de Processos

3.3 Exemplo (1)



● Especificação do processo

“AVALIAR USUÁRIO” do Software “BIBLIOTECA”:

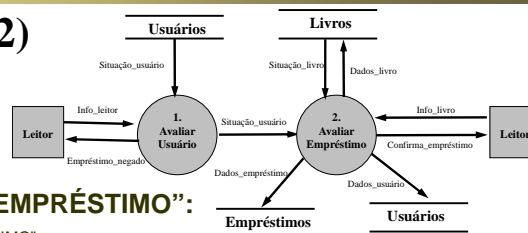
```

Processo “AVALIAR USUÁRIO”
  Obtenha(Info_leitor)
  Leia (USUÁRIOS, Status)
  Se Status = “Não OK”
    Então retorne ao LEITOR “Usuário em débito ou já retirou 8 livros”
    Senão retorne ao Programa que Situação_usuario = “Ok”
      armazene Nro_USP de forma global
  Fim_Se
  Fim_Processo
  
```

21

3. Especificação de Processos

3.3 Exemplo (2)



● Especificação do processo “AVALIAR EMPRÉSTIMO”:

```

Processo “AVALIAR EMPRÉSTIMO”
  Se Situação_usuario = “Ok”
    Então
      Obtenha(Info_livro)
      Leia(LIVROS, Disponibilidade)
      Se Disponibilidade = “Ok”
        Então Qdade ← Qdade + 1
          Escreva(USUÁRIOS, Qdade)
          Disponibilidade ← “Não OK”
          Escreva(LIVROS, Disponibilidade)
          Obtenha(Data_Atual)
          Atribua Data_Devolução
          Atribua Identificação
          Escreva(EMPRÉSTIMOS, Identificação, Nro_USP, Cod_livro,
            Data_saída, Data_devolução)
          “Emite Recibo” ao LEITOR
        Senão
          “Comunica Indisponibilidade” ao LEITOR
      Fim_Se
    Senão
      Imprima(“Leitor Inadimplente – Empréstimo Recusado”)
  Fim_Processo
  
```

22

Fim da Apresentação

AVISOS

**Prova P1 (13/10/2020) // Aulas 01 a 06
– 08h10 às 10h00 –**

