

BibIme - Um Software Gerenciador de Bibliotecas Produzido de Forma Cooperativa

Dairton Bassi, Kelly Braghetto, Eduardo Colli,
Fabio Kon, João Eduardo Ferreira

Instituto de Matemática e Estatística
Universidade de São Paulo

Jun/05

Apresentação

Parte I - BibIme

- Motivação
- Arquitetura
- Banco de Dados
- Processo de Desenvolvimento

Apresentação

Parte II - Periódicos

- Motivação
- Características

Motivação

A grande quantidade de obras e pessoas envolvidas no dia-a-dia de uma biblioteca exige um software especializado que garanta a eficiência e a boa qualidade dos serviços.

Além disso, o software deve considerar os vários setores da biblioteca: Acervo, Periódicos, Gerenciamento de Usuários, Balcão de Empréstimos, Controle de Aquisição, etc. Cada um requer um sistema que apóie suas atividades.

Dificuldades

- Integração de dados entre os sistemas.
- Dependência entre sistemas
- Modelo de dados abrangente e flexível.

BibIme

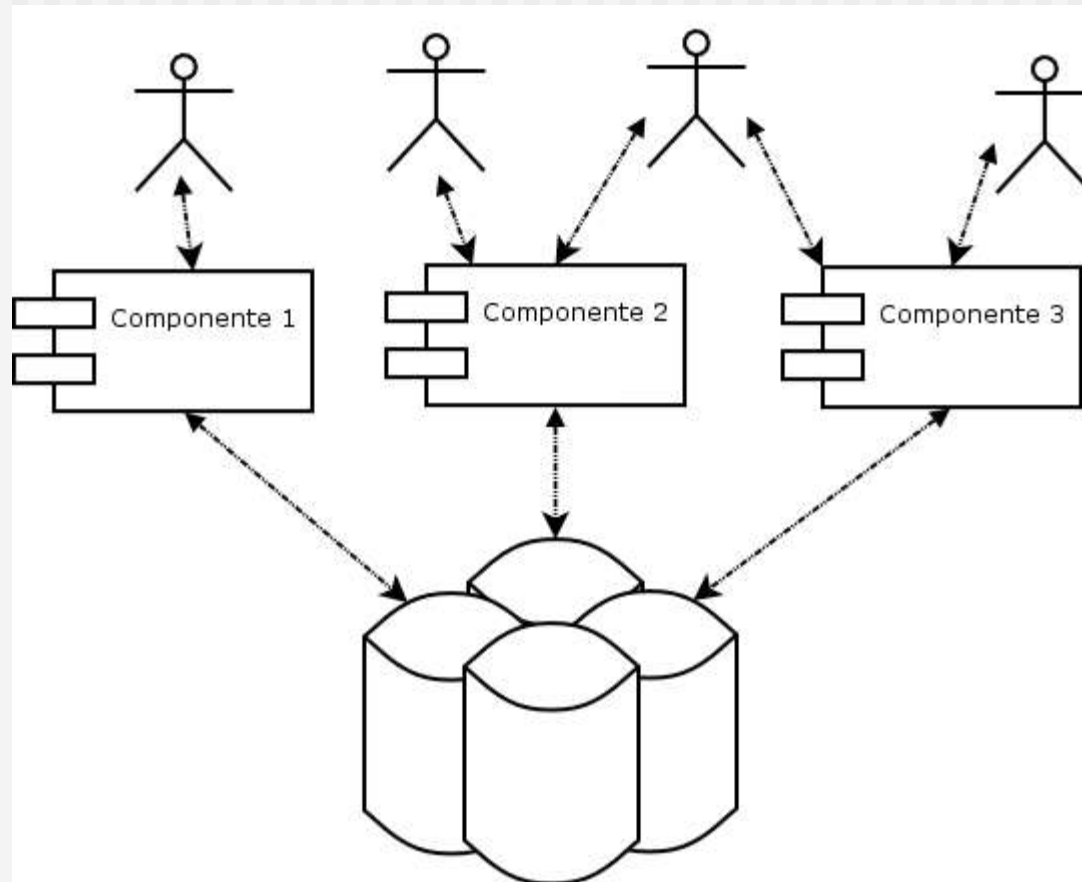
O BibIme é um sistema de código aberto para o gerenciamento de todos setores e processos da um biblioteca.

Desenvolvido no IME e implantado em sua biblioteca, usa banco de dados PostgreSQL e as camadas de negócios e apresentação codificadas em Java.

Arquitetura

- A camada de negócios e a interface são formados por **componentes** de software.
- Cada componente é o sistema que oferece um conjunto de serviços à biblioteca.
- Todos os componentes interagem com um único banco de dados.
- A estrutura do banco de dados define módulos independentes e relacionados.

Arquitetura Simplificada

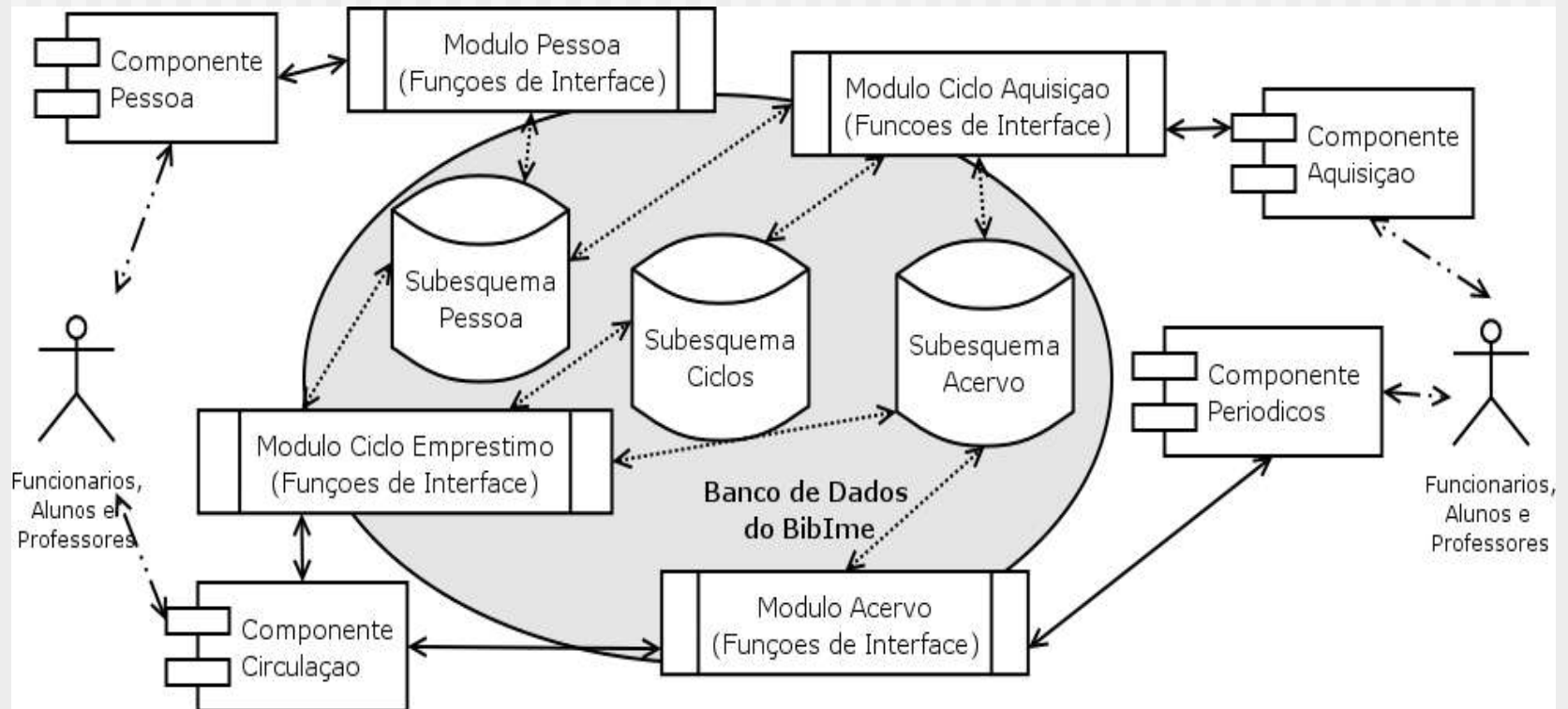


Banco de Dados

O banco de dados é dividido em módulos com uma interface formada por um conjunto de *stored procedures* que encapsulam o modelo de dados e garantem a flexibilidade.

Apesar das diferenças entre os tipos de itens da biblioteca, a modelagem usa uma hierarquia de especialização que permite tratar uniformemente todos os itens do acervo.

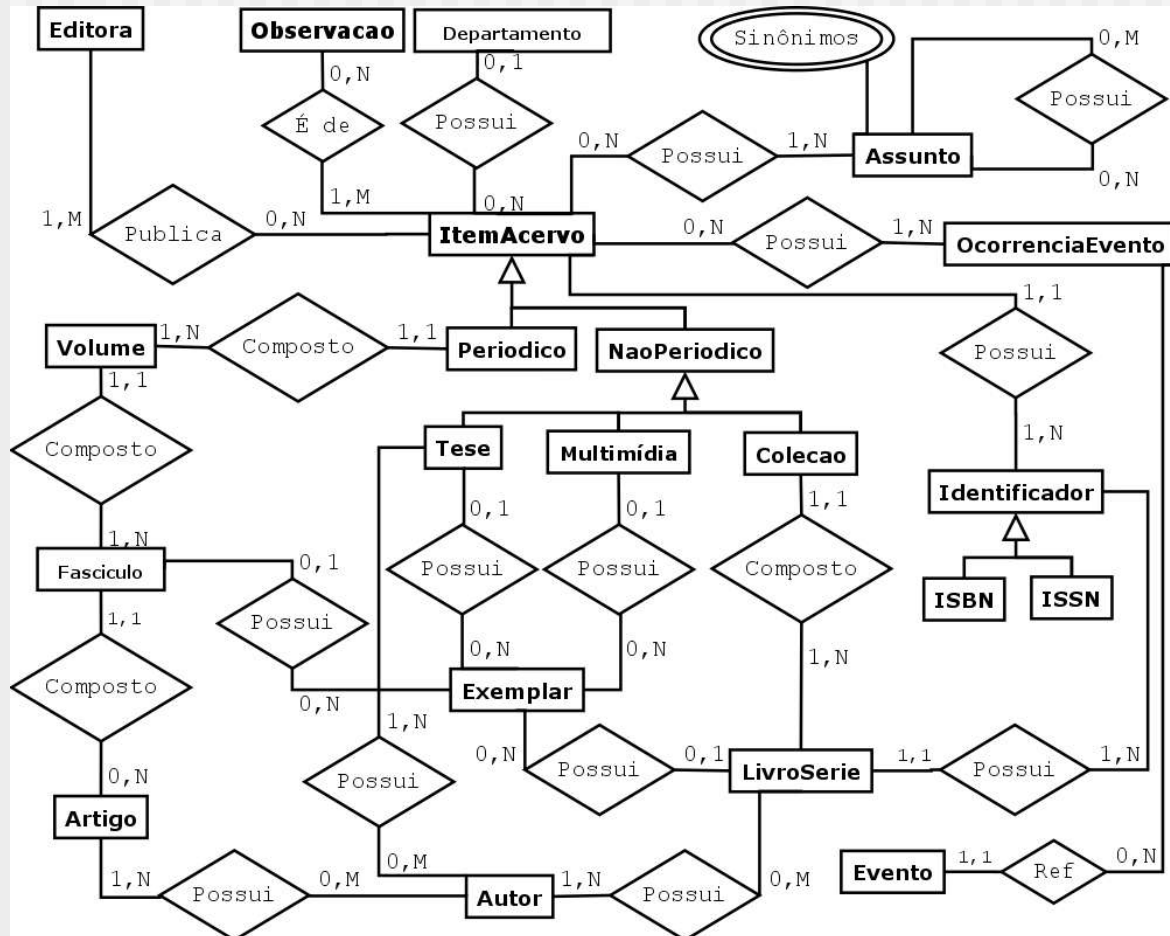
Arquitetura Geral



Módulos

- **Módulo Acervo** – catálogo de obras
- **Módulo Pessoa** – cadastro de usuários e funcionários.
- **Módulo Empréstimo** – gerenciamento da circulação dos itens do acervo.
- **Módulo Aquisição** – gestão do processo de aquisição de novos itens para o acervo.

Acervo - Modelo Conceitual



Módulo Acervo

- Hierarquia de Especialização para itens do acervo. Livros, séries, teses, materiais multimídia e Periódicos (com volumes e fascículos) podem ser tratados de forma uniforme.

Módulo Pessoa

- Compreende mais de 30 tipos de usuários.
- Identificação de usuários centralizada através de documentos.
- Permissões separadas por componente.

Desenvolvimento

O desenvolvimento é dividido em duas fases complementares:

- Fase Incremental
- Fase de Ajustes

Fase Incremental

- Trabalho em equipe praticando XP.
- Testes automatizados
- Funcionários e Coordenadores da biblioteca (*clientes*) presentes durante o desenvolvimento.

Fase de Ajustes

- Refinamento do software entregue na fase anterior.
- Adição de recursos complementares.
- Alunos de mestrado realizam e coordenam testes e correções que tornam o software pronto para entrar em produção.

Próximos Passos

1. Ajustes no *Componente Pessoa* para entrar em produção.
2. Implementação do *Componente Circulação*.
3. Implementação do *Componente Aquisição*.

Parte II

XoPs

Componente Periódicos

Padrões



- MVC
- Validator
- Observer
- Façade
- Singleton
- Data Transfer Object*
- Proxy



Funcionalidades

- Cadastro e busca de periódicos, volumes, fascículos e pagamentos
- Gerador automático de coleção
- Estatísticas de uso
- Relatório de completeza
- Perfis *administrativo, cadastral* e de *visualização*

Bibliografia

- Barrera, J., Junior, R. M. C., Ferreira, J. E., and Gubitoso, M. D. (2004). An environment for knowledge discovery in biology. *Journal of Computers in Biology and Medicine*, 34(5):427–447.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stad, M. (1996). *Pattern-Oriented Software Architecture — A System of Patterns*. John Wiley Press.
- Ferreira, J. E. and Busichia, G. (1999). Database modularization design for the construction of flexible information systems. *Proceedings IEEE for the IDEAS99*, pages 415–422.
- Ferreira, J. E. and Finger, M. (2000). Controle de concorrência e distribuição de dados: Teoria clássica, suas limitações e extensões modernas. *XII Escola de Computação*, pages 73–83.
- Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns*. Addison-Wesley.
- Grossman, F., Bergin, J., Leip, D., Merritt, S., and Gotel, O. (2004). One xp experience: Introducing agile (xp) software development into a culture that is willing but not ready. *IBM Centre for Advanced Studies Conference*, pages 242–254.
- MARC XML. The library of congress. <http://www.loc.gov/standards/marcxml>. Último acesso em: 11/04/2005.
- Pawson, R. and Matthews, R. (2002). *Naked Objects*. Wiley and Sons.
- Pressman, R. S. (1992). *Software Engineering: a Practitioner's Approach*. McGraw-Hill, Inc.

Projeto BibIme

malariadb.ime.usp.br/mac439/projetobiblioteca