

Sistema de Informação para Atividades de Extensão Universitária

Discente: Matheus Pereira dos Passos Oliveira

Disciplina: MATA60 – Banco de Dados

Professor: Robespierre Pita

Universidade: UFBA – Instituto de Computação

Repositório: <https://github.com/mppassos/mata60-extensao-ufba>

Abstract

This work presents the development of an Information System to support university extension activities promoted by the Computer Science Institute at UFBA. The system was modeled and implemented in a PostgreSQL database, focusing on privacy, performance, and governance, following the PPP1, MAD1, and PBR2 policies. The design used entity-relationship methodology and standardized nomenclature, with implementation of stored procedures, materialized views, and transactions to enable critical operations, along with audit routines and incremental backup. Analytical and operational dashboards were designed to support decision-making and track the impact of activities. The project prioritizes compliance with LGPD (Brazil's General Data Protection Law), traceability, and SQL query performance.

Keywords: Information System, University Extension, PostgreSQL, LGPD, Data Governance, Audit, Query Performance.

Resumo

Este trabalho apresenta o desenvolvimento de um Sistema de Informação para apoiar as atividades de extensão universitária promovidas pelo Instituto de Computação da UFBA. O sistema foi modelado e implantado em um banco de dados PostgreSQL, com foco em privacidade, desempenho e governança, conforme as políticas PPP1, MAD1 e PBR2. A modelagem seguiu a metodologia entidade-relacionamento e nomenclatura padronizada, com implementação de stored procedures, materialized views e transações para viabilizar operações críticas, além de rotinas de auditoria e backup incremental. Os dashboards analíticos e operacionais foram projetados para apoiar a tomada de decisão e acompanhar o impacto das ações. O projeto prioriza conformidade com a LGPD, rastreabilidade e performance das consultas SQL.

Palavras-chave: Sistema de Informação, Extensão Universitária, PostgreSQL, LGPD, Governança de Dados, Auditoria, Desempenho de Consultas.

1. Descritivo do Projeto

Esta entrega dá continuidade ao projeto iniciado na Entrega 1, cujo objetivo é informatizar a gestão de atividades de extensão do Instituto de Computação da UFBA. Nesta fase, foram realizadas:

- Correções e melhorias no modelo lógico e conceitual, alinhando-se à metodologia MAD1.
- Refinamento dos requisitos com base em novas evidências práticas.

- Documentação atualizada dos modelos lógico (docs/LinkModeloLogico.docx) e conceitual (docs/Conceptual model - BRMW.pdf).

Principais alterações:

- Inclusão de *materialized views* para dashboards.
- Implementação de *stored procedures* e transações com lógica de rollback/commit.
- Adoção das políticas MAD1 (nomenclatura), PPP1 (privacidade/auditoria) e PBR2 (backup/recuperação).

2. Artefatos

2.1. Tela 1: Cadastro com Validação

- **SQL Operations:**
 - `SELECT` : tipos de participante, atividades.
 - `INSERT` : participante e inscrição.
 - `UPDATE` : presença e dados do participante.
- **Rotinas:** uso de `SP_INSERE_PARTICIPANTE_COM_INSCRICAO` .

2.2. Tela 2: Cadastro ou Validação de Feedback

- **SQL Operations:**
 - `SELECT` , `INSERT` , `UPDATE` , `DELETE` .
- **Rotinas:** `SP_CADASTRA_OU_ATUALIZA_FEEDBACK` .

2.3. Dashboard Estratégico

Consultas em `03_Indices_Consultas.sql` :

- Participação por atividade.
- Emissão de certificados.
- Feedback agregado por participante.
- Parceiros com maior investimento.

2.4. Dashboard Operacional

- Atividades por tipo (`GROUP BY`).
- Participantes por categoria.
- Instrutores por atividade.
- Presença e avaliações por atividade.
- Participantes acima da média (janela + subconsulta).

3. Rotinas

3.1. Materialized Views

Scripts em `sql/04_Materialized_Views.sql` :

- `mv_atividade_por_tipo`
- `mv_parceiro_resumo`
- `mv_feedback_participante`
- `mv_instrutores_por_atividade`

View	Descrição

mv_atividade_por_tipo	Conta atividades por tipo e calcula duração média
mv_parceiro_resumo	Resume contribuições dos parceiros
mv_feedback_participante	Agrega feedbacks por participante
mv_instrutores_por_atividade	Lista instrutores envolvidos por atividade

3.2. Stored Procedures

SP_INSERE_PARTICIPANTE_COM_INSCRICAO

- **Parâmetros:** p_nm_participante, p_ds_email, p_id_tp_categoria, p_id_atividade
- **Lógica:** insere participante (se novo) + inscrição (sem duplicação)
- **Transacional:** BEGIN...COMMIT
- **Script:** sql/procedures/SP_INSERE_PARTICIPANTE_COM_INSCRICAO.sql

SP_CADASTRA_OU_ATUALIZA_FEEDBACK

- **Parâmetros:** p_id_atividade, p_id_participante, p_cd_nota, p_ds_comentario
- **Lógica:** verifica nota, atualiza ou insere feedback
- **Transacional**
- **Script:** sql/procedures/SP_CADASTRA_OU_ATUALIZA_FEEDBACK.sql

SP_ATUALIZA_CERTIFICADOS

- **Parâmetros:** p_id_atividade, p_presenca_minima (default 75)
- **Lógica:** verifica presença, gera hash SHA-256, insere certificado, atualiza views
- **Segurança:** executado pelo papel Sistema (PPP1)
- **Script:** sql/procedures/SP_ATUALIZA_CERTIFICADOS.sql

3.3. Transações

Todas as procedures são transacionais (BEGIN/COMMIT) garantindo atomicidade e rollback automático.

3.4. Atualização de Views

Local: sql/05_REFRESH_VIEWS.sh

Função: Executa REFRESH MATERIALIZED VIEW nas views:

1. mv_atividade_por_tipo
2. mv_parceiro_resumo
3. mv_feedback_participante
4. mv_instrutores_por_atividade

4. Políticas de Governança

4.1. PPP1 – Privacidade e Auditoria

- Trigger TG_A_IU_TB_PARTICIPANTE_LOG registra alterações em TB_PARTICIPANTE .
- Tabela TA_AUDITORIA_LOG com operação, registro, valores antes/depois, timestamp e usuário.
- Separação de papéis: DBA, Sistema, Análise, Backup.

4.2. MAD1 – Metodologia de Nomenclatura

Prefixos padronizados: | Tipo | Prefixo | Exemplo | |-----|-----|-----| |
Tabelas | TB_ | TB_TP_ATIVIDADE | | Relacionamentos | RL_ |
RL_ATIVIDADE_PARTICIPANTE | | Tipos | TP_ | TP_NOTA | | Índices | IDX_ |
IDX_MV_ATIV_TIPO | | Chaves Primárias | PK_ | PK_TB_PARTICIPANTE |

4.3. PBR2 – Backup e Recuperação

Script de backup incremental diário:

- Logs, validação mínima de integridade.
- Com usuário `pg_bkpincrem`, acesso ao diretório `./backups`.
- Scripts envolvidos:
 - `sql/08_Backup_PBR2.sh`
 - Requer permissões no PostgreSQL.

5. Desempenho das Consultas

Execução 20x com/sem índices. Resultados em `TemplatePerformance.xlsx`.

Consulta	Média s/ índice	Desvio s/ índice	Média c/ índice	Desvio c/ índice	Speedup
Consulta 1	1.4153 ms	1.2197	1.2232 ms	0.3883	1.16×
Consulta 2	2.5488 ms	6.0685	1.4575 ms	0.8161	1.75×
Consulta 3	3.6065 ms	3.8056	2.7008 ms	2.2035	1.34×
Consulta 4	0.9785 ms	0.6328	0.9826 ms	0.1673	1.00×

Análise:

Consulta 2 teve maior ganho em performance (1.75×). Consulta 3 também melhorou significativamente. Consulta 4 manteve eficiência, e a Consulta 1 teve ganho leve.

6. Anexos

• Scripts SQL:

- `01_DDL_Eschema_Tabelas.sql`
- `02_DML_Populacao_Dados.sql`
- `03_Indices_Consultas.sql`
- `04_Materialized_Views.sql`
- `05_REFRESH_VIEWS.sh`
- `06_Stored_Procedures.sql`
- `07_Triggers_Auditoria.sql`
- `08_Backup_PBR2.sh`

• Modelos:

- `docs/Conceptual model - BRMW.pdf`

- docs/LinkModeloLogico.docx

Artefatos visuais:

- Prints de consultas (prints/Resultados_Consultas/Consulta*.png)
- Arquivo de desempenho: docs/TemplatePerformance.xlsx

7. Dicionário de Dados e Objetos Técnicos

Objeto / Coluna	Tipo	Descrição / Restrições
TB_TP_ATIVIDADE	Tabela	Tipo de atividade
• ID_TP_ATIVIDADE	INTEGER GENERATED ALWAYS AS IDENTITY	PK, identificação automática
• DS_TP_ATIVIDADE	VARCHAR(30) NOT NULL	Descrição do tipo
TB_TP_CATEGORIA	Tabela	Tipo de categoria de participante
• ID_TP_CATEGORIA	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• DS_TP_CATEGORIA	VARCHAR(20) NOT NULL	Nome da categoria
TB_TP_PARCEIRO	Tabela	Tipo de parceiro
• ID_TP_PARCEIRO	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• DS_TP_PARCEIRO	VARCHAR(30) NOT NULL	Descrição de tipo de parceiro
TB_ATIVIDADE	Tabela	Atividades da extensão
• ID_ATIVIDADE	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• DS_TITULO	VARCHAR(100) NOT NULL	Título da atividade
• DT_INICIO	TIMESTAMP NOT NULL	Data de início
• DT_FIM	TIMESTAMP NOT NULL	Data de fim (DT_FIM > DT_INICIO)
• ID_TP_ATIVIDADE	INTEGER NOT NULL - FK → TB_TP_ATIVIDADE	Tipo da atividade
• DS_LOCAL	VARCHAR(100)	Local da atividade
TB_PARTICIPANTE	Tabela	Participantes
• ID_PARTICIPANTE	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• NM_PARTICIPANTE	VARCHAR(100) NOT NULL	Nome completo
• DS_EMAIL	VARCHAR(100) NOT NULL	E-mail

	UNIQUE	
• ID_TP_CATEGORIA	INTEGER NOT NULL - FK → TB_TP_CATEGORIA	Categoria
TB_INSTRUTOR	Tabela	Instrutores
• ID_INSTRUTOR	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• NM_INSTRUTOR	VARCHAR(100) NOT NULL	Nome
• DS_AFILIACAO	VARCHAR(100)	Afiliação
• DS_ESPECIALIDADE	VARCHAR(100)	Especialidade
TB_PARCEIRO	Tabela	Parceiros
• ID_PARCEIRO	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• NM_PARCEIRO	VARCHAR(100) NOT NULL	Nome do parceiro
• ID_TP_PARCEIRO	INTEGER NOT NULL - FK → TB_TP_PARCEIRO	Tipo do parceiro
• VL_CONTRIBUICAO	DECIMAL(10,2)	Valor da contribuição
TB_CERTIFICADO	Tabela	Certificados
• ID_CERTIFICADO	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• CD_HASH	VARCHAR(64) NOT NULL UNIQUE	Hash SHA-256 do certificado
• DT_EMISSAO	DATE NOT NULL	Data de emissão
• ID_ATIVIDADE	INTEGER NOT NULL - FK → TB_ATIVIDADE	Atividade
• ID_PARTICIPANTE	INTEGER NOT NULL - FK → TB_PARTICIPANTE	Participante
TP_NOTA	Tabela	Notas possíveis
• CD_NOTA	INTEGER PRIMARY KEY	Código da nota
• DS_NOTA	VARCHAR(20) NOT NULL UNIQUE	Descrição da nota
TB_FEEDBACK	Tabela	Feedbacks
• ID_FEEDBACK	INTEGER GENERATED ALWAYS AS IDENTITY	PK
• CD_NOTA	INTEGER NOT NULL - FK → TP_NOTA	Nota atribuída
• DS_COMENTARIO	TEXT	Comentário
• ID_ATIVIDADE	INTEGER NOT NULL - FK →	Atividade

	TB_ATIVIDADE	
• ID_PARTICIPANTE	INTEGER NOT NULL - FK → TB_PARTICIPANTE	Participante
RL_ATIVIDADE_PARTICIPANTE	Tabela	Relação atividade-participante
• ID_ATIVIDADE	INTEGER NOT NULL - FK → TB_ATIVIDADE	Atividade
• ID_PARTICIPANTE	INTEGER NOT NULL - FK → TB_PARTICIPANTE	Participante
• DT_INSCRICAO	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	Data de inscrição
• IS_PRESENCA	BOOLEAN DEFAULT FALSE	Presença confirmada
RL_ATIVIDADE_INSTRUTOR	Tabela	Relação atividade-instrutor
• ID_ATIVIDADE	INTEGER NOT NULL - FK → TB_ATIVIDADE	Atividade
• ID_INSTRUTOR	INTEGER NOT NULL - FK → TB_INSTRUTOR	Instrutor
• DS_PAPEL	VARCHAR(30)	Papel desempenhado
RL_ATIVIDADE_PARCEIRO	Tabela	Relação atividade-parceiro
• ID_ATIVIDADE	INTEGER NOT NULL - FK → TB_ATIVIDADE	Atividade
• ID_PARCEIRO	INTEGER NOT NULL - FK → TB_PARCEIRO	Parceiro
• VL_CONTRIBUICAO	DECIMAL(10,2)	Valor contribuição
TA_AUDITORIA_LOG	Tabela	Log de auditoria
• ID_LOG	SERIAL PRIMARY KEY	Identificador do log
• TX_ACAO	VARCHAR(50) CHECK ("INSERT"... "DELETE")	Tipo de operação
• TX_TABELA	VARCHAR(50)	Tabela afetada
• ID_REGISTRO	INTEGER	Registro alterado
• TX_VALOR_ANTIGO	JSONB	Valores antes da modificação
• TX_VALOR_NOVO	JSONB	Valores após a modificação
• DT_EVENTO	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	Data/hora do evento

• TX_USUARIO	VARCHAR(50)	Usuário executante
--------------	-------------	--------------------

8. Apêndices Técnicos

A. Glossário

- **LGPD**: Lei Geral de Proteção de Dados
- **Stored Procedure**: rotina SQL armazenada
- **Trigger**: gatilho automático
- **Materialized View**: view armazenada e atualizada periodicamente
- **MAD1 / PPP1 / PBR2**: políticas adotadas
- **Speedup**: ganho de performance
- **Transação**: sequência atômica com rollback/commit

B. Exemplo de Log de Auditoria

ID_LOG	TX_ACAO	TX_TABELA	ID_REGISTRO	TX_VALOR_ANTIGO	
101	UPDATE	TB_PARTICIPANTE	17	{"NM_PARTICIPANTE":"Ana"}	{"I B".
102	INSERT	TB_PARTICIPANTE	25	null	{"I

8. Detalhamento de Objetos Técnicos

8.1. Views Materializadas

View	Script	Objetivo	Frequência de Atualização
mv_atividade_por_tipo	04_Materialized_Views.sql	Quantifica atividades por tipo e calcula duração média	Diária
mv_parceiro_resumo	04_Materialized_Views.sql	Consolida o total de contribuições financeiras por parceiro	Semanal
mv_feedback_participante	04_Materialized_Views.sql	Calcula média de avaliações e quantidade de feedbacks por participante	Sob demanda
mv_instrutores_por_atividade	04_Materialized_Views.sql	Relaciona instrutores alocados em cada atividade	Diária

Script de Atualização:

05_REFRESH_VIEWS.sh (execução automática via cron)

8.2. Triggers de Auditoria

TG_A_IU_TB_PARTICIPANTE_LOG

Tabela Alvo: TB_PARTICIPANTE

Eventos:

- BEFORE INSERT OR UPDATE

Ações:

1. Registra operação em TA_AUDITORIA_LOG com:
 - Tipo de operação (INSERT/UPDATE)
 - Valores antigos/novos em JSONB
 - Timestamp automático
 - Usuário responsável

Políticas Aplicadas:

- PPP1 (Rastreabilidade)
 - MAD1 (Padronização de nomenclatura)
-

8.3. Stored Procedures

8.3.1. SP_INSERE_PARTICIPANTE_COM_INSCRICAO

Parâmetros: | Nome | Tipo | Obrigatório | Descrição | |-----|-----|-----|---
-----| | p_ds_email | VARCHAR(100) | Sim | Email único do participante |

Fluxo Transacional:

```
```sql BEGIN
```

- Verifica existência por e-mail
- Se novo: INSERT em TB\_PARTICIPANTE
- Valida e insere em RL\_ATIVIDADE\_PARTICIPANTE
- Atualiza MV\_FEEDBACK\_PARTICIPANTE COMMIT;