

Sprint 1 – Banco de Dados: Planejamento, Modelagem e Estrutura Inicial do Sistema

Período: 15/09 a 02/10

A Sprint 1 do grupo de Banco de Dados foi dedicada ao planejamento inicial, definição da arquitetura de dados e criação dos modelos necessários para sustentar as funcionalidades do sistema de gestão de notícias para condomínios. O foco desta etapa foi estabelecer a base conceitual e estrutural do banco, garantindo consistência, integridade, segurança e integração eficiente com o backend e as demais equipes.

1. Levantamento de Ferramentas e Tecnologias

Após análise dos requisitos funcionais e não funcionais fornecidos pelo grupo de Análise e pelo Product Owner, foram selecionadas as seguintes ferramentas e tecnologias:

- **Modelo de Dados:** Relacional
- **Ferramenta de Modelagem:** MySQL Workbench
- **Estratégia de Integração:** API REST (consumida pelo backend em FastAPI)
- **Controle de Versão:** GitHub (estrutura compartilhada com os demais grupos)
- **Sistema Gerenciador de Banco de Dados (SGBD):** MySQL

Bibliotecas e frameworks selecionados:

- **Normalização até 3FN:** para evitar redundância e garantir performance adequada.
- **Relacionamentos 1:N e N:N:** mapeados conforme regras de negócio.
- **Padrões de nomenclatura:** snake_case para tabelas e atributos.
- **Chaves Primárias e Estrangeiras:** para garantir integridade referencial.

Essas ferramentas e metodologias foram definidas visando clareza estrutural, segurança no armazenamento dos dados, escalabilidade e fácil integração com o backend.

2. Criação do minimundo

Nesta etapa, o grupo elaborou o Minimundo, documento narrativo que descreve em linguagem natural todos os elementos envolvidos no sistema, servindo como base para o modelo conceitual. Foram identificadas as principais entidades, seus relacionamentos e regras gerais do domínio:

- Definição de perfis (padrão, morador, administrador).
- Publicação e edição de notícias.
- Registro de comentários pelos moradores.
- Associação entre usuários e conteúdos publicados.
- Cadastro de usuários do condomínio.

3. Modelagem Conceitual – Modelo Entidade-Relacionamento

Com base no Minimundo, foi criado o Modelo Conceitual, representado graficamente em um Diagrama E-R, contendo:

Entidades Identificadas

- **Usuário**
- **Notícia**
- **Comentário**
- **Perfil de Acesso**

Relacionamentos

- Um *usuário* pode publicar várias *notícias*.
- Um *usuário* pode fazer vários *comentários*.
- Uma *notícia* pode ter vários *comentários*.
- Cada *usuário* possui exatamente um *perfil de acesso*.

Principais Atributos definidos

- Usuário: nome, email, senha, apartamento, bloco.
- Notícia: título, corpo, data_publicação.
- Comentário: conteúdo, data, id_usuario, id_noticia.
- Perfil: tipo (padrão, condômino, admin).

O modelo conceitual foi validado com as demais equipes antes de prosseguir para a modelagem lógica.

4. Modelagem Lógica – Diagrama UML de Banco de Dados

Nesta etapa, o grupo converteu o Modelo Conceitual em um modelo lógico normalizado:

- Definição de tabelas e atributos.
- Identificação de chaves primárias (PK) e chaves estrangeiras (FK).
- Aplicação das regras de normalização.
- Ajuste dos relacionamentos para estruturas relacionais.

5. Criação da Estrutura Física do Banco de Dados

Com o modelo lógico aprovado, iniciou-se a construção do **script SQL** contendo:

- Estrutura inicial das tabelas.
- Relacionamentos com FOREIGN KEY.
- Regras de exclusão e alteração como:

Essas regras foram utilizadas para manter integridade referencial, especialmente em casos como:

- Usuário removido → comentários permanecem, porém ficam sem autor.
- Atualização de IDs → propagação automática.

Sprint 2 – Banco de Dados: Implementação das Tabelas, Relacionamentos e Testes

Período: 02/10 a 13/10

A Sprint 2 do grupo de Banco de Dados foi dedicada à implementação prática do modelo lógico criado na Sprint anterior. O foco desta etapa foi transformar o diagrama UML em tabelas físicas no SGBD, definir chaves primárias e estrangeiras, estabelecer relacionamentos, testar restrições e validar o funcionamento do banco com dados reais. Nesta fase, todas as entidades planejadas foram oficialmente criadas e populadas, garantindo a base funcional.

1. Criação das Tabelas e Estruturação Inicial do Banco

Com base no modelo lógico aprovado, foram criadas todas as tabelas necessárias para o funcionamento do sistema. As atividades incluíram:

- Criação da tabela Usuário/Morador
- Criação da tabela Usuário_sistema
- Criação da tabela Perfis de Usuário
- Criação da tabela Notícias
- Criação da tabela para relacionamentos quando necessário
- Cada tabela foi criada seguindo padrões de nomenclatura, tipos de dados adequados e definição correta de PKs.

2. Implementação de Chaves Primárias e Chaves Estrangeiras

Nesta etapa, foram implementados os mecanismos de integridade referencial:

- Criação de PKs em todas as tabelas principais.
- Definição das FKs que conectam Usuário → Perfil, Notícia → Usuário, Comentários → Notícias e Usuários.
- Uso de constraints seguindo o padrão:

Essas regras garantiram:

- Persistência de dados mesmo quando um usuário é excluído.
- Propagação automática de atualizações em chaves referenciadas.
- Controle seguro sobre a integridade entre tabelas.

3. Junção e Relacionamento entre as Tabelas

Após criar as tabelas individualmente, a equipe realizou a **junção lógica**:

- Teste de queries com JOIN para validar a integridade.
- Consultas envolvendo Usuário ↔ Notícias ↔ Perfil.

- Verificação da coerência dos dados retornados.
- Checagem de cardinalidade 1:N e N:N conforme o modelo.

Esses testes foram essenciais para garantir que o banco estava preparado para atender a API que seria desenvolvida.

4. Popular o Banco de Dados (Testes)

Com as tabelas concluídas, iniciou-se o processo de popular o banco com dados simulados:

- Criação de usuários fictícios.
- Inserção de notícias de teste.
- Inserção de perfis.
- Testes de comentários e relacionamentos.

Esses dados foram utilizados para:

- Verificar consistência.
- Testar restrições.
- Validar joins complexos.
- Auxiliar o backend na construção da lógica de consulta.

5 Testes de Relacionamentos e Constraints

Durante a Sprint, foram realizados diversos testes focados em garantir o comportamento correto do banco:

- Testes de exclusão envolvendo FKs.
- Testes de integridade referencial.
- Testes com dados inválidos para verificar se as constraints impediam erros.
- Ensaios com UPDATE, DELETE, SET NULL e CASCADE.

Resultados observados:

- Todas as constraints funcionaram conforme planejado.
- Nenhum erro crítico foi encontrado.
- Pequenos ajustes de tipo e estrutura foram feitos para otimizar certas consultas.

6. Conferência e Revisão do Código SQL

Ao final da sprint, toda a estrutura foi revisada coletivamente:

- Identificação de possíveis inconsistências.
- Ajustes de sintaxe.
- Padronização de nomes de tabelas e colunas.

- Melhoria da legibilidade do script SQL.

Essa revisão garantiu que o banco estivesse limpo, seguro e pronto para uso.

7. Criação das Tabelas do Sistema

Com base nas tarefas listadas no Trello, as seguintes tabelas foram construídas:

1.1 Tabela de Notícias

- Criação da estrutura com campos como: título, conteúdo, autor, data de publicação e status.
- Definição do ID como chave primária.
- Ajustes nos tipos de dados conforme necessidade do sistema.

1.2 Tabela de Usuários/Admin

- Estrutura destinada ao controle de administradores do sistema.
- Campos definidos: nome, e-mail, senha, permissões e status.
- Chave primária configurada (ID).

1.3 Tabela de Moradores/Usuários

- Criação da tabela destinada aos moradores do condomínio.
- Campos como: nome, apartamento, bloco, telefone, e-mail etc.
- Definição da chave primária.

8. Definição de Chaves Primárias e Relacionamentos

Durante a Sprint 2, foram aplicadas:

- Chaves primárias em todas as tabelas criadas.
- Preparação do ambiente para inclusão de chaves estrangeiras (FK) em sprints futuras, caso relacionamentos entre entidades fossem necessários.

9. Ajustes e Classificação dos Tipos de Dados

As tarefas da sprint incluíram:

- Determinar e ajustar o tipo adequado para cada campo conforme o uso esperado.
- Uso de tipos como INT, VARCHAR, TEXT, DATE, TIMESTAMP, entre outros.
- Garantia de compatibilidade com as operações do backend.

10. Estratégias e Configurações de Backup

As seguintes definições foram tratadas e concluídas durante esta sprint:

- Seleção do tipo de backup utilizado (full, incremental ou diferencial).
- Escolha da ferramenta de backup mais adequada ao ambiente.
- Registro do processo padrão de backup para manutenções futuras.

11. Ferramenta Utilizada para Criação do Banco

Nesta sprint foi formalizada a ferramenta utilizada para criar e gerenciar as tabelas, conforme tarefa listada no Trello:

- Ferramenta selecionada: *(MySQL Workbench / DBeaver / outra – posso ajustar se quiser)*
- Padronização do uso da ferramenta por toda a equipe.
- Integração com o GitHub para versionamento dos scripts SQL.

Sprint 3 – Banco de Dados: Prototipação, Relacionamentos e Testes

Período: 13/10 a 30/10

A Sprint 3 da equipe de Banco de Dados teve como foco principal a **validação prática do banco**, a construção de protótipos funcionais, a implementação de consultas, os testes iniciais do fluxo de informações e o alinhamento com outras áreas do projeto.

Nesta etapa, o sistema começou a ganhar forma funcional, permitindo visualizar tabelas, testar relacionamentos e garantir que futuras integrações ocorreriam de forma estável e coerente.

1. Visualização e Validação da Estrutura do Banco

As primeiras tarefas da sprint envolveram:

1.1 Visualização do Banco Pronto

- Revisão da estrutura criada nas sprints anteriores.
- Conferência das tabelas, tipos de dados e organização interna.
- Ajustes finos quando necessário.

1.2 Visualização das Tabelas

- Conferência individual das tabelas implementadas.
- Verificação de campos, chaves primárias e relacionamentos iniciais.

2. Protótipos Funcionais de Segurança e Backup

Durante a Sprint 3, foram desenvolvidos e testados protótipos essenciais para garantir segurança, integridade e recuperação das informações do sistema:

2.1 Protótipo de Login e Senha

- Criação do fluxo inicial de autenticação.
- Teste com dados fictícios do banco.
- Preparação para integração com o Backend.

2.2 Protótipo de Backup Incremental

- Estruturação de um modelo simples de backup incremental.
- Registro das alterações ocorridas no banco para validação futura.

2.3 Protótipo de Backup Completo

- Execução de um protótipo de backup completo.
 - Validação da restauração do banco.
 - Comparação entre o backup incremental e completo.
-

3. Relacionamentos e Dependências

3.1 Apresentação dos Relacionamentos entre Tabelas

- Identificação de entidades que se relacionam diretamente.
- Esclarecimento de potenciais vínculos (1:N, N:N).
- Atualização da documentação interna com base nesses relacionamentos.

3.2 Alinhamento com Outras Áreas

- Comunicação com Backend, Frontend e Segurança.
- Ajustes na modelagem conforme necessidades identificadas por outras equipes.
- Sincronização de expectativas antes das próximas sprints.

4. Consultas e Fluxo Inicial de Dados

4.1 Implementar Consultas de Dados

- Criação de consultas SQL básicas e intermediárias.
- Teste de filtros, buscas e retornos específicos.
- Verificação da performance das consultas.

4.2 Testar Rotas de Informações

- Simulação do fluxo de informações entre banco e demais camadas.
- Testes com endpoints simulados ou reais.
- Confirmação de que consultas e dados retornam corretamente.

5. Verificações e Correções

5.1 Registrar Falhas e Aplicar Correções

- Registro detalhado dos erros identificados durante os testes.
- Correção imediata de falhas estruturais, de dados ou de consulta.
- Atualização da documentação técnica.

Sprint 4 – Banco de Dados: Testes Finais, Documentação e Integração

Período: 15/11 a 03/12

1. Documentação da Configuração do Banco

Nesta etapa, a equipe realizou:

- Registro detalhado da configuração utilizada no banco de dados.
- Documentação das ferramentas, versões, estrutura e padrões aplicados.
- Registro das configurações de segurança, permissões e padrões de acesso.

Essa documentação garante que qualquer membro da equipe possa replicar ou auditar o banco em instâncias futuras.

2. Testes de Integração

2.1 Testar Funcionamento Após Integração

- Execução de testes após integração com Backend e demais áreas.
- Verificação de rotas, consultas e operações realizadas pelo sistema.
- Identificação e resolução de inconsistências entre camadas.

Esse processo assegurou que o banco respondia corretamente às solicitações externas.

3. Documentação das Etapas do Processo

A equipe registrou:

- Todo o fluxo de construção do banco realizado desde a Sprint 1.
- Decisões importantes tomadas durante o desenvolvimento.
- Mudanças estruturais e justificativas técnicas.

Essa documentação auxilia em auditorias, provas acadêmicas e continuidade do projeto.

4. Testes de Backup e Recuperação

4.1 Testar Recuperação do Backup

- Teste completo do processo de restauração do banco.
- Validação de dados restaurados com base nos backups full e incrementais.
- Garantia de que o banco poderia ser recuperado em caso de falhas.

5. Testes de Armazenamento de Dados

Nessa tarefa, foram executados:

- Testes de inserção, leitura, atualização e exclusão de dados (CRUD).
- Avaliação de performance e tempo de resposta.
- Testes com diferentes volumes de dados para verificar estabilidade.

Esses testes confirmaram que o banco está funcional e capaz de lidar com a demanda esperada do sistema.

6. Entregável Complementar – Relatório de Requisitos

Uma tarefa adicional incluída foi:

6.1 Elaborar Relatório de Requisitos

- Compilação dos requisitos relacionados ao banco de dados.
- A ligação entre requisitos e tabelas, campos e funcionalidades.
- Uso do relatório como base para apresentações e métricas de entrega.