SUMÁRIO

DESCRIÇÃO DO PROJETO

A Hamm API (nomeada carinhosamente em homenagem ao porquinho-cofrinho do filme *Toy Story*) é uma aplicação desenvolvida em ASP.NET Core Web API com Entity Framework Core, cujo principal objetivo é atuar como o motor de um sistema de gestão financeira pessoal.

Ela é responsável por centralizar e disponibilizar, de forma segura e organizada, todas as informações relacionadas a:

- Gastos pessoais
- Metas financeiras
- Budgets mensais
- Relatórios consolidados

A API funciona como o ponto central de comunicação entre um aplicativo cliente (por exemplo, um app mobile) e o banco de dados, garantindo que todas as operações de cadastro, consulta, atualização e exclusão de dados sejam realizadas de forma consistente.

Além disso, a Hamm API implementa a lógica de negócios necessária para:

- Processar e integrar dados financeiros.
- Gerar relatórios (JSON).
- Executar consultas avançadas utilizando LINQ.
- Integrar informações externas, como taxas de câmbio em tempo real, por meio de APIs públicas.

Com isso, o projeto busca oferecer uma solução modular e extensível, capaz de evoluir futuramente para um sistema completo de gestão financeira pessoal com interface mobile ou web, mantendo a API como núcleo central.

ANÁLISE DE REQUISITOS

Requisitos Funcionais

RF01 – Gerenciar Usuários (CRUD Usuários)

- A API deve permitir cadastrar, editar, listar e excluir usuários.
- A API deve armazenar dados básicos do usuário (nome, e-mail, senha).

RF02 – Gerenciar Transações (CRUD Transações)

- A API deve permitir o cadastro, edição, listagem e exclusão de transações financeiras (entrada e saída).
- Cada transação deve estar associada à um usuário e conter: valor, data, categoria, descrição, tipo (receita ou despesa) e forma de pagamento.

RF03 – Gerenciar Categorias de Despesas/Receitas

- A API deve permitir cadastrar e consultar categorias (ex.: alimentação, transporte, lazer, contas, investimentos, metas, etc).
- As categorias devem ser utilizadas para classificar transações.

RF04 – Gerenciar Orçamentos Mensais (Budgets)

- A API deve permitir definir um orçamento mensal por categoria de despesa.
- A API deve permitir consultar quanto do orçamento já foi utilizado e quanto ainda está disponível para cada categoria e no total.

RF05 – Gerenciar Metas Financeiras

- A API deve permitir cadastrar, editar, listar e excluir metas financeiras (ex.: juntar R$ 5.000 para uma viagem).
- A API deve calcular o progresso das metas com base nas contribuições feitas.

RF06 – Relatórios Financeiros

- A API deve permitir gerar relatórios consolidados dos dados de determinado mês ou determinado ano em .json.
- A API deve gerar estatísticas, como:
    - Total de receitas e despesas no mês ou ano fornecido.
    - Orcamento total do período (somando todos os orçamentos definidos para aquele período).
    - Top 5 maiores despesas e Top 5 maiores receitas (com base no Tipo das transações).
    - Gasto por categoria.
    - Status e detalhes de todas as metas com data alvo futuro

RF08 – Consultar Taxas de Câmbio (Integração Externa)

- A API deve integrar-se com uma API externa de câmbio para obter taxas de conversão de moedas (API escolhida: https://br.dolarapi.com/v1/cotacoes).
- O usuário deve poder consultar quanto ele poderia comprar de certa moeda com determinado valor em reais (as moedas disponíveis são: ARS, USD, EUR, CLP, UYU)

Regras de Negócio

RN01 – Usuários

- Cada usuário deve ter **e-mail único**; não é permitido cadastrar dois usuários com o mesmo e-mail.
- Senhas devem ser armazenadas de forma **criptografada**.
- Um usuário só pode acessar e manipular suas próprias transações, categorias, budgets e metas.

RN02 – Transações

- Transações **não podem ter valor negativo** para receitas ou despesas; apenas o tipo define se é entrada ou saída.
- A data da transação não pode ser futura (não é possível registrar transações em datas que ainda não ocorreram).
- Cada transação deve estar vinculada a uma **categoria existente**.
- O tipo de pagamento (dinheiro, cartão, transferência) deve ser consistente e validado.

RN03 – Categorias

- Categorias padrão (alimentação, transporte, lazer, contas, investimentos) já podem vir pré-cadastradas na API.
- Usuários podem criar categorias, mas não podem alterar ou excluir categorias padrão do sistema.

RN04 – Orçamentos Mensais (Budgets)

- O orçamento definido para uma categoria **não pode ser negativo**.

RN06 – Relatórios

- Relatórios gerados devem refletir apenas as transações **pertencentes ao usuário** que solicitou.
- Estatísticas (top 5 despesas, saldo consolidado, gasto por categoria) devem ser calculadas **em tempo real** com base nos dados atuais do banco.

RN08 – Exportação de Arquivos

- Usuário pode escolher período ou categorias para exportação.
- Arquivos exportados devem incluir **todas as informações relevantes**: valor, data, categoria, tipo, forma de pagamento.

RN09 – Consultar Taxas de Câmbio

- A conversão deve usar a **taxa mais recente disponível** na API externa.

DESIGN DO SISTEMA

Diagrama de componentes

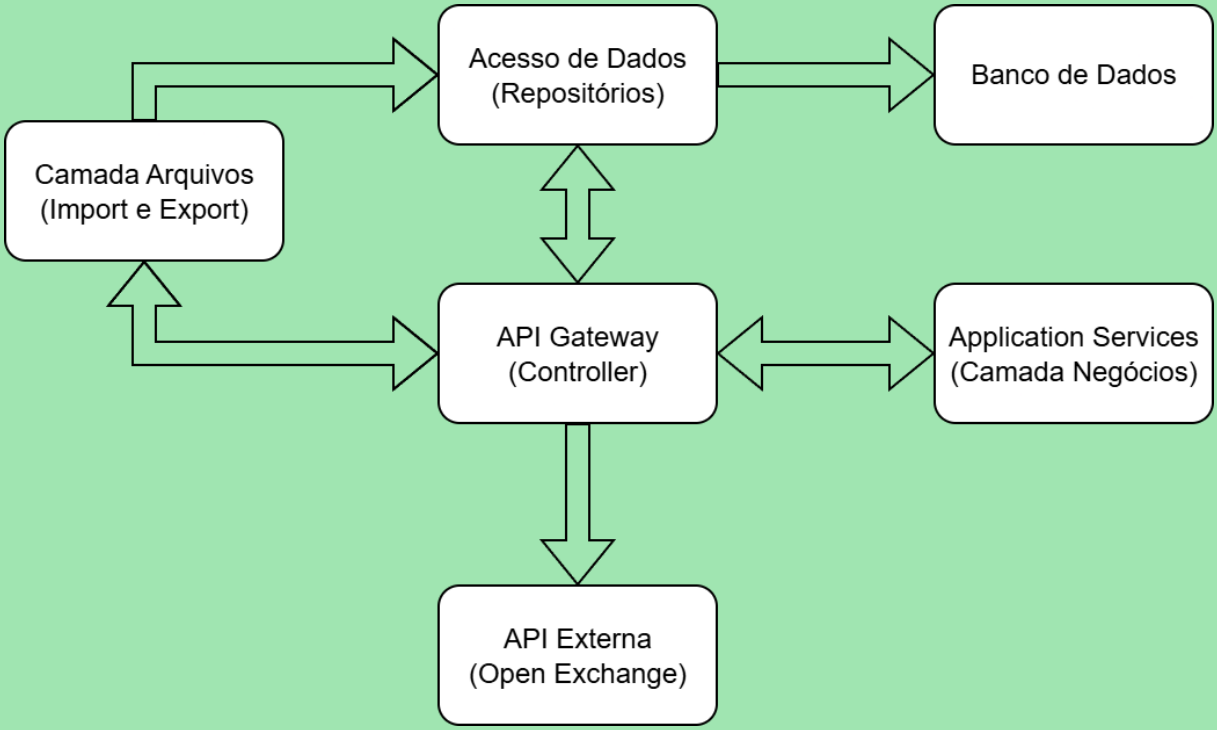
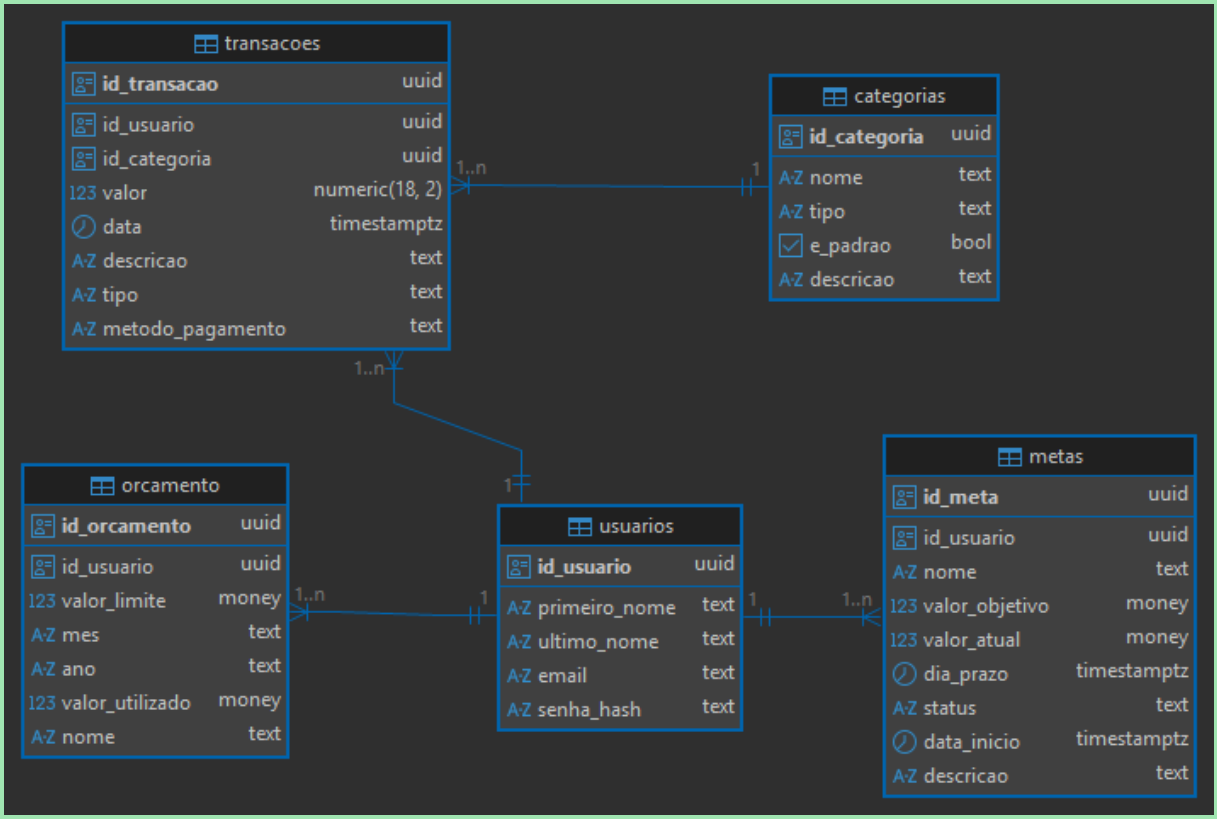


Diagrama de Entidades (Banco de Dados)



API ENDPOINTS

Class	Method	HTTP request	Description
AuthApi	**apiAuthLoginPost**	**POST** /api/Auth/login	Realiza o login de um usuário existente e gera um token JWT.
AuthApi	**apiAuthRegisterPost**	**POST** /api/Auth/register	Registra um novo usuário no sistema.
CambiosApi	**apiCambiosConverterMoedaGet**	**GET** /api/Cambios/converter/{moeda}	Converte um valor em reais para a moeda especificada
CambiosApi	**apiCambiosCotacaoMoedaGet**	**GET** /api/Cambios/cotacao/{moeda}	Consulta a cotação atual de uma moeda (ARS, USD, EUR, CLP, UYU)
CategoriasApi	**apiCategoriasGet**	**GET** /api/Categorias	Retorna todas as categorias cadastradas.
CategoriasApi	**apiCategoriasIdDelete**	**DELETE** /api/Categorias/{id}	Remove uma categoria existente.
CategoriasApi	**apiCategoriasIdGet**	**GET** /api/Categorias/{id}	Retorna uma categoria específica pelo seu identificador único.
CategoriasApi	**apiCategoriasIdPut**	**PUT** /api/Categorias/{id}	Atualiza os dados de uma categoria existente.
CategoriasApi	**apiCategoriasPost**	**POST** /api/Categorias	Cria uma nova categoria.
MetasApi	**apiMetasGet**	**GET** /api/Metas	Retorna todas as metas cadastradas.
MetasApi	**apiMetasIdDelete**	**DELETE** /api/Metas/{id}	Remove uma meta existente.
MetasApi	**apiMetasIdGet**	**GET** /api/Metas/{id}	Retorna uma meta específica pelo seu identificador único.
MetasApi	**apiMetasIdPut**	**PUT** /api/Metas/{id}	Atualiza os dados de uma meta existente.

Class	Method	HTTP request	Description
MetasApi	**apiMetasPost**	**POST** /api/Metas	Cria uma nova meta vinculada a um usuário.
OrcamentosApi	**apiOrcamentosGet**	**GET** /api/Orcamentos	Retorna todos os orçamentos cadastrados.
OrcamentosApi	**apiOrcamentosIdDelete**	**DELETE** /api/Orcamentos/{id}	Remove um orçamento pelo ID.
OrcamentosApi	**apiOrcamentosIdGet**	**GET** /api/Orcamentos/{id}	Retorna um orçamento específico pelo seu ID.
OrcamentosApi	**apiOrcamentosIdPut**	**PUT** /api/Orcamentos/{id}	Atualiza os dados de um orçamento existente.
OrcamentosApi	**apiOrcamentosPost**	**POST** /api/Orcamentos	Cria um novo orçamento.
RelatoriosApi	**apiRelatoriosDownloadGet**	**GET** /api/Relatorios/download	Gera e baixa o relatório financeiro como arquivo JSON.
RelatoriosApi	**apiRelatoriosGet**	**GET** /api/Relatorios	Gera um relatório financeiro para o usuário informado.
TransacoesApi	**apiTransacoesGet**	**GET** /api/Transacoes	Retorna todas as transações do usuário autenticado, com paginação.
TransacoesApi	**apiTransacoesIdDelete**	**DELETE** /api/Transacoes/{id}	Remove uma transação do usuário autenticado.
TransacoesApi	**apiTransacoesIdGet**	**GET** /api/Transacoes/{id}	Retorna uma transação específica do usuário autenticado.
TransacoesApi	**apiTransacoesIdPut**	**PUT** /api/Transacoes/{id}	Atualiza uma transação do usuário autenticado.

Class	Method	HTTP request	Description
TransacoesApi	**apiTransacoesPost**	**POST** /api/Transacoes	Cria uma nova transação para o usuário autenticado.
UsuariosApi	**apiUsuariosMeDelete**	**DELETE** /api/Usuarios/me	Exclui a própria conta.
UsuariosApi	**apiUsuariosMeGet**	**GET** /api/Usuarios/me	Retorna o perfil do usuário autenticado.
UsuariosApi	**apiUsuariosMePut**	**PUT** /api/Usuarios/me	Atualiza o perfil do usuário autenticado.

Bearer

- **Type**: HTTP Bearer Token authentication (JWT)