



## **GRUPO 07**

### **INTEGRANTES**

**BRUNA LUIZA DA SILVA GONÇALVES**  
**DANIEL ANDRADE**  
**JULIANA RODRIGUES MATOS**  
**KEVEN DOS SANTOS COSTA**  
**LUCAS DE OLIVEIRA MAFRA**  
**NAYANE RODRIGUES MATOS SANTOS OI**

### **PROJETO INTEGRADOR** **ESCOPO DO PROJETO**

**TEMA: SEGURO DE VIDAS**



## **GRUPO 07**

### **INTEGRANTES**

BRUNA LUIZA DA SILVA GONÇALVES  
DANIEL ANDRADE  
JULIANA RODRIGUES MATOS  
KEVEN DOS SANTOS COSTA  
LUCAS DE OLIVEIRA MAFRA  
NAYANE RODRIGUES MATOS SANTOS OI

## **PROJETO INTEGRADOR ESCOPO DO PROJETO**

**TEMA: SEGURO DE VIDAS**

Relatório solicitado pela Generation Brasil para compor o projeto final. O relatório refere-se ao escopo do projeto integrador.

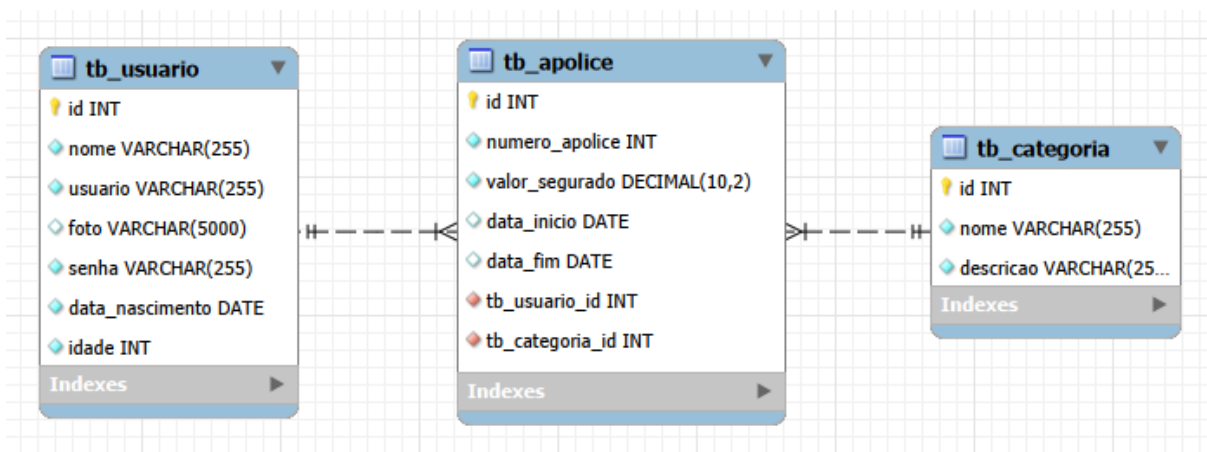
# 1 Título do Projeto e Modelo de Negócio escolhido

GenLife - Sistema de cadastro de RH.

## 2 Descrição Geral do Projeto

O projeto tem como objetivo desenvolver um sistema de cadastro de RH, focado em empresas de pequeno e médio porte para gerenciar usuários e suas apólices de seguro de vida.

## 3 Descrição da Entidade/Model foi criada e seus atributos



O sistema é composto por três entidades principais mapeadas na base de dados:

### 3.1 Usuário (**tb\_usuarios**)

Representa os funcionários ou clientes cadastrados no sistema.

**Campos:**

- **id:** Identificador único (Chave Primária).
- **nome:** Nome completo (Não nulo, 255 caracteres).
- **usuario:** Nome de utilizador/email (Único, Não nulo).
- **foto:** URL da foto (Opcional, 5000 caracteres).
- **senha:** Palavra-passe encriptada (Não nulo).
- **data\_nascimento:** Data de nascimento (Tipo Date).
- **idade:** Idade calculada (Numérico).
- **Relacionamentos:**
  - **OneToMany:** Um utilizador pode ter várias apólices (**apolice**).

### 4.2. Categoria (**tb\_categorias**)

Define os tipos de seguros ou categorias de classificação.

## Campos:

- **id:** Identificador único (Chave Primária).
- **nome:** Nome da categoria (Único, Não nulo).
- **descricao:** Descrição detalhada da categoria (Não nulo).
- **Relacionamentos:**
  - **OneToMany:** Uma categoria pode estar associada a várias apólices ([apólice](#)).

### 4.3. Apólice ([tb\\_apólice](#))

Representa o contrato de seguro vinculado a um utilizador e a uma categoria.

#### 4 Campos:

- **id:** Identificador único (Chave Primária).
- **numero\_apólice:** Número identificador da apólice (BigInt, Não nulo).
- **valor\_segurado:** Valor monetário do seguro (Decimal 15,2, Não nulo).
- **data\_inicio:** Data de início da vigência (Atualizada automaticamente).
- **data\_fim:** Data de término da vigência (Opcional).
- **Relacionamentos:**
  - **ManyToOne:** Pertence a uma [Categoria](#) (com eliminação em cascata).
  - **ManyToOne:** Pertence a um [Usuario](#) (com eliminação em cascata).

## 5 Funcionalidades Principais (CRUD) implementadas

O sistema backend disponibiliza uma API RESTful completa com operações de Criação (Create), Leitura (Read), Atualização (Update) e Exclusão (Delete) para as três entidades principais do negócio: Usuários, Categorias e Apólices.

### 4.1. Gestão de Usuários (CRUD de Usuário)

Responsável por gerir o cadastro de funcionários ou clientes do sistema.

**Criar Usuário (POST):** Permite o cadastro de um novo utilizador no sistema. E como regra de negócio ele verifica automaticamente se o utilizador é maior de 18 anos antes de permitir o cadastro.

**Listar Todos (GET):** Retorna uma lista completa de todos os utilizadores cadastrados, incluindo as apólices associadas a cada um.

**Consultar por ID (GET):** Permite a busca detalhada de um utilizador específico através do seu identificador único (id).

**Atualizar (PUT):** Permite a alteração dos dados cadastrais de um utilizador existente.

**Apagar (DELETE):** Remove um utilizador da base de dados.

## 4.2. Gestão de Categorias (CRUD de Categoria)

Responsável pela classificação dos tipos de seguros (ex: Seguro de Vida, Seguro Acidentes Pessoais).

**Criar Categoria (POST):** Cadastrar um novo tipo de categoria de seguro no sistema.

**Listar Todas (GET):** Exibe todas as categorias disponíveis para vincular às apólices.

**Consultar por ID (GET):** Busca os detalhes de uma categoria específica.

**Atualizar (PUT):** Altera o nome ou descrição de uma categoria existente.

**Apagar (DELETE):** Exclui uma categoria do sistema.

## 4.3. Gestão de Apólices (CRUD de Apólice)

Responsável pela administração dos contratos de seguro vinculados aos utilizadores.

**Criar Apólice (POST):** Gera um novo contrato de seguro. E como regra de negócio: ele valida se o Utilizador e a Categoria informados existem antes de criar a apólice.

**Listar Todas (GET):** Retorna todas as apólices registradas, trazendo os dados completos do utilizador titular e da categoria do seguro.

**Consultar por ID (GET):** Busca uma apólice específica pelo seu identificador interno.

**Consultar por Número da Apólice (GET):** Funcionalidade adicional que permite buscar contratos pelo número oficial da apólice (número apólice).

**Atualizar (PUT):** Atualiza informações do contrato, como valor segurado ou datas de vigência.

**Apagar (DELETE):** Remove uma apólice do sistema.

## 5 Tecnologias Utilizadas (banco de dados e backend)

O backend foi desenvolvido utilizando as seguintes tecnologias e bibliotecas principais, conforme definido no ficheiro package.json:

- **Framework:** NestJS (@nestjs/core, @nestjs/common).
- **Linguagem:** TypeScript.
- **ORM:** TypeORM.
- **Ambiente de Execução:** Node.js
- **Banco de dados:** MySQL
- **Validação:** class-validator e class-transformer.
- **Ferramenta de teste:** Insomnia