

Visão Geral do Sistema - Pop-ups de Avisos e Gestão de Interesses

1. Visão Geral

O objetivo deste sistema é disponibilizar um mecanismo de avisos em pop-up numa aplicação universitária, totalmente configurável através de um backoffice, permitindo aos administradores criar e gerir conteúdos, enquanto os alunos recebem informação personalizada com base nos seus interesses.

O sistema permite:

- Criar avisos pop-up configuráveis (texto, categoria, prioridade, duração e limites de interação).
- Apresentar um pop-up específico para seleção de interesses.
- Guardar as preferências de cada aluno para reutilização futura.
- Controlar quando os pop-ups aparecem, evitando repetições constantes.
- Garantir que toda a lógica e configuração são feitas no backend.

O frontend é responsável apenas por apresentar a interface de forma dinâmica com base nos dados fornecidos pelo backend.

2. Fluxo de Utilização (Exemplo Realista)

No início do semestre, os administradores accedem ao backoffice para configurar o sistema. Começam por definir quais as categorias de conteúdo existentes, como Erasmus, Eventos, Formações ou Notícias, ativando apenas as que fazem sentido naquele período.

De seguida, configuram um pop-up global de seleção de interesses, definindo um intervalo de datas durante o qual os alunos devem escolher ou confirmar os seus interesses. Desta forma, o pop-up não aparece em todos os logins, mas apenas quando necessário.

Os administradores criam também avisos pop-up normais, definindo o título, a descrição, a categoria, a prioridade, o texto do botão, a duração e o número máximo de vezes que cada aluno pode dispensar o aviso.

Quando um aluno abre a aplicação, o backend começa por verificar se o pop-up de seleção de interesses deve ser apresentado, tendo em conta as datas configuradas e se o aluno já respondeu anteriormente. Caso seja necessário, o frontend constrói dinamicamente o pop-up com base na lista de categorias ativas fornecida pelo backend.

Após o aluno selecionar os seus interesses, estas escolhas ficam guardadas no backend e passam a ser utilizadas para filtrar os avisos apresentados. Sempre que o aluno volta à aplicação ou acede à página de definições, as preferências anteriormente escolhidas são carregadas automaticamente, permitindo alterações.

Em paralelo, o backend controla a apresentação dos avisos normais, garantindo que apenas aparecem avisos ativos, dentro do período definido, relevantes para os interesses do aluno e que ainda não tenham ultrapassado o limite de interações configurado.

3. Coleções MongoDB e o Seu Propósito

Para suportar todo este funcionamento, o sistema utiliza cinco coleções MongoDB, cada uma com uma responsabilidade bem definida.

3.1 PopupNotifications

Esta coleção armazena todos os avisos pop-up criados no backoffice.

Cada documento representa um aviso que pode aparecer na aplicação.

Propósito:

- Guardar o conteúdo e a configuração dos avisos.
- Controlar a duração e a prioridade.
- Definir limites de interação por utilizador.

Campos principais e significado:

- **titulo**: Título apresentado no pop-up.
- **descricao**: Texto principal do aviso.
- **categoria**: Código da categoria associada ao aviso (**ligacao com InterestTypes**).

- **prioridade**: Valor numérico que define a cor e importância do aviso.
 - **texto_botao**: Texto do botão do pop-up (ex.: “Entendido”).
 - **max_dismiss_count**: Número máximo de vezes que o aviso pode ser dispensado por um aluno (**ligacao com UserPopUpState**).
 - **data_inicio**: Data a partir da qual o aviso é apresentado.
 - **duracao_dias**: Número de dias em que o aviso permanece ativo.
 - **data_fim**: Data final calculada automaticamente / opcional
 - **ativo**: Indica se o aviso está ativo.
 - **timestamps**: Datas de criação e atualização / opcional
-

3.2 InterestTypes

Esta coleção define os tipos de conteúdo disponíveis na aplicação.

Cada documento representa uma categoria que pode ser selecionada pelos alunos.

Propósito:

- Centralizar a definição das categorias.
- Evitar valores fixos no frontend.
- Permitir gestão dinâmica através do backoffice.

Campos principais e significado:

- **codigo**: Identificador único da categoria (ex.: “erasmus”).
 - **nome**: Nome apresentado ao utilizador.
 - **ativo**: Indica se a categoria está disponível.
 - **timestamps**: Datas de criação e atualização / opcional
-

3.3 UserPreferences

Esta coleção guarda as preferências de cada aluno.

Cada documento corresponde a um utilizador específico.

Propósito:

- Armazenar os interesses selecionados pelo aluno.
- Permitir reutilizar escolhas anteriores.
- Facilitar a edição das preferências nas definições.

Campos principais e significado:

- `id_utilizador`: Identificador único do aluno (obrigatório).
 - `interesses`: Lista de categorias escolhidas pelo aluno (array e só categorias selecionadas).(**ligacao com InterestTypes**)
 - `last_interest_popup_id`: Guarda o ID da campanha ativa (marca que o aluno respondeu).
 - `timestamps`: Datas de criação e atualização / opcional
-

3.4 InterestPopupConfig

Esta coleção controla quando o pop-up de seleção de interesses deve aparecer.

Cada documento representa uma configuração global.

Propósito:

- Definir períodos específicos para apresentação do pop-up.
- Evitar que o pop-up apareça em todos os logins.
- Permitir controlo total via backoffice.

Campos principais e significado:

- `_id`: Identificador único da campanha do pop-up de interesses (**ligacao com UserPreferences**).
 - `ativo`: Indica se a configuração está ativa.
 - `data_inicio`: Data de início da apresentação do pop-up.
 - `data_fim`: Data de fim da apresentação do pop-up.
 - `roles`: Define o público-alvo (ex.: alunos).
 - `timestamps`: Datas de criação e atualização / opcional
-

3.5 UserPopupState

Esta coleção guarda o estado de interação de cada aluno com cada aviso.

Cada documento representa a interação de um utilizador com um aviso específico.

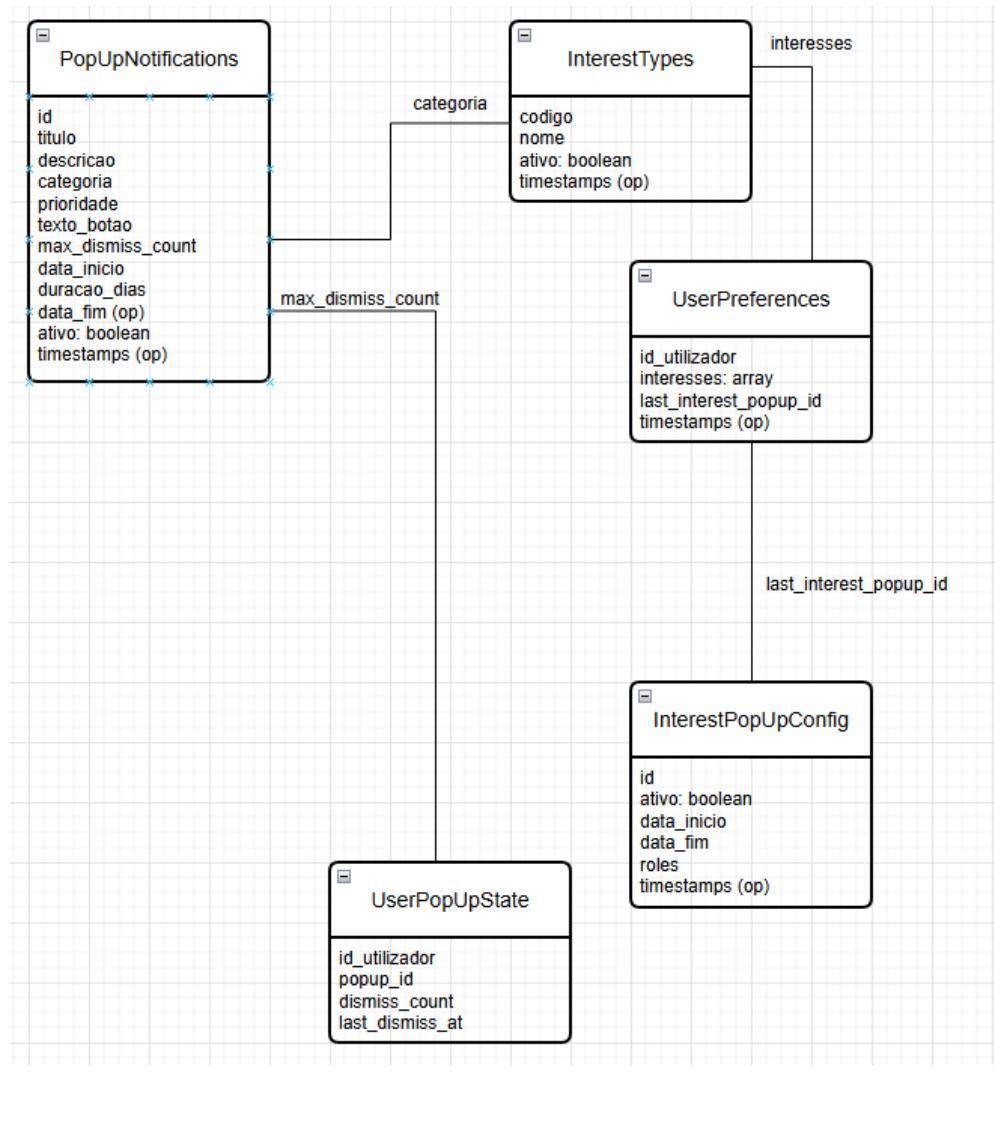
Propósito:

- Controlar o número de vezes que um aviso foi dispensado.
- Impedir a repetição excessiva de avisos.
- Manter histórico de interações.

Campos principais e significado:

- `id_utilizador`: Identificador do aluno.
 - `popup_id`: Identificador do aviso.
 - `dismiss_count`: Número de vezes que o aviso foi dispensado.
 - `last_dismiss_at`: Data da última interação.
-

4. Arquitetura



5. Resumo Final

Este sistema combina avisos pop-up configuráveis, seleção dinâmica de interesses, personalização de conteúdos e controlo de interações por utilizador. A separação clara das responsabilidades entre as diferentes coleções MongoDB garante um funcionamento robusto, extensível e alinhado com os requisitos do projeto.