Menus nas paginas

Originalmente os menus eram mais simples, mas também menus eficazes, sendo que nas primeiras versões eram compostos unicamente por botões que serviam de *links*.

Tal estrutura levantou um problema muito cedo e notável, o espaço ocupado por tantos botões. Usando os botões só para incluir os *links* já estabelecidos, cada menu ocuparia três vezes o espaço vertical atualmente ocupado pelo menu.

Uma possibilidade considerada para estes menus foi a utilização do *rich widget* Accordion, que também permite fazer algo semelhante. Mas devido algumas limitações visuais, o Ballon foi escolhido sobre o Accordion.

Tabs

Originalmente, antes de se descobrir a existência deste rich widget, foram consideradas duas possibilidades, ambas com limitações notáveis.

A primeira era incluir nas páginas com muita informação um menu extra, no topo da página, com vários botões que demonstram a parte da página desejada. Depois considerou-se passar esse menu para um ponto mais abaixo e usar *ballon* para esconder os botões.

Ambas possibilidades envolviam algum código extra da parte da aplicação, para esconder a parte das páginas que não se quer visualizar era necessário utilizar a característica Visible, associando a mesma a uma variável que era alterada com o pressionar dos botões já referidos.

Esta alteração seria sempre acompanhada por um *ajax refresh* que escondia as partes da página não desejadas e demonstra a parte da página desejada.

Então estas formas de visualização envolviam sempre incluir uma variável extra a página, como também uma Screen Action, elementos desnecessários com a utilização da Tabs.

List\_Navigation

Deve ser notado que a List\_Navigation não inclui em si a capacidade de alterar a tabela, para tal é necessário associar a mesma uma Screen Action. Esta deve obter a informação da tabela, normalmente realizando uma Resfesh Data sobre o *aggreate* que obteve a informação da tabela, seguindo de um Ajax Refresh sobre a tabela em si.

Esta forma de navegação não foi a primeira utilizada, sendo que originalmente foi utilizado dois botões um para obter a página seguinte e outra a página anterior.

Estes botões envolviam um quanto código extra da parte da aplicação, primeiro era utilizado quatro variáveis, o *current* que representava a página da lista verificada, nRows o numero de elemento por cada página da lista, needsNext e needsPrev que demonstram a necessidade do botão Next e o botão Prev.

As duas ultimas variáveis seriam utilizadas para definir as características de visibilidade e habilitar dos botões.

Para verificar a necessidade do Prev era bastante simples, sendo que desde que *current* fosse maior que zero então havia pelo menos uma página anterior a mostrar.

Já a necessidade de Next era mais complexo, primeiro a informação para a listagem teria de ocorrer com um SQLQuery em vez de um Aggregate (como é feito atualmente), e nesta *query* era obtida nRows mais 1 instancia da entidade.

Se o resultado dessa *query* for um numero de instancias igual a nRows mais 1 então é necessário o Next, para no mínimo obter a ultima instancia.

Com a utilização do List\_Navigation estas variáveis são desnecessárias e o *query* pode ser substituído por um Aggregate, que em si é otimizado, dai ter sido escolhida sobre o sistema dos botões Prev e Next.

Outro fator desta escolha é o facto que o List\_Navigation permite vários botões numerados, não sendo restinguido aos botões Next e Previous, que também inclui.

Create e Cancel

Deve ser notado que estas páginas partilham uma logica, a ideia que até o botão “Create” ser pressionado a instancia não é criada, nem as instancias relacionadas.

Esta lógica tem uma vantagem principal, ao não criar as instancias até ao momento desejado nunca são criadas instancias desnecessárias que são removidas de seguida. Também garante que se um utilizador iniciar o processo de criação, mas não o completar, saindo da página sem a utilização dos botões “Create” e “Cancel”, a instancia incompleta não ocupa espaço na base de dados.

Por outro lado, esta lógica levantará algumas dificuldades no processo de formação das instancias a adicionar.

O menu de colaboradores tem *links* para as seguintes páginas:

* Vacancies, vagas já estabelecidas, permite introduzir uma nova vaga,
* Applications, lista de aplicações de candidatos a vagas já estabelecidas,
* Projects, projetos iniciados, permite iniciar um novo projeto,
  + Nota: é possível criar vagas a partir de um projeto.
* Clients, clientes da PS Tec, permite adicionar um cliente,
  + Nota: é possível criar projetos a partir de um cliente.
* Events, demonstra os eventos em que o utilizador participa, permite introduzir um novo evento,
* Forms, lista de formulários existentes, permite adicionar e alterar formulários,
* Curriculums, demonstra as candidaturas espontâneas que ainda não foram recusadas ou aceitadas,
* Candidates, lista dos candidatos já existentes,
  + Nota: cada candidato demonstrado permite ver as informações gerais, o currículo, dossiê de capacidades, aplicações e até entrevista já realizadas.

Já o menu de candidatos tem links para as páginas:

* General, informação geral do candidato,
* Curriculum, currículo do candidato, permite alterar o mesmo,
* Profile, dossiê de capacidades do candidato, permite alterar o mesmo,
* Tutorial, uma página de tutorial que explica algumas características do currículo e dossiê,
* Applications, aplicações a vagas existentes que o candidato esta aplicado,
* Vacancies, vagas disponíveis que o candidato ainda não se aplicou,
  + Nota: a página permite uma pesquisa inteligente de vagas.
* Availability, com esta página o candidato pode demonstrar a sua disponibilidade para entrevista.