

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Departamento de Matemática - CFM

PET Matemática

Minicurso da Classe *EXAM*

Aldrovando Luis Azeredo Araujo
Mateus Souza Oliveira

Florianópolis, Março de 2016

Sumário

1	Apresentação	3
2	Introdução	4
2.1	Sobre o Modelo do Departamento	4
3	Questões	5
3.1	\code{questions}, as Questões da Prova	5
3.2	Pondo em prática	5
3.3	Questões com Itens e Subitens	6
3.4	Pontuação	6
3.5	Além das Questões	6
3.6	Bônus, para os Professores Bonzinhos!	7
4	Soluções de Questões	8
4.1	Tipos de Questões	8
4.2	Ambiente <code>solution</code>	9
4.3	Imprimindo a Prova	9

Capítulo 1

Apresentação

Quem comentou pela primeira vez comigo sobre o assunto foi o professor José Luiz Rosas Pinho, então tutor do PET Matemática. Ele me falou que o Professor Aldrovando Luis Azeredo Araujo estava querendo ministrar um minicurso sobre \LaTeX aos professores. Eu fiquei interessado e comentei com a equipe que havia na época no PET responsáveis pela elaboração de minicursos, mais ligado ao *MATLAB* naquele período. Nenhum deles era muito fã do programa de escrita, pelo seu caráter mais instável, comparado ao outro programa citado, e acabei indo conversar com o então chefe do departamento sozinho.

O plano do professor Aldrovando era modernizar e padronizar as provas do departamento de matemática da UFSC, e logo compreí a ideia. Conversamos no fim do segundo semestre de 2015 pela primeira vez, mas coloquei a mão na massa mesmo no começo do semestre de 2016, fazendo esta apostila e ajudando no modelo padrão desejado pelo professor desde o início.

O professor Aldrovando também comentou que havia professores no departamento que ainda nem conheciam o \LaTeX , e seria uma boa chance de começarem a usá-lo. Para vocês que ministram aulas como Álgebra, Cálculo, entre outras, vão tirar isto de letra. não acreditam? Tomem um exemplo: eu. O professor Pinho gosta de falar uma coisa a todos que perguntam sobre mim no PET: “O Mateus entrou no PET sem saber ligar um computador e hoje é um de nossos colaboradores em minicursos do PET!” (Claro que havia um pouco de exagero do Pinho: ligar o computador eu sabia). Hoje já tenho de bagagem participação na editoração Revista da Olimpíada Regional de Matemática de Santa Catarina, ter digitado alguns treinamentos da mesma e 3 minicursos de \LaTeX e um de *MATLAB* ministrados (todos pelo PET), e talvez estou ministrando mais um para vocês hoje.

Mateus Souza Oliveira

Aluno de Graduação em Matemática Licenciatura
Aluno Voluntário na Revista da ORM
Ex-bolsista PET Matemática

Capítulo 2

Introdução

O objetivo desta apostila é auxiliá-lo no manuseamento do modelo de provas do departamento de Matemática da UFSC, feito a partir da classe *Exam* no \LaTeX . Em cada capítulo vamos abordar uma das funções necessárias para tal.

IMAGEM DO CABEÇALHO DO MODELO

Caso não esteja familiarizado com o \LaTeX , sugiro a leitura da apostila [UMA INTRODUÇÃO AO \$\text{\LaTeX}\$ \[2\]](#) feita por mim e colegas do PET, que aborda os primeiros passos no programa.

Agora se você procura algo mais aprofundado ou criar um modelo próprio, indico a apostila [Using the exam document class\[1\]](#), que usei de base para esta, onde há várias informações sobre layout de cabeçalho e rodapé, por exemplo, que não será abordado neste material.

2.1 Sobre o Modelo do Departamento

A ideia é deixar o modelo fixo, que os professores não se preocupem em mexer no cabeçalho ou rodapé, e sim nas questões de sua prova e completamente customizá-las como quiserem, e ainda as disponibilizarem com gabaritos a seus alunos depois. Para tal vamos passar todas as informações que julgamos necessário para realização disto nos capítulos seguintes.

Capítulo 3

Questões

Vamos imaginar que nosso modelo seja um arquivo diferente, focando apenas nas partes “editáveis”, sendo assim vamos dizer que nosso `\begin{document}` agora seja o comando `\begin{questions}`. Este comando é muito parecido com os ambientes `enumerate` e `itemize`, onde o comando `\item` é substituído por `\questions`. Então não precisamos nos preocupar com a quebra de linhas porém devemos lembrar que o primeiro comando do ambiente deve ser o `\questions`, se não haverá problemas na hora da compilação do documento.

3.1 `\questions`, as Questões da Prova

Além de ser extremamente parecido com o comando `\item`, esse comando também apresenta suas funcionalidades únicas, como:

- Pontuação: exprime o valor da questão (antes ou após a questão, ou no local desejado pelo usuário);
- Solução: permite digitar a solução da questão logo após a mesma, sem se preocupar em comentá-la ao final e ainda deixa um espaço em branco para o aluno resolvê-la (ou criar linhas para escrever a resposta, ou simplesmente deixa o espaço em branco para o aluno assinalar o item que julga correto);

3.2 Pondo em prática

Para fazer suas questões basta inserir o comando `\question` e sua questão logo após, dentro do ambiente `question`. ele numera as questões automaticamente e é definida por um número arábico seguido de um ponto. Para trocar isto devemos usar os comandos: `\renewcommand{\thequestion}{alteração{question}}`, para trocar do modo arábico para as seguintes opções:

- `\arabic` para números arábicos;
- `\alph` para letras minúsculas;
- `\Alph` para letras maiúsculas;
- `\roman` para números romanos minúsculos;
- `\Roman` para números romanos maiúsculos;
- `\greeknum` para letras gregas minúsculas.

Agora para mudar o contorno devemos usar o comando:

`\renewcommand{\questionlabel}{alteração\thequestion alteração}`. Por exemplo para colocar as questões como letras maiúsculas e entre parênteses devemos usar os seguintes comandos:

```
\renewcommand{\thequestion}{\Alph{question}}
\renewcommand{\questionlabel}{(\thequestion)}
```

3.3 Questões com Itens e Subitens

Caso deseje criar uma questão com itens e subitens basta criar outro ambiente dentro da questão, o ambiente *parts* e *subparts* (e *subsubparts* caso seja a questão mais longa do mundo). Então caso deseje uma questão com dois itens e um subitem devo criar primeiro o ambiente *parts* e para criar o item o comando `\part`, e para criar o subitem, criar o ambiente *subparts* dentro de um ambiente *parts* e usar o comando `\subpart` (e o mesmo pensamento vale para o *subsubparts*).

Para customizar os itens devemos usar uma sintaxe muito parecida com a que usamos nas questões:

- Itens: para mudar algo nos itens devemos usar a mesma sintaxe usada nas questões, exceto por onde houver `\thequestion`, `question` e `\questionlabel` por `\thepartno`, `partno` e `\partlabel`, respectivamente;
- Subitens: para mudar algo nos subitens devemos usar o mesmo pensamento acima, mas trocando por `\thesubpart`, `subpart` e `\subpartlabel`;
- Subsubitens: e também vale `\subsubpart`.

3.4 Pontuação

Agora vem a parte legal, dar valor as questões. Também é bem simples, para fazê-lo basta depois do comando de questão (ou subitem), entre chaves colocar o valor. Então, por exemplo, para ter uma questão valendo um ponto usamos o comando `\question[1]`.

Mas precisamos ter alguns cuidados, o primeiro é que não podemos usar qualquer valor em nossas questões, nosso conjunto possível (até agora pelo menos, em futuras atualizações isto talvez mude) é: $0,5k$, $k \in \mathbb{N}$. E também não podemos colocar [1.5], temos que colocar a parte inteira seguida do comando `\half`.

O outro ponto que temos que tomar cuidado é quando damos valor aos subitens. Por exemplo se faço uma questão com dois subitens, a questão valendo 1 ponto e cada subitem 0,5 ponto, para que a pontuação da questão não seja contada duas vezes devemos usar os comandos `\addpoints` e `\noaddpoints`. Para tal, antes de iniciar o ambiente subitem usamos o comando `\noaddpoints`, para não contar os pontos dos subitens, e logo após o fechamento deste ambiente usamos o comando `\addpoints` para voltar a contar os pontos (caso esteja se perguntado onde esses pontos são contados, isto acontece na tabela do cabeçalho, chamada de *gradetable* pela classe *exam*, mas como não é foco deste material não iremos abordá-la, caso sintá-se interessado sugiro a apostila[1]).

E também podemos alterar a aparência dos pontos, com os seguintes comandos:

- `\pointpoints{alteração 1}{alteração 2}`: este altera as palavras *point* e *points* por outras desejadas, respectivamente;
- `\pointname{alteração}`: parecido com o anterior, porém este troca tanto o singular quanto o plural da palavra, mais indicado se for usar um símbolo ao invés de palavras na indicação da pontuação;
- `\bracketedpoints`: altera o *layout* original de entre parenteses para entre colchetes;
- `\boxedpoints`: e este altera o *layout* para em volta em um retângulo.

Além desses comandos para alterar o *layout*, existem outros que não serão citados, os quais poderíamos escolher onde os pontos aparecerão (antes da questão, ao final da questão, por exemplo).

3.5 Além das Questões

Caso queira inserir um texto antes de uma questão, ou uma instrução para ela, sugiro inserir os comandos `\fullwidth{texto}` ou `\uplevel{texto}`, que irão prevenir de haver algum problema por estarem fora de um ambiente *question*. Estes comandos também podem ser usados para aumentar os espaços deixados pelo ambiente *solution* (que será tratado no próximo capítulo).

3.6 Bônus, para os Professores Bonzinhos!

Além das questões normais também há a possibilidade de fazermos questões bônus, as quais sua pontuação não será incluída na tabela de notas do cabeçalho. Para criar uma questão bônus usamos o mesmo ambiente das questões, porém usamos o comando `\bonusquestion`.

Como estamos trabalhando dentro do ambiente *questions*, algumas destas sintaxes valem para as questões bônus, como por exemplo os comandos `\bracketedpoints` e `\boxedpoints`. Já para alterar a aparência da questão bônus os comandos são bem parecidos, com quase todos sendo iguais aos das questões, mas com a palavra *bonus* na frente, por exemplo: `\bonuspointpoints{alteração 1}{alteração 2}`, para trocar o singular e plural do que irá seguir a pontuação bônus.

Capítulo 4

Soluções de Questões

Depois de fazer as questões é hora de fazer as soluções delas (caso esteja se perguntando se é obrigatório fazer a solução: não é). Para fazê-las basta criar o ambiente `solution[tamanho]`, em que o tamanho é o espaço que será deixado livre para o aluno resolver a questão e não o tamanho da solução, o qual você não deve se preocupar.

4.1 Tipos de Questões

O que vai ditar qual o tipo de questão vai ser a sua solução (objetiva, discursiva ou aberta). Para questões discursivas ou abertas o comando será o próprio ambiente `solution`, mas para objetivas vamos usar um dos seguintes ambientes:

- `choices`: Cria múltiplas escolhas numeradas, pulando linha uma para a outra, a partir do comando `\choice` e `\correctchoice`, para a resposta;
- `oneparchoices`: igual ao anterior mas uma do lado da outra, ideal para quando as alternativas são curtas;
- `checkboxes`: Cria múltiplas escolhas com espaços para o aluno preencher a resposta, pulando linha uma para a outra, também a partir do comando `\choice` e `\correctchoice`, para a resposta;
- `oneparcheckboxes`: igual ao anterior mas um do lado da outro.

E também podemos customizar a aparência delas, com os comandos:

- `\correctchoiceemphasis{alteração}`: aqui você pode mudar a cor e estilo da resposta certa (negrito, itálico, etc);
- `\renewcommand{\thechoice}{alteração{choice}}`: aqui você altera pelo o que as alternativas serão presididas (números, letras, etc), nos ambientes `choices` e `oneparchoices`;
- `\renewcommand{\choicelabel}{alteração\thechoice alteração}`: aqui você altera pelo o que os números das alternativas serão envoltos (parenteses, colchetes, etc), também nos dois ambiente citados no item anterior;
- `\checkboxchar{alteração}`: aqui você muda o símbolo que irá presidir as opções, nos ambientes `checkboxes` e `oneparcheckboxes`;
- `\checkedchar{alteração}`: troca o símbolo que irá presidir a opção correta, também nos dois ambientes citados no item anterior.

Observação: os comandos `\checkboxchar` e `\checkedchar` devem ser preenchidos com símbolos, portanto você deve precisar usar o ambiente matemático neles (cifrões).

Além desses citados, ainda existem outros dois tipos de questões, usando os comandos `\fillin[resposta][tamanho]`, que cria uma linha para completar com a resposta, e o comando `\answerline`, que cria ao final da questão uma linha à direita, ideal para colocar apenas o valor final das contas.

4.2 Ambiente *solution*

Voltando ao tema principal, o ambiente *solution* cria um espaço (o qual seu tamanho nós definimos, como explicado anteriormente), quando não queremos a solução, ou a própria solução.

Não há segredo quanto a digitação da solução, porém podemos mudar o espaço deixado quando não queremos que a solução apareça, com os seguintes ambientes:

- *solutionorlines*: preenche o espaço com linhas;
- *solutionordottedlines* preenche o espaço com linhas pontilhadas;
- *solutionorbox* envolve o espaço em um retângulo;
- *solutionorgrid* preenche o espaço com linhas horizontais e verticais (como um quadriculado).
Observação: todos eles usados ao invés do ambiente solution. Nenhum deles muda o estilo da solução, apenas o estilo do espaço deixado.

E também podemos customizar as aparência das soluções com os seguintes comandos:

- `\renewcommand{\solutiontitle}{alteração}`: troca o título da sua solução e seu estilo;
- `\SolutionEmphasis{alteração}`: troca o estilo do retângulo que contorna a solução, como sua cor.

4.3 Imprimindo a Prova

Se você quiser que sua prova não apareça com as soluções (para dar aquela dificultada nos alunos), basta usar o comando `\noprintanswers` no preâmbulo, e compilar. Em alguns testes que fiz, as vezes as soluções por serem muito extensas, exprimiam o cabeçalho. Caso isto aconteça, um simples comando de pular página no ambiente da solução deve solucionar o problema.

Referências Bibliográficas

- [1] Philip Hirschhorn, *Using the exam document class*.
- [2] Jean Gengnagel; Mateus Oliveira; Miguel bauschat; Priscilla Oliveira, *UMA INTRODUÇÃO AO L^AT_EX*.