

Faremos no horário da aula (turma completa) de divulgação dessa atividade o cadastro das equipes. **A responsabilidade de cadastramento da equipe é de cada estudante** e a equipe deverá ser composta por até **3 (três) integrantes da mesma turma de CSBA** ( mesmo período ).

### Objetivo geral:

Desenvolver um aplicativo orientado a objetos em **Python** com sua interface gráfica em **Kivy** que satisfaça as solicitações descritas abaixo.

### Sobre as equipes e os prazos:

1. Equipes: com até **3** estudantes. Conforme a quantidade de estudantes nas turmas, poderá existir, no máximo, 1 (UMA) dupla se for necessário.
2. Itens de entrega no Canvas:
  - a. arquivo (**py**) com o código fonte orientado a objetos em Python, bem-organizado e com nomes significativos para métodos, classes e variáveis.
  - b. arquivo (**kv**) com a interface gráfica em Kivy e referente ao programa Python entregue (obrigatório o desenvolvimento do arquivo .kv separado do arquivo .py). Nomes significativos para elementos gráficos e eventos. Os componentes gráficos **OBRIGATÓRIOS** a serem utilizados são:

Layouts, Labels, TextInput, Spinner, Buttons, CheckBoxes (RadioButtons)  
Sendo Spinner, Button e Check(Radio)Buttons contendo evento que chama método implementado no programa em Python

- c. arquivo (**pdf**) com nome completo dos membros da equipe, descrição resumida (um parágrafo de, no máximo, 10 linhas) do aplicativo e **instruções passo a passo (com exemplos de uso do programa)** . **Incluir aqui o Diagrama de Classes do aplicativo..**
3. Avaliação:
    - a. três arquivos solicitados e entregues no Canvas (local específico para isso) até **15/10 (domingo)** para receber NOTA INTEGRAL
    - b. entregas atrasadas serão aceitas, no máximo, até 16/10 (segunda-feira) com nota máxima 7,0 (sete)
    - c. defesa do projeto SOMENTE no horário da aula na semana de **17/10 a 20/10 nas aulas de turma modulada**
  4. Período para desenvolvimento do projeto pelas equipes:  
**a partir de 05/10 até a data da entrega**, todos os estudantes poderão usar todos os horários de aula para reunirem-se com suas equipes, tirarem dúvidas com seus professores e focarem no desenvolvimento da sua solução

### Sobre os descontos previstos:

ATENÇÃO às penalidades a serem aplicadas para entregas indevidas:

Descrição dos problemas considerados:	Desconto
Entrega feita em local indevido (diferente do INDICADO no Canvas)	2 pontos
Entrega feita após data limite: <b>15/10 às 23h59</b>	3 pontos
Plágio entre resultados entregues por equipes distintas	10 pontos

Arquivo(s) entregue(s) pela equipe diferentes do conteúdo solicitado	10 pontos
O resultado entregue pela equipe não segue o solicitado (formatos dos arquivos a serem entregues conforme descrito no item (2) Itens de entrega no Canvas:)	5 pontos
O aplicativo entregue pela equipe não é executado com sucesso (não roda - apresenta algum erro de sintaxe)	5 pontos

- Se houver descontos, estes serão computados inicialmente na correção. Na sequência, a correção seguirá rubrica disponibilizada no Canvas, considerando a nota máxima **com desconto**.
- Casos especiais serão tratados individualmente.

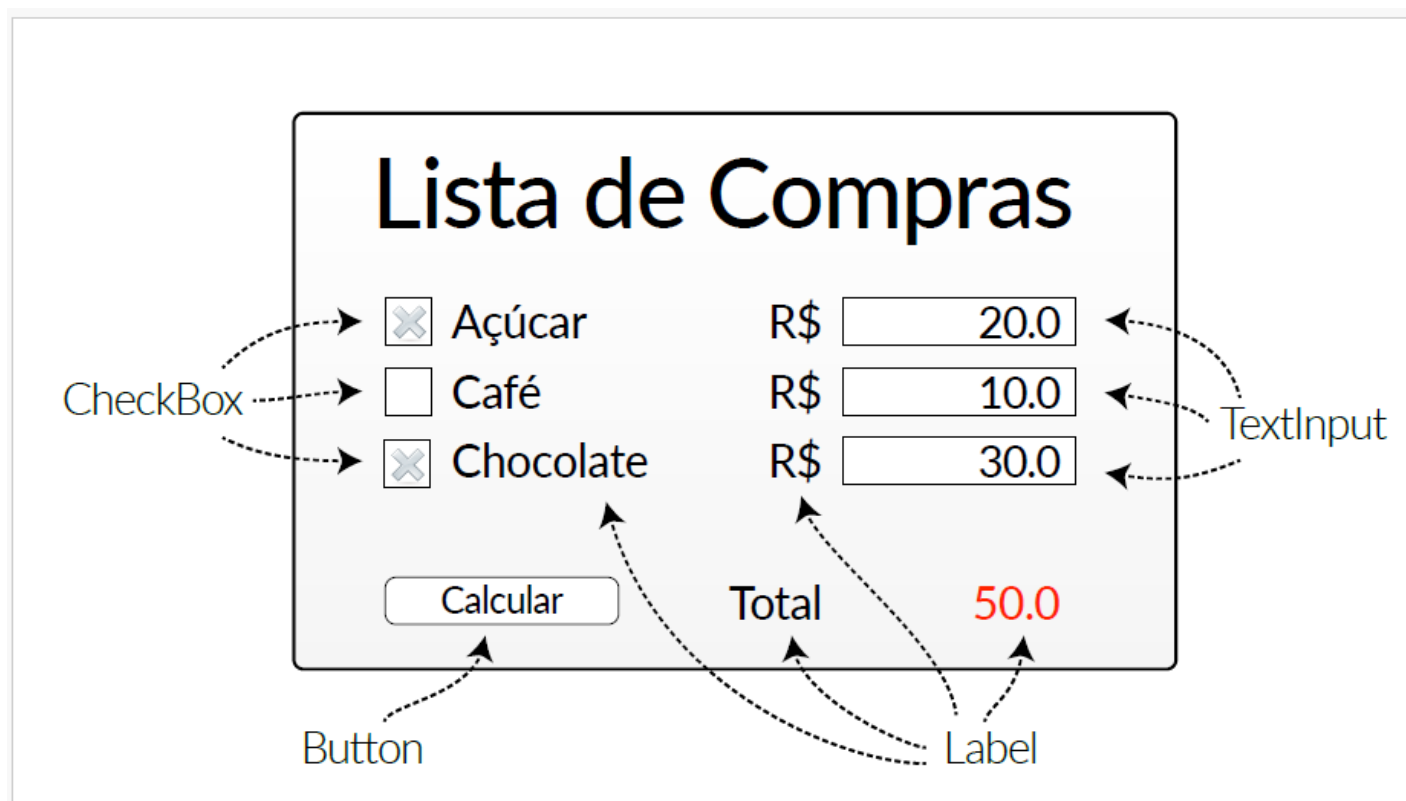
### Sobre a GUI - interface gráfica com o usuário em Kivy:

1. Elementos gráficos a serem usados:

**No mínimo: Label, Spinner, Botões, TextInput, Layouts e Check(Radio)Buttons.**

Caso a equipe queira utilizar mais alguns elementos gráficos distintos dos citados aqui, deve, primeiramente, confirmar sua validade com o Professor.

2. Layout de uma Interface gráfica que pode ser usada como base para a interface a ser desenvolvida (lembrando que, nesse exemplo base não há Spinner, porém, na interface a ser desenvolvida é obrigatório o uso do Spinner):



### Sobre a codificação dos programas:

1. O programa deve ser escrito em linguagem Python e com o paradigma orientado a objetos, importando todas as bibliotecas necessárias e disponibilizadas pelo próprio Python e pelo Kivy
2. O programa deve ser rigorosamente estruturado em classes e seus respectivos atributos e métodos, respeitando argumentos e valores de retorno necessários para o correto funcionamento do aplicativo como um todo
3. Fazer uso devido das regras de sintaxe definidas para a integração dos programas Python e Kivy, suas classes, eventos e métodos, variáveis comuns, argumentos
4. Devem ser usados NOMES SIGNIFICATIVOS para variáveis, métodos, classes e argumentos
5. Deve ser feita a necessária e suficiente verificação da codificação dos programas, integração entre eles, verificação de dados válidos como entrada, a fim de evitar erros de execução devido a dados inválidos ou ausentes.

### Sobre a lógica a ser desenvolvida:

- **Elaboração de um PEDIDO de compra de um estabelecimento com produtos bem específicos. Os estabelecimentos poderão ser:**
  - Pizzaria (tipos ou nomes de pizzas)
  - Sanduicheria (tipos ou nomes de sanduíches)
  - Fornecedor de bebidas (tipos ou nomes de bebidas)
  - Fornecedor automotivo (tipos ou nomes de produtos automotivos)
  - Loja de roupas (tipos ou nomes de roupas)
  - Loja de brinquedos (tipos ou nomes de brinquedos)
- **Funcionamento geral:**
  1. Aparece o nome do estabelecimento escolhido (nome fantasia)
  2. O usuário digita o seu nome
  3. O usuário deverá selecionar uma ou mais opções dentre 6 (seis) itens vendidos pelo estabelecimento escolhido
  4. O usuário deverá digitar a quantidade desejada de cada item selecionado
  5. O programa deverá apresentar um Spinner contendo os tipos de pagamento que serão: à vista (espécie), à vista (pix), à vista (cartão de crédito) ou parcelado 2 vezes no cartão de crédito
  6. Cada equipe deverá estipular o valor de acréscimo ou desconto referente aos tipos de pagamento, por exemplo: compra em espécie tem 5% de desconto no valor total, pix tem 8% de desconto no valor total, e assim por diante. Esses valores deverão ser digitados internamente no programa **.py** dentro de uma lista.
  7. Conforme os itens selecionados, as quantidades especificadas por item e o tipo de pagamento, o programa deverá apresentar na tela como resposta: ***O valor da compra de <nome do usuário> totalizou RS <valor total calculado pelo programa>***
  8. Cada equipe deverá estipular o valor individual de cada item definido para o estabelecimento escolhido. Os valores deverão ser digitados internamente no programa **.py** dentro de um dicionário de dados
  9. O programa deverá apresentar dois botões (obrigatoriamente): um para limpar campos (reset) e outro para sair (fechar aplicativo).