

# AGILIDADE NA PRÁTICA

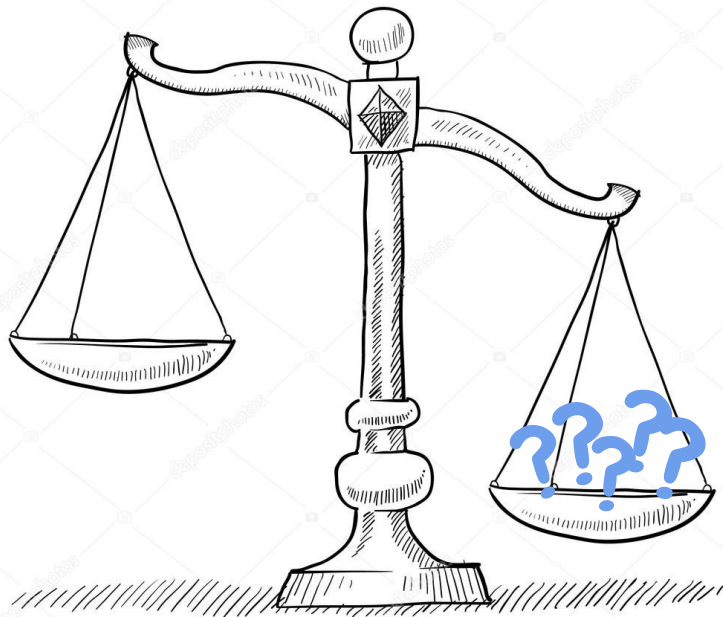
COM ANDROID 

DIEGO LISBOA  
LUCAS MONTEIRO  
CARLOS SANTIAGO  
ARMANDO ALMEIDA

# Agenda

- Lightning Talk sobre Métodos Ágeis (**Diego Lisboa**)
- Lightning Talk sobre Planejamento Ágil (**Lucas Monteiro**)
- Lightning Talk sobre o Desafio Prático (**Carlos Santiago**)
- Prática 1 - Mão na Massa de Planejamento
- Lightning Talk sobre Android (**Carlos Santiago e Armando Almeida**)
- Prática 2 - Mão na Massa: 1º Ciclo de Desenvolvimento
- Prática 3 - Retrospectiva / Planos de Ação - 1º Ciclo de Desenvolvimento
- Prática 4 - Mão na Massa: 2º Ciclo de Desenvolvimento
- Prática 5 - Apresentação dos Resultado - Review
- Prática 6 - RETROSPECTIVA FINAL
- Encerramento

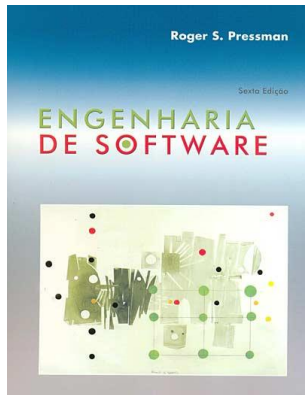
# Balança



- O que lhe motivou a fazer este mini-curso?
- Quem já fez a disciplina de Engenharia de Software?
- Quem já ouviu falar sobre Métodos Ágeis?
- Quem já trabalhou e/ou trabalha com Métodos Ágeis?
- Quem já programou em par?
- Quem já participou de um Dojo de Programação?



# Engenharia de Software



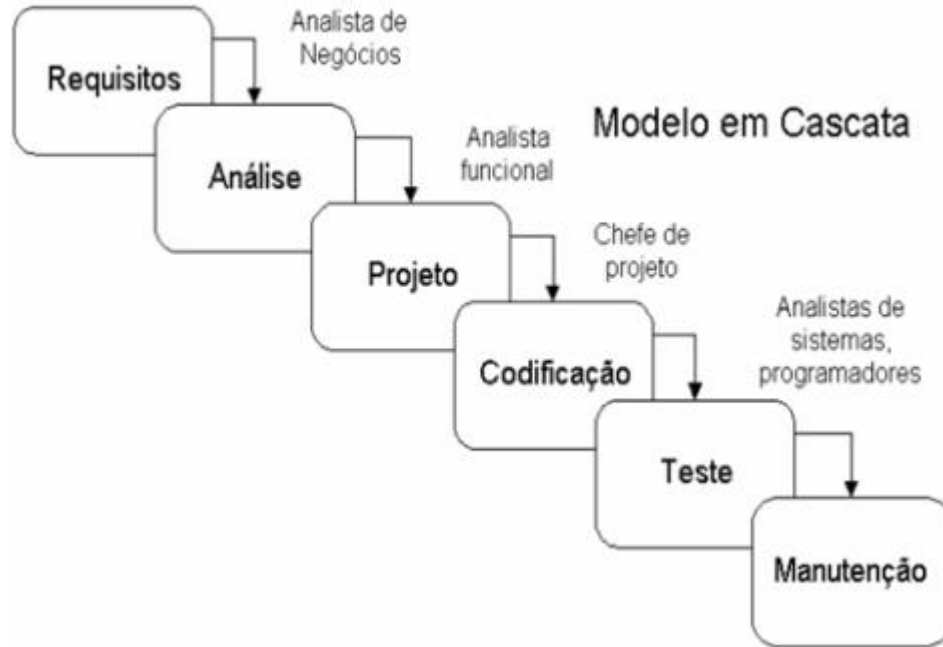
“Engenharia de *software* é uma abordagem sistemática e disciplinada para o desenvolvimento de *software*.” (PRESSMAN, 2006)

“Engenharia de Software é uma disciplina de engenharia relacionada com todos os aspectos de produção de software, desde os estágios iniciais de especificação do sistema até a sua manutenção, ou seja, mesmo depois que este entrar em operação.” (SOMMERVILLE, 2003)



# Fases da Engenharia de Software

## Modelo Cascata





Como o cliente explicou



Como o líder de projeto entendeu



Como o analista planejou



Como o programador codificou



O que os beta testers receberam



Como o consultor de negócios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistência técnica instalou



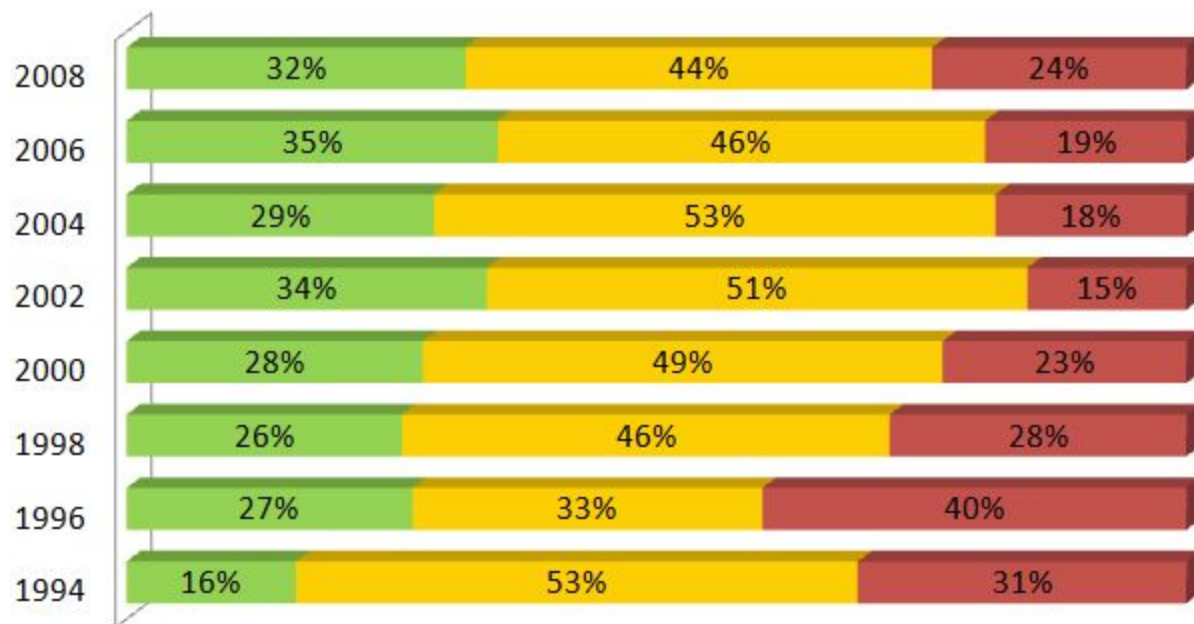
Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava



■ Sucesso ■ Atraso/Prejuízo ■ Falha

Fonte: Standish Group, CHAOS Summary for 2010 e Extreme CHAOS 2001



**MODERN RESOLUTION FOR ALL PROJECTS**

	2011	2012	2013	2014	2015
<b>SUCCESSFUL</b>	29%	27%	31%	28%	29%
<b>CHALLENGED</b>	49%	56%	50%	55%	52%
<b>FAILED</b>	22%	17%	19%	17%	19%

*The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.*

# Manifesto para o desenvolvimento ágil de software

---

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

**Indivíduos e interações** mais que processos e ferramentas  
**Software em funcionamento** mais que documentação abrangente  
**Colaboração com o cliente** mais que negociação de contratos  
**Responder a mudanças** mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

Steve Mellor

Ken Schwaber

Jeff Sutherland

Dave Thomas

*Métodos Ágeis*



# O que é ser ágil?



## Responder a Mudanças, Sem Traumas

Mudanças são bem vindas  
Foco em Valor

## Entregar mais Rápido (Frequência)

Visualização antecipada dos resultados  
Feedback Antecipado

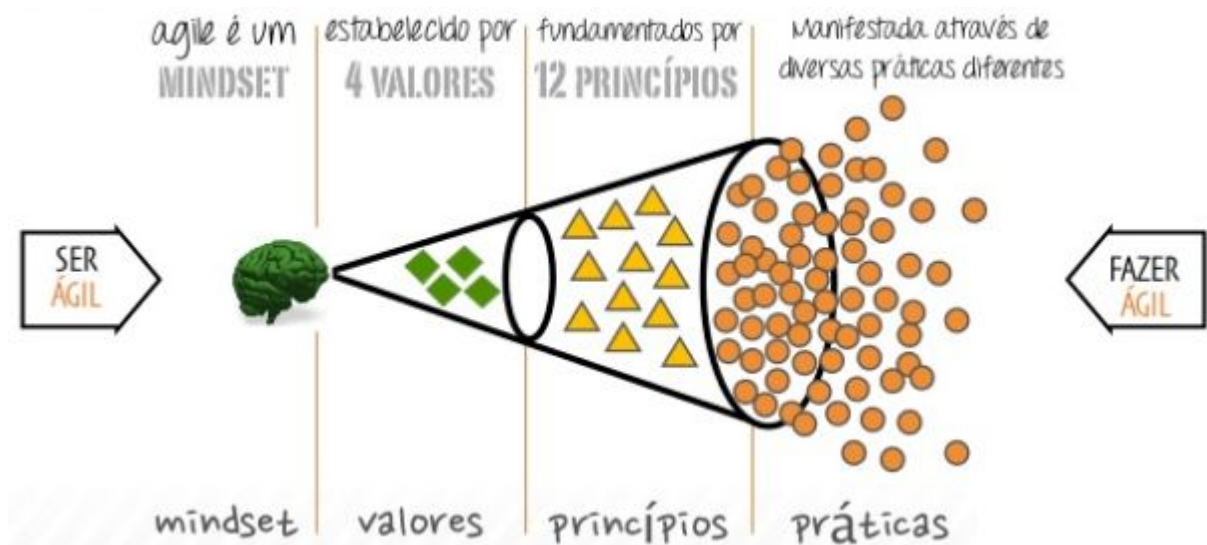


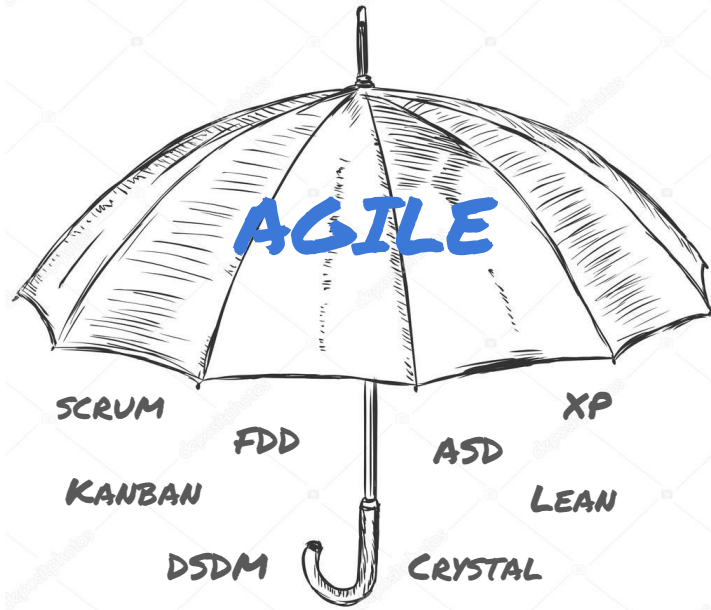
## Assertividade

Maior frequência de planejamento  
Maior Qualidade das Entregas

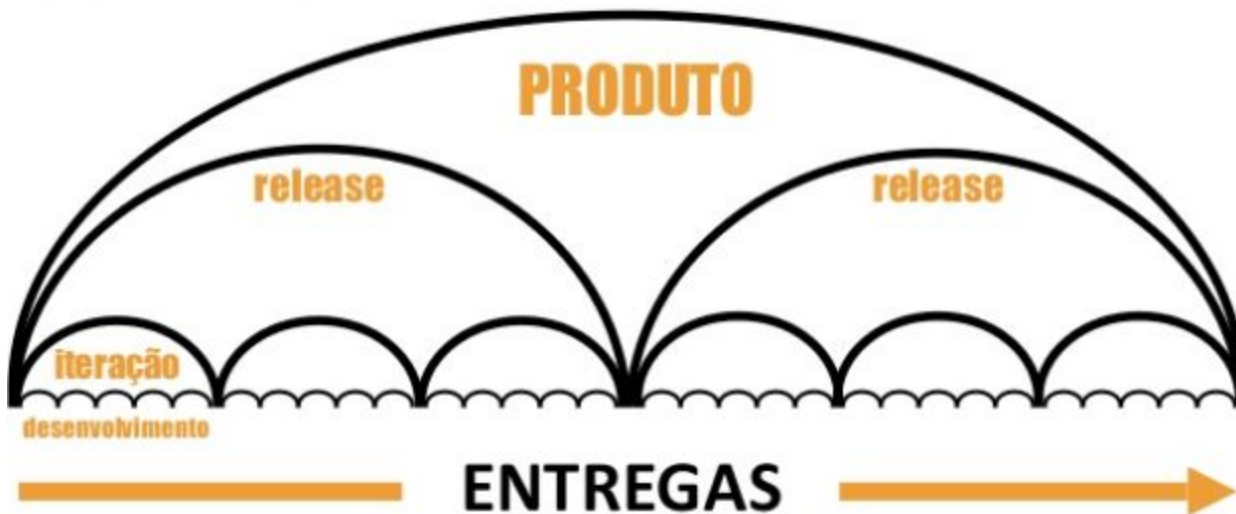


# Mindset Ágil



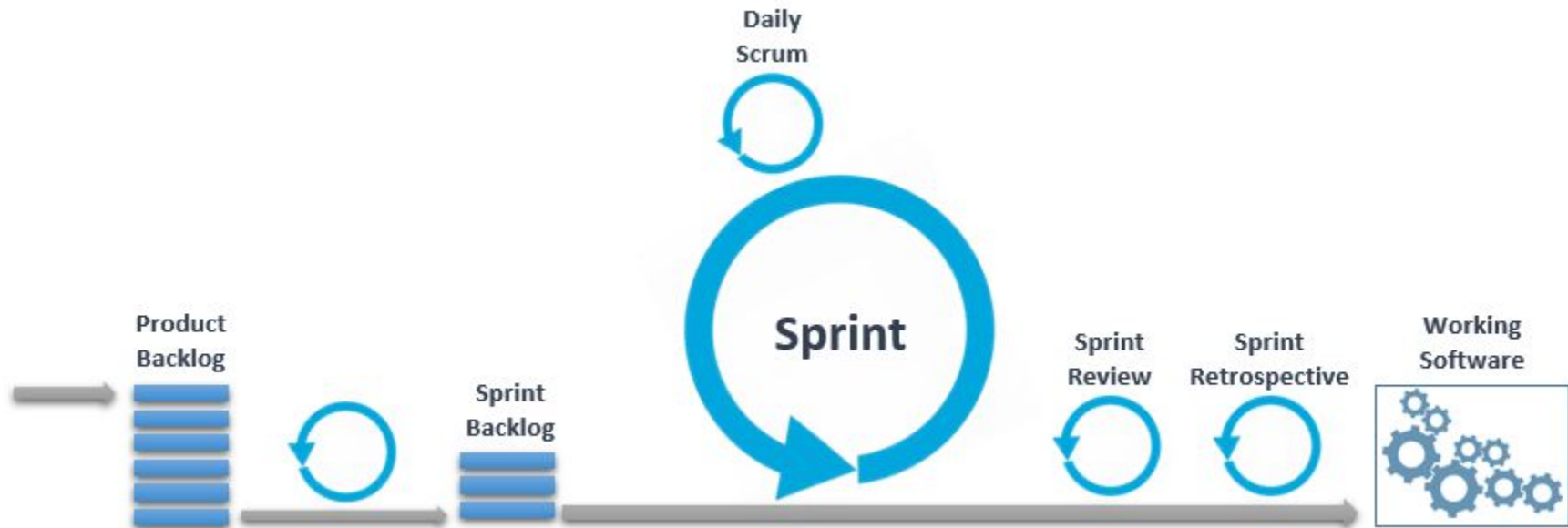


# Iterativo e Incremental



Processos ágeis são fortemente baseados em uma abordagem iterativa e incremental

# Ciclo de vida do Scrum

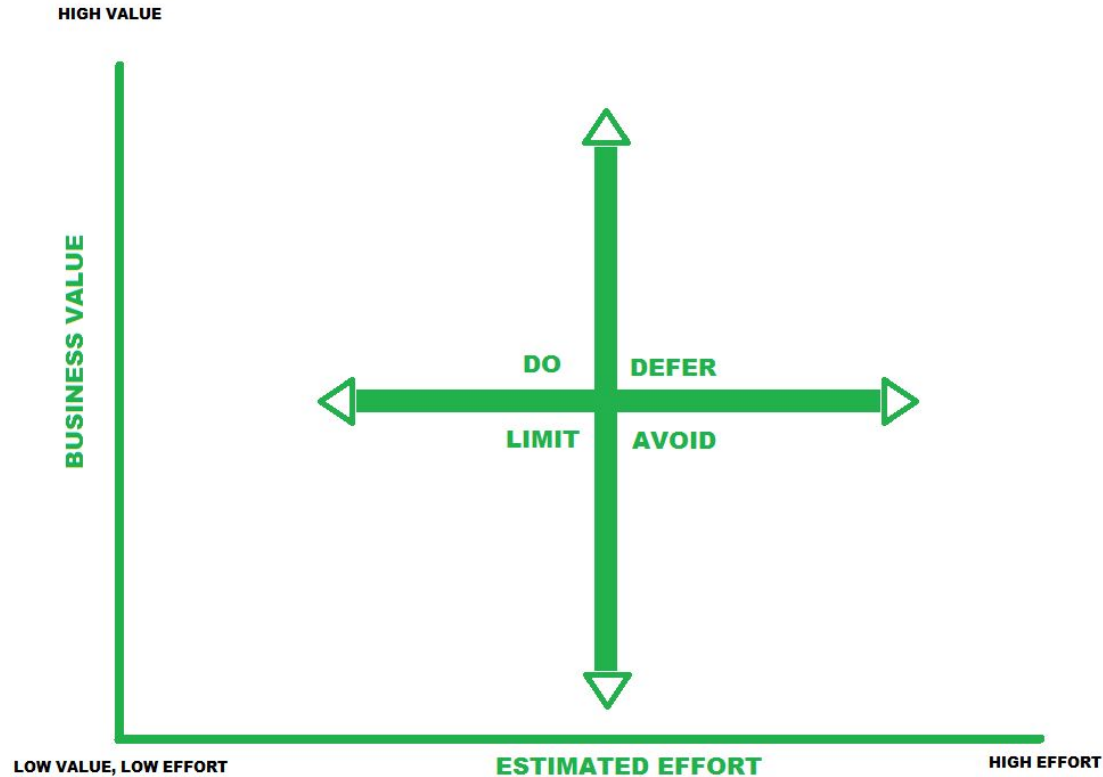




# SE VOCÊ NECESSITA COMEÇAR UM PROJETO, VOCÊ JÁ FALHOU

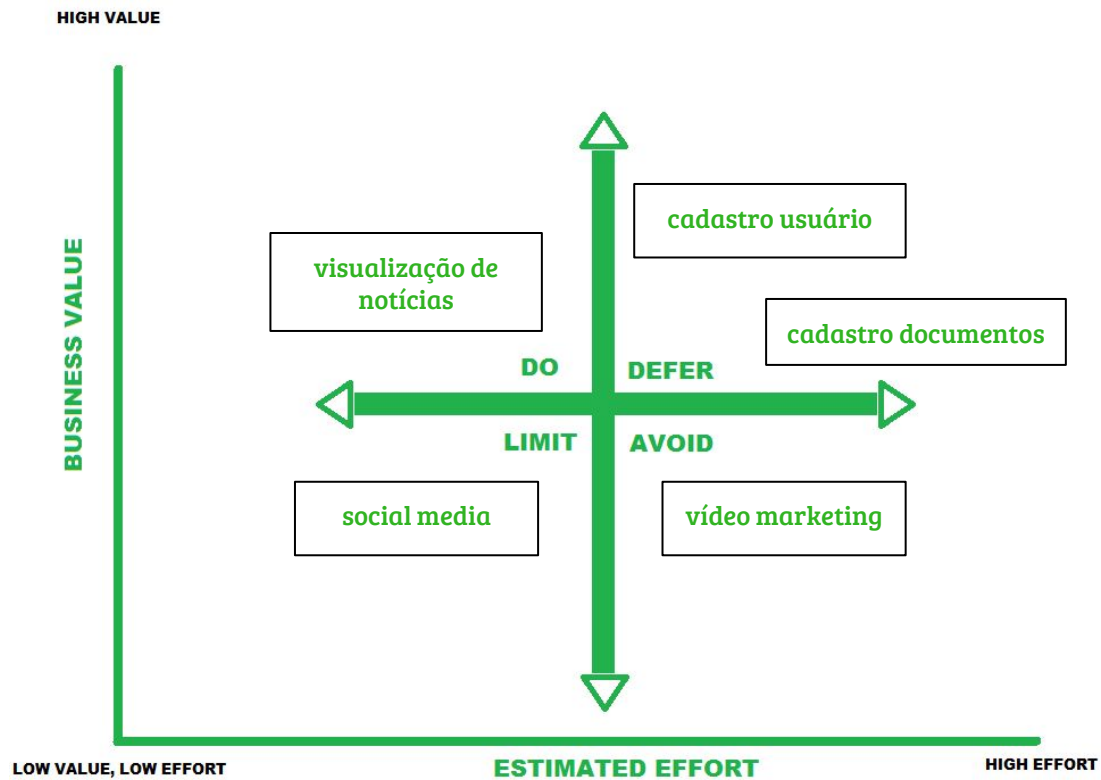
- Prioriza resultados, ao invés de saídas;
- Verifica o maior valor entregue possível, não distinguindo entre produtos;
- Activity Canva baseado no valor e esforço.

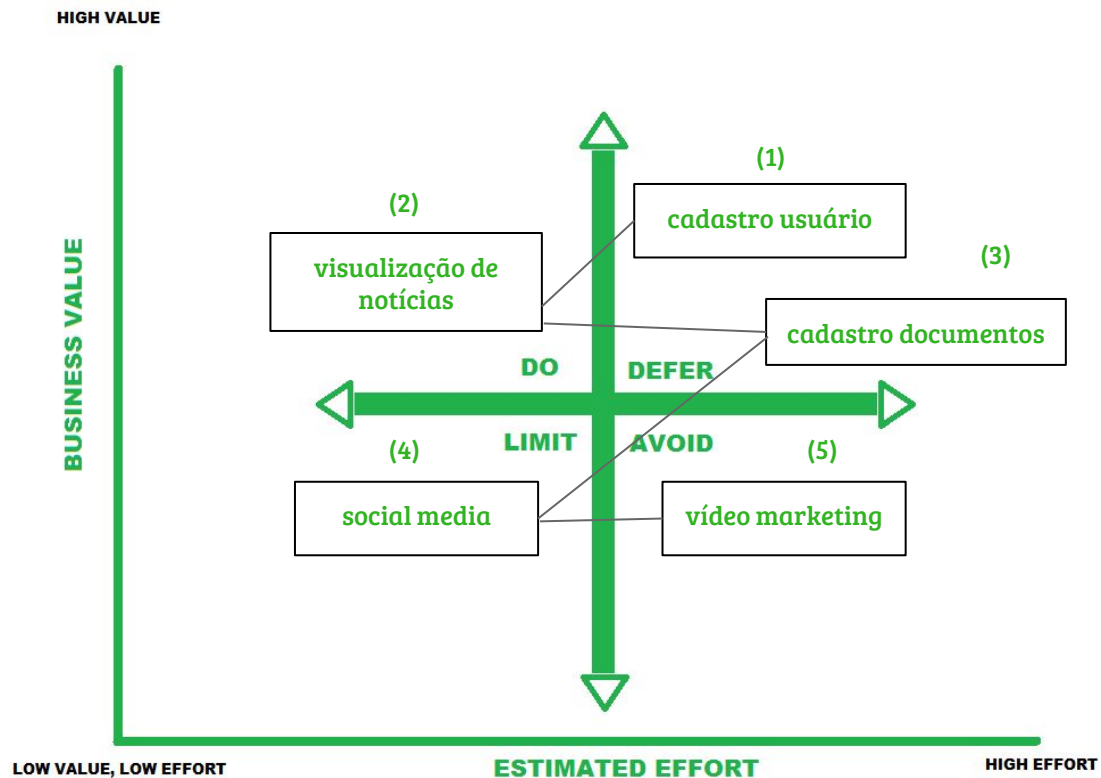
# ACTIVITY CANVA



# ACTIVITY CANVA

- Estima-se valor e esforço de atividades (nesse caso, U.S) comparando-as entre si;
- Uma atividade pode mudar de quadrante caso seja necessário;
- Os quadrantes não impõem ordem de prioridade.






# Desafio Proposto

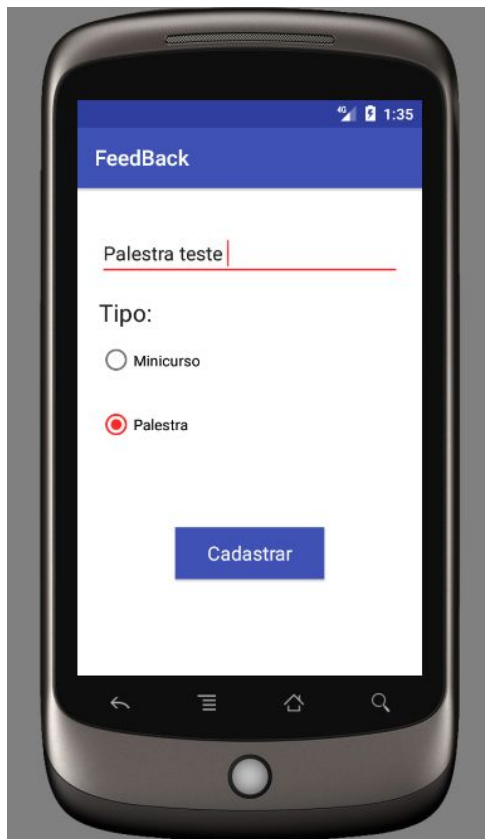
Feedback do evento

# App Mobile para Feedback do evento

O app deverá ser desenvolvido em **Android** 



# Cadastrar e Listar Palestras/Minicursos





# Visualizar Palestra/Minicurso



# Excluir Palestra/Minicurso



# Avaliar Palestra/Minicurso



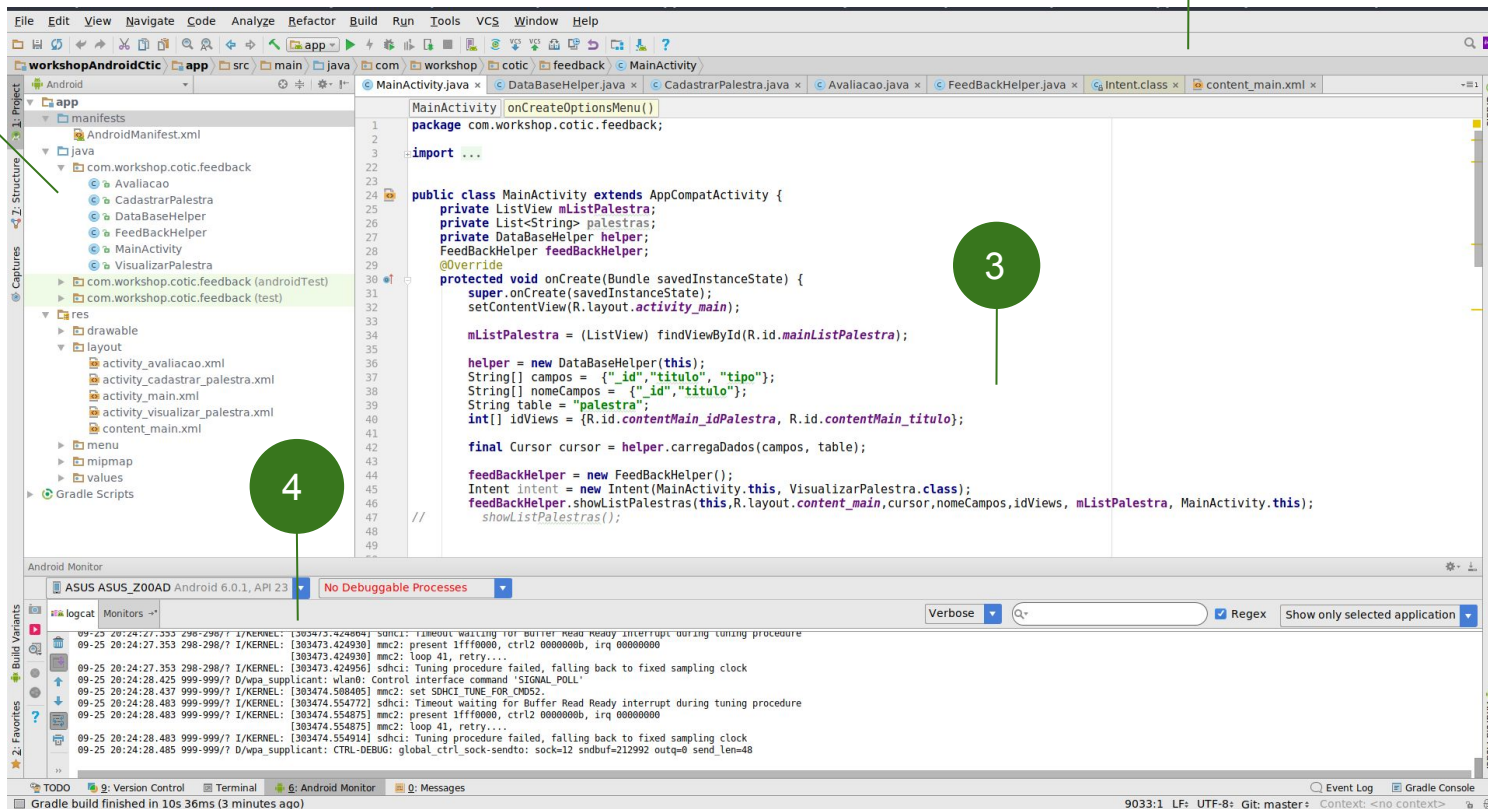
Outras Features serão serão detalhadas  
a seguir...



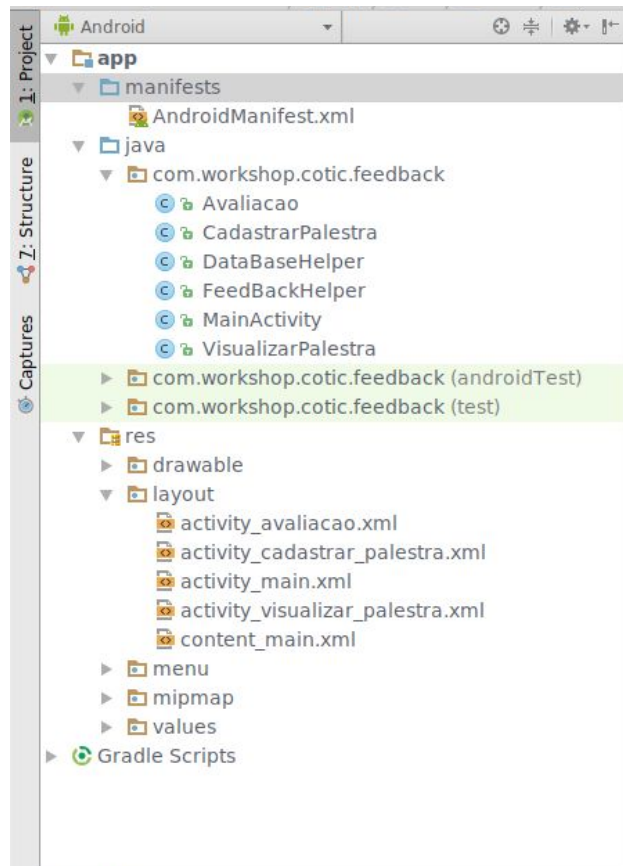
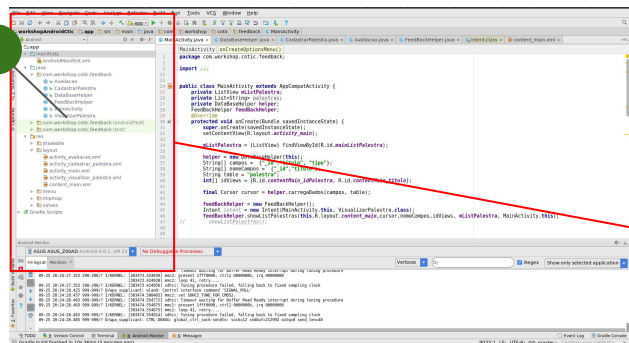


**E ai? Vai encarar?**

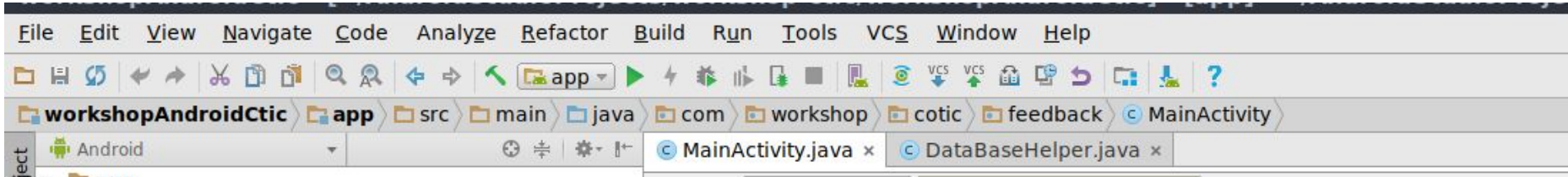
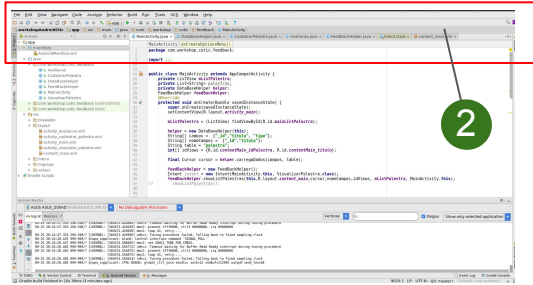
# Apresentação android Studio



# Estrutura de Diretórios

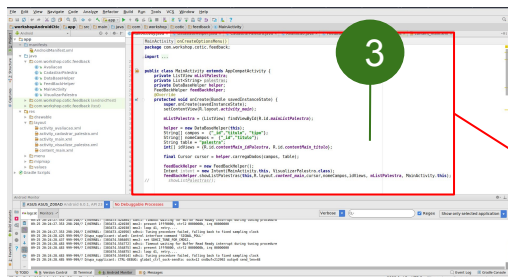


# Menus e Toolbars



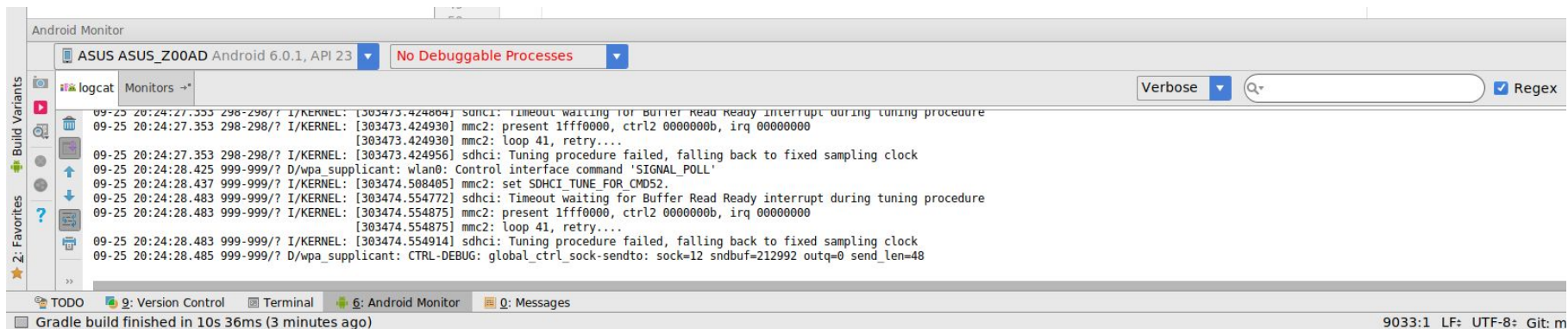
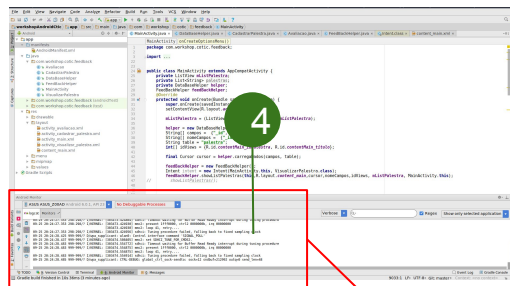


# Editor de Texto

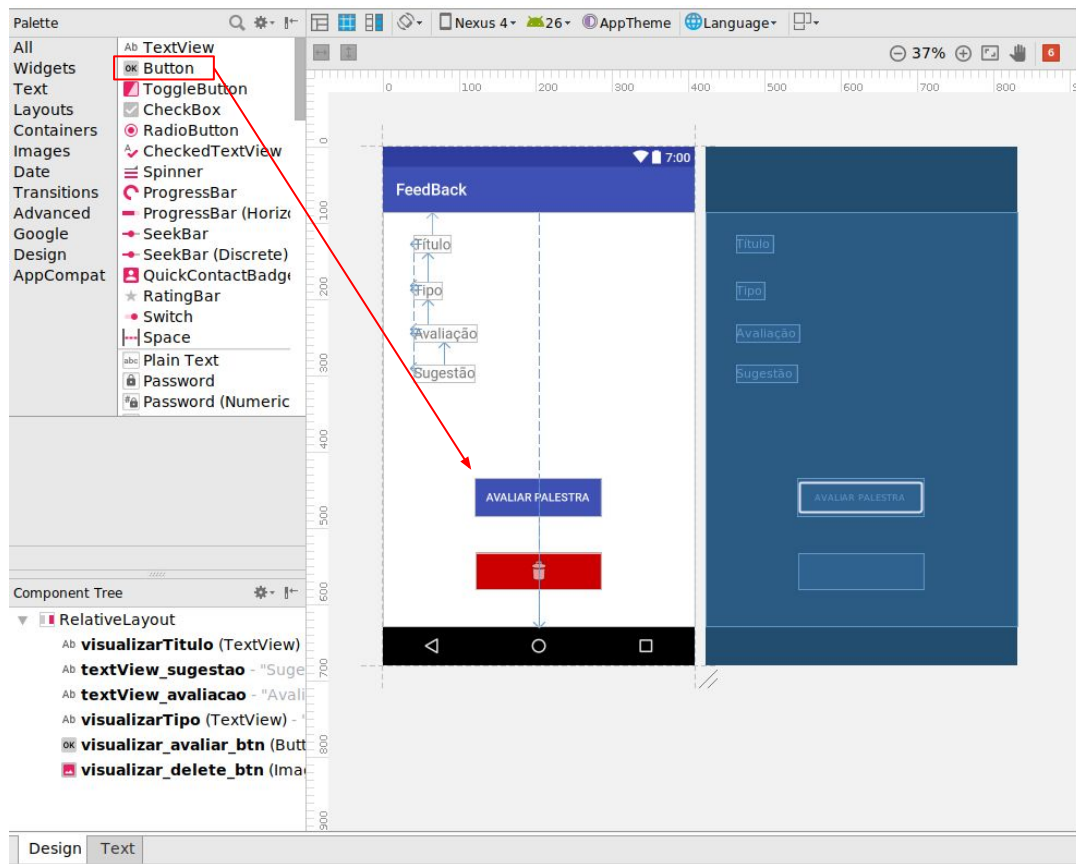


```
MainActivity onCreateOptionsMenu()  
package com.workshop.cotic.feedback;  
  
import ...  
  
public class MainActivity extends AppCompatActivity {  
    private ListView mListPalestra;  
    private List<String> palestras;  
    private DataBaseHelper helper;  
    FeedBackHelper feedBackHelper;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mListPalestra = (ListView) findViewById(R.id.mainListPalestra);  
  
        helper = new DataBaseHelper(this);  
        String[] campos = {"_id", "titulo", "tipo"};  
        String[] nomeCampos = {"_id", "titulo"};  
        String table = "palestra";  
        int[] idViews = {R.id.contentMain_idPalestra, R.id.contentMain_titulo};  
  
        final Cursor cursor = helper.carregaDados(campos, table);  
  
        feedBackHelper = new FeedBackHelper();  
        Intent intent = new Intent(MainActivity.this, VisualizarPalestra.class);  
        feedBackHelper.showListPalestras(this, R.layout.content_main, cursor, nomeCampos, idViews, mListPalestra, MainActivity.this);  
        showListPalestras();  
    }  
}
```

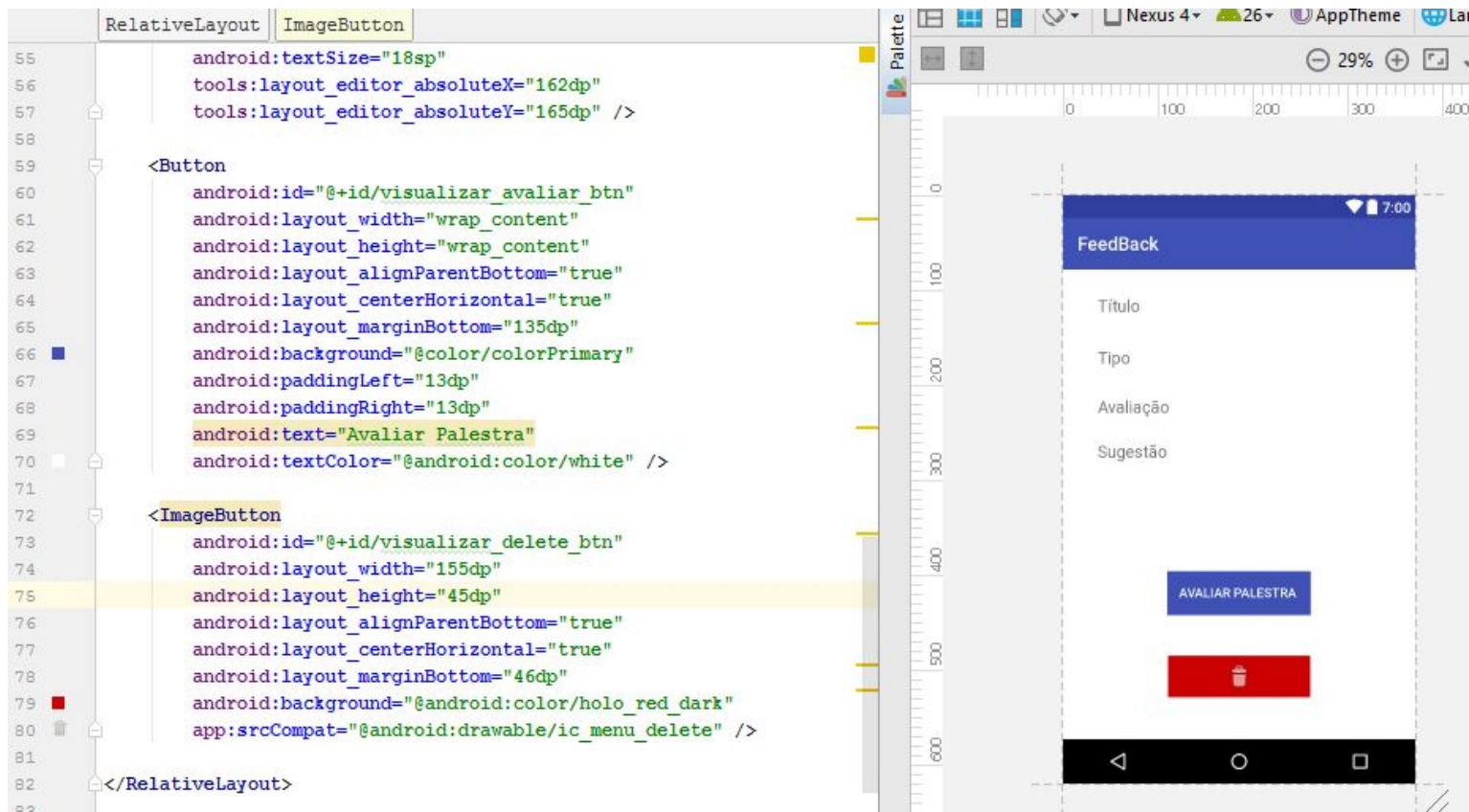
# Debug



# Prototipagem de tela



# Prototipagem de tela



# Activities



Título

Tipo

Avaliação

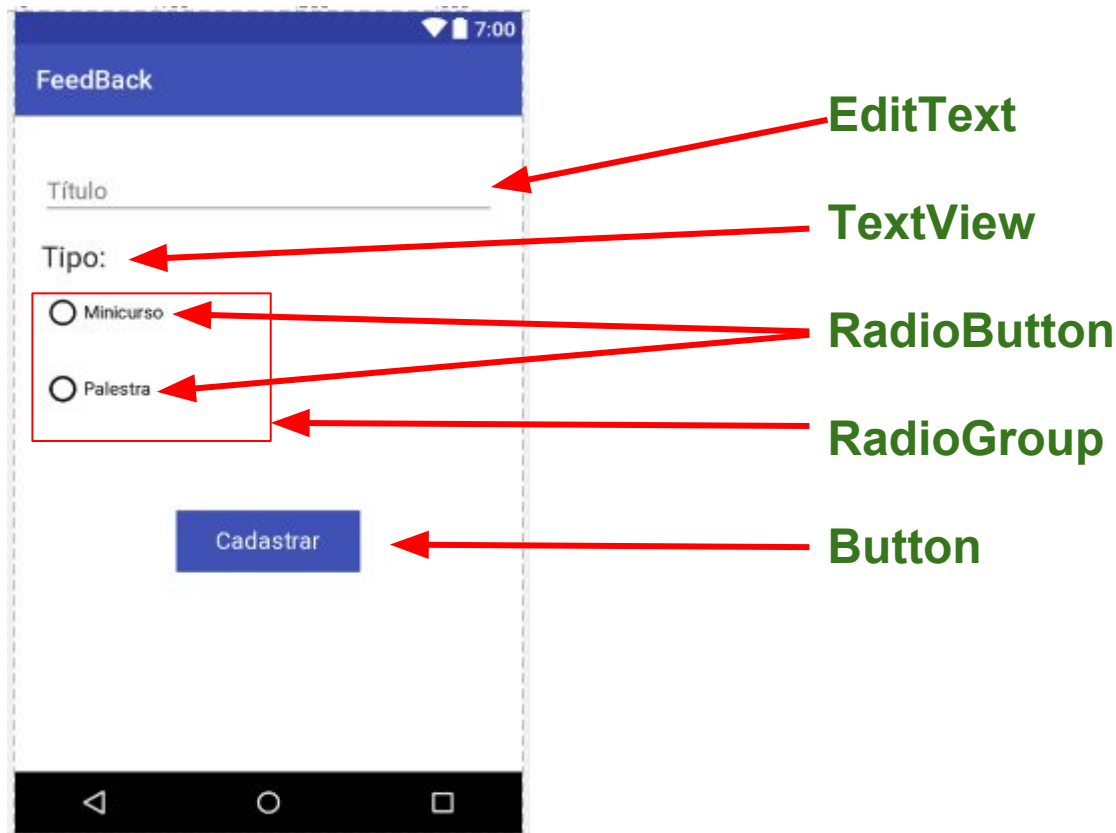
Sugestão

AVALIAR PALESTRA



```
VisualizarPalestra  onClick()
1  package com.workshop.cotic.feedback;
2
3  +import ...
19
20  public class VisualizarPalestra extends AppCompatActivity implements View.OnClickListener {
21
22      Bundle extras;
23      TextView textViewTitulo;
24      TextView textViewTipo;
25      TextView textViewAvaliacao;
26      TextView textViewSugestao;
27
28      DataBaseHelper helper;
29      private AlertDialog.Builder dialog;
30      private Button mAvaliarButton;
31      private ImageButton mApagarPalestra;
32
33      @Override
34      protected void onCreate(Bundle savedInstanceState) {...}
55
56      private void buildInformationAvaliacao(int idPalestra) {...}
82
83      private void buildInformationPalestra(int idPalestra) {...}
100
101
102      @Override
```

# Componentes de UI



# Navegar entre Activities (Intents)



```
Intent navegar = new Intent(Activity1.this, Activity2.class);  
navegar.putExtra("textoExtra", texto); //Informação mandada para segunda activity  
startActivity(navegar);
```

SQLite







**VAMOS  
CODAR!!!!**



# Obrigado



[contato.coyoti@gmail.com](mailto:contato.coyoti@gmail.com)