Android: ListView

Prof. Rodrigo Barbosa rodrigo.bezerra@unibratec.edu.br

ListView

- Componente visual utilizado para representar um conjunto de elementos em forma de Lista
- É o componente utilizado para exibir lista para o usuário
- O ListView, tem como vantagem, a possibilidade de criar um layout customizado para aplicar em cada linha da lista

ListView

- Para criar listas que contém apenas textos, pode ser usado a classe ArrayAdapter

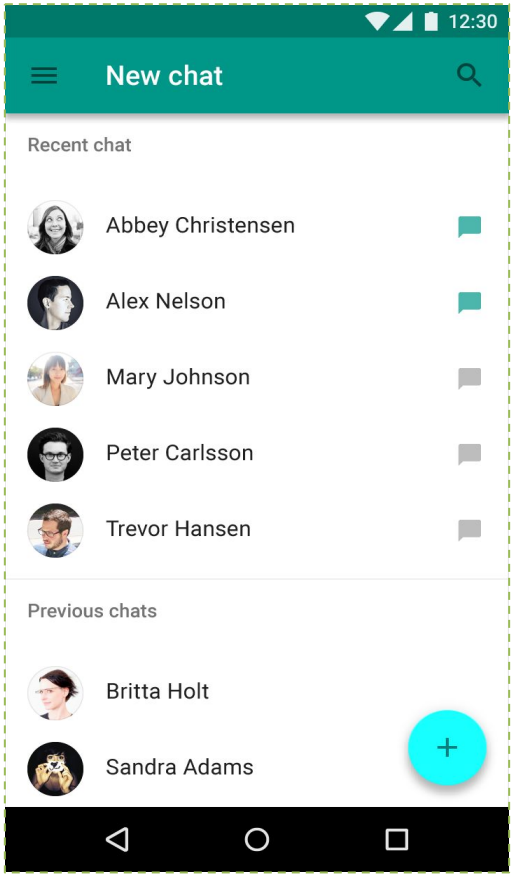
```
ArrayAdapter<Team> lista = new ArrayAdapter<>(
                this, android.R.layout.simple_list_item_1, itens);

itensListView.setAdapter(lista);
```

FACULDADE IBRATEC®

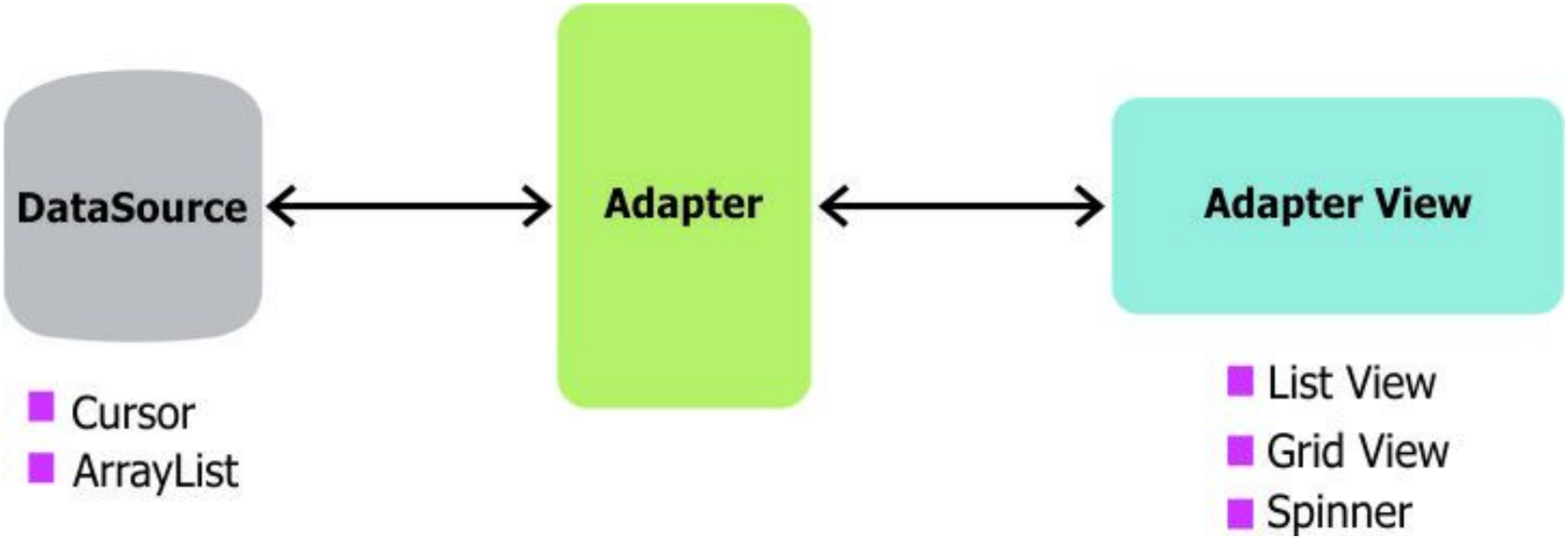
Exercício 15

Criar uma ListView

ListView Customizada



Lista e Adapter



Customizando o Adapter

- Definir um layout para representar nossa Linha
- Criar nosso Adapter de forma eficiente
- Usar a classe ViewHolder
- Atribuir nosso Adapter customizado ao ListView

LayoutInflater

- Permite carregar um arquivo de Layout em um objeto da classe android.view.View e opcionalmente atribuí-lo a um ViewGroup

```
View view = LayoutInflater.from(context).inflate(R.layout.activity_detalhe, null);
```

```
public class MeuAdapter extends BaseAdapter {

    public int getCount() {
        return <qtde_itens_lista>;
    }

    public Object getItem(int position) {
        return <objeto_da_posicao>;
    }

    public long getItemId(int position) {
        return <id_do_objeto>;
    }

    public View getView(int position, View convertView, ViewGroup parent) {
        return <view_que_representa_o_objeto>;
    }
}
```

```
public class MyCustomAdapter extends BaseAdapter {

    private Context context;
    private ArrayList<Language> languages

    public MyCustomAdapter(Context context, ArrayList<Language> languages) {
        this.context = context;
        this.languages = languages;
    }

    public int getCount() {
        return languages.size();
    }

    public Object getItem(int i) {
        return languages.get(i);
    }

    public long getItemId(int i) {
        return i;
    }
```

```
    @Override
    public View getView(int position, View view, ViewGroup viewGroup) {
        Language currentLanguage = languages.get(position);

        LayoutInflater inflater = LayoutInflater.from(context);
        View listLine = inflater.inflate(R.layout.list_item, null);

        ImageView logo = listLine.findViewById(R.id.img_logo);
        TextView name = listLine.findViewById(R.id.txt_language);

        logo.setImageResource(currentLanguage.getIcon());
        name.setText(currentLanguage.getLanguageName());

        return listLine;
    }
}
```

FACULDADE IBRATEC®

Exercício 16

Criar uma ListView Customizada

ViewHolder

- Mecanismo criado para otimização da criação das views
- Cria apenas as views que estão sendo exibidas. A medida que realizar um scroll novos itens serão criados
- Através da Tag é possível recuperar as views já criadas anteriormente

```
    @Override
    public View getView(int position, View view, ViewGroup viewGroup) {
        Language currentLanguage = languages.get(position);

        LayoutInflater inflater = LayoutInflater.from(context);
        View listLine = inflater.inflate(R.layout.list_item, null);

        ImageView logo = listLine.findViewById(R.id.img_logo);
        TextView name = listLine.findViewById(R.id.txt_language);

        logo.setImageResource(currentLanguage.getIcon());
        name.setText(currentLanguage.getLanguageName());

        return listLine;
    }
}
```

```
public View getView(int position, View view, ViewGroup viewGroup) {
    Language currentLanguage = languages.get(position);
    ViewHolder holder;

    if (view == null) {
        LayoutInflater inflater = LayoutInflater.from(context);
        view = inflater.inflate(R.layout.list_item, null);
        holder = new ViewHolder();
        holder.logo = view.findViewById(R.id.img_logo);
        holder.name = view.findViewById(R.id.text_language);
        view.setTag(holder);
    } else {
        holder = (ViewHolder) view.getTag();
    }

    holder.logo.setImageResource(currentLanguage.getIcon());
    holder.name.setText(currentLanguage.getLanguageName());

    return view;
}

static class ViewHolder {
    ImageView logo;
    TextView name;
}
```

FACULDADE IBRATEC®

Exercício 17

Utilizar o ViewHolder