

APLICAÇÃO MOBILE

ARM

PROGRAMAÇÃO III

Alunas: Adriana Mateus 11029
Carolina Abade 10992
Carolina Saraiva 10971

DEFINIÇÃO DO TEMA

Optámos por desenvolver uma aplicação mobile com a finalidade de ser utilizada nos jardins de infância como “auxiliar de memória” aos educadores/secretaria.

Esta aplicação designa-se ARM (aplicação de registo de medicação), tal como o nome indica é para registar/consultar os dados pessoais de cada criança da instituição e em diversos casos, quando alguma se encontra a tomar medicação, seja registado.

OBJETIVO/UTILIDADE DA APLICAÇÃO MOBILE

Esta aplicação serve para que os educadores e as auxiliares tenham um maior controlo sobre a situação de cada criança. Ou seja, é preenchida uma ficha com os dados pessoais de cada uma e em caso de estarem a ser medicadas por algum motivo é inserido um registo, pela auxiliar/educador(a) ou secretaria, que vai guardar o medicamento que foi administrado, as suas características, a quantidade e o motivo pelo qual foi administrado bem como quem deu e as horas e data a que foi dado.

Este registo é guardado numa tabela que contém todos os registos de medicação referentes à criança, onde pode ser consultado a qualquer altura e sempre que necessário.

Conclui-se que com esta aplicação mobile, existirá maior controlo sobre a administração de medicação a cada criança pois serve como auxiliar para, entre educadores e auxiliares de sala, saber se foi dado ou não algum medicamento à criança e também para esclarecimento de alguma dúvida aos pais.

TEMPLATE E DESIGN

Por se tratar de uma profissão de algum stress e muita confusão, para o site escolhemos cores mais neutras, como o branco e tons de cinza, e utilizamos alguns apontamentos de azul, que segundo a psicologia das cores transmite tranquilidade e confiança, de formar a eliminar todo o barulho da aplicação mobile e tornar as operações claras ao utilizador e ser mais intuitivo.

O logótipo é da nossa autoria, foi criado no programa Adobe Illustrator. Na sua construção optámos por utilizar a mesma paleta de cores para se enquadrar no site e porque se trata de uma aplicação, não de entretenimento, mas administrativa. No logótipo está a ideia de uma folha com o símbolo do termómetro, que nos remete à medicação, e de uma chupeta que nos remete às crianças. Nas imagens seguintes pode observar-se o logótipo e os ícones utilizados na aplicação mobile, assim como a paleta de cores.

LOGÓTIPO / ÍCONES MENÚ



LOGÓTIPO / ÍCONES MENÚ

Menú funcionários



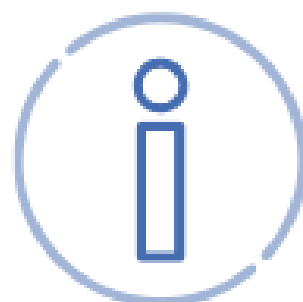
Dados pessoais



Lista de Crianças



Medicação



Sobre



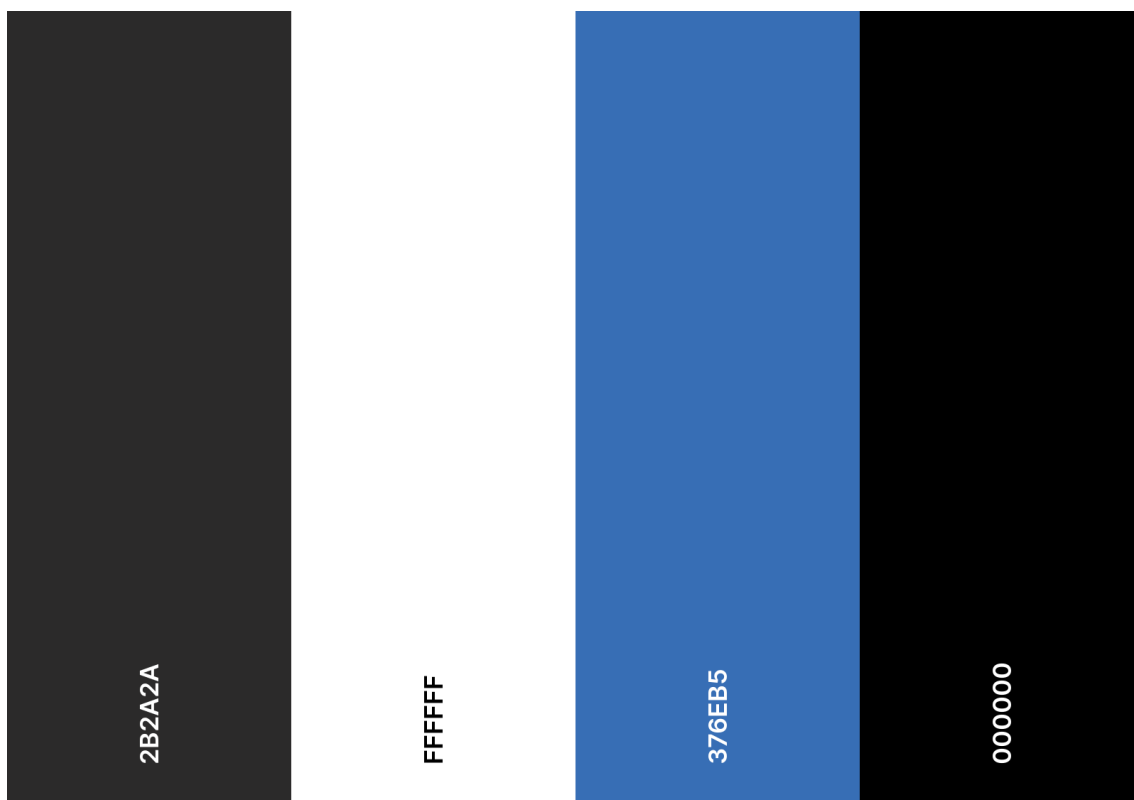
Ajuda

LOGÓTIPO / ÍCONES MENÚ

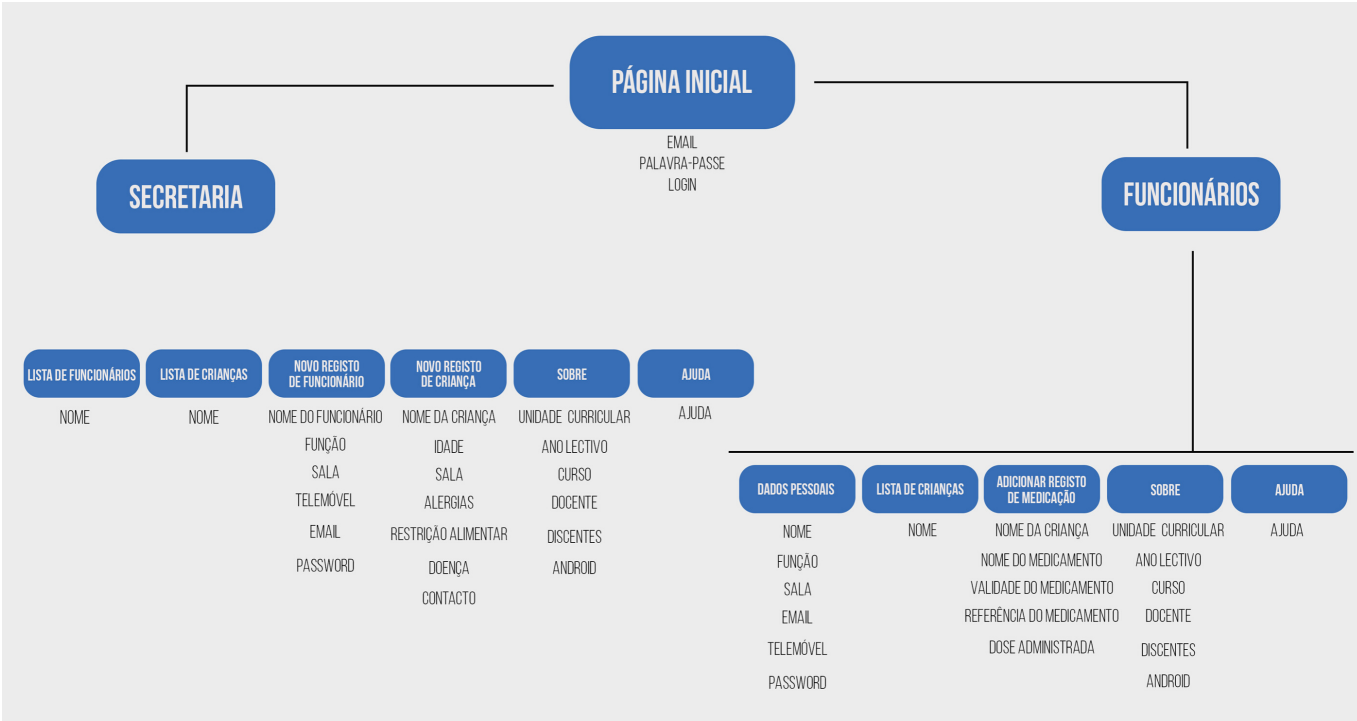
Menú secretaria



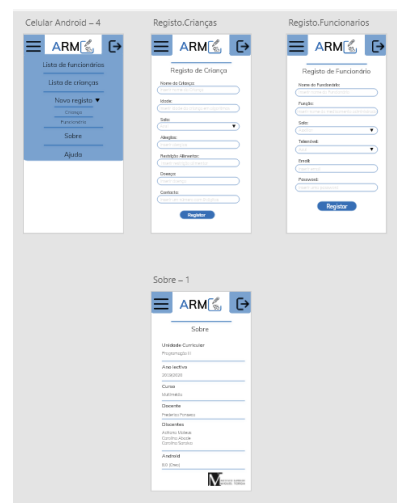
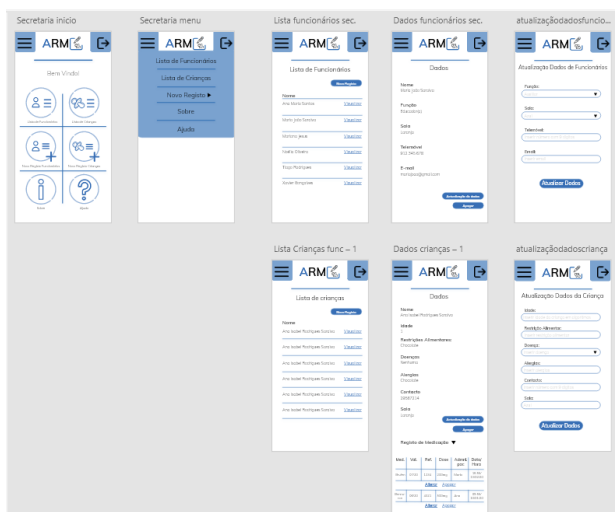
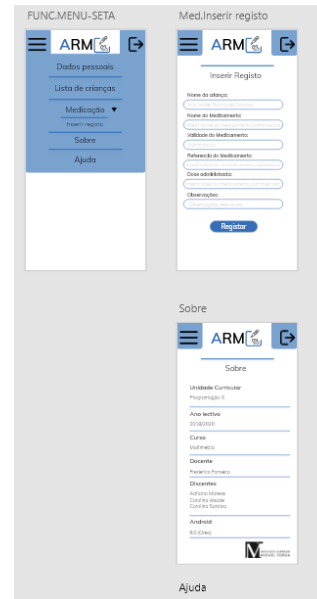
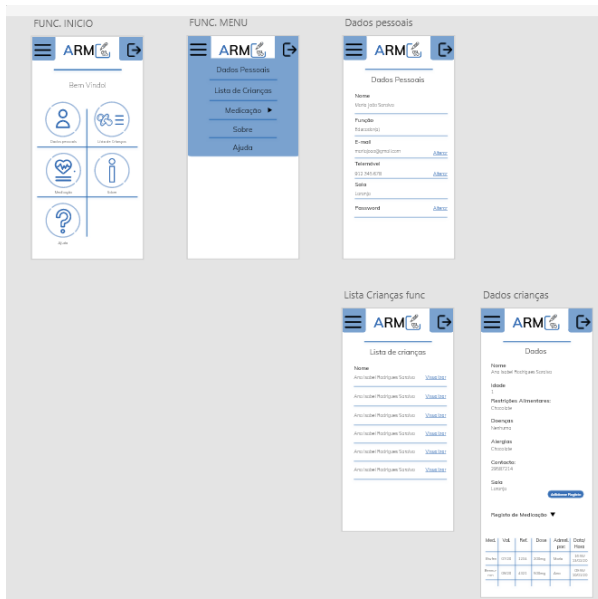
PALETA DE CORES



SITE MAP



DESIGN DA APLICAÇÃO



ANDROID

CÓDIGO QUE PERMITE FAZER TRANSIÇÃO PARA A ÁREA DO FUNCIONÁRIO OU PARA A ÁREA DA SECRETARIA:

```
private void validate (String userName){  
  
    if (userName.equals("staff")) {  
  
        Intent entrarActivity = new Intent(this, FuncionarioActivity.class);  
startActivity(entrarActivity);    }else if(userName.equals("sec")) {  
  
        Intent entrarActivity = new Intent(this, SecretariaActivity.class);  
startActivity(entrarActivity);  
  
    }else {  
  
        }  
}
```

ANDROID

CÓDIGO QUE, AO CARREGAR NO BOTÃO, PERMITE FAZER A TRANSIÇÃO ENTRE FRAGMENTS:

```
Button btn_listcri = (Button)
root.findViewById(R.id.btn_listcri);btn_listcri.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View root) {
        ListaCriançasFragment listaCriançasFragment = new
        ListaCriançasFragment();
        FragmentManager manager = getFragmentManager();
        manager.beginTransaction().replace(R.id.nav_host_fragment,
        listaCriançasFragment, listaCriançasFragment.getTag()).commit();
    }
});
```

NODE

DEFINIÇÃO DE ROTAS:

//endpoints colaboradores

```
router.get('/colaboradores',  
  armController.colab_list);router.get('/colaborador/:id_colaborador',  
  armController.colab_detail);router.post('/colab',  
  armController.colab_create);router.put('/colaborador/:id_colaborador',  
  armController.colab_update);router.delete('/colaborador/:id_colaborador',  
  armController.colab_delete);
```

//endpoints criancas

```
router.get('/criancas',  
  armController.crianca_list);router.get('/crianca/:id_crianca', armController.  
  crianca_detail);router.post('/crianca',  
  armController.crianca_create);router.put('/crianca/:id_crianca',  
  armController.crianca_update);router.delete('/crianca/:id_crianca',  
  armController.crianca_delete);
```

//endpoints registo_medicacao

```
router.get('/registos',  
  armController.registo_list);router.get('/registo/:id_registo', armController.r  
  egisto_detail);router.post('/registo',  
  armController.registo_create);router.put('/registo/:id_registo',  
  armController.registo_update);router.delete('/registo/:id_registo',  
  armController.registo_delete); module.exports = router;
```

NODE

CÓDIGO QUE CHAMA AS ROTAS:

```
//rotas API (arm)  
Const armApi = require('./routes/arm.route');  
app.use('/api/v1', armApi);
```

CÓDIGO QUE PERMITE CRIAR UMA NOVA CRIANÇA:

```

Button inserir = (Button) root.findViewById(R.id.button13);
inserir.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //validação das caixas de texto
        if (txt_nome_c.getText().toString().length() == 0) {
            txt_nome_c.setError("É necessário preencher o nome do aluno.");
            txt_nome_c.requestFocus();
        } else if (txt_idade.getText().toString().length() == 0) {
            txt_idade.setError("É necessário preencher o email do aluno.");
            txt_idade.requestFocus();
        } else if (txt_sala.getText().toString().length() == 0) {
            txt_sala.setError("É necessário preencher a morada do aluno.");
            txt_sala.requestFocus();
        } else if (txt_alergias.getText().toString().length() == 0) {
            txt_alergias.setError("É necessário preencher o telefone do aluno.");
            txt_alergias.requestFocus();
        } else if (txt_restricao.getText().toString().length() == 0) {
            txt_restricao.setError("É necessário preencher o telefone do aluno.");
            txt_restricao.requestFocus();
        } else if (txt_doenca.getText().toString().length() == 0) {
            txt_doenca.setError("É necessário preencher o telefone do aluno.");
            txt_doenca.requestFocus();
        } else if (txt_contacto.getText().toString().length() == 0) {
            txt_contacto.setError("É necessário preencher o telefone do aluno.");
            txt_contacto.requestFocus();
        } else {
            //dados a enviar para a API (formato BODY)
            String dados =
                "nome_crianca=" + txt_nome_c.getText() + "&idade=" + txt_idade.getText() +
                "&sala=" + txt_sala.getText() + "&alergias=" + txt_alergias.getText() + "&restricao="
                + txt_restricao.getText() + "&doenca_cronica=" + txt_doenca.getText() +
                "&contacto=" + txt_contacto.getText();

            //executa o pedido à API
            _api = new ApiConnection();
            SecaddCriançasFragment.this;
            _api._activity =
                _api._listaCrianças = new ArrayList();
            _api.execute("http://10.0.2.2:3001/api/v1/crianca", "1", dados);
        }
    }
}

```

CÓDIGO QUE IMPRIME UMA LIST VIEW A LISTA DE TODAS AS CRIANÇAS E PERMITE MUDAR DE FRAGMENT AO CLICAR

```
public void updateUI() {
    //lista onde irá ficar armazenado a string para mostrar na lista (listView)
    final ArrayList<String> dadosLista = new ArrayList<>();

    //ciclo que percorre todos as crianças retornados pela API
    for (int i = 0; i < _api._listaCrianças.size(); i++) {
        //variável que guarda os dados da criança (no formato key-value-pair)
        HashMap<String, String> crianca = _api._listaCrianças.get(i);
        lv.setOnItemClickListener(new android.widget.AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id)
            {

                SecDadosCriançasFragment secDadosCriançasFragment = new
                SecDadosCriançasFragment();
                FragmentManager manager = getFragmentManager();
                manager.beginTransaction().replace(R.id.nav_host_fragment_sec,
                secDadosCriançasFragment, secDadosCriançasFragment.getTag()).commit();
            }
        });

        //string com os dados a mostrar na lista no formato - nome
        String nome_crianca = crianca.get("nome_crianca");
        dadosLista.add(nome_crianca);
    }

    lv.setAdapter(new ArrayAdapter<String>(getContext(),
    android.R.layout.simple_expandable_list_item_1, dadosLista));
}
```


CÓDIGO QUE PERMITE ACTUALIZAR OS DADOS DE UMA CRIANÇA:

```

Button atualizar = (Button) root.findViewById(R.id.button13);

atualizar.setOnClickListener(new View.OnClickListener() {           public void
onClick(View v) {
//validação das caixas de texto
if (txt_idade.getText().toString().length() == 0) {
txt_idade.setError("É necessário preencher o email do aluno.");
txt_idade.requestFocus();
} else if (txt_sala.getText().toString().length() == 0) {
txt_sala.setError("É necessário preencher a morada do aluno.");
txt_sala.requestFocus();
} else if (txt_alergias.getText().toString().length() == 0) {
txt_alergias.setError("É necessário preencher o telefone do aluno.");
txt_alergias.requestFocus();
} else if (txt_restricao.getText().toString().length() == 0) {
txt_restricao.setError("É necessário preencher o telefone do aluno.");
txt_restricao.requestFocus();
} else if (txt_doenca.getText().toString().length() == 0) {
txt_doenca.setError("É necessário preencher o telefone do aluno.");
txt_doenca.requestFocus();
} else if (txt_contacto.getText().toString().length() == 0) {
txt_contacto.setError("É necessário preencher o telefone do aluno.");
txt_contacto.requestFocus();
} else {
//dados a enviar para a API (formato BODY)
String dados = "idade=" + txt_idade.getText() + "&sala=" + txt_sala.getText() +
"&alergias=" + txt_alergias.getText() + "&restricao=" + txt_restricao.getText() +
"&doenca_cronica=" + txt_doenca.getText() + "&contacto=" + txt_contacto.getText();

//executa o pedido à API
_api = new ApiConnection();
_api._activity = SecActCriancaFragment.this;
_api._listaCrianças = new ArrayList();
_api.execute("http://10.0.2.2:3001/api/v1/crianca/2", "2", dados);
}
}

```

CONCLUSÃO

Com o desenvolvimento desta aplicação mobile, utilizamos os conhecimentos que adquirimos ao longo das aulas, do suporte dos slides que nos foi fornecido através do moodle, e alguns tutoriais referidos na bibliografia. A parte do back end foi realizada com base no exercício dado em aula. Na parte android sentimos dificuldade na transição entre as activities e fragments, contudo conseguimos resolver esse problema. Na conexão da API, sentimos também dificuldades, pois apesar de acharmos que tínhamos todo o código necessário para funcionar, não conseguimos detetar de imediato o problema. Após a ajuda do professor, conseguimos estabelecer essa conexão mas, deparámo-nos novamente com dificuldades em aplicar o código necessário para a execução de operações na base de dados através do Android. Conseguimos arranjar soluções para a consulta, actualização e criação de dados referente à tabela da crianças. Ficou em falta a aplicação do delete, para apagar um registo na base de dados, fazer a transição de dados entre fragments, aplicar condições e repetir estes processos para as restantes tabelas. Em suma, apesar de todo o esforço, para tornar a aplicação funcional, não nos foi possível, pois não conseguimos ultrapassar os problemas apresentados e, consequentemente não tivemos tempo para replicar o código para as restantes situações.

BIBLIOGRAFIA

- <https://www.youtube.com/watch?v=CVME9yW54mY>
- <https://www.youtube.com/watch?v=TY-2Cx4IW9A>
- <https://www.youtube.com/watch?v=oHnWuTeaTu4>
- <https://stackoverflow.com/questions/1051004/how-to-send-put-delete-http-request-in-httpurlconnection?fbclid=IwAR26Q7hSTeN9pGcTwiffVUpuidDd10VBhdt7-lzWaVzkFCTfYZ0NXftg38g>
- https://www.youtube.com/watch?v=eKYBUuxM3aA&fbclid=IwAR1O45C0aF51C-iRu_C1drcMtZCz7yAf5UMnf-ypUE7yEoYxUDSfFRN_AwY
- <https://stackoverflow.com/questions/15480857/adding-button-listener-in-arraylist?fbclid=IwAR0aNTJ2FglnfQn3sruxK9P7ODfNntnfvg1629WfyM-VJVam-VRrYdJTcwI>