

# **Interação Pessoa-Computador**

## **Trabalho de Grupo 02**

Elaborado por:

Fábio Gonçalves nº 17646

Luís Silva nº 19314

Docente:

Luis Garcia

30/06/2021

## Conteúdo

Lista de figuras.....	1
1. Introdução .....	2
2. Protótipo de Alta Fidelidade .....	2
3. Conclusões .....	16

## Lista de figuras

Figura 1 "BottomNavigationView" para troca de ecrãs.....	2
Figura 2 - Código da interface de navegação.....	3
Figura 3 - Interface principal da aplicação.....	4
Figura 4 – Registo.....	5
Figura 5 - Login .....	5
Figura 6 - Código implementado (Registo) .....	6
Figura 7 - Código implementado (Login).....	7
Figura 8 - Procurar Surf Camp's .....	8
Figura 9 - Visualizar Informações sobre o Surf Camp .....	9
Figura 10 - Consultar Ofertas.....	10
Figura 11 - Pagamento .....	11
Figura 12 - Exemplo de código da interface de Procurar.....	12
Figura 13 - Exemplo de código da interface de visualizar informações .....	12
Figura 14 - Código implementado na interface de Ofertas .....	13
Figura 15 - Código implementado na interface de Pagamento .....	14
Figura 16 Ecrã de Suporte.....	15
Figura 17 Ecrã de Edição de Perfil .....	15

## 1. Introdução

Com este trabalho de grupo pretende-se realizar um protótipo parcial do sistema interativo que foi desenhado no trabalho anterior.

Não era pedido que fosse desenvolvida e implementada toda a interface do sistema, mas sim algumas das partes mais importantes, de forma que o funcionamento geral da aplicação fosse correto. O sistema foi desenvolvido na plataforma Android com o emulador Pixel 4.

Neste trabalho era ainda necessário realizar uma primeira avaliação do protótipo construído.

## 2. Protótipo de Alta Fidelidade

A elaboração do projeto foi um resultado de vários altos e baixos. Inicialmente o primeiro passo foi a implementação de uma ferramenta que fizesse com que fosse possível trocar de ecrãs, assim sendo, encontramos a “BottomNavigationView”. Esta ferramenta permite criar uma espécie de menu horizontal com botões, para que torne a navegação mais agradável.



*Figura 1 "BottomNavigationView" para troca de ecrãs*

Esta ferramenta ficou sempre no mesmo ecrã, mudando apenas a parte restante do ecrã, recorrendo a “fragments”. O código para que esta ferramenta pudesse mudar de “fragments” foi o seguinte:

Como podemos visualizar na figura abaixo podemos verificar o código que foi usado para a implementação da interface de navegação que se encontra presente em toda a aplicação para fornecer uma simples e fácil navegação ao utilizador que navega pela aplicação.

```
public class HomePageActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_homepage);

        BottomNavigationView bottomNav = findViewById(R.id.bottom_navigation);

        bottomNav.setOnNavigationItemSelectedListener(navListener);

        getSupportFragmentManager().beginTransaction().replace(R.id.homepage_layout, new HomeFragment()).commit();
    }

    private BottomNavigationView.OnNavigationItemSelectedListener navListener =
        new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                Fragment selectedFragment = null;

                switch (item.getItemId()) {
                    case R.id.nav_home:
                        selectedFragment = new HomeFragment();
                        break;
                    case R.id.nav_search:
                        selectedFragment = new SearchFragment();
                        break;
                    case R.id.nav_support:
                        selectedFragment = new SuporteFragment();
                        break;
                    case R.id.nav_profile:
                        selectedFragment = new ProfileFragment();
                        break;
                }
                getSupportFragmentManager().beginTransaction().replace(R.id.homepage_layout, selectedFragment).commit();

                return true;
            }
        };
}
```

Figura 2 - Código da interface de navegação

Após a criação e implementação da interface de navegação no sistema, foi feita a criação de seguida das interfaces de registo assim como a de login para que cada utilizador possa ter acesso à aplicação, com os seus determinados dados de acesso em que após isso os dados recolhidos serão inseridos no sistema.

Na figura abaixo podemos verificar a interface que o utilizador irá ser confrontado ao abrir a aplicação em que terá a opção de fazer o seu registo caso ainda não o tenha efetuado ou o login caso seja um utilizador recorrente.



*Figura 3 - Interface principal da aplicação*

De seguida podemos verificar os dois cenários, em que estes são caso o utilizador opte por fazer o registo ou o login, ambos são abordados nas figuras abaixo, em que ao fazer o registo o utilizador terá de inserir algumas informações básicas (nome de utilizador único assim como um endereço de email e uma palavra-passe), após isso só terá de introduzir o nome de utilizador e palavra-passe previamente criados para efetuar o login e assim entrar na aplicação.



SurfCampsApp

Utilizador:

E-mail:

Password:

Confirmar Password:

**REGISTAR**

*Figura 4 – Registo*

Utilizador:

Password:

**ENTRAR**

*Figura 5 – Login*

Para o login, foi usado um ficheiro de texto, que irá ser gerado em cada smartphone, com os dados que foram colocados no registo.

Nas figuras abaixo podemos visualizar o principal código que foi implementado em cada uma das interfaces de tanto login como registo na aplicação.

```
public void register(){
    String FILENAME = "logins_file";
    String username = user.getText().toString() + "\n";
    String mail = email.getText().toString() + "\n";
    String password = pass.getText().toString() + "\n";
    String confPassword = confPass.getText().toString() + "\n";
    String division = "- \n";

    if (username.length() > 1 && mail.length() > 1 && password.length() > 1) {
        if (password.equals(confPassword)) {
            FileOutputStream fos = null;
            try {
                File directory = getFilesDir();
                File file = new File(directory, FILENAME);
                fos = new FileOutputStream(file, append: true);

                fos.write(username.getBytes());
                fos.write(mail.getBytes());
                fos.write(password.getBytes());
                fos.write(division.getBytes());

                fos.flush();
                fos.close();
            } catch (FileNotFoundException e) {
                e.printStackTrace();
                Toast.makeText( context this, text: "O ficheiro falhou!", Toast.LENGTH_SHORT).show();
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText( context this, text: "Nao encontrou file", Toast.LENGTH_SHORT).show();
            }
            //EditText email = (EditText) findViewById(R.id.);
            Intent intent = new Intent( packageContext this, LoginActivity.class);
            startActivity(intent);
        }else{
            Toast.makeText(getApplicationContext(), text: "As passwords não correspondem!", Toast.LENGTH_SHORT).show();
        }
    }else{
        Toast.makeText(getApplicationContext(), text: "Preencha os campos!", Toast.LENGTH_SHORT).show();
    }
}
```

Figura 6 - Código implementado (Registo)

Figura 7 - Código implementado (Login)

```
public void makeLogin(){
    String FILENAME = "logins_file";
    String user = username.getText().toString();
    String pass = password.getText().toString();
    File path = getFilesDir();

    boolean b = false;
    String aBuffer = "";
    int counter = 0;
    try {
        File myFile = new File(path, FILENAME);
        BufferedReader myReader = new BufferedReader(new FileReader(myFile));
        String aDataRow = "";

        while ((aDataRow = myReader.readLine()) != null) {
            aBuffer += aDataRow + " ";
            if(counter % 4 == 0 && user.equals(aDataRow)){
                b = true;
                aDataRow = myReader.readLine();
                aBuffer += aDataRow + " ";
                aDataRow = myReader.readLine();
                aBuffer += aDataRow + " ";
                if(pass.equals(aDataRow)){
                    Intent login = new Intent( packageContext, HomePageActivity.class);
                    Toast.makeText( context, this, "Login Efetuado com Sucesso!", Toast.LENGTH_SHORT).show();
                    startActivity(login);
                }else{
                    Toast.makeText(getApplicationContext(), "As credenciais estão inválidas!", Toast.LENGTH_SHORT).show();
                }
                break;
            }
            counter++;
        }
        myReader.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    if(!b){
        Toast.makeText(getApplicationContext(), "As credenciais estão inválidas!", Toast.LENGTH_SHORT).show();
    }
}
```

Após o correto e devido funcionamento de ambas as interfaces registo e login,

seguimos para a implementação da interface de procura, em que esta tem a funcionalidade de procurar Surf Camps na proximidade do utilizador que utiliza a aplicação e com isto pode ver quais Surf Camp's estão disponíveis e prontos a visualizar.





*Figura 8 - Procurar Surf Camp's*

Após o utilizador visualizar os Surf Camp's que se encontram disponíveis a poderá então escolher o qual pretende visualizar e poderá então verificar as informações sobre o mesmo, em que esta informação inclui as condições da devida praia da localização onde se encontra o Surf Camp, como também uma breve descrição do mesmo.

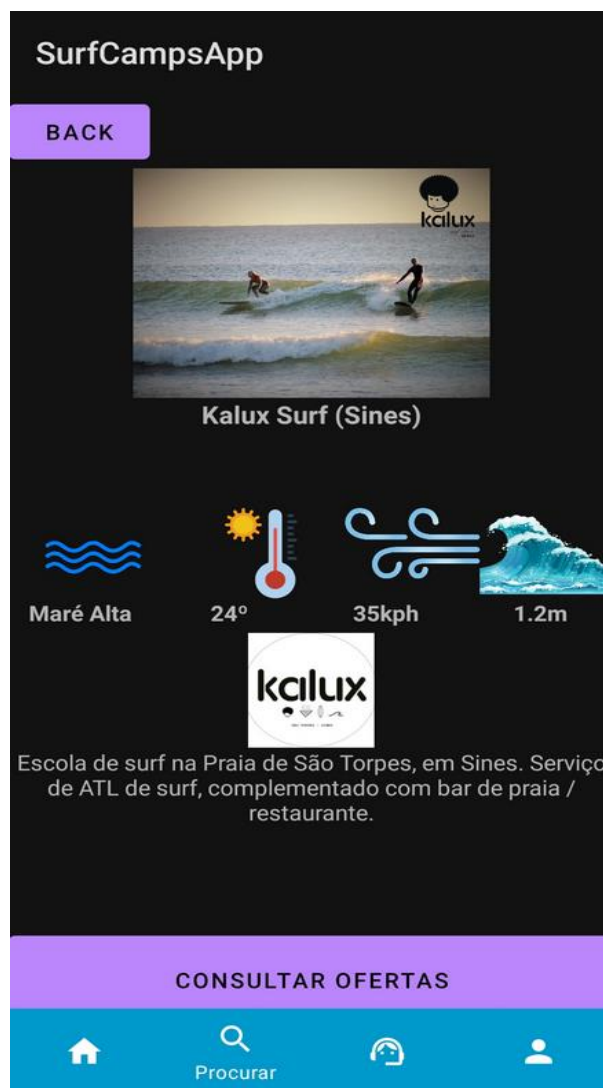


Figura 9 - Visualizar Informações sobre o Surf Camp

De seguida (se o utilizador assim pretender) o utilizador poderá então consultar as ofertas disponíveis para esse mesmo Surf Camp em que nestas ofertas estão incluídos diferentes pacotes em que estes disponibilizam de um número diferente em que o utilizador poderá passar a sua estadia em um deles, em que poderá também visualizar tudo o que inclui o preço para o qual estará a pagar, assim como fazer a marcação ao escolher o dia que deseja que a mesma seja efetuada.

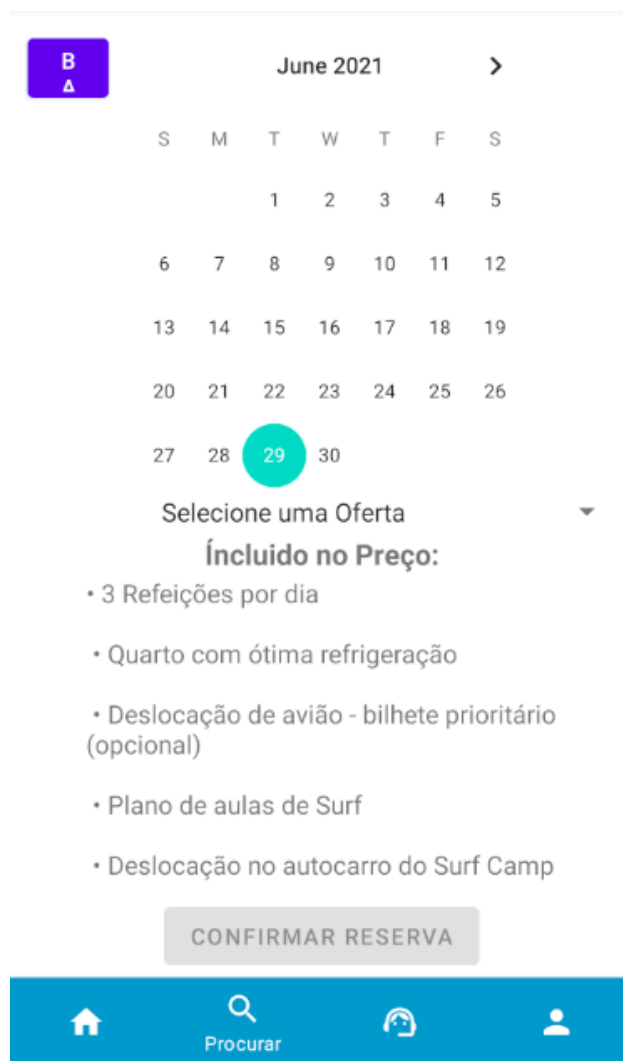


Figura 10 - Consultar Ofertas

Após o utilizador selecionar a data desejada que pretende marcar a reserva e seleciona o pacote desejado, poderá confirmar a reserva selecionando o botão “Confirmar Reserva” e assim o utilizador será então levado à próxima interface em que esta será para o utilizador efetuar o pagamento utilizando um dos meios de pagamento disponibilizados pela aplicação, em que poderá também visualizar a data e o pacote que escolheu no passo anterior, para em caso de erro poderá voltar atrás e alterar os mesmos.

Primeiro Nome:

Último Nome:

Data da Reserva: 23/6/2021

Pacote: 710€ (6 dias / 6 noites)

☒ Cartão de Crédito



☐ Paypal



☐ Google Pay



2135512553121523

07

2028

125

Utilizador Teste

PAGAR COM CARTÃO



Figura 11 - Pagamento

Nas figuras abaixo podemos visualizar o principal código que foi implementado em cada uma das quatro interfaces previamente apresentadas.

```

public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

    final View view = inflater.inflate(R.layout.fragment_search, container, attachToRoot: false);

    txtVkalux = view.findViewById(R.id.txtKalux);
    txtVAlentejo = view.findViewById(R.id.txtAlentejoSurfCamp);
    txtVericeira = view.findViewById(R.id.txtEriceiraSurfCamp);
    txtVmilfontes = view.findViewById(R.id.txtSurfMilfontes);

    kalux = view.findViewById(R.id.imgBtnStorpes);
    kalux.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Bundle bundle = new Bundle();
            nomeSurfCamp = "" + txtVkalux.getText().toString();
            bundle.putString("NomeSurfCamp", nomeSurfCamp);
            SurfCampInfoFragment surfCampInfoFragment = new SurfCampInfoFragment();
            surfCampInfoFragment.setArguments(bundle);
            FragmentTransaction transaction = getFragmentManager().beginTransaction();
            transaction.replace(R.id.homepage_layout, surfCampInfoFragment);
            transaction.commit();
        }
    });
}

```

Figura 12 - Exemplo de código da interface de Procurar

```

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

    final View view = inflater.inflate(R.layout.fragment_surf_camp_info, container, attachToRoot: false);

    nomeSurfCamp = view.findViewById(R.id.txtNomeSurfCamp);
    imagemSurfCamp = view.findViewById(R.id.imgSurfCamp);
    logoSurfCamp = view.findViewById(R.id.imgLogoSurfCamp);
    mare = view.findViewById(R.id.txtMare);
    temperatura = view.findViewById(R.id.txtTemperatura);
    vento = view.findViewById(R.id.txtVento);
    ondas = view.findViewById(R.id.txtAlturaOndas);
    descricao = view.findViewById(R.id.txtDescricao);

    Bundle bundle = getArguments();
    nomeSurfCamp.setText(String.valueOf(bundle.getString( key: "NomeSurfCamp")));
    nomeSurfCamp.toString();

    if (nomeSurfCamp.getText().equals("Kalux Surf (Sines)")) {
        imagemSurfCamp.setImageResource(R.drawable.kalux);
        logoSurfCamp.setImageResource(R.drawable.kaluxlogo);
        descricao.setText("Escola de surf na Praia de São Torpes, em Sines. Serviço de ATL de surf, complementado com bar de praia / restaurante.");
    }

    else if (nomeSurfCamp.getText().equals("Alentejo Surf Camp (Vila Nova de Milfontes)")) {
        imagemSurfCamp.setImageResource(R.drawable.alentejosurfcamp);
        logoSurfCamp.setImageResource(R.drawable.alentejosurfcamp_logo);
        descricao.setText("Reconhecido como um dos melhores Surf Camps em Portugal, oferecemos cursos de surf de alto rendimento para iniciantes e intermediários para garantir o seu R");
    }

    else if (nomeSurfCamp.getText().equals("Ericeira SurfCamp (Ericeira)")) {
        imagemSurfCamp.setImageResource(R.drawable.ericeirasurf);
        logoSurfCamp.setImageResource(R.drawable.ericeiralogo);
        descricao.setText("A Ericeira Surf School promove surf tours, aulas de grupo, privadas e para crianças no âmbito do ensino e do treino, para todos os surfistas.");
    }

    else if (nomeSurfCamp.getText().equals("SurfMilfontes (Vila Nova de Milfontes)")) {
        imagemSurfCamp.setImageResource(R.drawable.surfmilfontes);
        logoSurfCamp.setImageResource(R.drawable.surfmilfonteslogo);
        descricao.setText("Queremos também promover o interesse dos jovens da zona de Vila Nova de Milfontes por um desporto que valorize não só as potencialidades do local onde vivem");
    }
}

```

Figura 13 - Exemplo de código da interface de visualizar informações

```

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
    Spinner spinner2 = (Spinner) view.findViewById(R.id.spinnerOffers);

    pacote = parent.getItemAtPosition(position).toString();

    if (pacote.equals("Selecione uma Oferta")){
        Toast.makeText(getContext(), text: "Selecione uma Oferta.", Toast.LENGTH_SHORT).show();
        buttonReserva.setEnabled(false);
        //buttonReserva.setClickable(false);
    }

    else {
        buttonReserva.setEnabled(true);
        //buttonReserva.setClickable(true);
    }
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
}

public void onClickReserva(View view)
{
    Bundle payment = new Bundle();
    payment.putString("date", date);
    payment.putString("pack", pacote);
    PaymentFragment paymentFragment = new PaymentFragment();
    paymentFragment.setArguments(payment);
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.homepage_layout, paymentFragment);
    transaction.commit();
}

public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

    final View view = inflater.inflate(R.layout.fragment_offers, container, attachToRoot: false);

    Spinner spinner = (Spinner) view.findViewById(R.id.spinnerOffers);
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(getContext(), R.array.ofertas, android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
    spinner.setOnItemSelectedListener(this);

    buttonReserva = view.findViewById(R.id.buttonReserva);
    calendarView = view.findViewById(R.id.calendarView);
    buttonBack = view.findViewById(R.id.buttonBack);

    calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
        @Override
        public void onSelectedDayChange(@NonNull CalendarView view, int year, int month, int dayOfMonth) {
            date = dayOfMonth + "/" + month + "/" + year;
        }
    });

    buttonBack.setOnClickListener(v -> {
        onClickBack(view);
    });
}

```

Figura 14 - Código implementado na interface de Ofertas

```

public void onClickPay(View view)
{
    String confirmed = "A sua encomenda foi confirmada com sucesso!\nEncomenda ID: E092X8M2B1";
    Bundle message = new Bundle();
    message.putString("suporte", confirmed);
    SuporteFragment suporteFragment = new SuporteFragment();
    suporteFragment.setArguments(message);
    Toast.makeText(getContext(), text: "Pagamento feito com Sucesso.", Toast.LENGTH_SHORT).show();
    SearchFragment searchFragment = new SearchFragment();
    FragmentTransaction transaction = getFragmentManager().beginTransaction();
    transaction.replace(R.id.homepage_layout, searchFragment);
    transaction.commit();
}

```

```

        radioButton.setOnCheckedChangeListener(v -> {
            onClickRB(view);
        });

        btnPay.setOnCheckedChangeListener(v -> {
            onClickPay(view);
        });

        return view;
    }

    public void onClickRB(View view)
    {
        int radioId = radioGroup.getCheckedRadioButtonId();
        radioButton = view.findViewById(radioId);
        btnPay.setEnabled(true);

        if (radioButton.getText().equals("Cartão de Crédito")){
            //btnPay.getOffsetForPosition(10, 3);
            txtNumber.setVisibility(View.VISIBLE);
            spinnerMes.setVisibility(View.VISIBLE);
            spinnerAno.setVisibility(View.VISIBLE);
            cvv.setVisibility(View.VISIBLE);
            nome.setVisibility(View.VISIBLE);
            btnPay.setText("Pagar com Cartão");
        }

        else if (radioButton.getText().equals("Paypal")) {
            hidden();
            btnPay.setText("Pagar com Paypal");
        }

        else {
            hidden();
            btnPay.setText("Pagar com Google Pay");
        }
    }
}

```

Figura 15 - Código implementado na interface de Pagamento

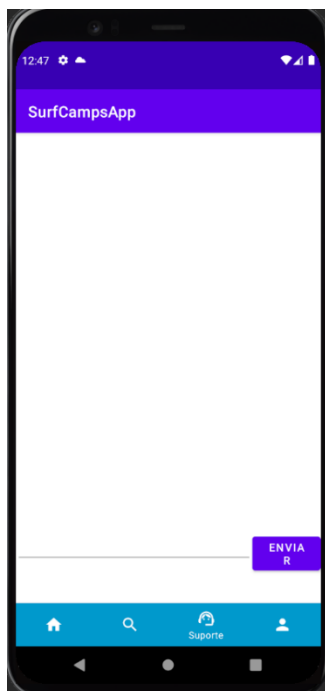


Figura 16 Ecrã de Suporte

Neste ecrã irá ser possível receber uma mensagem após efetuar uma reserva com a sua confirmação, e ser possível falar com o possível instrutor/escola, na qual fizemos a reserva.

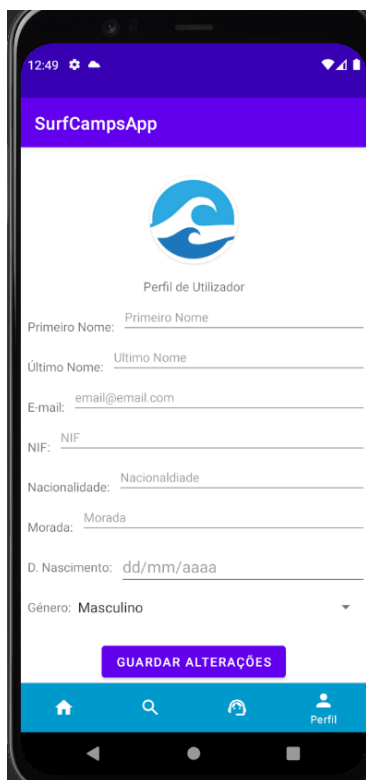


Figura 17 Ecrã de Edição de Perfil

Neste ecrã podemos alterar o perfil de utilizador, preenchendo as informações pessoais.



### **3. Conclusões**

Concluimos que com este trabalho fomos capazes de melhorar e aplicar os conteúdos dados nas aulas no desenho de aplicações, ainda que não tenhamos conseguido fazer testes com utilizadores, assumimos que poderíamos ter implementado mais algumas funcionalidades que, neste trabalho, não estão a funcionar, ou outras que não estão a funcionar da melhor forma.

Ainda assim no geral pensamos que os objetivos principais foram atingidos e desenvolvidos com o projeto “SurfCampsApp”, visto termos conseguido implementar a maioria das funcionalidades pretendidas.