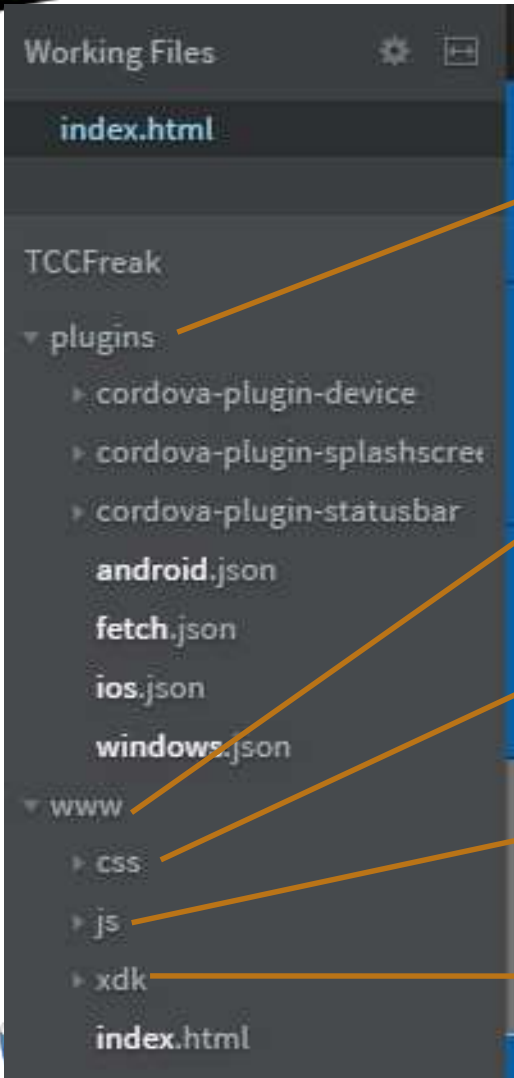


ESTRUTURA E FUNCIONAMENTO DO INTEL XDK + CORDOVA



**“Todo mundo tem cliente.
Só traficante e analista de
sistemas é que tem
usuário.” (Bill Gates)**

ESTRUTURA DO PROJETO: Intel XDK + Cordova



Diretório com os plug-ins instalados.

Diretório com a estrutura web da aplicação.

Diretório com os arquivos CSS. Podem ser adicionados arquivos customizados.

Diretório com os arquivos JavaScript. Podem ser adicionados arquivos customizados.

Diretório com arquivos JavaScript específicos do Intel XDK.

ESTRUTURA DO PROJETO(1): Intel XDK + Cordova

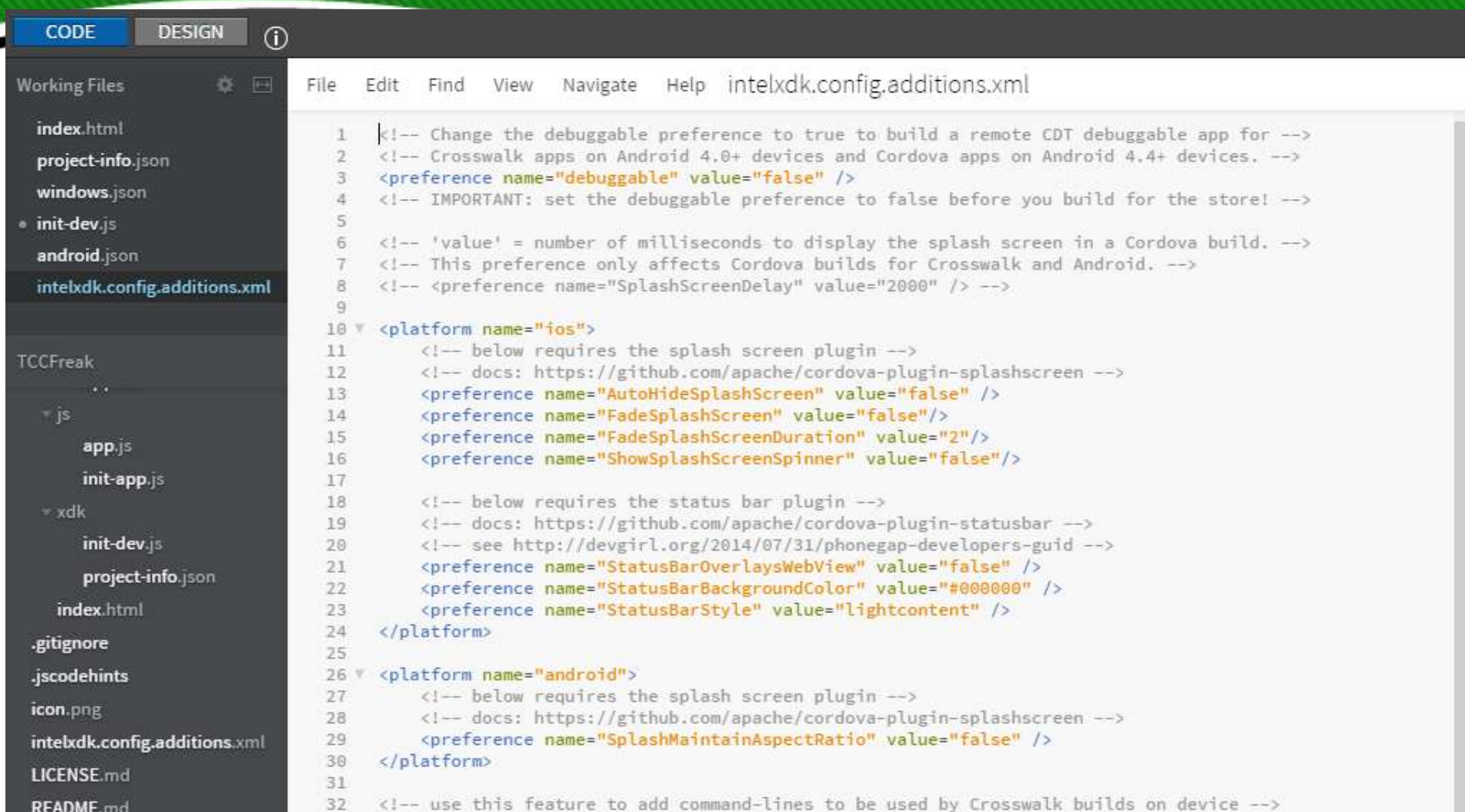
Arquivo	Função
index.html	Arquivo HTML principal da aplicação.
app.css	Arquivo CSS padrão da aplicação. Contém definições CSS globais.
app.js	Contém os manipuladores de eventos da aplicação.
init-app.js	É parte do arquivo app.js.
init-dev.js	Contém funções para detectar <i>ready events</i>. Função device-ready, por exemplo.
index-user-scripts.js	Código JS específico do usuário.

ESTRUTURA DO PROJETO(2): Intel XDK + Cordova

Arquivo	Função
intelxdk.config.additions.xml	Configurações especiais de BUILD da aplicação.
intelxdk.config.<platform>.xml	Configurações específicas de cada plataforma. Arquivo gerado automaticamente pelo XDK.
platform.json	Configurações específicas de cada plataforma com relação aos plug-ins instalados.



ESTRUTURA DO PROJETO: Intel XDK + Cordova

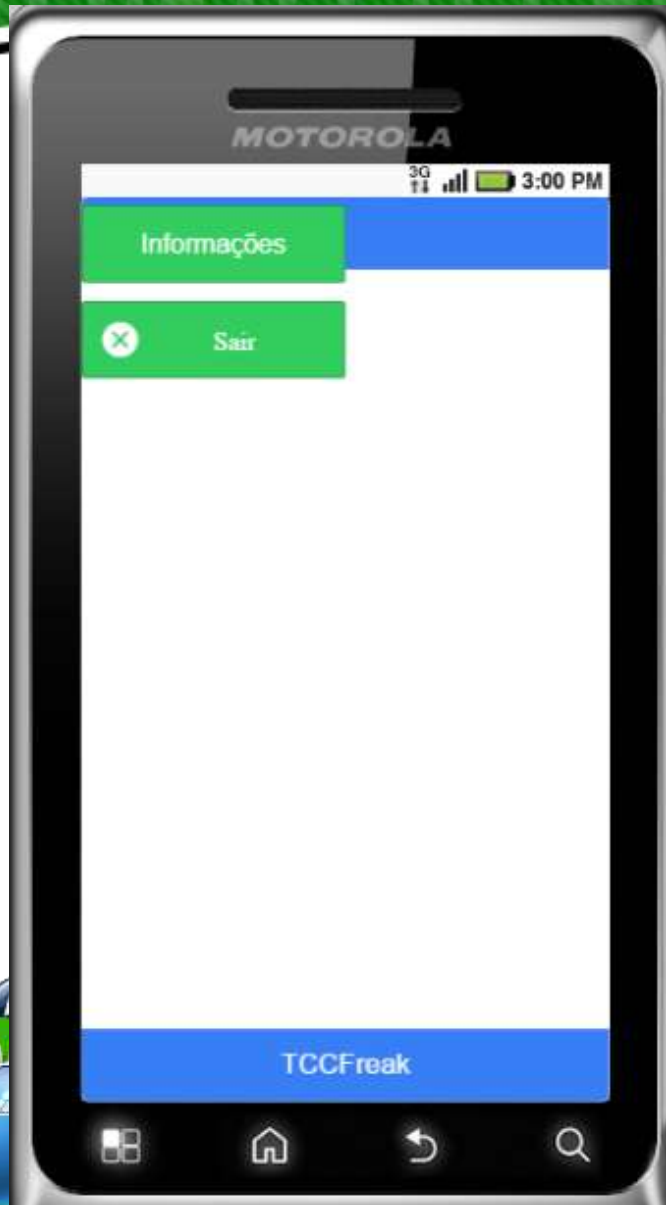


PROJETO TCCFreak

- Criando o projeto TCCFreak
 - HTML5 + Cordova
 - Nome do projeto: *TCCFreak*
 - Ativar plugins: *Device, Camera*
 - Ionic Framework



PROJETO TCCFreak



SPLASHSCREEN, DIÁLOGOS E EVENTOS



EVENTOS DO CORDOVA

- Fazem parte do ciclo de vida do Cordova;
- O evento **“deviceready”**, por exemplo, precisa ser recebido antes de acessar as funcionalidades do dispositivo.



EVENTOS DO CORDOVA(1)

- **deviceready**
- Essencial para qualquer aplicação;
- API do Cordova carregada e pronta para uso;
- “Atachar” o evento a um *event listener*.

```
document.addEventListener("deviceready", onDeviceReady, false);  
  
function onDeviceReady() {  
    // Now safe to use device APIs  
}
```



EVENTOS DO CORDOVA(2)

- **pause**
- Disparado quando a aplicação é colocada em segundo plano;
- “Atachar” o evento a um *event listener*.

```
document.addEventListener("pause", onPause, false);  
  
function onPause() {  
    // Handle the pause event  
}
```



EVENTOS DO CORDOVA(3)

- **resume**
- Ativado quando a aplicação é recuperada do segundo plano;
- “Atachar” o evento a um *event listener*.

```
document.addEventListener("resume", onResume, false);  
  
function onResume() {  
    // Handle the resume event  
}
```



EVENTOS DO CORDOVA(4)

- **online**
- Disparado quando a aplicação está conectada a internet;
- “Atachar” o evento a um *event listener*.

```
document.addEventListener("online", onOnline, false);  
  
function onOnline() {  
    // Handle the online event  
}
```



EVENTOS DO CORDOVA(5)

- **offline**
- Disparado quando a aplicação é desconectada da internet;
- “Atachar” o evento a um *event listener*.

```
document.addEventListener("offline", onOffline, false);  
  
function onOffline() {  
    // Handle the offline event  
}
```



PROJETO TCCFreak

- Finalizando o aplicativo:

```
// finalizando o aplicativo  
function sair() {  
    navigator.app.exitApp();  
}
```



EVENTOS DO CORDOVA(6)

- **EXEMPLO**

```
...  
<head>  
  <title>Exemplo de eventos</title>  
  <script type="text/javascript" charset="utf-8" src="cordova.js"></script>  
  <script type="text/javascript" charset="utf-8">  
  
    function onLoad() {  
      document.addEventListener("deviceready", onDeviceReady, false);  
      document.addEventListener("pause", onPause, false);  
    }  
  
    function onDeviceReady() {  
      console.log("onDeviceReady()");  
    }  
  
    function onPause() {  
      console.log("onPause()");  
    }  
  </script>  
</head>  
...
```



MENSAGENS COM O CORDOVA

- O Cordova possui suporte a mensagens visuais e audíveis;
- Essas mensagens, sejam alertas, confirmações, *prompts*, *beeps* e vibrações podem ser emitidas por meio do objeto *notification*;
- É necessário ativar os plugins chamados de *org.apache.cordova.dialogs* e *org.apache.cordova.vibration*;



MENSAGENS COM O CORDOVA(1)

- **navigator.notification.alert**
- Exibe um alerta ou caixa de diálogo customizada.

```
navigator.notification.alert(message, alertCallback, [title], [buttonName])
```

- **message:** Dialog message (String)
- **alertCallback:** Callback to invoke when alert dialog is dismissed. (Function)
- **title:** Dialog title (String) (Optional, Default: "Alert")
- **buttonName:** Button name (String) (Optional, Default: "OK")

```
// Android / BlackBerry WebWorks (OS 5.0 and higher) / iPhone / Tizen
//
function alertDismissed() {
    // do something
}

navigator.notification.alert(
    'You are the winner!', // message
    alertDismissed,       // callback
    'Game Over',          // title
    'Done'                // buttonName
);
```



MENSAGENS COM O CORDOVA(2)

- **navigator.notification.confirm**
- **Exibe uma caixa de diálogo de confirmação.**

```
navigator.notification.confirm(message, confirmCallback, [title], [buttonLabels])
```

- **message:** Dialog message (String)
- **confirmCallback:** - Callback to invoke with index of button pressed (1, 2 or 3) or when the dialog is dismissed without a button press (0), (Function)
- **title:** Dialog title (String) (Optional, Default: "Confirm")
- **buttonLabels:** Comma separated string with button labels (String) (Optional, Default: "OK,Cancel")

```
// process the confirmation dialog result
function onConfirm(buttonIndex) {
    alert('You selected button ' + buttonIndex);
}

// Show a custom confirmation dialog
//
function showConfirm() {
    navigator.notification.confirm(
        'You are the winner!', // message
        onConfirm,             // callback to invoke with index of button pressed
        'Game Over',           // title
        'Restart,Exit'         // buttonLabels
    );
}
```



MENSAGENS COM O CORDOVA(3)

- **navigator.notification.prompt**
- Exibe uma caixa de diálogo de entrada de dados.

```
navigator.notification.prompt(message, promptCallback, [title], [buttonLabels])
```

- **message:** Dialog message (String)
- **promptCallback:** - Callback to invoke when a button is pressed (Function)
- **title:** Dialog title (String) (Optional, Default: "Prompt")
- **buttonLabels:** Array of strings for the button labels (Array) (Optional, Default: ["OK","Cancel"])

```
// process the prompt dialog results
function onPrompt(results) {
    alert("You selected button number " + results.buttonIndex + " and entered " + results.input1);
}

// Show a custom prompt dialog
//
function showPrompt() {
    navigator.notification.prompt(
        'Please enter your name', // message
        onPrompt,                // callback to invoke
        'Registration',          // title
        ['Ok','Exit']            // buttonLabels
    );
}
```



MENSAGENS COM O CORDOVA(4)

- **navigator.notification.bEEP**
- Emite som de um beep.

```
navigator.notification.bEEP(times);
```

- **times:** The number of times to repeat the beep (Number)

```
// Beep twice!  
navigator.notification.bEEP(2);
```



MENSAGENS COM O CORDOVA(1)

- **navigator.notification.vibrate**
- Ativa a vibração do dispositivo por um certo tempo.

```
navigator.notification.vibrate(milliseconds)
```

- **time:** Milliseconds to vibrate the device. 1000 milliseconds equals 1 second (Number)

```
// Vibrate for 2.5 seconds  
//  
navigator.notification.vibrate(2500);
```



MENSAGENS NA CONSOLE

- **console.log('...');**
- O Cordova suporta o envio de mensagens para a console do dispositivo por meio do objeto console.

```
function onBatteryStatus(info) {  
    // Handle the online event  
    console.log("Level: " + info.level + " isPlugged: " + info.isPlugged);  
}
```



SPLASH COM O CORDOVA

- **SplashScreen**
- Permite exibir/esconder a tela de splash;
- É necessário ativar o *plugin* **org.apache.cordova.splashscreen**.
- *navigator.splashscreen.show()*: exibe a tela de splash da aplicação;
- *navigator.splashscreen.hide()*: esconde a tela de splash da aplicação.



SPLASH COM O CORDOVA(1)

- Setando a imagem da tela de splash(Android):

```
super.setIntegerProperty("splashscreen", R.drawable.splash);  
super.loadUrl("file:///android_asset/www/index.html", 10000);
```



SPLASH COM O CORDOVA(2)

- **EXEMPLO**

```
...  
<head>  
  <title>Exemplo de SplashScreen</title>  
  
  <script type="text/javascript" charset="utf-8"  
src="cordova.js"></script>  
  <script type="text/javascript" charset="utf-8">  
  
    // Wait for Cordova to load  
    //  
    document.addEventListener("deviceready", onDeviceReady, false);  
  
    // Cordova is ready  
    //  
    function onDeviceReady() {  
      navigator.splashscreen.hide();  
    }  
  </script>  
</head>  
...
```

