



# VW Play Especificações Técnicas

Detalhes de hardware e software

# Especificações Técnicas

<b>Sistema Operacional</b>	Android Automotive 9.0 (Pie)
<b>Processador</b>	ARM based architecture
<b>Armazenamento</b>	10GB para Apps
<b>Memória RAM</b>	2GB
<b>Tela</b>	10.1 polegadas com multi-touch
<b>Resolução</b>	160 DPI 1540 x 720 pixels
<b>Área para APP</b>	1332 x 636 (sem barras)
<b>Antena</b>	GPS
<b>Wireless</b>	2.4 GHz and 5 GHz (modo cliente)



# Princípios de Design

**Funções suportadas**

- Atualização remota de apps
- Velocidade do Carro
- GPS
- Acelerômetro
- Microfone
- Áudio



# Princípios de Design

## Não suportado

Google Play Services

Maps

Google Play Store

Gmail

Hangout

Google Docs



# Princípios de Design

## Recomendado

Não distraia o motorista

Pense na rotina do motorista

Interações curtas e rápidas

Interações gestuais

Evite propaganda no app

Fácil acesso



# Primeiros passos

**VW Play** utiliza como base o Android Automotive version 9.0 (Pie) e utiliza o padrão AOSP (Android Open Source Project) API e interface.

Tudo disponível in AOSP pode ser considerado disponível para uso.

**Software**                      Android Studio

**Aparelho**                      Tablet

**Orientação**                   Paisagem

**Nível de API**                 28

Não inclua ou utilize Google Services. Aplicações terceiras externas do Google Automotive Services (GAS) e Google Mobile Services (GMS) não estão disponíveis.



# DPI configuração

Para evitar mudanças inesperadas de DPI baseado na lógica por trás da aplicação de gerenciamento do VW Play é necessário definir o DPI explicitamente no início das atividades do Launcher seguinte o código de exemplo:

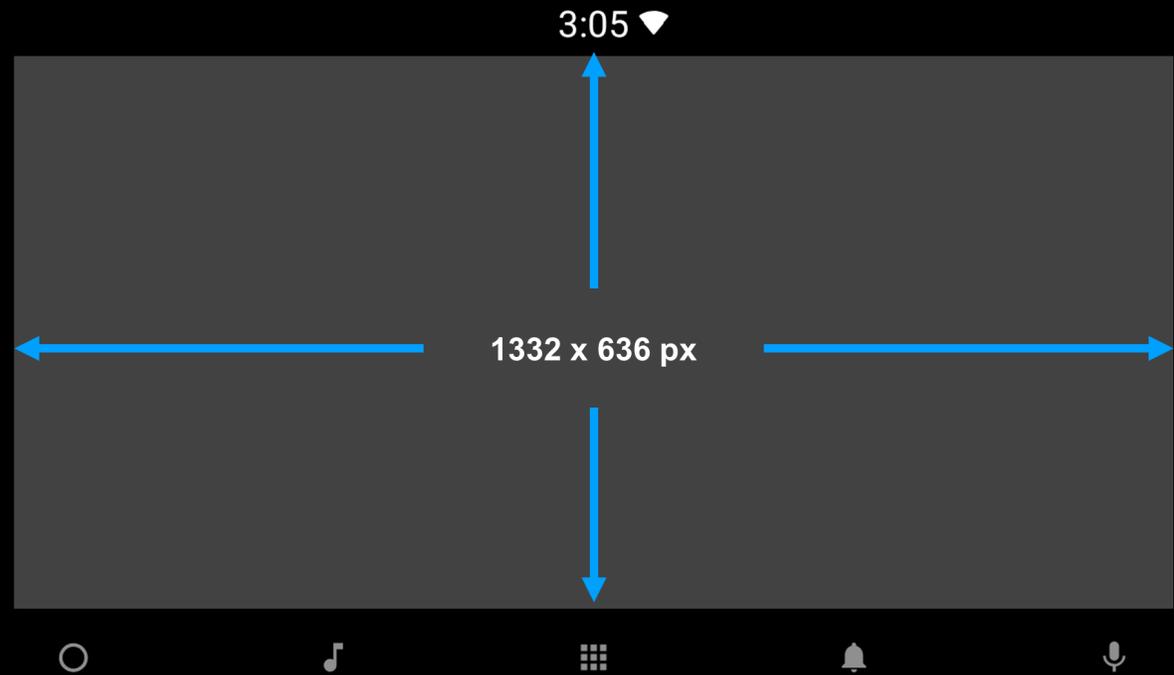
```
protected void onCreate(Bundle savedInstanceState) {  
    mContext = this;  
    //Set DPI  
    DisplayMetrics displayMetrics = mContext.getResources().getDisplayMetrics();  
    Configuration config = mContext.getResources().getConfiguration();  
    displayMetrics.densityDpi = 160;  
    config.densityDpi = 160;  
    displayMetrics.setTo(displayMetrics);  
    config.setTo(config);  
    mContext.getResources().updateConfiguration(config, displayMetrics);  
}
```



# Android aparelho virtual

Atualmente não existe um emulador para VW Play capacitado para plataformas x86/x64, no entanto para validar e conferir a aplicação durante o desenvolvimento nós recomendamos utilizar um aparelho virtual com a seguinte configuração:

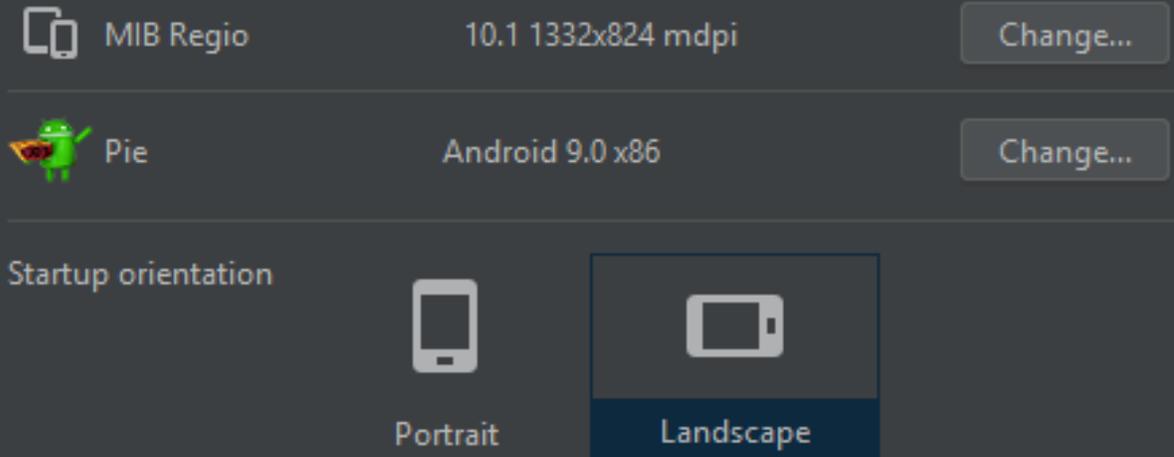
Emulador Android Automotive utiliza duas barras de navegação: topo e base. Ignore ambas as barras, a área disponível para aplicação no VW Play é de 1332x636 pixels.



# Configuração Emulador

No final a configuração AVD deve seguir a imagem abaixo:

<b>OS</b>	Automotive Android compilation PIE 9.0 x86
<b>Resolution</b>	1332 x 824 pixels
<b>Orientation</b>	Landscape



# Acessos comuns de **API**

Disponível para aplicativos coletarem.

**Local de Serviço**

**Bluetooth**

**Notificação**

**Foco do Áudio**

**Configuração do WiFi**

**Configuração do Bluetooth**

**Concessionário**

**Navegação**

**Sugestões de busca**

**... e mais**



# Local de Serviço

Para acessar a localização GPS o aplicativo deve solicitar a permissão dos módulos a seguir e o código exemplo deve solicitar os dados:

```
//Request Location
permissionManifest.permission.ACCESS_FINE_LOCATIONManifest.permission.ACCESS_COARSE_LOCATION
...
LocationManager locationManager = (LocationManager)
getApplicationContext().getSystemService(LOCATION_SERVICE);
Location location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
if (location != null){
    gpsPosition.setText(Double.toString(location.getLatitude())+" x
"+Double.toString(location.getLongitude()));}
```

**Nota:** Não utilize “fused location provider” do Google ou outro aplicativo terceiro para acessar a localização GPS.



# Bluetooth

API padrão do Android pode ser utilizada para escutar as modificações de Bluetooth e coletar a lista de aparelhos conectados por Bluetooth.

```
//Bluetooth ListenerlastBTStatus = findViewById(R.id.last_bt_status);
IntentFilter filter = new IntentFilter();
filter.addAction(BluetoothDevice.ACTION_ACL_CONNECTED);
filter.addAction(BluetoothDevice.ACTION_ACL_DISCONNECT_REQUESTED);
filter.addAction(BluetoothDevice.ACTION_ACL_DISCONNECTED);
this.registerReceiver(mReceiver, filter);...//Bluetooth Broadcasts
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        ...
    }
};//Bluetooth Connected Device
ListBluetoothAdapter.getDefaultAdapter().getBondedDevices();
```



# Notificação

VW Play possui uma central de notificações no menu de configuração. Aplicações terceiras podem utilizar “Notification Manager” do Android para criar notificações, no entanto não é possível incluir ações ou mensagem contextual. Por favor, confira a documentação de Design para mais detalhes. Código de exemplo:

```
String channelId = "notify_001";
NotificationManager mNotificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
NotificationCompat.Builder mBuilder =
    new NotificationCompat.Builder(getApplicationContext(), channelId)
        .setSmallIcon(R.drawable.common_full_open_on_phone)
        .setContentTitle("User can read TitleContent, but can't read TextContent")
        .setContentText("N/A");

NotificationChannel channel = new NotificationChannel(
    channelId,
    "Main Channel",
    NotificationManager.IMPORTANCE_DEFAULT);
mNotificationManager.createNotificationChannel(channel);
mBuilder.setChannelId(channelId);

mNotificationManager.notify(001, mBuilder.build());
```



# Foco do Áudio

Utilizando o código de exemplo as aplicações terceiras podem gerir os recursos de áudio do carro em concorrência com a saída e entrada de áudio disponível no ecossistema do VW Play.

```
//Requesting Audio Focus -
AudioAttributes mVoiceAudioAttrib = new AudioAttributes.Builder()
    .setUsage(AudioAttributes.USAGE_MEDIA)
    .build();
AudioFocusRequest focusRequest = new AudioFocusRequest.Builder(AudioManager.AUDIOFOCUS_GAIN_TRANSIENT)
    .setAudioAttributes(mVoiceAudioAttrib)
    .setOnAudioFocusChangeListener(mFocusListener)
    .build();

status = mAudioManager.requestAudioFocus(focusRequest);
if (status == AudioManager.AUDIOFOCUS_REQUEST_GRANTED) {
    Log.d(TAG, "Audio Focus Request Granted for USAGE_MEDIA");
} else {
    Log.d(TAG, "Audio Focus Request Rejected for USAGE_MEDIA");
}

//Abandon Audio Focus
mAudioManager.abandonAudioFocusRequest(focusRequest); // AudioFocusChange callback listener

AudioManager.OnAudioFocusChangeListener mFocusListener =
    new AudioManager.OnAudioFocusChangeListener() {
        public void onAudioFocusChange(int focusChange) {
            if (focusChange == AudioManager.AUDIOFOCUS_LOSS) {
                // Permanent loss of audio focus
                // Pause/Stop playback immediately
            } else if (focusChange == AudioManager.AUDIOFOCUS_LOSS_TRANSIENT) {
                // Pause playback
            } else if (focusChange == AudioManager.AUDIOFOCUS_LOSS_TRANSIENT_CAN_DUCK) {
                // keep playing, Volume will be ducked by the platform
            } else if (focusChange == AudioManager.AUDIOFOCUS_GAIN) {
                // Your app has been granted audio focus again (which may be lost due to TRANSIENT_LOSS)
                // resume the playback if necessary
            }
        }
    }
};
```



# Configuração do WiFi

Se o aplicativo requer conexão com a internet e possui lógica para detectar e notificar o usuário, é possível redirecionar o usuário para tela de configuração do WiFi no VW Play com a seguinte ação:

```
startActivity(new Intent(Settings.ACTION_WIFI_SETTINGS));
```



# Configuração do Bluetooth

A mesma opção também está disponível para Bluetooth, é possível redirecionar o usuário para tela de configuração do Bluetooth no VW Play com a ação a seguir:

```
startActivity(new Intent("com.vw.car.connectivity.intent.action.OPEN_DEVICE_MANAGER"));
```



# Concessionário

Para abrir a tela de concessionário no VW Play, a aplicação deve enviar a seguinte solicitação:

```
Intent intent = new Intent(Intent.ACTION_DIAL);  
intent.setData(Uri.parse("tel:12345678901"));  
startActivity(intent);
```

**Nota:** Se nenhum aparelho Bluetooth estiver conectado, o usuário será redirecionado para tela de configuração das conexões Bluetooth



# Navegação

Para enviar uma intenção para um aplicativo terceiro, o padrão Android funciona. Se não existir nenhum app registrado para processar as intenções de geo localização, a intenção será ignorada.

```
String uri = "geo: -21,-47";  
startActivity(new Intent(android.content.Intent.ACTION_VIEW, Uri.parse(uri)));
```

**Nota:** VW Play não possui aplicativos terceiros de navegação pré-instalados. Usuário final deve baixar um app na loja de aplicativos.



# Sugestões de busca

Teclado padrão para buscas pode ser utilizado acessando a interface de API, porém se for solicitado o teclado com sugestões (Campo de busca), o código utilizado deve ser:

XML Layout Code:

```
<EditText
```

```
    android:id="@+id/et_settings_search"  
    style="@style/Widget.SearchBar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center_vertical"  
    android:hint="@string/search_for"  
    android:importantForAutofill="no"  
    android:autofillHints="@null"  
    android:focusable="false"  
    android:focusableInTouchMode="false"  
    android:cursorVisible="false"  
    android:inputType="text"  
    android:maxLines="1"/>
```

```
<com.vw.car.input.minikeyboardlib.KeyboardView
```

```
    android:id="@+id/keyboard"  
    style="@style/Keyboard"  
    app:font="@*android:font/font_montserrat_bold"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="start" />
```

Java Code:

```
KeyboardBinder.bind( keyboardView , etSearchView );
```

