

Catarina Criativa

Manual Técnico

IDE

Para o desenvolvimento dos protótipos e o formato do projeto no repósito foi usado o FlashDevelop na versão 4.6.3.2 para .NET 2.0 com o Flex SDK 4.6.0 e Adobe Air 14.0.

SERVIDORES

Na infra estrutura de servidor foi usado o Ubuntu Desktop na versão 14.04.1 LTS com LAMP-SERVER instalado rodando o servidor APACHE2 com PHP5 e MySQL5.5.

REPOSITÓRIO

Para repositório dos arquivos do projeto foi usado o GITHUB em formato público sobre a licença APACHE 2.0 de Janeiro de 2014.

Clone URL

<https://github.com/cafunduestudio/catarinacriativa>

PROTÓTIPO 1

DOCUMENTARY

O projeto foi desenvolvido usando a linguagem ActionScript3 com Adobe Air e estruturado em cima do padrão Model-View-Controller, usando de um Framework no modelo Data-Driven criado pelo Cafundó Estúdio Criativo, de uso livre.

Para criação da marca digital de áudio e comparação do áudio com um banco de dados foi usado a biblioteca OPEN-SOURCE Echoprint

<http://echoprint.me/>

Echoprint UBUNTU install

```
sudo apt-get update
sudo apt-get install git
sudo apt-get install python-pip python-dev python-setuptools build-essential
sudo pip install --upgrade pip
sudo pip install --upgrade virtualenv
sudo pip install git+https://github.com/echonest/pyechonest.git
sudo add-apt-repository ppa:jon-severinsson/ffmpeg
sudo apt-get update
sudo apt-get install ffmpeg libboost1.55-dev libtag1-dev zlib1g-dev
git clone https://github.com/echonest/echoprint-codegen.git
cd echoprint-codegen/src/
make
cd ..
./echoprint-codegen <AUDIO FILE NAME>
```

TokyoTyrant UBUNTU install

```
sudo apt-get install tokyotyrant
wget
http://sourceforge.net/projects/tokyocabinet/files/tokyotyrant/1.1.33/tokyotyrant-1.1.33.tar.gz
tar xvf tokyotyrant-1.1.27.tar.gz
cd tokyotyrant-1.1.33
```

Echoprint-Server UBUNTU install

```
sudo apt-get install default-jre
git clone https://github.com/echonest/echoprint-server
cd echoprint-server/solr/solr
java -Dsolr.solr.home=/home/path/to/echoprint-server/solr/solr/solr/ -Djava.awt.headless=true
-jar start.jar
```

```
cd <PATH>/tokyotyrant-1.1.33
sudo ./ttservctl start
```

----

ADD MUSIC TO DATABASE

----

```
cd <PATH>/echoprint-codegen
find -name "*.mp3" > music_to_ingest
./echoprint-codegen -s < music_to_ingest > allcodes.json
cd <PATH>/echoprint-server/utils/
python fastingest.py allcodes.json
```

----

SEARCH MUSIC IN DATABASE

----

```
cd <PATH>/echoprint-server/examples/
python lookup.py Robyn\ 'Call\ Your\ Girlfriend\'\' Official\ Video\ \'(HQ)\'.mp3
```

PROTÓTIPO 2

ANIMATION

O projeto foi desenvolvido usando a linguagem ActionScript3 com Adobe Air e estruturado em cima do padrão Model-View-Controller, usando de um Framework no modelo Data-Driven criado pelo Cafundó Estúdio Criativo, de uso livre.

Para o sistema de PUSH foi usado a **ANE** de Adobe Air “Local Notification”

<https://bitbucket.org/juankpro/jklocalnotifications-ane/overview>

Para sincronia com o servidor e o desbloqueio de novos episódios foram usadas requisições para um arquivo contendo SCRIPTS PHP no servidor.

Application.XML MANIFEST Permissions

Android > ManifestAddtions > Manifest

```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<application>
    <service android:name="com.juankpro.ane.localnotif.LocalNotificationIntentService" />
    <receiver android:name="com.juankpro.ane.localnotif.AlarmIntentService" />
    <receiver android:name="com.juankpro.ane.localnotif.AlarmRestoreOnBoot">
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
    </receiver>
</application>
```

Extensions

```
<extensions>
    <extensionID>com.juankpro.ane.LocalNotification</extensionID>
</extensions>
```

EXEMPLO DE CHAMADAS DA API

```
if (true || NotificationManager.isSupported) {
    var notificationManager:NotificationManager = new NotificationManager();

    var notification:Notification = new Notification();
```

```

        notification.actionLabel = "OK";
        notification.body = "Confira nosso novo episódio!";
        notification.title = "Bruxinha Catarina";
        notification.fireDate = new Date((new Date()).time + (10 * 1000)); // 10 Seconds
        notification.numberAnnotation = 1;
        notification.actionData = { };

        notificationManager.notifyUser("NOTIFICATION_CODE_001", notification);

    }

```

SCRIPT DE SINCRONIA

```

<?php
$EPISODES = array(
    'episodes' => array(
        'datetime' => mktime(15, 49, 0, 11, 04, 2014),
        'route' => 'episodes_7'
    ),
    'episodes_locked' => array(
        'datetime' => mktime(15, 49, 0, 11, 04, 2014),
        'route' => 'episodes_locked_3'
    )
);

$JSON = array();
foreach ($EPISODES as $episode => $information) {
    if ($information['datetime'] <= time()) {
        $JSON['episodes'][$episode] = $information['route'];
    }
}

echo json_encode($JSON);
?>

```

PROTÓTIPO 3

LIVEACTION

O projeto foi desenvolvido usando a linguagem ActionScript3 com Adobe Air e estruturado em cima do padrão Model-View-Controller, usando de um Framework no modelo Data-Driven criado pelo Cafundó Estúdio Criativo, de uso livre.

Para o sistema de integração social com o Facebook foi usado a **ANE** de Adobe Air "ANE-Facebook"

<https://github.com/freshplanet/ANE-Facebook>

Para uso do mesmo foi criado um Facebook APP Graph 2.0 com a opção ANDROID.

Application.XML MANIFEST Permissions

Android > ManifestAddtions > Manifest

```
<application>
    <activity android:name="com.facebook.LoginActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:label="@string/app_name" />
    <activity android:name="com.freshplanet.ane.AirFacebook.LoginActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen">
        </activity>
    <activity android:name="com.freshplanet.ane.AirFacebook.DialogActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen">
        </activity>
    <service android:name="com.juankpro.ane.localnotif.LocalNotificationIntentService" />
    <receiver android:name="com.juankpro.ane.localnotif.AlarmIntentService" />
    <receiver android:name="com.juankpro.ane.localnotif.AlarmRestoreOnBoot">
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
    </receiver>
</application>
```

Extensions

```
<extensions>
    <extensionID>com.freshplanet.AirFacebook</extensionID>
</extensions>
```

EXEMPLO DE CHAMADAS DA API

```
if (Facebook.isSupported) {  
    //Init facebook api.  
    facebook = Facebook.getInstance();  
    facebook.init(APP_ID);  
  
    //Check if have a open session.  
    if (!facebook.isSessionOpen) {  
        //Open new session passing a callback function.  
        facebook.openSessionWithReadPermissions(READ_PERMISSIONS,  
callbackFunction);  
    }  
    else {  
        // USER IS LOGGED  
    }  
}  
  
//Share parameters  
const PARAMS:Object = {  
    message: "Here is my Big Bang Self!",  
    caption: "Big Bang Self in Quest"  
};  
  
//Open a dialog requesting the share confirmation.  
facebook.dialog("feed", PARAMS, callbackFunction, true);
```