



THE DEVELOPER'S CONFERENCE

Trilha – TV Digital

Recursos para enriquecer sua aplicação interativa

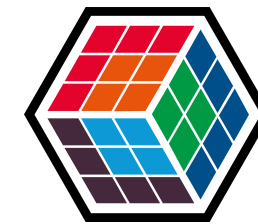
Fabio Eduardo Angeli

Coordenador de Desenvolvimento TV Digital - TV Record

Vice-Diretor de Interatividade da SET

Membro do Módulo Técnico do Fórum SBTVD

Agenda



THE
DEVELOPER'S
CONFERENCE

- Audiência
- Enquete
- Previsão do Tempo
- Otimização de Imagens
- Animações

Audiência



THE
DEVELOPER'S
CONFERENCE

- Google Analytics (<http://www.google.com/analytics/>)
- Web Tracking (client .js)
- Possibilita Server Side Tracking (.php)
- Tracking via HTTP GET

Algumas possibilidades



THE
DEVELOPER'S
CONFERENCE

- Tempo de uso da aplicação
- Cidade do usuário
- Rank com conteúdos mais acessados
- Fluxo de navegação

Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Client Side Tracking

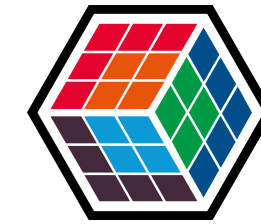
```
<script type="text/javascript">

var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-XXXXX-X']);
_gaq.push(['_trackPageview']);

(function() {
  var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
  ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www')
    + '.google-analytics.com/ga.js';
  var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
})();

</script>
```

Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Server Side Tracking

```
<?php

use UnitedPrototype\GoogleAnalytics;

$tracker = new GoogleAnalytics\Tracker('MO-XXXXX-X', 'exemplo.com.br');

// Informação do Visitante
$visitor = new GoogleAnalytics\Visitor();
$visitor->setIpAddress($_SERVER['REMOTE_ADDR']);
$visitor->setUserAgent($_SERVER['HTTP_USER_AGENT']);

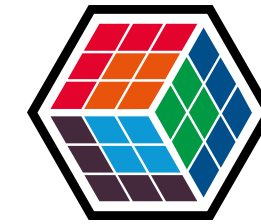
// Informação da Sessão
$session = new GoogleAnalytics\Session();

// Informação da Pagina
$page = new GoogleAnalytics\Page('/page.html');
$page->setTitle('My Page');

// Track Page View
$tracker->trackPageview($page, $session, $visitor);

?>
```

Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Preparando o servidor

```
<?php

use UnitedPrototype\GoogleAnalytics;

$tracker = new GoogleAnalytics\Tracker('MO-XXXXX-X', 'exemplo.com.br');

// Informação do Visitante
$visitor = new GoogleAnalytics\Visitor();
$visitor->setIpAddress($_SERVER['REMOTE_ADDR']);
$visitor->setUserAgent($_SERVER['HTTP_USER_AGENT']);

// Informação da Sessão
$session = new GoogleAnalytics\Session();

// Informação da Pagina
$page = new GoogleAnalytics\Page('/'.$_GET['content']);
$page->setTitle('Aplicação - '.$_GET['content']);

// Track Page View
$tracker->trackPageview($page, $session, $visitor);

?>
```

Ginga + Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
-- Classe responsável pela conexão de dados
local connection = nil;

-- Criar handler de eventos TCP
function handlerTCP(pEvento)

    if (pEvento.type == 'connect') then

        self.connection = pEvento.connection;

    end

end

-- Registrar handler
event.register(handlerTCP, 'tcp')

-- Criar/testar conectividade
event.post( { class='tcp', type='connect', host='192.168.0.1', port=80, timeout=5 } );
```


Ginga + Google Analytics



THE
DEVELOPER'S
CONFERENCE

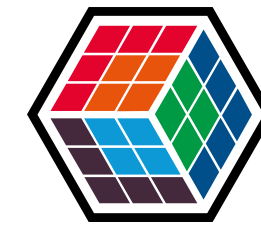
➤ Integrando a aplicação

```
function trackAcesso(pConteudo)

    event.post( { class='tcp', type='data', connection=self.connection,
        value='GET http://192.168.0.1/audiencia.php?content='
            ..pConteudo..' HTTP/1.0\nHost: 192.168.0.1\n\n', timeout=5 } );

end
```

Ginga + Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
function defaultHandler(pEvento)

  if (pEvento.class == 'key' and pEvento.type == 'press') then

    if (pEvento.key == 'RED') then

      -- Track acesso ao conteudo
      conexao:trackAcesso('Foto')

      -- Abrir conteudo
      conteudo:start()

    end

  end

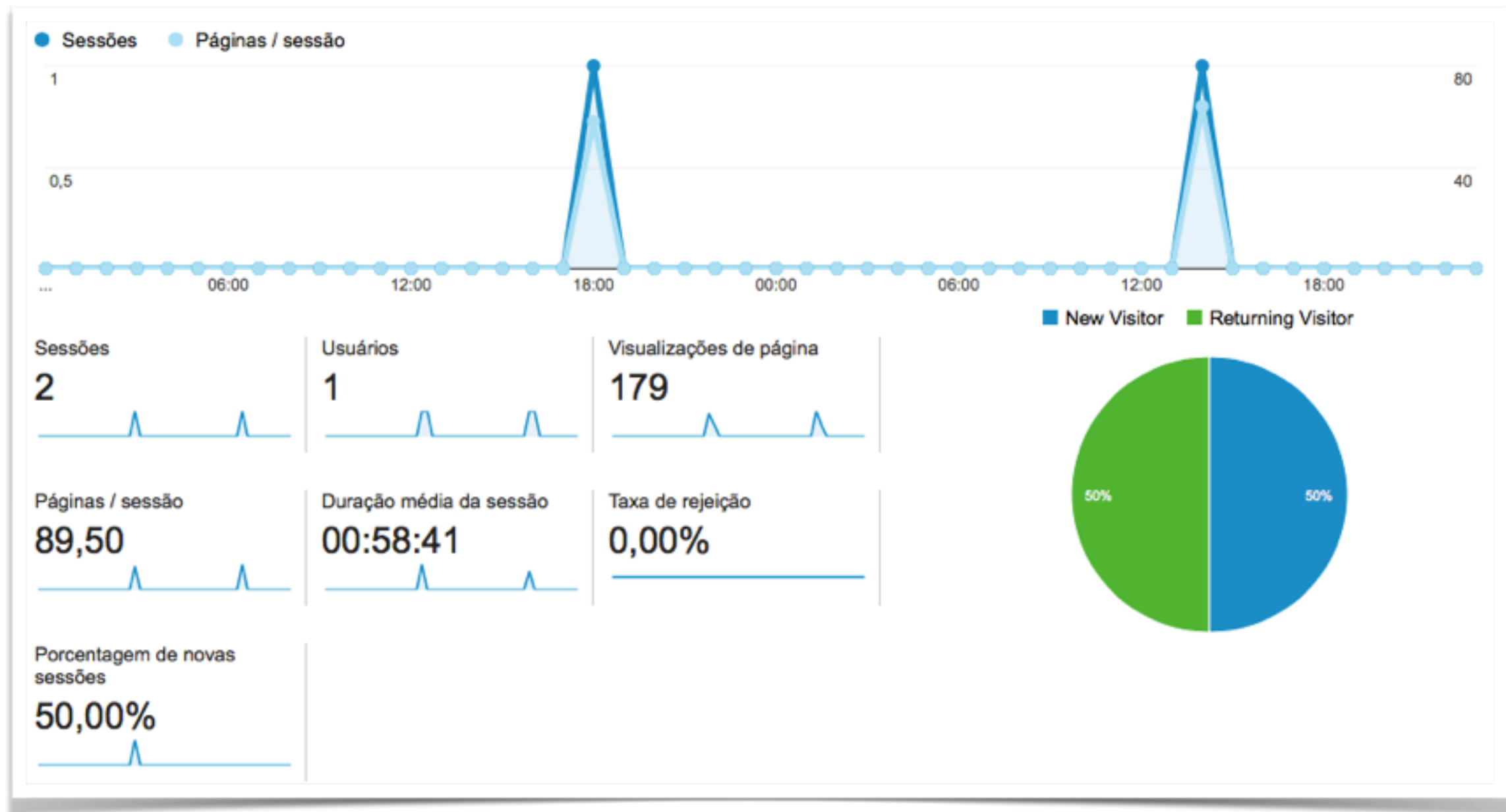
end
```

Ginga + Google Analytics

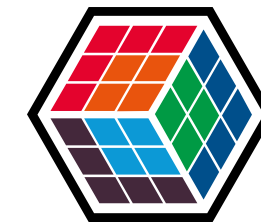


THE
DEVELOPER'S
CONFERENCE

➤ Resultados

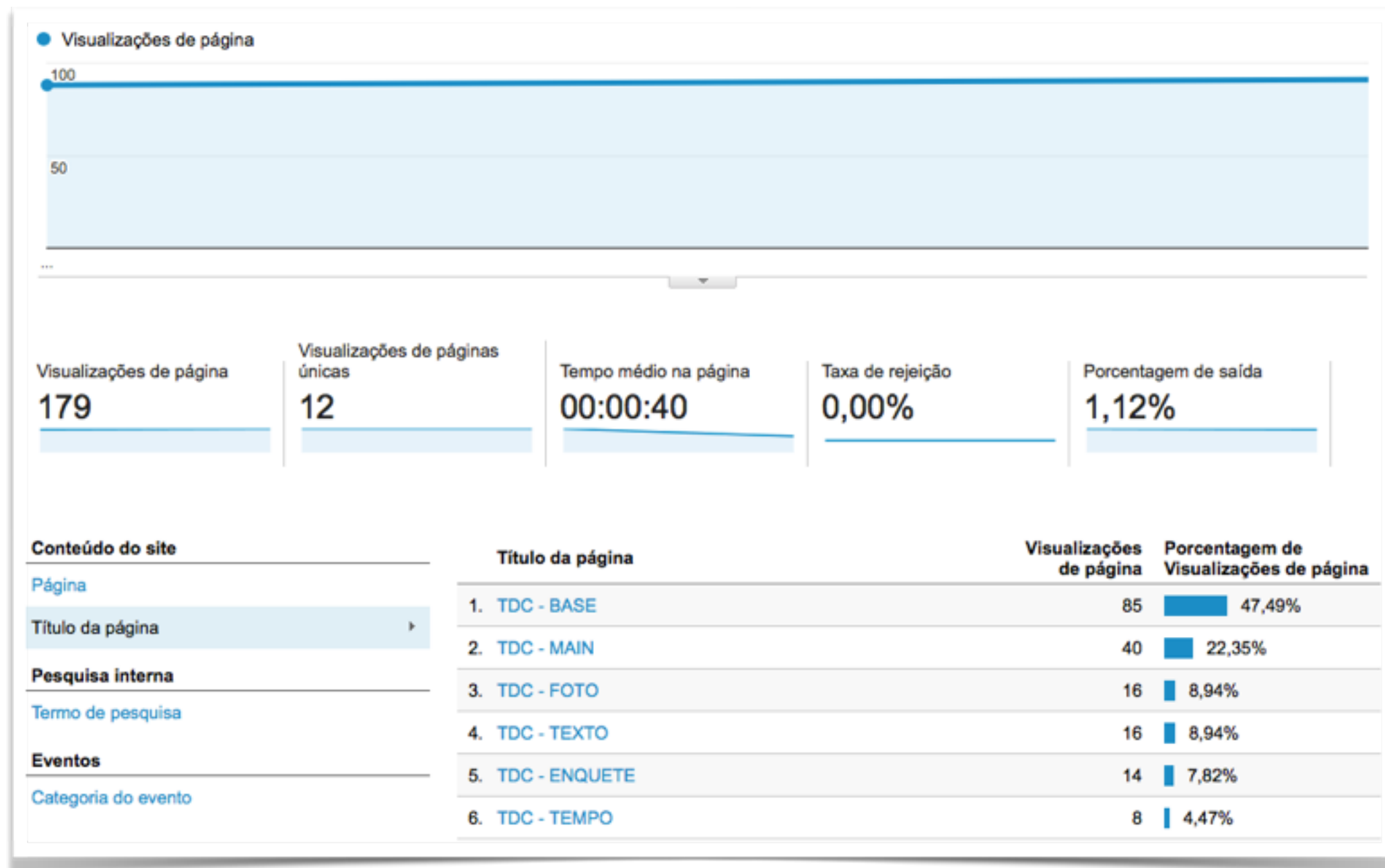


Ginga + Google Analytics

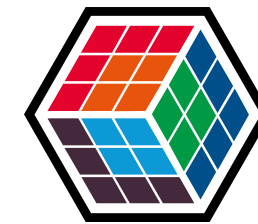


THE
DEVELOPER'S
CONFERENCE

➤ Resultados

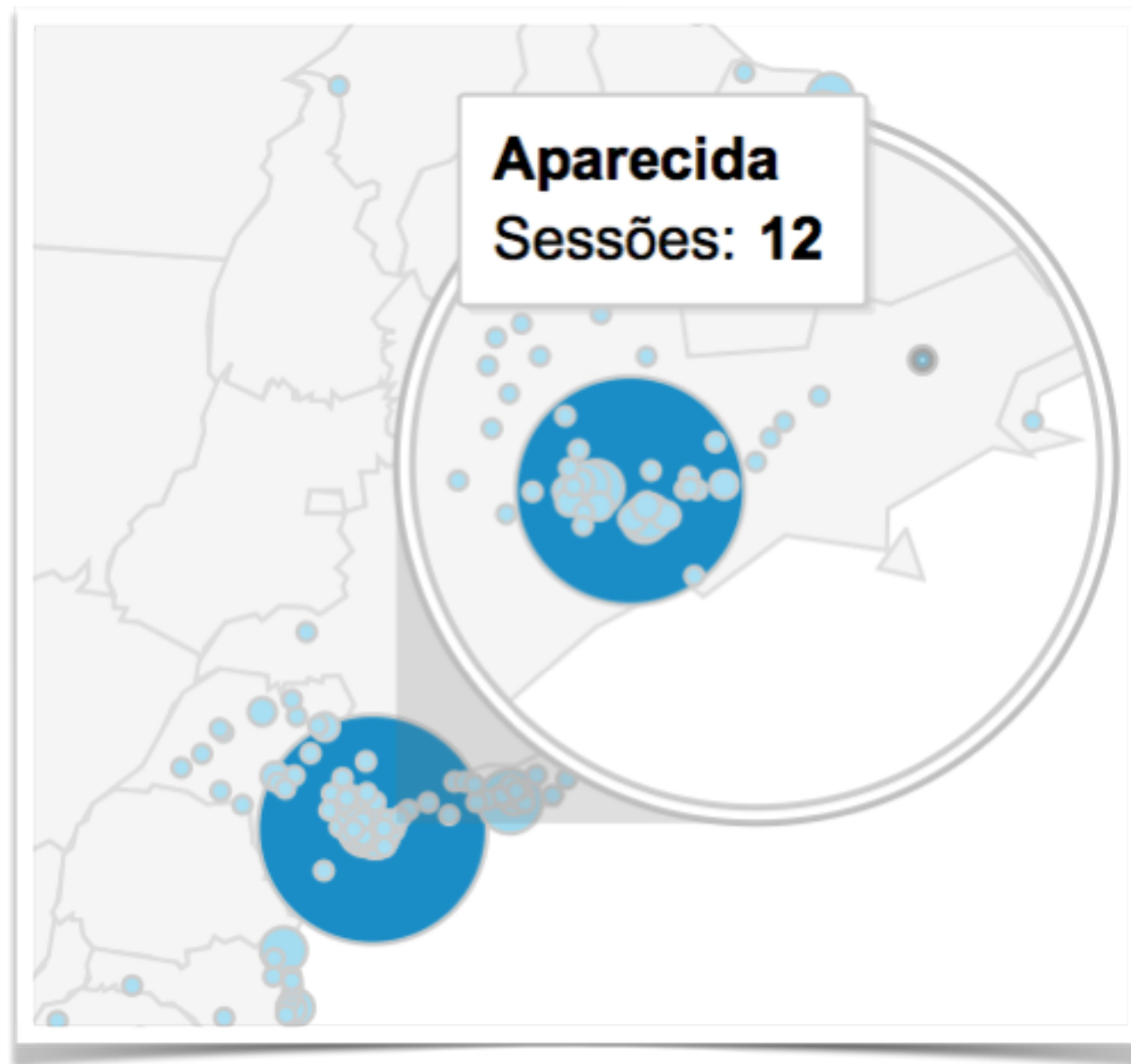


Ginga + Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Resultados

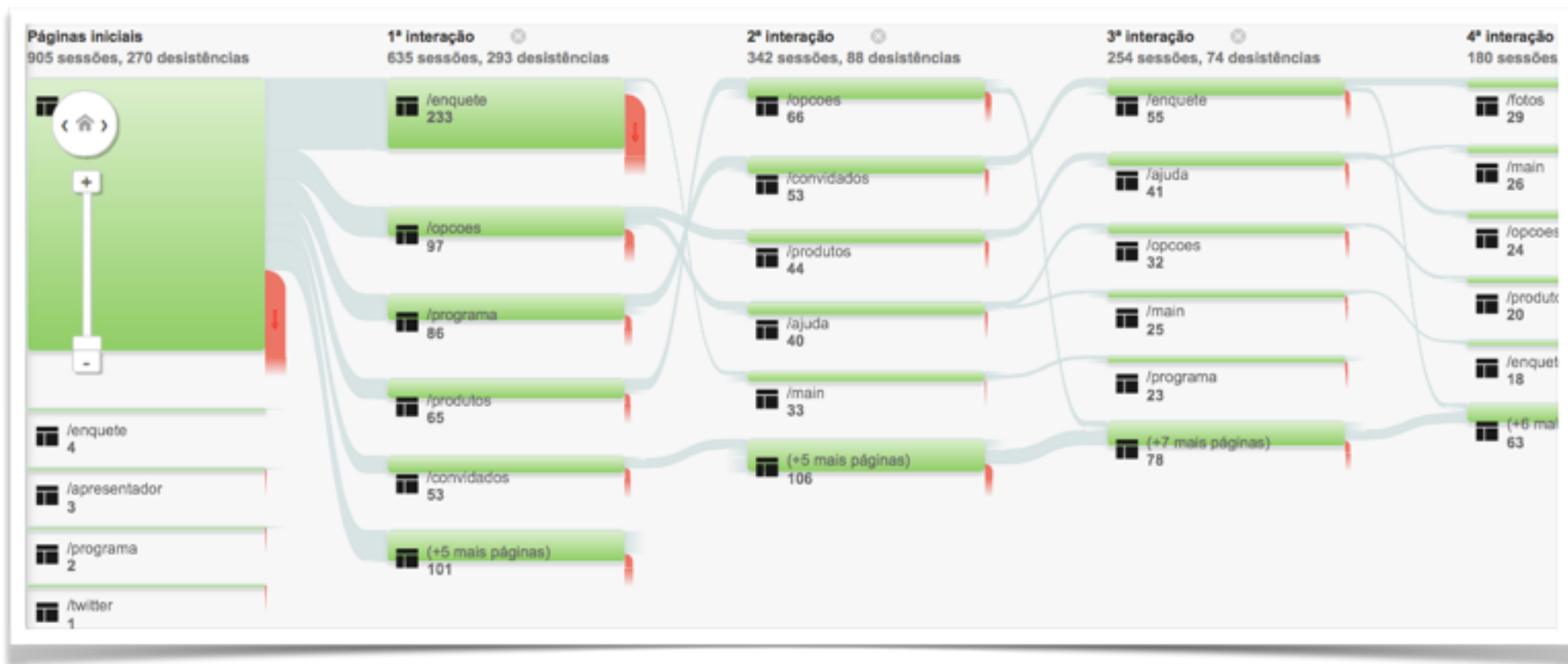


Ginga + Google Analytics



THE
DEVELOPER'S
CONFERENCE

➤ Resultados



Enquete



THE
DEVELOPER'S
CONFERENCE

- Já estamos enviando dados, por que não receber?
- Aproveitar a mesma requisição HTTP GET
- Servidor ao receber o voto já responde com resultado
- Utilizar o mesmo servidor da audiência

Enquete



THE
DEVELOPER'S
CONFERENCE

➤ Preparando o servidor

```
<?php

$saltNumero = $GET['param'];
$idEnquete = $_GET['eparam'];

// Obter ID alternativa de acordo com o ID da enquete e NUMERO da alternativa
$alternativaID = $dbManager->getIDAlternativaParaEnqueteNumero($idEnquete, $saltNumero);

// Adicionar voto para alternativa ID
$dbManager->AdicionarVoto($alternativaID);

// Obter resultado da votação
$result = $dbManager->resultadoEnquete($idEnquete);

// 'Exibe' o resultado da votação
print($result[0].";".$result[1]);

?>
```


Enquete



THE
DEVELOPER'S
CONFERENCE

➤ Enviando voto

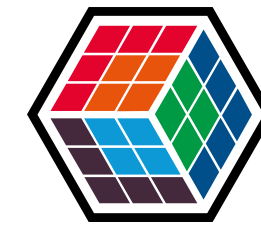
```
function callbackFunctionEnquete(pDados)

    -- Passa o resultado da enquete para o componente
    -- responsável por exibir seu conteúdo na tela
    resultado:processarExibirDados(pDados);

end

-- Ao selecionar a alternativa e pressionar OK
conexao:enquete(alternativaSelecionada, '1', callbackFunctionEnquete);
```

Enquete



THE
DEVELOPER'S
CONFERENCE

➤ Enviando voto

```
local callbackFunction = nil;
local callbackDados = nil;

function enquete(pAlternativa, pEnquete, pCallbackFunction)

    -- Função que deverá ser chamada ao receber os dados
    self.callbackFunction = pCallbackFunction;

    -- Onde os dados serão armazenados
    self.callbackDados = '';

    -- Envia requisição com o voto
    event.post( { class='tcp', type='data', connection=self.connection,
        value='GET http://192.168.0.1/enquete.php?eparam='..pEnquete..'&param='
            ..pAlternativa..' HTTP/1.0\nHost: 192.168.0.1\n\n', timeout=5 } );

end
```

Enquete

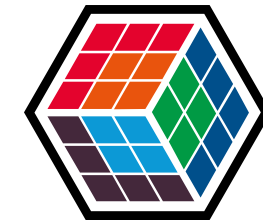


THE
DEVELOPER'S
CONFERENCE

➤ Enviando voto

```
if (pEvento.type == 'connect') then ...  
elseif (pEvento.type == 'data') then  
  
    if (self.callbackFunction ~= nil) then  
        self.callbackDados = self.callbackDados .. pEvento.value;  
    end  
  
elseif (pEvento.type == 'disconnect') then  
  
    if (self.callbackFunction ~= nil) then  
        self.callbackFunction(self.callbackDados);  
        self.callbackFunction = nil;  
    end  
  
end
```

Previsão do Tempo



THE
DEVELOPER'S
CONFERENCE

- OpenWeatherMap (openweathermap.org)
- Acesso a API diretamente pela aplicação
- Ou podemos acessar nosso servidor que fará as requisições à API do OpenWeatherMap

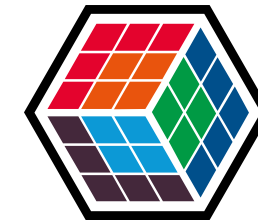
OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

- Tempo atual para uma ou mais cidades em um mesma requisição
- Previsão do tempo para 5 dias com intervalos de 3 horas
- Previsão do tempo para até 16 dias

OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

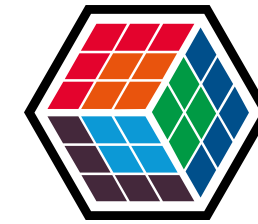
➤ API - Request x Response

Tempo atual

[http://api.openweathermap.org/data/2.5/
weather?q=Sao
%20Paulo,SP&mode=json&units=metric](http://api.openweathermap.org/data/2.5/weather?q=Sao%20Paulo,SP&mode=json&units=metric)

```
{
  "coord": {
    "lon": -46.65, "lat": -23.56
  },
  "sys": {
    "type": 1,
    "id": 4575,
    "message": 0.2213,
    "country": "Brazil",
    "sunrise": 1407317891,
    "sunset": 1407358002
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "Sky is Clear",
      "icon": "01d"
    }
  ],
  "base": "cmc stations",
  "main": {
    "temp": 22.04,
    "pressure": 1024,
    "humidity": 18,
    "temp_min": 20.2,
    "temp_max": 23
  },
  "wind": {
    "speed": 4.6,
    "deg": 100
  },
  "clouds": {
    "all": 0
  },
  "dt": 1407350134,
  "id": 3448439,
  "name": "Sao Paulo",
  "cod": 200
}
```

OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

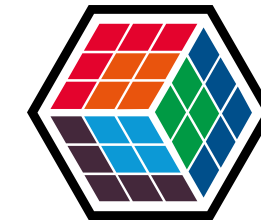
➤ API - Request x Response

Previsão do tempo para 7 dias

<http://api.openweathermap.org/data/2.5/forecast/daily?q=Sao%20Paulo,SP&mode=json&units=metric&cnt=7>

```
{
  "cod": "200",
  "message": 0.0611,
  "city": {
    "id": "3448439",
    "name": "Sao Paulo",
    "coord": {
      "lon": -46.6546,
      "lat": -23.5628
    },
    "country": "Brazil",
    "population": 0
  },
  "cnt": 7,
  "list": [
    {
      "dt": 1407337200,
      "temp": {
        "day": 25.59,
        "min": 11.97,
        "max": 27.21,
        "night": 11.97,
        "eve": 21.45,
        "morn": 17.27
      },
      "pressure": 942.32, "humidity": 35,
      "weather": [
        {
          "id": 800,
          "main": "Clear",
          "description": "sky is clear",
          "icon": "01d"
        }
      ]
    },
    { "speed": 3.12, "deg": 99, "clouds": 0 },
    { },
    { },
    { },
    { },
    { },
    { }
  ]
}
```


Ginga + OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

➤ Preparando o servidor

```
<?php
// Obtém os dados JSON da chamada a API OpenWeather
$jsonData = fopen('http://api.openweathermap.org/data/2.5/forecast/daily?q=Sao%20Paulo,SP&units=metric&cnt=6', "r");
echo stream_get_contents($jsonData);

?>
```


Ginga + OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
function callbackFunctionPrevisao(pJsonPrevisao)

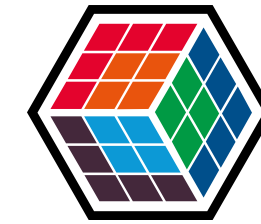
  -- Transforma o JSON em uma tabela
  tablePrevisao = json.decode(pJsonPrevisao);

  -- Passa a tabela com a previsão do tempo para o
  -- componente responsável pela exibição desse conteúdo
  previsao:exibir(tablePrevisao);

end

-- Iniciar download de dados
conexao:previsao(callbackFunctionPrevisao);
```

Ginga + OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
function previsao(pCallbackFunction)

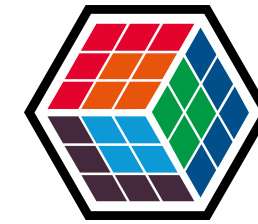
-- Função que deverá ser chamada ao receber os dados
self.callbackFunction = pCallbackFunction;

-- Onde os dados serão armazenados
self.callbackDados = '';

-- Envia requisição para download dos dados
event.post( { class='tcp', type='data', connection=self.connection,
  value='GET http://192.168.0.1/previsao.php HTTP/1.0\nHost: 192.168.0.1\n\n', timeout=5 } );

end
```

Ginga + OpenWeatherMap

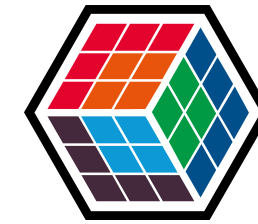


THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
if (pEvento.type == 'connect') then ...  
elseif (pEvento.type == 'data') then  
  
    if (self.callbackFunction ~= nil) then  
        self.callbackDados = self.callbackDados .. pEvento.value;  
    end  
  
elseif (pEvento.type == 'disconnect') then  
  
    if (self.callbackFunction ~= nil) then  
        self.callbackFunction(self.callbackDados);  
        self.callbackFunction = nil;  
    end  
  
end
```

Ginga + OpenWeatherMap



THE
DEVELOPER'S
CONFERENCE

➤ Integrando a aplicação

```
function exibir(tablePrevisao)

  -- Nome da Cidade
  local cidade = tablePrevisao.city.name;

  -- Tabela com os dados para o primeiro dia (hoje)
  -- Para obter os próximos dias incrementar o indice da lista
  local tableDiaAtual = tablePrevisao.list[1];

  -- Temperatura atual
  local temperatura = tableDiaAtual.temp.day;

  -- Umidade atual
  local umidade = tableDiaAtual.humidity;

  -- Temperatura máxima para hoje
  local temperaturaMax = tableDiaAtual.temp.max;

  -- Temperatura mínima para hoje
  local temperaturaMinima = tableDiaAtual.temp.min;

  -- Icone com a condição climática
  local iconeCondicao = tableDiaAtual.weather[1].icon .. '.png';

end
```

Otimização de Imagens



THE
DEVELOPER'S
CONFERENCE

- Os recursos não são infinitos
- O receptor precisa fazer download de todo conteúdo
- Quanto menos arquivos, melhor
- Quanto mais leve, mais rápido o download
- Menos imagens = mapa de imagens (TexturePacker)
- Imagens mais leves = otimização (IMAGEmini)

TexturePacker



THE
DEVELOPER'S
CONFERENCE

> Demo

Ícones temperatura - 59KB

18 arquivos



Ícones temperatura - 23KB

2 arquivos



```
"01d.png":{
  "frame": {"x":94,"y":24,"w":24,"h":24},
  "rotated": false,
  "trimmed": true,
  "spriteSourceSize": {"x":13,"y":12,"w":24,"h":24},
  "sourceSize": {"w":50,"h":50}
},
"01n.png":{
  "frame": {"x":94,"y":2,"w":28,"h":20},
  "rotated": false,
  "trimmed": true,
  "spriteSourceSize": {"x":13,"y":13,"w":28,"h":20},
  "sourceSize": {"w":50,"h":50}
},
"02d.png":{
  "frame": {"x":84,"y":64,"w":40,"h":22},
  "rotated": false,
  "trimmed": true,
  "spriteSourceSize": {"x":7,"y":11,"w":40,"h":22},
  "sourceSize": {"w":50,"h":50}
},
```


TexturePacker



THE
DEVELOPER'S
CONFERENCE

➤ Demo

Antes

Bandeiras Olimpíadas de Sochi - 402KB

205 arquivos

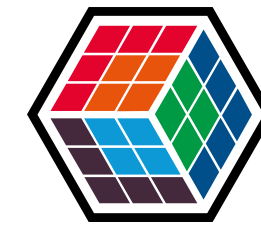
Depois

Bandeiras Olimpíadas de Sochi - 339KB

2 arquivos



IMAGEmini



THE
DEVELOPER'S
CONFERENCE

➤ Demo

Antes

Bandeiras Olimpíadas de Sochi - 303KB

Depois

Bandeiras Olimpíadas de Sochi - 287KB



IMAGEmini



THE
DEVELOPER'S
CONFERENCE

> Demo

Ícones temperatura

Antes: 19KB

Depois: 16KB



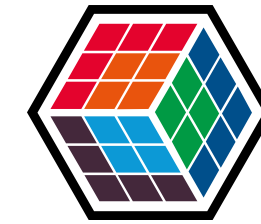
Demais imagens da aplicação (fundo, botões, etc)

Antes: 848KB

Depois: 706KB



Ginga + Mapa de Imagens



THE
DEVELOPER'S
CONFERENCE

➤ Preparando a aplicação

```
local ImageCacheCanvas = nil;
local ImageCacheIndexTable = nil;

function OBJDEF:startCache(JSONfile, IMAGEfile)
    local json = require("json")
    local raw_json_text = readAll(JSONfile)
    ImageCacheIndexTable = json.decode(raw_json_text)
    ImageCacheIndexTable = ImageCacheIndexTable.frames

    ImageCacheCanvas = canvas:new(IMAGEfile);
end

function OBJDEF:getImage(imageFileName)
    local newImageParams = ImageCacheIndexTable[imageFileName]

    local newImageCanvas = canvas:new(newImageParams.sourceSize.w, newImageParams.sourceSize.h)
    local trimmed = newImageParams.trimmed
    local spriteSourceX = 0;
    local spriteSourceY = 0;

    if (trimmed) then
        spriteSourceX = newImageParams.spriteSourceSize.x;
        spriteSourceY = newImageParams.spriteSourceSize.y;
    end
    newImageCanvas:compose(spriteSourceX, spriteSourceY, ImageCacheCanvas,
        newImageParams.frame.x, newImageParams.frame.y, newImageParams.frame.w, newImageParams.frame.h)
    newImageCanvas:flush()

    return newImageCanvas;
end
```

```
"01d.png":{
    "frame": {"x":94,"y":24,"w":24,"h":24},
    "rotated": false,
    "trimmed": true,
    "spriteSourceSize": {"x":13,"y":12,"w":24,"h":24},
    "sourceSize": {"w":50,"h":50}
},
```

Ginga + Mapa de Imagens



THE
DEVELOPER'S
CONFERENCE

➤ Preparando a aplicação

```
-- Icone
local nomeDoIcone = "01d.png";

-- (SEM MAPA) Pasta '/imagens/weather/' contém as 18 imagens dos ícones
local conditionTempImage = canvas:new("../images/weather/" .. nomeDoIcone);

-- (COM MAPA) Objeto 'TemperaturaImageCache' contém o mapa de imagens
local conditionTempImage = TemperaturaImageCache:getImage(nomeDoIcone);

-- Desenhar na tela
canvas:compose(100, 100, conditionTempImage);
```

Animações



THE
DEVELOPER'S
CONFERENCE

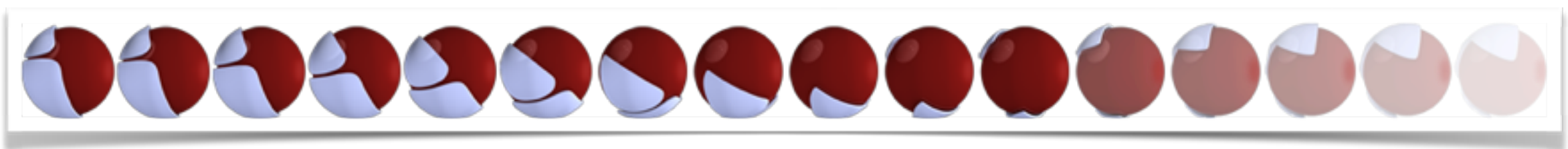
- As animações em Lua dependem da integração entre designer e desenvolvedor
- As animações podem ser feitas através keyframes ou manipulações da imagem

Animações

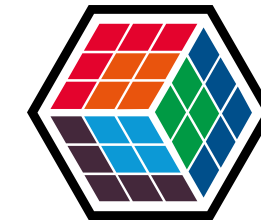


THE
DEVELOPER'S
CONFERENCE

➤ Keyframes



Animações



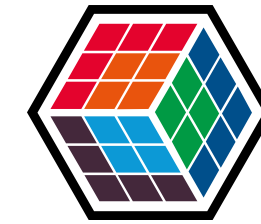
THE
DEVELOPER'S
CONFERENCE

➤ Keyframes

```
imagemLogo = {  
  "../images/invite/logo01.png",  
  "../images/invite/logo02.png",  
  "../images/invite/logo03.png",  
  "../images/invite/logo04.png",  
  "../images/invite/logo05.png",  
  "../images/invite/logo06.png",  
  "../images/invite/logo07.png",  
  "../images/invite/logo08.png",  
  "../images/invite/logo09.png",  
  "../images/invite/logo10.png",  
  "../images/invite/logo11.png",  
  "../images/invite/logo12.png",  
  "../images/invite/logo13.png",  
  "../images/invite/logo14.png",  
  "../images/invite/logo15.png",  
  "../images/invite/logo16.png",  
};  
imagemLogoW = 62;  
imagemLogoH = 64;
```

```
co = coroutine.create(  
  function (posX, posY, sizW, sizH, imagens)  
    frameAtual = 0;  
    lastRun = 0;  
    delay = 50; --ms  
  
    while frameAtual < #imagens do  
      actualTime = event.uptime();  
      if (delay < (actualTime - lastRun)) then  
        lastRun = actualTime;  
        frameAtual = frameAtual + 1;  
  
        canvas:attrColor(0, 0, 0, 0);  
        canvas:attrClip(posX, posY, sizW, sizH);  
        canvas:clear(posX, posY, sizW, sizH);  
        canvas:compose(posX, posY, canvas:new(imagens[frameAtual]));  
        canvas:flush();  
      end  
      coroutine.yield();  
    end  
  end  
)  
  
while coroutine.status(co) ~= 'dead' do  
  coroutine.resume(co, 100, 100, self.imagemLogoW, self.imagemLogoH, self.imagemLogo);  
end
```

Animações



THE
DEVELOPER'S
CONFERENCE

➤ Manipulações

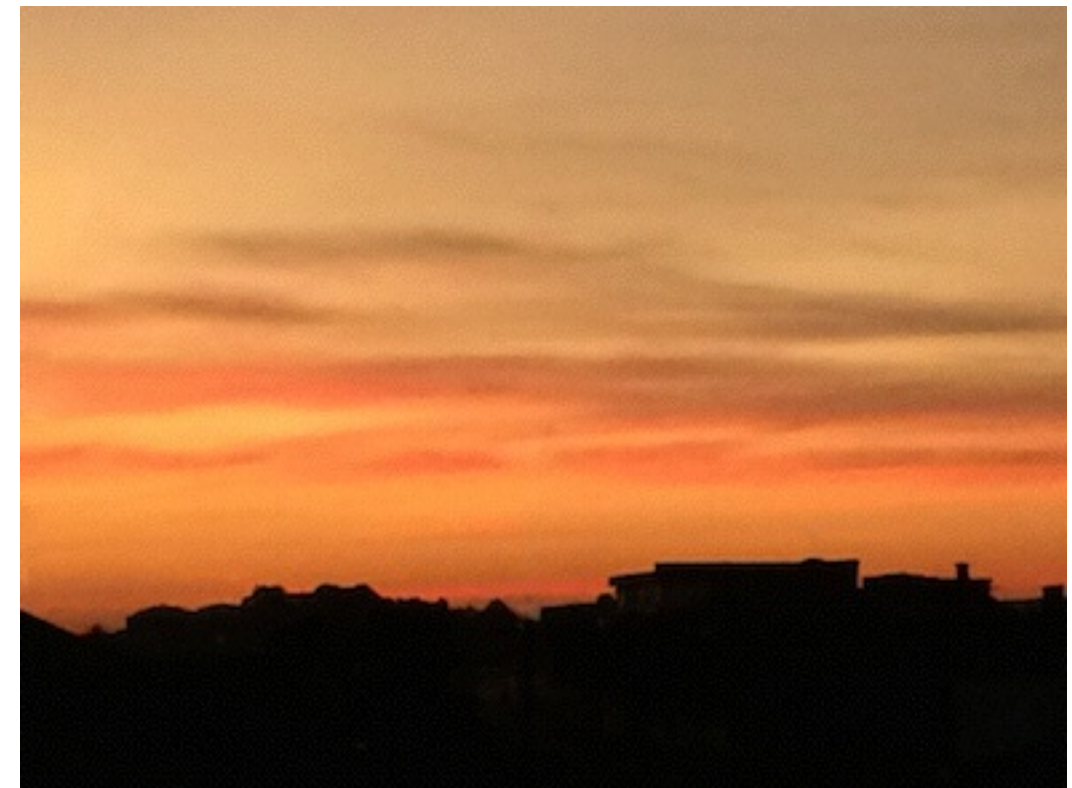
```
local imagem = canvas:new('foto.jpg');
local imagemUnder = canvas:new('fundo');
local canvasPai = canvas;
local sizeW, sizeH = sizeW, sizeH;
local x, y = 100, 500;
local step = 30;

co = coroutine.create(
  function (posX, posY, sizW, sizH, stp, img, imgUnder, cvsPai)
    frameAtual = 0;
    lastRun = 0;
    delay = 10; --ms

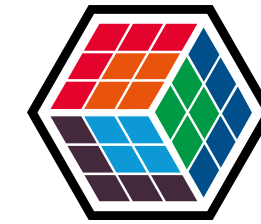
    while frameAtual < sizH do
      actualTime = event.uptime();

      if (delay < (actualTime - lastRun)) then
        lastRun = actualTime;
        frameAtual = frameAtual + stp;
        if (frameAtual > sizH) then
          frameAtual = sizH;
        end
        -- Seleciona apenas a área afetada
        cvsPai:attrClip(posX, posY, sizW, frameAtual);
        -- Limpa a área afetada
        cvsPai:clear(posX, posY, sizW, frameAtual);
        -- Desenha o fundo
        cvsPai:compose(posX, posY, imgUnder, 0, 0, sizW, frameAtual);
        -- Desenha o conteúdo
        cvsPai:compose(posX, posY, img, 0, sizH - frameAtual, sizW, frameAtual);
        -- Envia para a tela
        cvsPai:flush();
      end
      coroutine.yield();
    end
  end
)

while coroutine.status(co) ~= 'dead' do
  coroutine.resume(co, x, y, sizeW, sizeH, step, imagem, imagemUnder, canvasPai);
end
```



Animações

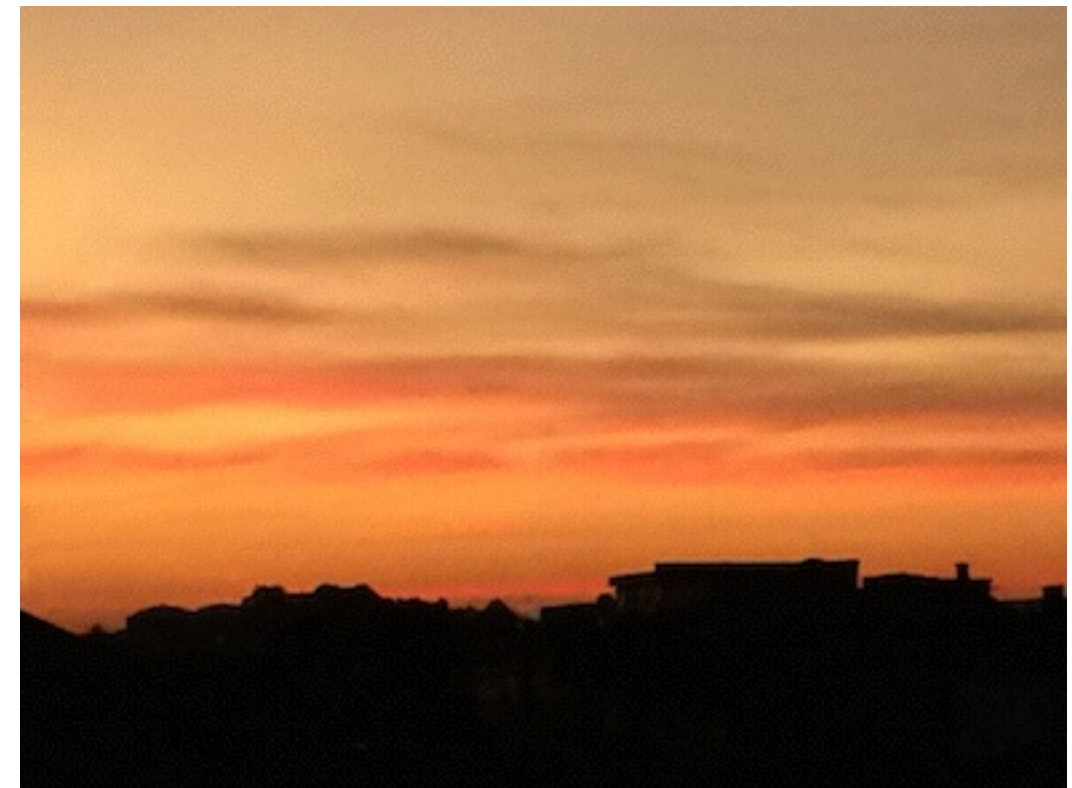


THE
DEVELOPER'S
CONFERENCE

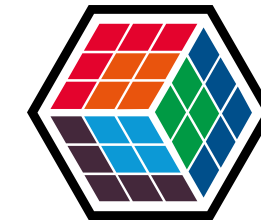
➤ Manipulações

```
local imagem = canvas:new('foto.jpg');
local imagemUnder = canvas:new('fundo');
local canvasPai = canvas;
local sizW, sizH = sizeW, sizeH;
local x, y = 100, 500;
local step = 30;

co = coroutine.create(
  function (posX, posY, sizW, sizH, stp, img, imgUnder, cvsPai)
    local totalW = 1280 - self.inParentPositionX;
    frameAtual = totalW * -1;
    lastRun = 0;
    delay = 10; --ms
    while frameAtual < 0 do
      actualTime = event.uptime();
      if (delay < (actualTime - lastRun)) then
        lastRun = actualTime;
        frameAtual = frameAtual + stp;
        if (frameAtual > 0) then
          frameAtual = 0;
        end
        cvsPai:attrClip(posX, posY, totalW, sizH);
        cvsPai:clear(posX + (frameAtual * -1), posY, totalW + frameAtual, sizH);
        cvsPai:compose(posX, posY, imgUnder);
        cvsPai:compose(posX + (frameAtual * -1), posY, img);
        cvsPai:flush();
      end
      coroutine.yield();
    end
  end
)
while coroutine.status(co) ~= 'dead' do
  coroutine.resume(co, x, y, sizeW, sizeH, step, imagem, imagemUnder, canvasPai);
end
```



Animações



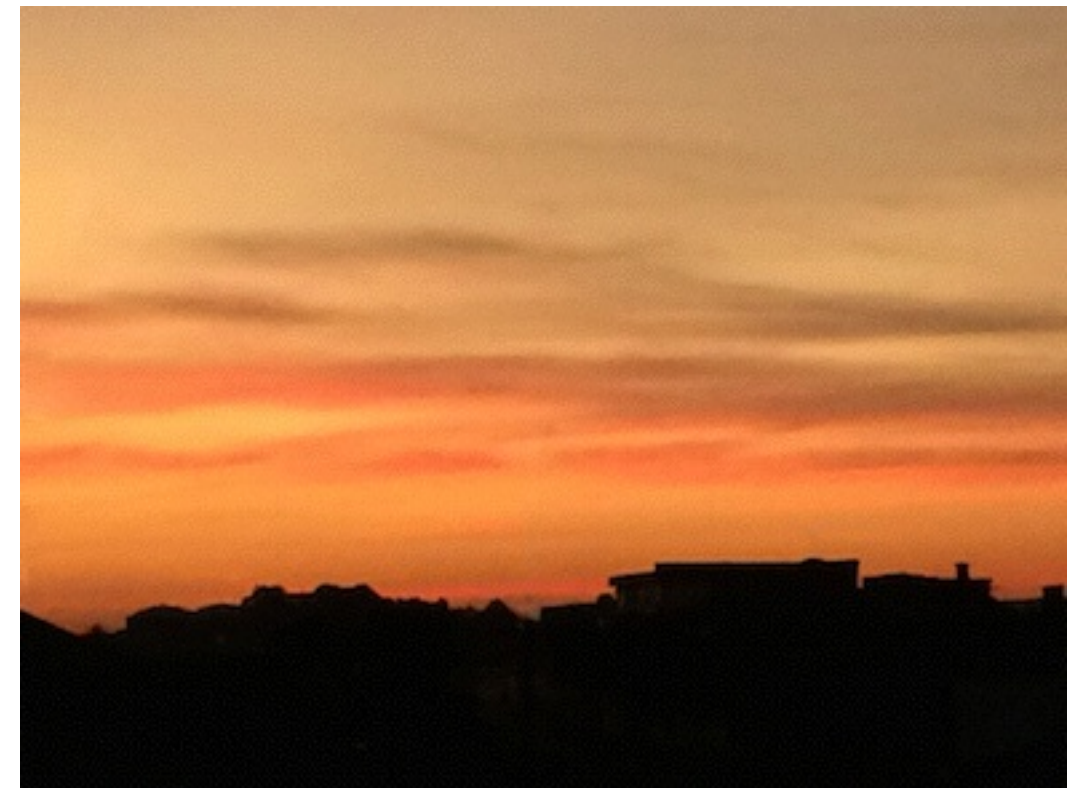
THE
DEVELOPER'S
CONFERENCE

➤ Manipulações

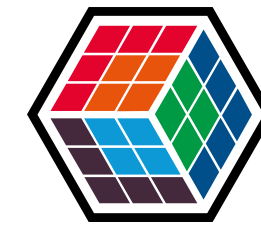
```
local imagem = canvas:new('foto.jpg');
local imagemUnder = canvas:new('fundo');
local canvasPai = canvas;
local sizeW, sizeH = sizeW, sizeH;
local x, y = 100, 500;
local step = 20;
co = coroutine.create(
  function (posX, posY, sizW, sizH, stp, img, imgUnder, cvsPai)
    lastRun = 0;
    delay = 10; --ms
    transparency = 0;

    while transparency < 255 do
      actualTime = event.uptime();
      if (delay < (actualTime - lastRun)) then
        lastRun = actualTime;
        transparency = transparency + step;
        if (transparency > 255) then
          transparency = 255;
        end
        cvsPai:attrClip(posX, posY, sizW, sizH);
        cvsPai:clear(posX, posY, sizW, sizH);
        cvsPai:compose(posX, posY, imgUnder);
        img:attrOpacity(transparency);
        cvsPai:compose(posX, posY, img);
        cvsPai:flush();
      end
      coroutine.yield();
    end
  end
)

while coroutine.status(co) ~= 'dead' do
  coroutine.resume(co, x, y, sizeW, sizeH, step, imagem, imagemUnder, canvasPai);
end
```



Animações



THE
DEVELOPER'S
CONFERENCE

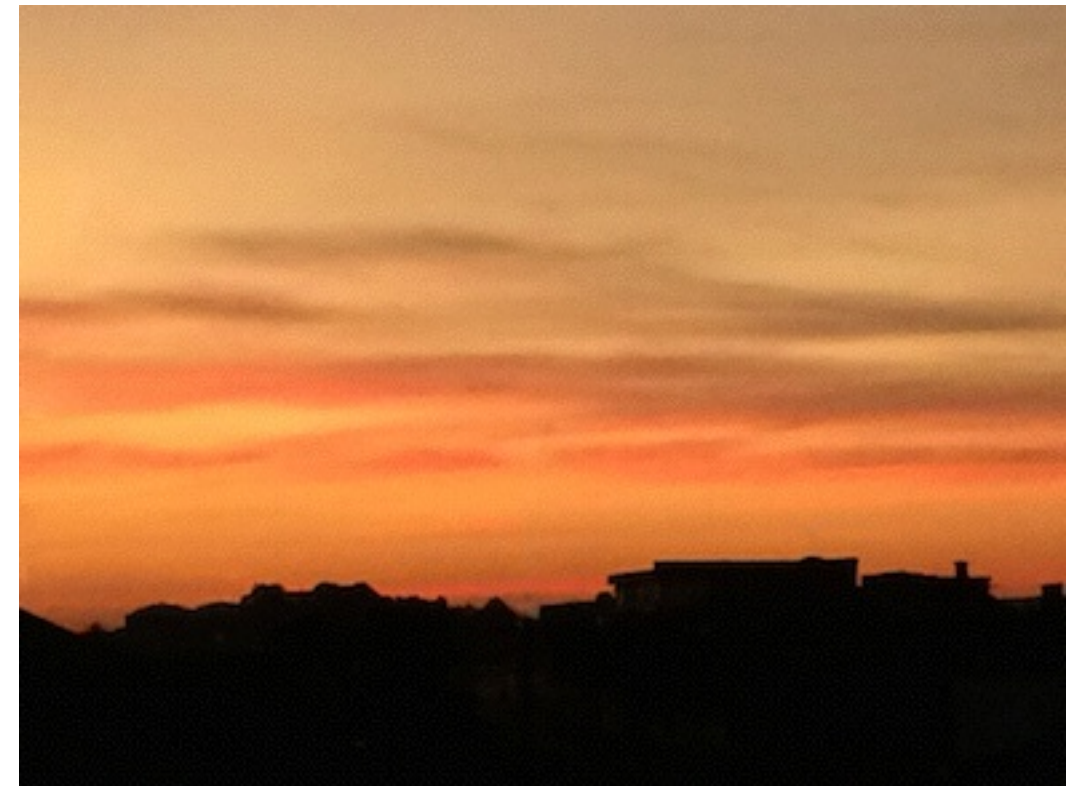
➤ Manipulações

```
local imagem = canvas:new('foto.jpg');
local imagemUnder = canvas:new('fundo');
local canvasPai = canvas;
local sizeW, sizeH = sizeW, sizeH;
local x, y = 100, 500;
local step = 0.1;
co = coroutine.create(
  function (posX, posY, sizW, sizH, stp, img, imgUnder, cvsPai)
    lastRun = 0;
    delay = 10; --ms
    scale = 0;
    transparency = 0;
    while scale < 1 do
      actualTime = event.uptime();
      if (delay < (actualTime - lastRun)) then
        lastRun = actualTime;
        scale = scale + step;
        if (scale > 1) then
          scale = 1;
        end
        resizeW = sizW * scale;
        resizeH = sizH * scale;
        img:attrScale(resizeW, true);

        transparency = transparency + 26;
        if (transparency > 255) then
          transparency = 255;
        end
        img:attrOpacity(transparency);

        cvsPai:attrClip(posX + ((sizW - resizeW) / 2), posY + ((sizH - resizeH) / 2), resizeW, resizeH);
        cvsPai:compose(posX + ((sizW - resizeW) / 2), posY + ((sizH - resizeH) / 2), img);
        cvsPai:flush();
      end
      coroutine.yield();
    end
  end
)

while coroutine.status(co) ~= 'dead' do
  coroutine.resume(co, x, y, sizeW, sizeH, step, imagem, imagemUnder, canvasPai);
end
```



Links



THE
DEVELOPER'S
CONFERENCE

➤ Google Analytics

➤ www.google.com/analytics

➤ OpenWeatherMap

➤ openweathermap.org

➤ TexturePacker

➤ www.codeandweb.com/texturepacker

➤ IMAGEmini

➤ www.effectmatrix.com/mac-appstore/image-mini.htm

Obrigado



THE
DEVELOPER'S
CONFERENCE

- Fabio Eduardo Angeli
- fangeli@sp.rederecord.com.br