21212 Academy

GUIA DE LANDING PAGES PARA DESENVOLVEDORES

Intro

Na 21212 sempre ensinamos o jeito "mais simples" de se criar Landing Pages (mais sobre elas adiante). Porém, já ouvi de colegas geeks dúvidas do tipo: "o unbounce.com é muito complicado, só pra fazer 1 paginazinha!". Então resolvi escrever este post para explicar de um jeito mais "direto ao ponto" pra galera técnica o que é uma Landing Page, pra que ela serve e algumas sugestões de como implementá-las.

O que é uma Landing page?

Pra quem segue a metodologia "Lean Startup", uma Landing Page é uma das formas mais simples de se validar hipóteses antes de sair desenvolvendo um produto (ou um MVP) baseado em "achismo".
Volta e meia recebo contato de pessoas que estão precisando de um back ou frontend developer ou precisa de indicação de um desenvolvedor Android ou iOS.
Minha resposta inicial é sempre a mesma: "precisa mesmo?" já validou se a sua solução resolve realmente a dor de alguém?
Não vou explicar o que qualifica uma landing page para uma validação mais eficiente, pois há muitos artigos online. Se precisar de um ponto de partida, comece com o básico.

Ferramentas para criação de LPs

Creio que a ferramenta mais popular para se criar uma Landing Page da internet seja o unbounce.com, por ser bem antigo e também pela facilidade de uso. Com um editor visual você monta páginas de layouts variados, com teste A/B e analytics integrado. Molezinha.
Entretanto, utilizar uma ferramenta visual pra validar uma hipótese é matar barata com tiro de canhão se você for um designer ou developer. Existem maneiras muito mais "simples" (relativamente falando, claro) para quem é iniciado em tecnologia.

A estrutura de uma landing page

Uma landing page nada mais é do que uma página estática (a principio sem conteúdo dinâmico), com analytics para medir a conversão (número de visitas x número de capturas) e um meio de armazenar os emails cadastrados.
A estrutura é normalmente uma ou mais imagens, uma proposta de valor e um form de contato. Franciscanamente simples.

Landing page em qualquer lugar free (heroku, github pages, s3, etc)

Vou separar a nossa Landing Page em 2 partes:

1. parte estática - HTML + JS + CSS com a cara da página e analytics
2. webservice - backend responsável por armazenar os emails cadastrados

Para disponibilizar nossa Landing Page existem várias formas gratuitas hoje em dia:

- heroku.com - só criar um projeto vazio com um arquivo index.php, sem código PHP, só com HTML mesmo
- github pages - as instruções dos caras é super simples, com 5 passos
- s3 - moleza caso você já tenha conta na Amazon AWS (free por 1 ano, caso não tenha conta ainda)
- para o DNS recomendo usar o cloudflare.com por ser gratuito ou o route 53 caso já tenha conta na Amazon AWS

Para fazer o armazenamento dos emails enviados no form de contato também existem várias soluções gratuitas hoje em dia:

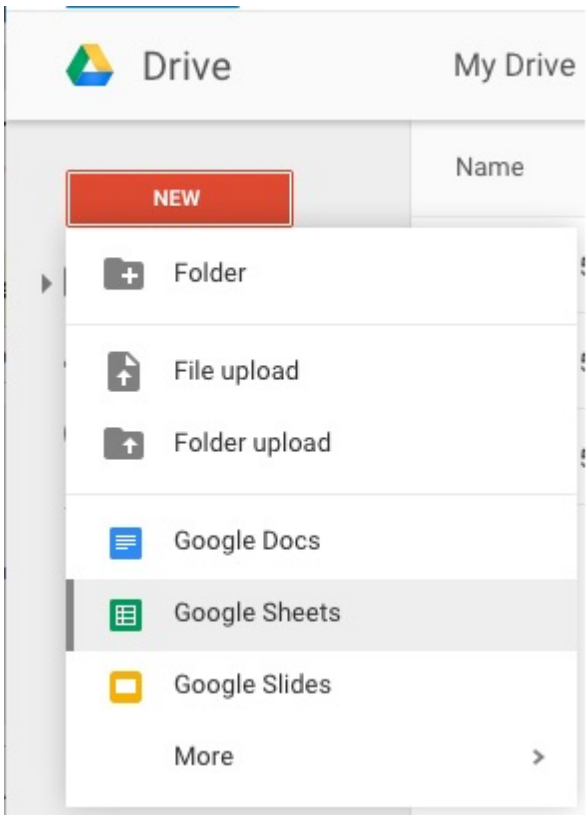
- heroku.com - limite de 25Mb de storage
- firebase.com - limite de 100Mb de storage
- parse.com - limite de 20Gb de storage
- Google Spreadsheets - ilimitado
- etc

As três primeiras opções são super fáceis de aprender, com documentação abundante. Já a utilização de Google Spreadsheet como webservice não há muita literatura disponível, então vamos focar nesta solução a partir daqui.

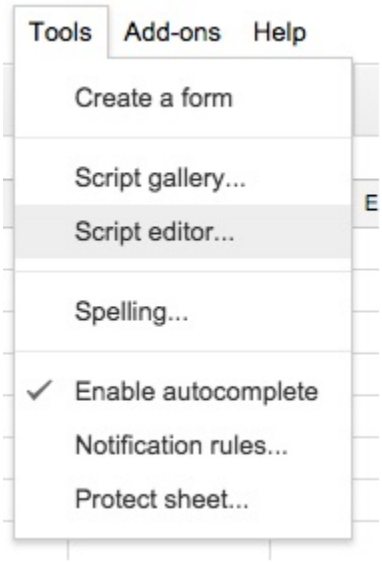
Armazenamento free

Segundo a Wikipedia, hoje em dia não há limitações no tamanho dos arquivos armazenados no Google Docs, provavelmente vão estar limitados pelo seu plano (business ou pessoal).
O que vamos fazer então é criar uma planilha que vai possuir algumas funções (scripts) que recebem as requisições vindas da web. Uma vez recebida uma requisição, vamos nos certificar de que a planilha possui espaço e vamos armazenar nela os dados recebidos. Vamos lá.

1. Crie uma planilha no Google Spreadsheet:



2. Crie a função do post

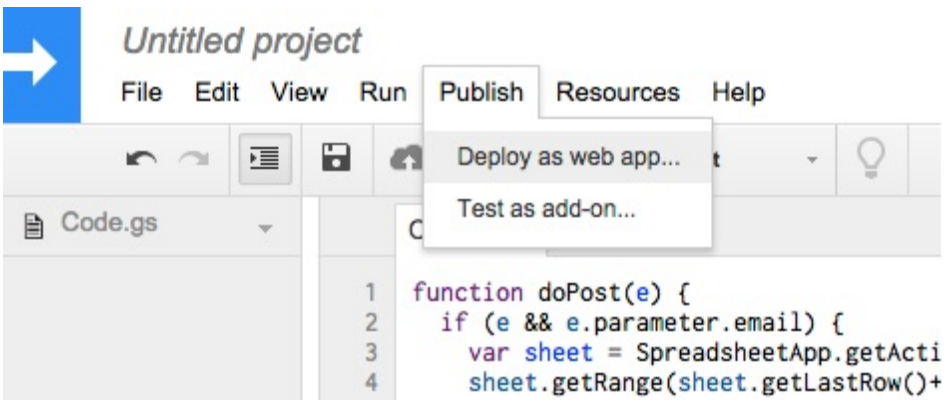


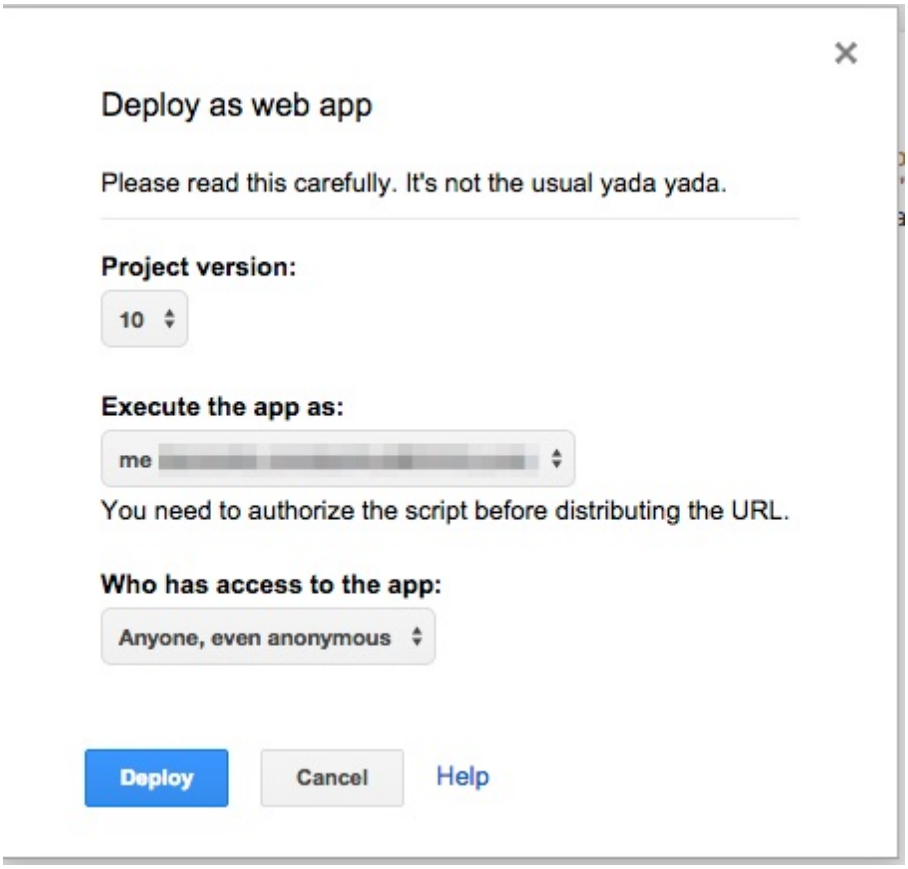

```
Code.gs
1  function doPost(e) {
2    if (e && e.parameter.email) {
3      // veja aqui como pegar o ID da planilha:
4      // https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app#openById(String)
5      var sheet = SpreadsheetApp.openById("1cA                                    g8Y8").getSheetByName("sheet1");
6      sheet.getRange(sheet.getLastRow()+1,1,1,2).setValues([[new Date(), e.parameter.email]]);
7    }
8    return ContentService.createTextOutput(JSON.stringify({result: "OK"}))
9      .setMimeType(ContentService.MimeType.JSON);
10 }
```

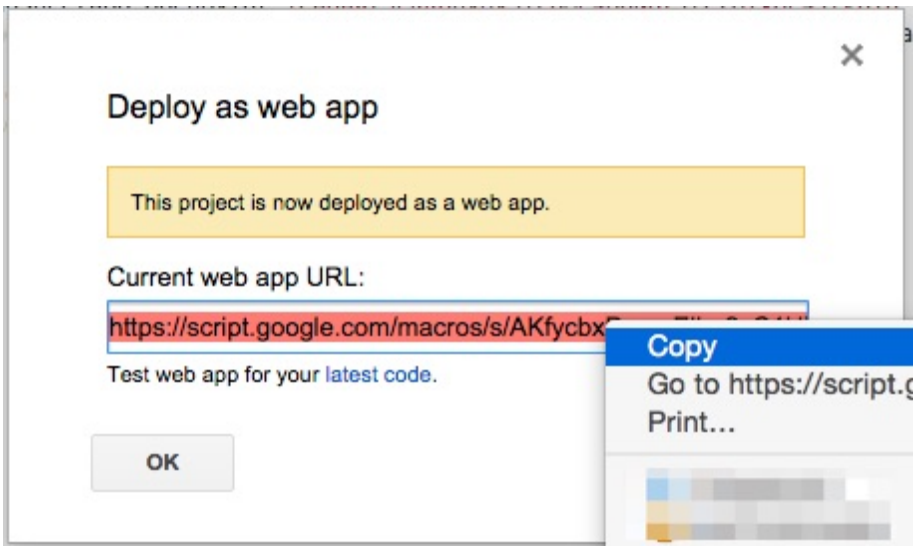
obs.: Para pegar o ID da planilha:

For example, the spreadsheet ID in the URL https://docs.google.com/spreadsheets/d/abc1234567/edit#gid=0 is "abc1234567".

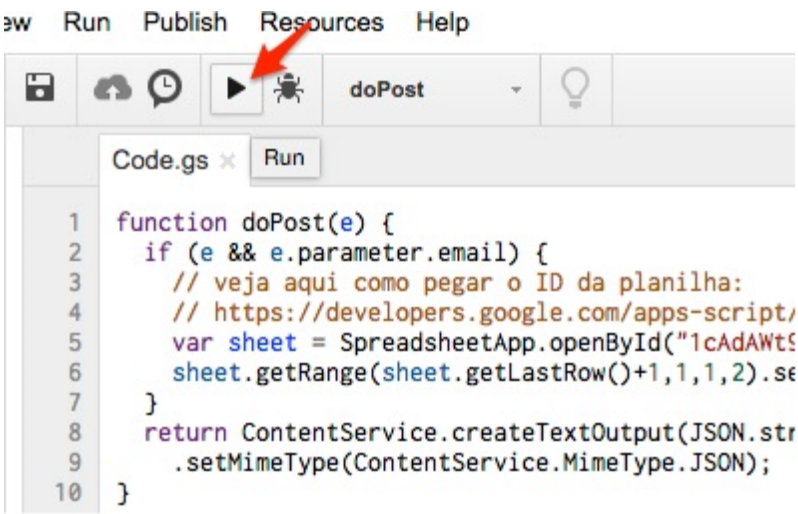
3. Especifique que esta planilha estará disponível como um webservice anonimo (qualquer um na internet pode acessar):



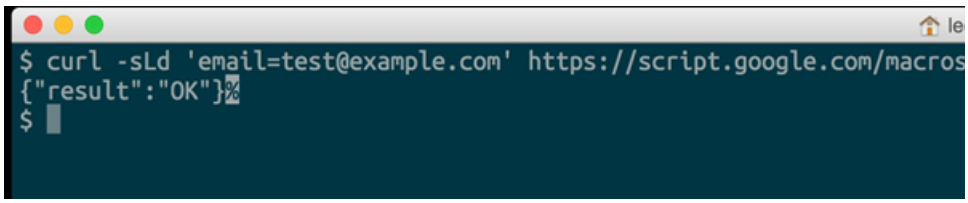




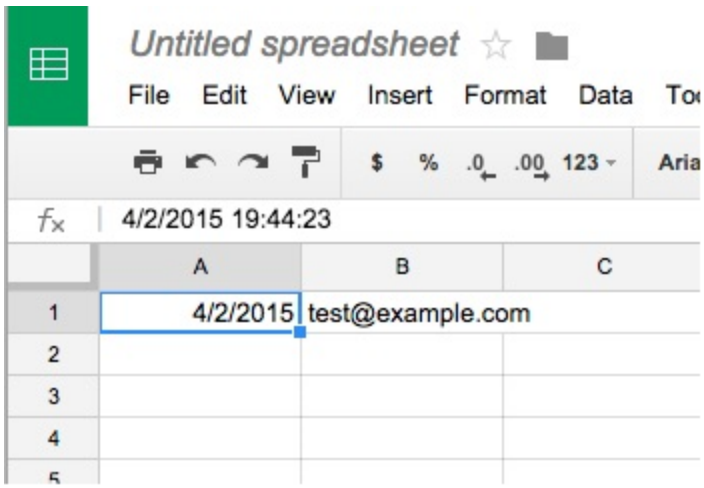
Pulo do gato: Após publicar seu script, clique no botão de executar (“play”) para poder autorizar a execução do script utilizando as suas credenciais:



4. Vamos testar nossa planilha/webservice:



e voilá…



Bônus - aonde buscar hipóteses para validar

Agora que você tem uma "máquina de validação", que tipo de soluções/dores você pode validar?

Seguem algumas idéias de onde buscar inspiração para sua próxima killer app / website:

- **Os 7 pecados capitais**
  - Gula
  - Avareza
  - Luxúria
  - Ira
  - Inveja
  - Preguiça
  - Soberba
- **Pirâmide de Maslow**
  - necessidades fisiológicas
  - necessidades de segurança
  - necessidades sociais ou de amor
  - necessidades de estima
  - necessidades de auto-realização
- **As 6 necessidades humanas básicas**
  - Certeza
  - Incerteza / variedade
  - Significado
  - Conexão / amor
  - Crescimento
  - Contribuição
- **Hierarquia de níveis lógicos**
  - ambiente
  - comportamentos
  - capacidades
  - crenças e valores
  - identidade
  - missão / espiritualidade