

## Prueba Técnica

Ante todo, muchas gracias por tu tiempo. Esperamos que estas pruebas te resulten entretenidas y te sirvan para demostrarnos tus habilidades como desarrollador.

Por favor, sube el código de cada prueba a un repo de github. En función de tu disponibilidad de tiempo, elige 1, 2 o 3 de las siguientes pruebas y llega hasta donde puedas.

En caso de duda, eres libre de interpretar las instrucciones como mejor te parezca. Si aun así tienes alguna pregunta, escíbeme a [hector@stylesage.co](mailto:hector@stylesage.co)

Por favor, no compartas con nadie este documento ni los ficheros de datos.

### Cat-Clicker - Frameworks Front-End

Desarrollar en cualquiera de los frameworks de Javascript comúnmente utilizados (AngularJS, Angular, Vue.js, React, ember, Backbone...) una aplicación con la siguiente funcionalidad:

#### Paso 1:

Una app que muestre la foto de un gato y que cuente el número de veces que se hace click sobre el gato. El contador se debe mostrar al lado del gato. No importa el diseño o maquetación.

#### Paso 2:

La aplicación debe mostrar 2 gatos. Para cada gato se debe mostrar: foto, nombre, número de clicks sobre el gato.

#### Paso 3:

La aplicación pasa a tener dos áreas. (A) una lista de gatos por nombre, (B) un área de detalle de gato. La aplicación debe mostrar al menos 5 gatos. En el detalle de gato se debe mostrar el nombre, foto y número de clicks (sobre la foto). El usuario puede hacer click sobre los nombres de la lista de gatos para cambiar el gato visible en el área de detalle.

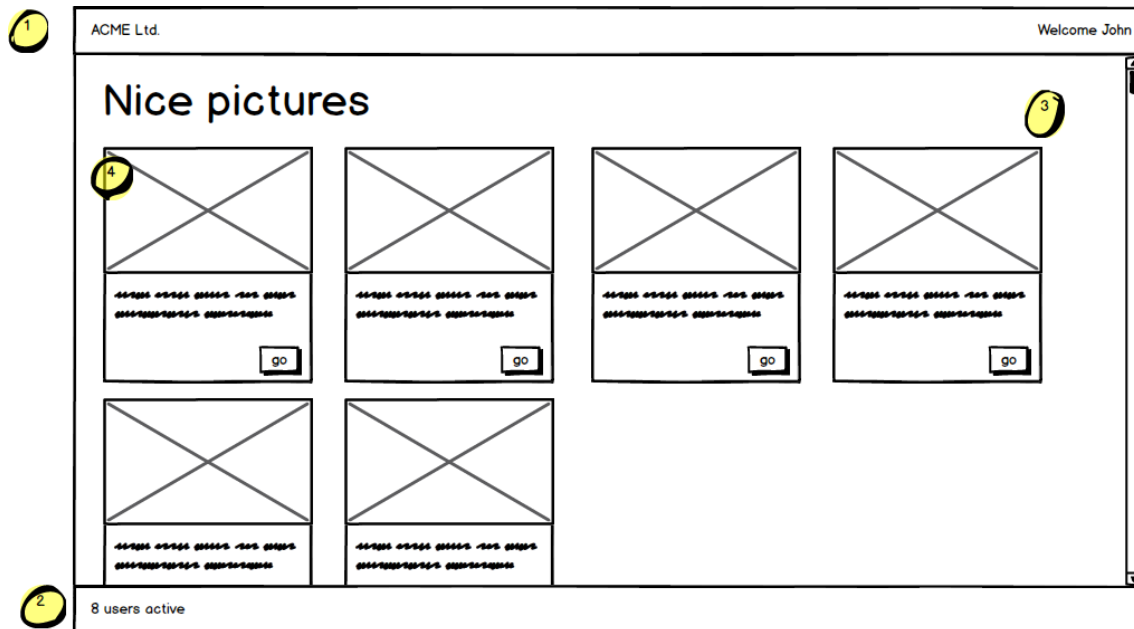
Puntos a valorar:

- estructura y claridad del código
- separación de responsabilidades y componentes
- uso adecuado del framework elegido
- progresión del código de más simple a más complejo paso por paso
- mantenibilidad

## Gifs Animados - Maquetación CSS

Desarrolla, con HTML y CSS una vista según las especificaciones de más abajo. Puedes usar Javascript (con cualquier framework) para crear la vista.

Puedes partir de Bootstrap o cualquier otro framework CSS con el que estés familiarizado. Puedes usar CSS o LESS/SASS/Stylus/...:



- 1. un encabezado de alto fijo, con dos secciones una alineada a la izquierda y otra a la derecha
- 2. un pie de página de alto fijo
- 3. un área principal con scroll vertical que muestre un título y una “rejilla” de bloques de imagen. Es deseable que tenga scroll vertical, que las imágenes se coloquen bien independientemente de cuantas haya, que la composición quede equilibrada en distintos tamaños de pantalla...
- 4. cada bloque de imagen debe tener: la imagen (sin deformar), un texto (que puede ser de longitud variable, incluso ocupar varias líneas) y un botón. Es deseable que todas las imágenes ocupen lo mismo (aunque los ficheros tengan tamaño distinto), que los bloques tengan el mismo tamaño (aunque imágenes y texto sean distintos), que los botones queden alineados abajo a la derecha...

¿Lo puedes hacer responsive?

Adjuntamos un fichero de urls de imagen y texto para que te sirvan de datos de prueba (gifs.json).

Puntos a valorar:

- uso correcto del marcado HTML
- uso correcto de las propiedades CSS
- limpieza y mantenibilidad del código CSS, estructura de clases, etc.

## Cupones - Visualización y Análisis de Datos

Dado el fichero cupones.json que contiene datos de cupones de descuento puedes elegir entre a) una visualización de datos y b) un análisis de los mismos datos (o las dos cosas!)

Los datos que se incluyen para cada cupón son:

```
"coupons": [  
  {  
    "country_code": "us",  
    "coupon_id": 259887,  
    "coupon_webshop_name": "Macy's & Promo Codes",  
    "description": "25% OFF Sale Save 25% Off Petite Tops ...",  
    "first_seen": "2017-09-06",  
    "last_seen": "2017-09-06",  
    "promotion_type": "percent-off",  
    "title": "25% Off Petite Tops",  
    "value": 25,  
    "webshop_id": "macys"  
  },  
  {  
    ...  
  }  
]
```

Explicación de los campos:

- country\_code: código de país, siempre es 'us'
- coupon\_id: id único de cada cupón
- coupon\_webshop\_name: nombre del retailer
- description: descripción del cupón
- first\_seen: fecha inicial
- last\_seen: fecha final
- promotion\_type: tipo de promoción, puede ser **percent-off**, **buy-one-get-one**, **dollar-off**, **free-gift**, **free-shipping**
- title: título del cupón
- value: valor numérico, para los cupones 'percent-off' es el porcentaje de descuento, para los cupones 'dollar-off' es el descuento en dólares
- webshop\_id: código del retailer

## Visualización de Datos

Presentar en un navegador web una visualización de datos que permita extraer conclusiones útiles. Puntos que vamos a valorar:

- librerías utilizadas (d3.js (v4 preferiblemente), o cualquier otra)
- claridad en la presentación de los datos
- limpieza y estructura del código
- interactividad

Tienes libertad para tomar las decisiones que quieras: limitar los datos, resumirlos, cambiar el enfoque...

## Análisis de los Datos

Usando Python o Javascript (node.js o en browser)

Analizar el fichero de cupones.json para obtener estadísticas, p.ej:

- cuántos cupones hay de cada tipo
- de los cupones percent-off, cuantos hay con cada descuento, mínimo, máximo, promedio
- de los cupones dollar-off, cuantos hay con cada descuento, mínimo, máximo, promedio
- las mismas estadísticas pero agrupado por retailer
- extraer las palabras clave relevantes de 'title' y 'description'

La salida puede ser un JSON o preferiblemente una pantalla web en la que se presenten los resultados.

Puntos a valorar:

- librerías utilizadas (lodash en JS?, pandas en Python?)
- limpieza y estructura del código
- claridad en la presentación
- interactividad?

## Bola Extra: Recomendado si no conoces Django

Haz el tutorial de Django: <https://docs.djangoproject.com/en/2.2/intro/tutorial01/>