Introdução

Como membro da equipa 5, com background em artes digitais e design gráfico e com os conhecimentos em front-end development adquiridos durante o curso na EDIT, estive envolvida deste o início no processo de criação da campanha para o desafio #1 proposto pela marca Marques Soares: "Relançamento do cartão cliente versão conta corrente, agora com o nome "cartão Marques Soares +", aumentando o número de clientes e realçando os seus benefícios".

Adoptando uma postura de entre-ajuda, além de cumprir com o meu papel de front-end developer, participei activamente nas fases de pesquisa, definição de conceito e estratégia, definição do claim de campanha, moodboard, e ajudei na criação e estruturação de alguns conteúdos textuais e no design de interfaces de alguns elementos para a Landing Page.

Processo de trabalho

Após definição do conceito e estratégia da campanha (com atraso de uma semana), chegou-se à conclusão que estas seriam as duas peças digitais centrais da nossa proposta: landing page do cartão Marques Soares + com página de adesão simplificada + PWA exclusiva a clientes com cartão Marques Soares + (Progressive Web App – resumidamente, tem praticamente o mesmo aspecto e comportamento de uma app nativa, mas é servida no browser, não requerendo download e instalação no dispositivo e tendo um custo de produção bem mais baixo do que uma app nativa). Tendo em conta o tempo que nos restava para desenhar e desenvolver estas duas peças, decidiu-se que seria melhor focarmos os esforços do front-end development na primeira peça, sendo que a segunda peça seria apresentada através de protótipo de InVision.

A divisão de trabalho pelos dois front-end developers foi definida da seguinte forma: eu fiquei responsável por desenvolver a página home da landing page, enquanto que o Jimmy ficou a cargo da página do formulário de adesão.

Processo de trabalho *(continuação)*

Partimos de uma mesma base de CSS, estruturada por mim e composta por: normalize.css (https://github.com/necolas/normalize.css), setup das fontes tipográficas escolhidas (@import da font "Libre Baskerville" do Google Fonts e @font-face da font-family "Gotham"), variáveis/classes com as cores escolhidas e grelha criada em sass contemplando 4 breakpoints (mobile first, min-width 600px, min-width 900px e min-width 1200px) com 12 colunas e espaçamento entre colunas de 24px (estes dois últimos valores foram definidos pelos designers responsáveis pelo UX&UI).

Dificuldades

Apesar dos esforços da minha parte nesse sentido, a designer responsável pelo layout da landing page decidiu não adoptar o conceito mobile first, tendo desenhado primeiro a versão desktop. Portanto, apesar de estar a desenvolver com base numa grelha mobile first, tive de ir criando primeiro o desktop, mas tendo sempre em mente que no final a organização do código teria de respeitar mobile first (à imagem do exercício desenvolvido no módulo de Responsive Development).

Apenas tive acesso ao design de layout "final" (desktop e mobile) no dia 7 de Fevereiro, ante-penúltimo dia da segunda semana dedicada ao desenvolvimento das peças, pelo que tive de trabalhar durante o fim-de-semana e ainda durante a semana dedicada à preparação da apresentação.

O layout desktop enviado pela designer através do InVision (https://invis.io/25QO97KA8YX#/345646288_landing_Mokup) continha 13 font-sizes diferentes (alguns mesmo muito próximos: 20px, 21px, 22px, 23px), e como ela não se mostrou disponível para uniformizar os font-sizes com a brevidade necessária para eu poder avançar com o trabalho, tive de tomar a decisão de ser eu a fazer essa uniformização, tendo em conta o layout por ela enviado e o meu bom senso.

Dificuldades
(continuação)

No dia 11 de Fevereiro, a 3-4 dias da apresentação final, durante a sessão de tutoria na EDIT, recebemos por parte do tutor Bruno Amorim o seguinte feedback em relação ao layout da landing page: o layout estava banal e pouco apelativo, tinha o problema de a parte das vantagens estar visualmente muito pesada e com demasiado texto, a utilização de um iPhone antigo tornava o layout datado, e era essencial incluir a imagem do cartão numa página cujo objectivo é comunicar esse mesmo cartão. Este feedback para mim já era de se esperar, e já há muito tempo que andava a tentar comunicar a minha opinião à designer acerca disso, principalmente em relação à inclusão da imagem do cartão no layout, mas a designer mostrou-se intransigente a esta ideia e a outras que eu e a pessoa responsável pela direcção de arte sugerimos.

Como a designer faltou a esta sessão de dia 11 de Fevereiro, não estando contactável, e o tempo era cada vez mais curto para o desenvolvimento, tivemos de em equipa decidir que as alterações ao layout sugeridas pelo tutor (ou pelo menos algumas delas) seriam feitas por mim em concordância com a aluna de direcção de arte. A designer foi posteriormente avisada desta decisão.

Apesar dos imensos esforços e longas horas dedicadas ao desenvolvimento front-end da landing page, no dia da apresentação final ao cliente e restante júri, apenas consegui ter uma versão desktop concluída com o slideshow de vídeos funcional (conteúdo textual a alterar consoante o vídeo) e algumas animações. O desenvolvimento responsivo já tinha sido iniciado, mas longe de estar finalizado. E ainda tinha ficado a faltar a funcionalidade da timeline do vídeo a indicar a duração do mesmo. Nas duas semanas seguintes à apresentação, concedidas às equipas para finalizar e organizar todo o material a entregar, terminei o desenvolvimento responsivo e consegui colocar a timeline do vídeo a funcionar (ainda que não tenha conseguido corrigir um erro que aparece na consola do browser assim que clicamos no vídeo seguinte do slideshow).

Soluções

Apesar de não ter recebido uma abordagem mobile first por parte da designer, tentei ainda assim ter em conta o mobile first na estruturação do código.

Para tornar o logo mais responsivo, utilizei a tag <figure> contendo a tag <img> com a versão símbolo do logo para os tamanhos de ecrã mais pequenos e com um srcset que inclui a versão completa do logo quando o ecrã é maior do que 900px. Tenho ideia que é mais comum utilizar-se JavaScript para estas situações, mas pareceu-me perfeitamente válido e até mais semântico resolver antes desta forma.

Com vista a tornar a secção das vantagens mais leve e lhe conferir algum dinamismo e interação por parte do utilizador, optei por esconder a descrição das 4 vantagens destacadas, sendo mostrada através de uma animação/transição assim que se faz hover sobre cada elemento.

Utilizei o Sass para programar/configurar os estilos do meu documento, porque realmente acredito que tem super-poderes e nos poupa imenso tempo, permitindo-nos, por exemplo, reutilizar código e optimizar tarefas repetitivas.

Em vez de utilizar a unidade REM para gerir os meus font-sizes ao longo dos vários breakpoints, optei por experimentar um método diferente, criando um @mixin que recebe como variáveis o tamanho e o valor a incrementar a cada breakpoint, retornando o font-size inicial (mobile first) igual ao valor passado no tamanho e iterando sobre cada um dos breakpoints para retornar em cada um deles o respectivo tamanho incrementado. O tamanho + o valor de incremento iria-me retornar sempre o mesmo valor para os 3 breakpoints, por isso, para obter o efeito pretendido tive de multiplicar a operação anterior pelo index do respectivo breakpoint.

Soluções
(continuação)

Para sistematizar os tamanhos das margins e paddings optei por utilizar sempre valores que fossem múltiplos ou divisores do valor da gridgutter (24px). E para optimizar a sua responsividade ao longo dos vários breakpoints, utilizei um @mixin muito parecido com o que usei para os font-sizes, mas que recebe, além do valor inicial da margem, a propriedade de css como variável, para poder ser mais dinâmico, permitindo utilizar tanto para margin como para padding, nas suas várias direcções.

Utilizei maioritariamente jQuery para programar as funcionalidades da página, no entanto quando tive de aceder directamente ao objecto do vídeo para utilizar os media events, tive de aceder através de JavaScript porque através do jQuery estava a aceder a um objecto de jQuery. Por esse motivo, existem alguns fragmentos de código em JavaScript nativo.

Utilizei as funções do jQuery de fadeOut e fadeIn para as transições dos vídeos e conteúdos do slideshow por falta de tempo para experimentar uma solução mais interessante, como por exemplo um efeito slide da direita para a esquerda.