

SR. VAGNER DINIZ: Bom dia a todos. Bom dia a todas. É uma satisfação poder fazer esse evento comemorativo do HTML5. E quero que todos vocês que participem, que estão participando desse evento possam ter momentos de bastante satisfação e de também agregação de conhecimento. A nossa ideia é trazer aqui pessoas que colaboraram, que contribuíram bastante com o desenvolvimento da linguagem HTML5, que é, efetivamente, um padrão do W3C que transformou a maneira de se fazer a apresentação de conteúdo na Web.

Aqueles que acompanharam o desenvolvimento da Web, desde a sua organização, criação por Tim Berners-Lee sabe que prover conteúdo na Web é algo extremamente simples, porque as tecnologias que são utilizadas são tecnologias muito simples, né? Entre elas, é o HTML. E o HTML, ele vem evoluindo desde quando isso foi colocado à disposição da comunidade, lá atrás, né? Ainda no ano de 1989. Portanto, é uma linguagem que a própria comunidade vem também participando e vem ajudando a desenvolver.

E quando nós chegamos naquilo que é chamado da versão 5, né? Isso foi uma evolução significativa, porque a versão 5 passou a conter diferentes elementos que puderam adotar, puderam incorporar as novas tecnologias que foram surgindo ao longo desse tempo, né? E antes mesmo do HTML5 tornar-se padrão em 2014, nós, do W3C, no Brasil e também internacionalmente, nós já começamos a mostrar para a comunidade a evolução que viria a se tornar um padrão efetivo, né? E eu lembro muito bem que na época, isso em 2010, nós criamos um curso de HTML5 já para anunciar as novidades e também para já começar a preparar a comunidade de desenvolvedores para aquilo que viria a ser um padrão, né? Nós fizemos um curso em julho de 2010, em setembro de 2010 também, o segundo curso. Depois, nós apresentamos o HTML5 em Brasília e em Belém. Tudo isso antecipando uma nova tecnologia, um novo padrão que seria efetivo. Foram quatro turmas. E, com isso, a gente formou uma comunidade de pessoas com conhecimento de HTML5, que depois foram pessoas que disseminaram esse conhecimento, foram facilitadores de outros cursos. E a gente tem a satisfação muito grande de poder contar hoje com boa parte das pessoas que participaram desses cursos há dez anos, né? E é com eles que nós vamos poder conversar, vamos poder debater diferentes temas sobre a evolução do HTML5, sobre a importância do HTML5 para a comunidade e para a indústria, para o mercado, né?

Nós preparamos diferentes painéis. E teremos uma satisfação, assim, muito grande de conversar com essas pessoas e, desde já, viu? Eu quero agradecer a todos vocês que vão participar, vão falar ao longo desse debate, porque vocês que participaram desse primeiro curso, sem dúvida, foram a ponta de lança de algo extremamente importante para comunidade de desenvolvedores no Brasil, né? E vocês abraçaram o HTML5 e ajudaram também a desenvolver a linguagem, até que ela se tornasse um padrão efetivo quatro anos depois, em 2014. Eu quero, sinceramente, agradecer pela importância que vocês tiveram ao longo desses anos para o W3C no Brasil, para o NIC.br, que sempre apoiou e sempre também hospedou o W3C como algo de ponta, como algo de importância para o desenvolvimento da Internet no Brasil.

Portanto, esse evento é um evento promovido pelo Ceweb, o Centro de Estudos de Tecnologias Web no Brasil. Nós estamos cada vez mais comprometidos com a evolução de uma Web que seja continuamente aberta e para todos, né? O Ceweb, ele é também aquele que hospeda o capítulo do W3C no Brasil, e nós temos uma satisfação muito grande de fazer isso. Portanto, quero desejar a todos que estão participando, aos debatedores, aos moderadores e à audiência como um todo uma boa atividade, um bom evento, uma boa conversa. E não esqueçam: mantenham sempre os seus microfones desligados para que não haja ruído durante a apresentação. Se vocês não estão fazendo a apresentação, se vocês não estão falando, por favor, mantenham a câmera desligada e também o áudio desligado. E não esqueçam também, se vocês

quiserem fazer alguma pergunta, fiquem muito à vontade. Mas façam sempre as suas perguntas pelo chat do evento, no YouTube. E ali nós vamos ter nossas pessoas, nossas pessoas, nossos profissionais coletando essas perguntas para serem repassadas para os debatedores.

E agora, então, eu quero chamar a Ana Duarte, que é profissional de Ceweb, que vai fazer moderação dessa primeira conversa, desse primeiro painel, que é: A Web antes do HTML5. Ana Laura... desculpa. Ana Duarte, você tem a palavra, por favor. Siga em frente. Muito obrigado, gente.

SRA. ANA DUARTE: Obrigada, Vagner. Bom dia, gente. Então, a gente vai dar início a esse painel de abertura para falar um pouco sobre o que aconteceu com a Web nos últimos dez anos e como ela transformou a tecnologia que a gente usa até hoje.

E para esse painel a gente conta com a participação da Ana Laura Gomes, que, desde 2000, atua como consultora de soluções da Adobe Creative Cloud e Document Cloud, e em 2016 ela se juntou ao time do Adobe Solution Consultant, das ferramentas criativas. E hoje ela atua como customer success manager, cuidando da satisfação dos clientes da Adobe na América Latina. A Ana Laura, ela também é autora, cria e produz materiais educacionais e artigos para blogs e revistas especializadas. E ela também é colaboradora do escritório do W3C Brasil.

A gente também conta com presença do Pedro Rogério de Assis, que é full stack overflow developer da Vitta e, nas horas vagas, ele posta dicas sobre desenvolvimento Web no Twitter.

E também teremos a presença do Diego Eis, que criou PM Letter, que é um newsletter sobre construção de produtos digitais. Ele escreveu o livro Gestão Moderna de Produtos Digitais, onde ele fala sobre gestão de produtos como serviço. Atualmente, ele trabalha como head de produtos digitais na Sympla. Ele adora ler e perde muito tempo com jogos casuais no celular.

Então, para começar, eu gostaria de saber quais foram as principais mudanças na Web nos últimos dez anos na opinião de vocês. Eu passo a palavra.

SR. DIEGO EIS: Vai lá, Ana.

SRA. ANA LAURA: Vou começar. Sabe que eu estava lembrando... ah, me preparando para conversar com vocês hoje e era muito diferente, era muito diferente. Eu lembro que a nossa maior preocupação se era 1024 por 768 ou 800 por 600. Então a gente vivia outros tempos.

Mas, desculpa, bom dia para todo mundo. Começar com educação, desculpa. Mas é que, assim, esse é um tema que é muito querido, é muito do meu coração. Eu gosto muito de HTML justamente por causa disso, né? Justamente por causa da... principalmente o HTML5, né? Por causa dessa transição que aconteceu. A gente tinha outra mentalidade, a gente pensava outras coisas. Era... não sei, era muito diferente. Era tudo muito estático, muito focado, às vezes, na máquina e no device, e quando veio o HTML5, uma das coisas mais bacanas foi justamente isso, falou assim: não, gente, vocês não estão entendendo, a gente está se comunicando. Isso aqui é para comunicar. Então, como que a gente faz isso melhor? Como que a gente atende melhor o usuário, né? Apesar de na época o grande mote era que o conteúdo era rei, já tinha muito a pegada que a gente vê hoje, do foco no usuário, da experiência do usuário. Então, eu acho que isso nasceu lá atrás. E era uma preocupação que a gente não tinha. Assim, tinha, mas talvez a gente não tinha o ferramental tão estruturado quanto a gente tem hoje.

SR. PEDRO ROGÉRIO: O que mudou realmente mesmo foi que antigamente a gente desenvolvia focado, igual você citou, em resolução. Pega ali a resolução de 1024, a gente, por exemplo, desenvolvia usando tabela e tinha que funcionar naquela resolução específica, no browser específico, e, tipo, era basicamente um inferno isso. Era meio complicado isso. Foi o que realmente mudou bastante isso, mudou bastante dessa época para cá. O Diego pode dizer muito mais que isso. Foi um dos caras, o precursor... acho que foi o precursor, né? Por causa do blog dele. Ele que, meu, basicamente, acho que muita gente aprendeu com ele. Eu, basicamente, sou fã dele, aprendi muito com ele. Se eu cheguei onde eu cheguei, eu devo muito a ele também.

SR. DIEGO EIS: Pô, legal, cara. Você vai me fazer ficar emotivo aqui na frente de todo mundo? Para complementar o que vocês falaram aí, era uma época de muita restrição, né? O que hoje a gente não tem, né? Hoje a gente tem até abundância de tecnologia, né? Ontem eu estava numa thread no Twitter falando sobre escrever CSS no JavaScript. Então, para quem participou dez anos atrás, participou até antes, né? Isso, se a gente não estiver preparado, soa até como pecado, né? Isso soa como algo errado. Tipo, a galera que faz [ininteligível] hoje está maluca: como assim? Escrever CSS? E aquele esquema de separar a informação da formatação e do comportamento, né? Tudo o que a gente pregava muito e falava muito, né? Então, hoje a gente vive numa época que não é uma época de abundância talvez, assim, mas tem muito menos restrições e a gente consegue explorar muito mais as tecnologias e as ferramentas que a gente tem disponíveis, criando outras tecnologias em cima, muitos níveis de abstração. E antes não, a gente tinha restrição de tecnologia de trabalho, então a gente tinha restrição do HTML, sim, que era muito curto ainda, né? Do CSS também, que era muito curto. Do JavaScript, que a gente não podia fazer tudo que a gente queria. A gente tinha restrição de conexão das pessoas. Não era todo mundo que estava conectado, não era todo mundo que tinha uma conexão boa, né? A gente estava falando aí de 2G, 3G. Não estamos falando de 4G ou 5G, ontem a Apple lançando, né? A gente tinha uma restrição muito grande também de conhecimento das pessoas, né? Não era todo mundo que sabia o que era fazer sites naquela época, fazer sistemas naquela época, fazer sistemas com boas práticas e padrões e tudo mais.

Então, era uma série de restrições que a gente estava tentando quebrar para conseguir, bom, chegar onde a gente chegou hoje, né? Então, era muito mato alto, estava todo mundo cortando aí o mato, né? Estava alto para caramba. Não é só esticar e cortar. Tinha que criar uma escadinha para a gente subir lá em cima e sair cortando. Então, quando a gente fala dessa época de dez anos atrás, era realmente uma época, assim, eram menos restrições do que quando a gente, sei lá, o Tableless saiu em 2003, naquela época tinha mais restrições do que 2010, com certeza. Mas, ainda assim, 2010 era uma época que estava em plena mudança, né? Ainda tinha guerra dos browsers, era uma guerra fria, mas ainda existia, né? Então, eu acho que hoje a gente está em uma época bem menos restritiva, mais abundante de tecnologia, com mais possibilidades, que antes a gente não tinha esse luxo, a gente tinha que fazer o que tinha... o que estava na mão ali, né?

SRA. ANA DUARTE: Legal, gente. Então, assim, nesse cenário, como era criar uma aplicação na Web em 2010? Quais que eram as dificuldades, quais que eram as soluções que vocês arranjavam na época?

SRA. ANA LAURA: A gente ia lá no Tableless, ia lá [ininteligível], copiava e colava o código. É verdade, cara. Eu usei muito isso.

[falas sobrepostas]

SR. PEDRO ROGÉRIO: Não via a hora de sair post novo lá. Não via a hora de sair post novo para aprender coisa nova. Era basicamente assim.

SRA. ANA LAURA: Pois é, era isso mesmo, cara. Olha só que delícia.

Era... eu acho que a gente imitava muito a parte impressa, sabe? Pensando ainda no que a gente fazia. Normalmente a tecnologia seguinte sempre copia a tecnologia anterior. E talvez a gente ainda fizesse muito isso quando a gente estava nessa época. A gente queria um livro, né? Eu lembro aquela coisa pavorosa de fazer a página rodar, assim. Falei: Gente, eu não quero ver uma página rodar. Eu vejo a página rodar no livro. Eu não quero, eu não preciso disso na Internet. Mas tem gente que precisa, então, vamos lá, né?

E eu acho que a gente criava muito... algumas pessoas já tinham, e algumas empresas, porque normalmente você faz o site para alguém, né? A empresa já tinha virado também a mentalidade para essa coisa de foco no conteúdo, como que meu conteúdo se apresenta melhor com essas ferramentas que eu tenho aqui, que é exatamente isso que o Diego falou, a gente tinha pouca coisa. A gente não tinha a quantidade de coisas... o HTML não tinha maturidade que a gente tem hoje, e o navegador também não tinha. Então, a gente, nossa, a gente vivia mega nesse cobertor curto, né? Cobre a cabeça, descobre o pé, cobre o pé, descobre a cabeça. Mas a gente criava, com o pouco recurso que a gente tinha, a gente conseguia criar algumas coisas.

Então, cada vez menos... cada vez menos eu não sei, estou generalizando uma coisa que não dá para generalizar, mas, assim, a gente economizava um pouco o banco de dados, porque o conteúdo precisava estar presente, quando que o conteúdo está presente, quanto tempo esse conteúdo vai descarregar, porque naquela época, gente, Internet era... Nossa Senhora, era muito ruim. Hoje a pior Internet não é tão ruim quanto a Internet boa daquela época. Então, a gente tinha super essa preocupação de ficar toda hora medindo: o que vem primeiro? O que descarrega? O que não descarrega? Então, era realmente uma construção baseada naquela escassez que a gente tinha. Então, a gente ainda não... criavam-se coisas lindas e faziam coisas maravilhosas, a gente conseguia fazer isso, mas, assim, era raro o cliente que dava essa oportunidade para a gente. Então, era muito código, muito estudar, muito estudar, para realmente dar uma qualidade para aquele material que a gente estava apresentando.

SR. PEDRO ROGÉRIO: O que eu gastei de dinheiro com revista. Ia na banca comprar revista para poder estudar. Não está...

SR. DIEGO EIS: Nossa, revista...

[falas sobrepostas]

SRA. ANA LAURA: A gente estudava na banca, cara, [ininteligível].

SR. PEDRO ROGÉRIO: Pedia dinheiro para a minha mãe: Me dá dinheiro que eu preciso comprar revista nova para aprender.

SR. DIEGO EIS: Revista WWW, várias.

SR. PEDRO ROGÉRIO: Depois começaram a surgir os blogs, aí a gente foi correndo atrás e aprendendo. Mas basicamente era assim antigamente. Época boa.

SR. DIEGO EIS: Acho que para construir uma aplicação aí em 2010 eu acho que a gente ainda... eu acho que a gente seguia muito manual ainda, né? Era uma coisa um pouco... era relativamente mais simples do que é hoje. É botão direito, nova pasta, né? Index.HTML, pronto. Hoje tem uma série de caminhos aí. Mas a

gente ainda tinha algumas preocupações que hoje não se tem, exatamente por causa do avanço da tecnologia aí, hoje não se tem, porque já é meio que automático. Isso já foi, entre aspas, resolvido, né?

Então, por exemplo, entre aspas, bem aspas aí, tá? Então, antes a gente se preocupava muito mais com semântica, né? Hoje a gente não tem tanto mais esse apreço, assim, né? O que gerou lá, gerou, acabou, beleza, bota no React, tudo junto ali e vamos embora. Hoje a gente está muito mais preocupado em acoplar as tecnologias do que o resultado final, realmente, delas, né? Antes a gente tinha muito apreço de separação entre camadas, por exemplo. Então quando a gente ia criar uma aplicação de verdade era: Beleza, aqui no front a gente vai separar assim, assim, assado. A gente passava muito tempo planejando como que a gente ia fazer o CSS, como a gente ia fazer o HTML, como a gente ia fazer o JavaScript, e o back-end, como a gente ia acoplar. Naquele... dez anos atrás não era todo mundo que usava API o tempo todo. Ainda a gente estava passando na transição da galera o back-end ainda entrar no HTML e fazer as coisas lá, né? Aí, obviamente, a galera mais sênior já começava a fazer as coisas mais novas muito mais parecidas do que é hoje.

A gente acho que tinha mais apreço à acessibilidade. Então, acho que a gente parava, como a gente escrevia o código HTML mais na mão, ele não era gerado, acho que a gente conseguia parar mais, pelo menos era a nossa briga, na época, né? Parar mais para conseguir implementar [ininteligível], para conseguir implementar até shortcuts no HTML ali e etc.

Então, acho que era... acho que a gente tinha preocupações, assim, que eram mais, talvez mais fundamentais, né? Que hoje provavelmente isso já se resolve... é mais diluído no processo de construção do que antes, né? Antes a gente tinha realmente que parar, sentar e pensar como isso ia ser feito, e era aquilo, né? Se a gente perdesse o CSS de vista, já era, a gente não conseguia voltar atrás. Nossa, quantas vezes eu já refiz projetos enormes por causa do CSS mal feito, né?

SRA. ANA LAURA: Diego, mas eu estou pensando aqui, eu acho também que aquela nossa batalha lá atrás, com relação a todo esse tipo de conteúdo, é que trouxe essas coisas automáticas de agora. Você não tem um pouco disso? A gente já foi... a gente era tão caxias naquela época que a gente acabou deixando os sistemas que hoje estão aqui ajudando a gente caxias também nesse sentido. Sabe?

SR. DIEGO EIS: Talvez.

SRA. ANA LAURA: É isso que você fala: o problema já está resolvido. A gente já colocou esse bias. Um bom bias.

SR. DIEGO EIS: Talvez. Eu acho que naquele tempo a gente tinha mais apreço. Hoje não sei tanto. Você pega um site feito em Vue, React, Angular, o que for aí, dessas libraries, dessas frameworks, o que for, e você olha o código, ele não está tão mais semântico assim. A galera não parou... Está cheio de Div(F), cheio de atributo, data attribute, blábláblá e etc., e está uma loucura. Então, acho que, assim, acho que o apreço para algumas coisas realmente se perdeu, assim. Eu acho que foi embora. Mas eu acho que hoje as automatizações, eu concordo com você. Algumas das automatizações que tem, das facilidades que tem, realmente, acho que foi exatamente por causa das boas práticas anteriores. Acho que resolveram coisas para caramba. Assim, hoje, webpack, por exemplo, um monte de coisa que eles fazem é exatamente se preocupando para resolver esses problemas anteriores que a gente tinha e que não dava para resolver. Desde otimização de imagem, até compilação, minificação de [ininteligível]. Então, sei lá.

SRA. ANA LAURA: E aí também, eu fiquei pensando, depois, a hora que você estava falando dos pacotes e tal, de que a gente hoje tem, talvez, uma preocupação, que aí talvez case com isso que você está dizendo

agora, né? De fazer... a gente voltou a se preocupar com o sistema, né? Porque a gente tem métrica. Cada clique que o cara dá eu preciso saber de onde eu vim, para onde eu vou, sabe?

Então eu acho que talvez essa... talvez a próxima revolução seja essa, né? Falar assim: Como é que eu faço a métrica? Como é que eu gero essa linguagem, que é uma linguagem de máquina, que eu estou fazendo para a máquina, que eu estou pensando na máquina, ela volta para o ser humano de novo, né? Será que talvez a gente consiga... e talvez isso seja até a maturidade do HTML5. Talvez ele e as outras linguagens, o JS, tudo mais, eles vão caminhar, não sei, nesse sentido.

SR. DIEGO EIS: É, eu acho que a gente vai falar até um pouquinho mais disso, mas eu concordo com você. Porque, assim, por exemplo, o HTML5, para mim, o maior pulo dele foi exatamente a semântica, né? Foi a gente tentar trazer um pouco mais desse significado para o código e etc., né? Então, se a gente consegue automatizar isso hoje, o problema não é automatizar, né? O problema não é, sei lá, uma máquina fazer alguma coisa ou a gente automatizar alguma coisa via tecnologias que a gente tem atuais, né? Ou a gente colocar CSS no JS, mas é a gente conseguir fazer do modo certo automatizado, né? A gente conseguir entregar o resultado que a gente queria entregar naquela época, que era o único jeito que a gente tinha, era sentar e digitar as coisas, mas a gente conseguir entregar esse resultado. A gente conseguir entregar um código que seja semântico, um site que seja acessível, a gente conseguir fazer uma escala do sistema de código, entre as camadas, de forma que seja... que a gente consiga manter isso durante mais tempo, né? Então, o como eu acho que mudou bastante. O como a gente entrega isso, esse meio do caminho acho que mudou bastante, né? Antes, como eu falei, criar novo arquivo index.HTML. Hoje é diferente. Então, a forma de executar acho que mudou. Talvez a essência ali tenha se mantido. Mas eu acho que... é que talvez eu seja saudosista, mas acho que hoje ficou mais fácil de perder essa essência. Acho que fica mais simples da galera perder as preocupações que antes a gente tinha, talvez.

SRA. ANA DUARTE: Beleza. Então, mudando agora um pouquinho o foco de soluções e falando um pouquinho sobre o usuário. O que o usuário consegue fazer hoje na Web que ele não conseguia fazer em 2010? Quais foram... é, voltando a falar de soluções, quais foram as soluções que o HTML5 trouxe para isso?

SRA. ANA LAURA: Acho que se a gente for dar nome aos bois, acho que talvez o Diego seja mais... tenha mais nomes para dar nessa hora. Mas eu acho que a grande, para mim, a grande virada foi justamente essa, né? Realmente, Diego, a gente trabalhava super em camadas. Era o que é do usuário, eu brincava em sala de aula, né? O que é estrutura e o que é beleza, né? Você falar uma coisa é HTML e outra coisa é CSS. A gente trabalhava superseparado. E era exatamente isso. E aí a gente pensava: o HTML é a minha estrutura e a estrutura vai atender o quê? Ela vai atender ao usuário final. Tem uma parte dessa estrutura que é em prol do device que está vendo? Tinha. Lembrando que a gente não tinha essa miríade de telefone e dispositivos em geral, porque hoje você coloca HTML em qualquer lugar. Não tem limite, né? E a gente, talvez, ficasse pensando nisso, de como vai ser essa estrutura. Então, sim, o HTML segurava uma estrutura que era em função daquele conteúdo, e aquele conteúdo tinha que ser apresentado para o usuário. Aí entrava a beleza. Eu ia lá, colocava o CSS, colocava tudo isso organizado, bonitinho para que fosse acessível para o usuário, para que ele conseguisse manipular isso, e a gente tinha essas preocupações. E aí depois vinha o JavaScript com o comportamento. Isso no nível mais front, né? Fora o que rolava lá no back-end, porque a gente também tinha métrica, tinha um monte de coisa, mas tinha... nossa, não passava nem perto do que é hoje. E uma parte desse back-end ficava na mão do navegador. Então, por exemplo, hoje a gente continua tendo uma certa disparidade, né? Estão bem melhores, os navegadores em geral estão bem estáveis, mas o Chrome sempre renderiza mais, [ininteligível] sempre dá

mais velocidade. Tudo bem que trava mais em alguns momentos, mas ele também tem uma performance muito boa para o que é novo. Então, a gente ainda tem alguns gargalos. Não sei que navegadores eles estão colocando na geladeira em outros dispositivos aí que acessam o HTML, mas acho eu acho que a gente tinha uma coisa mais marcadinha mesmo, e cada um fazia o seu nome, né? Hoje está tudo misturado. O CSS no JavaScript confesso que ainda me dá um certo...

SR. DIEGO EIS: É, dá arrepio.

SR. PEDRO ROGÉRIO: Lembro que a gente defendia ferrenhamente isso. Meu, HTML é HTML, CSS e JavaScript. Era tudo separado. Era um crime você chegar e misturar tudo antigamente. Aí hoje você [interrupção no áudio] é tudo, CSS no meio do JS, HTML junto. Uma fonte que está se perdendo. Não que seja ruim, mas será que... como vai ser daqui para frente? Vai ser sempre assim? Ou daqui cinco anos a gente vai voltar a defender como era antigamente? Não sei. É uma incógnita.

SR. DIEGO EIS: Escrevi um artigo uns anos atrás aí com a minha visão, mas vamos ver. Estou brincando. Eu não sei, é muito difícil responder isso, o que o usuário faz hoje na Web que não fazia em 2010. Eu acho que tem coisas que, na verdade, na verdade, eu acho que o usuário faz a mesma coisa que ele fazia em 2010, só que hoje ele faz melhor. Então, a gente fala aí do HTML, né? A gente esquece que tem o H aí, né? O hipertexto, né? Hipertexto, lá atrás, né? Era porque a gente conseguia misturar várias... como vou dizer? Mídias, né? Vários tipos de conteúdo nesse texto. Imagem, a gente podia colocar ali texto, a gente podia colocar talvez vídeo e etc. Hoje a gente leva isso para um outro nível, né? Então, o usuário hoje, ele... Bom, a gente está aqui, né? Fazendo um streaming via YouTube, estamos conversando aqui. Vocês aí do YouTube não estão vendo, mas a gente está aqui dentro do Zoom, que está integrado com o YouTube fazendo um streaming com centenas de pessoas aí interagindo. Então, antes a gente já fazia isso, né? Web 2.0, né? Não sei se vocês lembram disso. Mas hoje a gente levou para um outro nível, né? Então, eu acho que a gente hoje realmente é hipertexto. A gente consegue misturar hoje diversas mídias num lugar só, de maneiras diferentes, e hoje a gente consegue realmente andar. Muito por causa do HTML5. Hoje, vídeo com HTML5 deu um salto. A gente consegue customizar quase o que a gente quiser ali.

Então, acho que, hoje, acho que essa evolução que teve do código, essa evolução da maneira com que a gente pensa nas tecnologias, né? Ajudou a potencializar exatamente as mesmas atividades que a gente já fazia antes, só que hoje a gente consegue fazer melhor e consegue resolver isso de uma maneira um pouco mais integrada, né? Um pouco mais compatível, um pouco mais acessível do que antes, né?

Então, 2010 até... bom, não, aí acho que aí eu vou apelar, 2010, que já estava resolvido. Ia falar do Gmail, né? A gente usando Ajax e tudo mais, né? Pois é, então. Aquilo mudou bastante coisa para a gente, né? Mas isso é antes de 2010. Então, acho que hoje é mais ou menos isso, né? A gente consegue fazer um pouco mais, mas acho que a gente consegue fazer as mesmas coisas que a gente fazia antes, só que muito melhor, sem tantos entraves e sem tantos problemas de tecnologia. Acho que não tem mais tantas barreiras como antes, né?

SRA. ANA LAURA: Só, cara, é impressionante isso, porque imagina a quantidade de coisa. Hoje a gente usa World, Excel e PowerPoint no navegador, né? Sabe? É impensável.

SR. PEDRO ROGÉRIO: O Photoshop no navegador aí.

SRA. ANA LAURA: Nossa, nem fala. Photoshop, XD, e assim, a gente tem um monte de lançamento chegando aqui agora, eu estou cropando, com inteligência artificial. É, assim, é insano. É absolutamente insano. Eu não sei se a gente faz só mais e melhor, a gente faz coisa 'pra dedéu', cara.

SR. DIEGO EIS: É, é, é.

SRA. ANA LAURA: A gente faz muito. Tudo bem que a gente subiu a régua do que a gente fazia, mas a gente extrapolou, e eu não sei que a gente vai fazer com 5G não, cara. A hora que tiver AD Computing e tudo mais, que o cálculo for feito mais [ininteligível] para armazenar tudo isso, né?

SR. DIEGO EIS: É verdade.

SRA. ANA DUARTE: Bom, então, assim, só para encerrar o painel, antes da gente fazer as perguntas da audiência. Há dez anos atrás (sic) não era possível ter um vídeo nativo na Web, por exemplo. O que o HTML5 resolveu nesses últimos dez anos e o que ainda precisa ser solucionado?

SR. DIEGO EIS: Legal. Essa pergunta é bacana, assim, e acho que está bem atual, porque ontem, realmente, eu estava... saiu um episódio do Dev na Estrada, onde ele estava falando exatamente sobre CSS no JS, né? E, assim, os front-ends de hoje, assim, eles conversam de forma bem fluida sobre isso. Mas eu fico pensando, mas aí é o tio falando, né? Acho que essas libraries, esses frameworks que surgiram hoje, eles vieram para resolver um problema, e esse, para mim, talvez seja a coisa que o HTML ainda não resolveu e que tentou se resolver posteriormente, mas não deu muito certo, né? Vide aí as tecnologias que existem agora. Mas o que todas essas frameworks, essas libraries tentaram resolver foi a componentização. Como a gente consegue componentizar elementos que a gente usa na interface, de forma que eles sejam isolados, os códigos deles sejam isolados. Então, eu tenho ali junto todo o JS que eu preciso, todo o CSS que eu preciso, todo o HTML que eu preciso, a conexão que eu tenho com o servidor para funcionar aquele pedaço da tela, e eu querer controlar aquele pedacinho da tela, não a tela inteira. Então, isso eu não estou falando nem só de desempenho, mas isso melhora muito a experiência do usuário como um todo, né? Então, acho que o HTML, assim, ele resolveu, ele deu muitas ferramentas para a gente conseguir chegar até onde a gente chegou hoje, mas existem problemas maiores ainda que não é só HTML sozinho que vai resolver, né? Não é só uma das tecnologias fundamentais da Web que vai resolver. Acho que são todas e talvez novas tecnologias tenham que surgir para conseguir resolver isso de verdade. Para mim, acho que o maior gargalo aí foi componentização.

A maior coisa, para mim, que o HTML5 resolveu foi realmente a parte da semântica, foi realmente a gente conseguir não só trazer significado para o código, mas também conseguir manipular elementos na tela, melhorar os elementos na tela sem quebrar a semântica. Com significado a gente conseguir customizar esses elementos, vide, por exemplo, a tag de vídeo. A gente conseguir manipular isso sem quebrar tudo ali, sem ter que fazer puxadinho com JavaScript ou fazer mágica com qualquer outra coisa que a gente queira fazer.

Mas acho que, para mim, acho que esses foram os dois grandes pontos, assim. Esse componentização é basicamente o que a gente tenta resolver hoje. A gente põe React para tentar resolver componentização da tela. Só que aí a gente quebra uma série de coisas, né? Então, a gente tem que ficar mexendo com roteamento, a gente tem que ficar colocando CSS no JS, a gente tem que ficar refazendo Dom(F), uma série de outras coisas para tentar resolver esse ponto aí da componentização, né? Acho que isso é legal de ser dito, e é por isso que a gente fala de CSS no JS, a gente fala de código não semântico, a gente fala de uma série de tecnologias amontoadas aí para rodar um simples projeto. Acho que é isso.

SRA. ANA LAURA: Deixa eu só colocar uma coisinha, pimentinha. Cada vez mais eu fico... estou começando a achar que é isso mesmo. A gente precisa voltar a ter foco no usuário, né? Parar de ter foco no sistema, porque eu acho que a hora que a voltar de novo para o usuário, a nossa mira, essas respostas acabam

vindo melhor, sabe? Tipo, eu estou fazendo isso? Sim. Eu tenho que quebrar isso? Sim. Mas isso vai ser benefício para o usuário? Então não faz. Ou então está na hora de pensar outra ferramenta. Ou então está na hora de pensar outro jeito. Então, talvez seja realmente essa a evolução, né? O próximo passo que a gente tem que dar, de voltar a olhar para o usuário.

SR. DIEGO EIS: É, e fazer o uso progressivo da tecnologia, né? É exatamente isso. Às vezes o usuário não precisa de React maravilhoso e etc. Fazer um site de conteúdo com React e etc. Às vezes não precisa. O bom e velho HTML ali, usando o PHP, está ok, já resolve o problema, né? Entrega muito bem o que usuário quer, ajuda o usuário a resolver problema dele, a cumprir com os objetivos, deixa ele melhor, né? Porque ele vai usufruir dos benefícios ali, e a gente conseguir usar as tecnologias corretas para os momentos corretos para resolver os problemas corretos, né? Concordo.

SRA. ANA LAURA: Super concordo.

SR. PEDRO ROGÉRIO: O bom e velho feijão com arroz resolve tudo, né?

SRA. ANA LAURA: Justo.

SRA. ANA DUARTE: Bom, pessoal, a gente já está com tempinho um pouco apertado. Então, a gente vai fazer algumas perguntas da audiência aqui que estão no chat do YouTube. E aí, só para não estourar o tempo dos outros painéis, a gente pede para que vocês façam as respostas mais sucintas, assim, tá?

Então, a primeira pergunta é da Poliana Vitorino: "A tecnologia CSS3 é uma tecnologia que deve incorporar o HTML5?".

SRA. ANA LAURA: CSS é CSS. HTML é HTML. Diego, não sei se você quer, depois, complementar com alguma coisa, mesmo o Pedro, mas, assim, as coisas caminham... estão sempre caminhando juntas, né? Não sei. Não sei como você avalia essa junção. O CSS ainda é a beleza e o HTML ainda é a estrutura do conteúdo. Eles têm que evoluir, têm que andar juntos. Eu não vejo eles sendo uma coisa e a outra. Eles são coisas separadas.

SR. PEDRO ROGÉRIO: Sim, vão caminhar um ao lado do outro, mas no final vão ser, o HTML... camadas de desenvolvimento separadas, né? HTML de um lado e o CSS de outro. Mesma coisa com JavaScript também.

SR. DIEGO EIS: Concordo. Assim, acho que é bom deixar separado, porque elas são coisas que resolvem problemas diferentes. E elas conseguem evoluir de maneira independente. Conseguem cada um ir para o seu lado, mas elas conseguem se manter integradas. O CSS não precisa ser usado só no HTML. Então, a gente pode usar ele para uma série de outras coisas. JavaScript não precisa fazer... ser usado só para manipular CSS e HTML. Pode fazer o API, vide [ininteligível] e tudo mais. Então, a gente consegue evoluir e usar essas tecnologias para coisas totalmente independentes, separadas e para cumprir outros objetivos, né? Então, se a gente incorpora o CSS no HTML, não faz muito sentido. A gente está dando muitas responsabilidades para uma linguagem, para uma tecnologia que foi feita para resolver uma coisa. Então, a gente consegue separar isso e fazer bom uso disso, né?

SRA. ANA DUARTE: Beleza.

Então, seguindo na pergunta que a gente pegou no YouTube, do Sérgio Eduardo Antônio. Quanto a incremento da segurança, quanto menos tecnologias envolvidas no processo de construção de páginas, menos variantes estão envolvidas na análise do código. O que vocês pensam a respeito?

SR. DIEGO EIS: Legal. É, quando a gente fala de segurança é difícil, porque tem o fator usuário aí, que eu acho que é o principal problema, talvez. Acho que a gente está muito maduro com segurança, né? A gente consegue, hoje, ter soluções aí que não deixam faltar nenhuma necessidade, principalmente para o usuário ou para as tecnologias que a gente tem hoje, né? Para o uso que a gente tem hoje e os objetivos que a gente tem hoje. Acho que o problema não está tanto na tecnologia. Acho que o problema realmente está realmente nesse gargalo do usuário que está aprendendo a lidar com a tecnologia hoje.

Então, sim, acho que quanto menos tecnologia, menos linguagens e menos coisas que a gente pendura no sistema, melhor. São, talvez, menos potenciais furos, mas acho que o principal problema ali é a educação do usuário e a forma como ele lida com a tecnologia, né? Acho que é o principal gargalo, talvez.

SRA. ANA DUARTE: Ainda do Sérgio Eduardo Antônio, ainda fazendo referência à segurança contida nas páginas, ele pergunta: Os links deveriam receber incrementos relativos à segurança pelas linguagens, especialmente no HTML5?

SR. DIEGO EIS: Alguém quer responder aí? Eu respondi a outra, hein.

SRA. ANA LAURA: Eu acho difícil responder essas coisas, porque não é muito a minha área de expertise, que eu pendo mais para o design. Mas você tem a segurança na ponta, essa que o Diego falou, que realmente está na mão do usuário, o usuário tem o comportamento inseguro, e você tem a segurança do código que você está oferecendo. Eu não sei onde é mais interessante para alguém roubar os dados. Se é dentro do código ou se é lá no back-end. Então, vai depender do dado que está oferecendo, que você está disponibilizando, porque, às vezes, sei lá, é como se você fosse proteger o seu quarto. Você não protege o seu quarto, você protege a sua casa. Então, sabe? E não é o usuário sentado na escrivaninha, talvez, naquele momento, tá? Pode ser que tenha que proteger o prédio onde ele está, o bairro onde ele está. Então, eu acho que conforme conteúdo, conforme o tipo de ataque, você tem um... vai colocar segurança de um lado ou de outro.

Eu não sei exatamente como seria um HTML seguro, mas um back-end seguro eu acredito que tenha mais força. E, de novo, a mão do usuário, né? Aquele filme de terror "a mão que balança o mouse". É esse cara que tem que ter e que tem que cuidar e ter uma atitude um pouco mais saudável, né? Saber o que ele está... aonde ele está entrando, aonde ele está indo, sabe? Coisas antigas que a gente se preocupava e que deveria continuar se preocupando. Não sei por aí com bluetooth aberto, sabe? Umas coisas assim. Se protege um pouco, sai da Internet, às vezes, quando fecha o computador, sabe? Algumas coisas que são simples e que às vezes a gente não faz, né?

Então, acho que tem níveis, e tem níveis de conteúdo e tem o que você lida, a preocupação de um banco às vezes não é necessariamente a preocupação de um blog, sabe? São coisas diferentes, conteúdos diferentes.

SRA. ANA DUARTE: Ok. O Sérgio Eduardo Antônio de novo, ainda com o tema de segurança. Ele pergunta se... ele fala assim: HTML5 e posteriores poderiam incorporar a criptografia além do protocolo HTTPS para validar os conteúdos interativos utilizados pelos usuários?

SR. DIEGO EIS: Tá. Acho que o Élcio até comentou aqui no nosso chat interno aqui, acho que faz muito sentido. Acho que o problema aí de segurança, talvez, e expandindo um pouco talvez a pergunta, talvez não seja nem das linguagens, etc. Talvez seja exatamente do que tange ali o limite do client side. Exatamente a gente poder entender melhor sobre conexão com o servidor. A gente não fala só de HTTPS, que hoje não é a única maneira de se defender, né? O Élcio até deu ali, falando de cores, falando de

bloqueio mix(F), de content, etc., e tudo mais que a gente já tem ali entre esse limite entre o front-end e conexão com o servidor.

Então, talvez sim. Talvez o browser poderia fazer um trabalho melhor de segurança. Acho que entre o browser e o servidor tem um trabalho que talvez possa ser feito. Eu não sei se quem desenvolve o HTML, o CSS, o JavaScript, o React, o Vue, o que for ali, ele precise se preocupar com problemas, talvez, do código front-end, mas talvez ali em toda essa camada de conexão com o servidor. Acho que, aí sim, aí é um trabalho grande que o front-end tem que fazer para conseguir pensar em segurança, e não é um trabalho sozinho, né? É um trabalho junto com toda a galera de back-end, Dev, risco, segurança, infra e etc.

SRA. ANA DUARTE: Tá. E a Poliana Vitorino, ela pergunta se ela consegue fazer um site HTML5 puro?

SR. PEDRO ROGÉRIO: Consegue, fácil. Acho que é até legal a pessoa não partir para utilizar só frameworks, começar com a base ali, desenvolvendo basicamente em camadas. HTML, CSS, JavaScript, acho que é a base, todo mundo tem que... aconselhável todo mundo começar por essa parte para depois começar a migrar para outras frameworks.

SRA. ANA LAURA: Nossa, não podia concordar mais com o Pedro. É como a gente aprender a ler e escrever. Você tem que aprender o que é um parágrafo, o que que... sabe? Quando você quebra a linha, o que é cada pedacinho de conteúdo que você está colocando. Eu acho que é... inclusive, a gente conversa muito, eu trabalho muito com instituição educacional dentro da Adobe, e a gente fala muito de digital literacy, que é fazer o cara aprender a lidar com o meio digital. Qual é a linguagem do meio digital? Para mim, a resposta é HTML. Você tem que saber o que é cada pedacinho do que você vê, do que está escrito, do que aparece na tela, é um IMG, é um vídeo, é um P, é um BR, é um B, é um I, sabe? Você tem que ter esse código. É o teu abecedário, é o teu vocabulário, e você vai ampliando. Quando a gente é criança, a gente desenha, depois aprende a escrever, aí a gente começa a fazer textinho pequenininho, cartinha para a mamãe, o papai, a tia, sabe? Vai aumentando, carta para os amigos, e aí você escreve as suas próprias histórias, e aí você vai ampliando e vai aumentando a sua bagagem, e cada vez mais você vai querer conversar com mais gente de formas diferentes, e aí vai aprendendo novas ferramentas. Então, o HTML5, para mim, é o teu abecedário, sabe? E é bem básico assim. É o teu abecedário, é a tua ferramentinha primeira, sabe? É aquela mais importante de todas, e aí depois cresce e vai colocando o resto.

SRA. ANA DUARTE: Beleza, pessoal. Eu agradeço pela participação de vocês, encerrando esse primeiro painel para... para o próximo, que Sel vai moderar. A Selma vai moderar um painel a respeito de mudanças no mercado e market shares.

SRA. SELMA DE MORAIS: Obrigada, Ana. Obrigada a todo mundo aí que participou desse debate bem legal. Antes de começar, eu quero me descrever: eu sou morena, eu tenho os cabelos compridos e uso óculos de grau.

Bom, como a Ana falou, eu sou a Selma, trabalho com ela lá no Ceweb, com o Wagner, com o Reinaldo e com mais um monte de gente boa, e eu vou fazer moderação desse próximo painel, que tem como foco apresentar as mudanças do HTML, o que ele trouxe de novo para a Web. Como a tecnologia mudou para os desenvolvedores e para nós, usuários. E nesse painel a gente vai contar com a participação do Clécio Bachini, que é presidente da UXPA de São Paulo e fundador da Soyuz. O Clécio é coautor do livro Open Web Platform. O Clécio também é palestrante em eventos nacionais e internacionais sobre UX, acessibilidade digital e desenvolvimento Web.

Não menos importante, a gente vai contar também com a participação do Thiago Prado, que é analista de TI na Universidade Tecnológica Federal do Paraná, é professor de pós-graduação na área de desenvolvimento Web. O Thiago também é mestre em ciência da computação e tem um MBA em engenharia de software. Ele é membro do GT de acessibilidade na Web e foi empreendedor no segmento de agência digital por mais de 12 anos.

Sejam muito bem-vindos, Clécio e Thiago. Muito obrigado pela participação de vocês hoje, no nosso evento. E eu gostaria de abrir esse painel com uma pergunta sobre o panorama que a gente tinha há dez anos atrás (sic), quando o curso do HTML5 foi realizado pela primeira vez pelo W3C. Como que era o mercado na perspectiva do desenvolvimento Web e como ele mudou? E as novas bibliotecas e frameworks, como elas também impactaram nesse mercado e no desenvolvimento Web?

SR. THIAGO PRADO: Bom dia, Selma, Clécio, bom dia a todos que nos acompanham. Eu vou tentar responder então, essa pergunta tentando contribuir um pouquinho mais com o que os colegas trouxeram no painel anterior.

Então, há dez anos atrás (sic) o mercado de desenvolvimento Web era bem focado em Websites e aplicações Web que, em sua maioria, eram utilizados em desktops. Mas já tínhamos sinais de mudança, né? Nós já tínhamos muitos outros dispositivos com telas de diferentes tamanhos, principalmente os mobiles, tablets, smartphones, e crescia, e surgia o termo Responsive Web Design. Então, principalmente para poder adaptar aqueles sites que eram feitos até então, aquelas aplicações para esses dispositivos, a gente tinha um crescimento, uma mudança de paradigma no desenvolvimento Web.

O Responsive Web Design, ele ganhou força também com mídias cares que foram trazidas pela CSS3, que veio junto com o HTML5. Então, quando a gente fala de HTML5, é comum a gente colocar no mesmo pacote a versão nova da linguagem HTML, que é a linguagem de marcação que substituiu a versão 4, a tecnologia CSS na versão 3, e todos os novos recursos que ela trouxe, e uma série de outras APIs que foram criadas e implementadas nos navegadores para que pudessem ser, então... para que o desenvolvedor tivesse acesso a mais recursos do browser e dispositivo que estaria usando.

Então, no contexto lá de 2010, como o Elcio colocou no chat, a gente já tinha uma evolução em relação aos anos 2000. Então, a gente já estava... Era até era muito mais recursos e possibilidades. Mas a gente ainda tinha algumas dúvidas, né? Alguns pontos que estavam obscuros, né? As aplicações nativas para smartphone, usando linguagem nativa de cada plataforma, ganhavam bastante espaço. E havia dúvida se a tecnologia aberta da Web iria sobreviver ao avanço dessas tecnologias e desses aplicativos.

Em termos de cenário, nós tínhamos, por exemplo, em 2010, o Internet Explorer com 50% de market share, depois, o Firefox tinha 30% e o Safari e o Chrome estavam ali brigando entre os 5 e 10%. O Chrome estava começando naquela época. E cada um dos navegadores traziam diferentes motores de layout e de execução JavaScript. Então, isso era uma dificuldade muito grande para quem desenvolvia na época. Em 2012 o Chrome virou o jogo, passou a ter a dianteira, e hoje, por exemplo, tem 70% de market share no mundo, né? Em termos de browser. O Safari e o Firefox vêm logo atrás e a Microsoft acabou até abandonando o desenvolvimento de um motor de layout e adotou a plataforma do Chrome no navegador Edge.

Existia um outro componente naquela época bastante popular, que era o Flash. Então, ele era muito adotado para a produção de animações, jogos e outras aplicações interativas, porque ele tinha algumas vantagens, como, por exemplo, acesso a câmera e microfone do usuário, possibilidade de desenho

vetorial, reprodução de vídeo full screen, um certo tipo de armazenamento de dados. Então, tudo isso dava uma vantagem e tornava o flash uma boa escolha para aplicações ricas, de conteúdo rico para na Web.

Na mesma linha, a gente tinha algumas bibliotecas JavaScript, como, por exemplo, a [ininteligível], que tentava também otimizar a criação de componentes, como o Diego também colocou no painel anterior, que é algo muito importante para você melhorar a experiência de uso, né? De aplicações na Web. HTML5 e todas essas outras tecnologias que vieram junto com ela, cobriu boa parte desses recursos que as outras bibliotecas, JavaScript e o Flash traziam. Então, tornou-se, de forma aberta, como plataforma aberta, uma alternativa mais viável para a adoção. Por isso, nos anos seguintes, o Flash acabou caindo em desuso, também por questões de mercado. Principalmente numa briga ali da adoção do Flash no iPhone, por exemplo, e outros fatores.

Então, do ponto de vista dos profissionais e das empresas que desenvolviam para Web, a gente tinha alguns desafios técnicos, por exemplo: o que o usuário passaria a adotar? Adotaria mais? O mobile ou o desktop? Em termos de ferramentas de desenvolvimento, a gente devia usar Flash? HTML5? Java para Android? Objective-C para IOS? Eram questões que permeavam aquela época. Então, tinha a necessidade de fazer o site de aplicação funcionar em diferentes tamanhos de tela, navegadores, versões, de navegadores, diferentes plataformas e tecnologias.

Também, os usuários, os clientes, eles queriam mais animações, queriam ter acesso a recurso de dispositivos. Então, quem trabalhava com o desenvolvimento Web ficava numa encruzilhada. Ele daria força a plataformas fechadas ou ele usaria a plataforma aberta? A aposta do... e no curso foi... que acabaram, realmente, prevalecendo em muitas situações, né? E a gente teve aí um grande crescimento de novos recursos, de novas aplicações que usam a base. E no caso da base, é importante lembrar que HTML, CSS e JavaScript continua sendo a base de todas as bibliotecas que existem hoje, né? Por mais que nesse caminho ali de 2010 para hoje tenha surgido Bootstrap, Angular, React, etc., tudo isso está sob a base de HTML, CSS e JavaScript, tá?

Então, apenas para encerrar e passar a palavra para o Clécio, as equipes de desenvolvimento, para ilustrar, elas começavam, lá nos anos 2000 a gente tinha de duas a três pessoas, uma especialista, que chamava de server side, outra em programação HTML, CSS, que fazia o recorte do layout, e era mais ou menos isso. E um designer, né? Que desenhava, o Web Design, que desenhava no Photoshop o layout do site ou da publicação. Nos anos 2010 a gente começou... a gente já tinha uma grande quantidade de especialização na área de experiência do usuário. Arquitetura de informação, design de interação, UX, etc., e começava também a fazer parte das equipes os profissionais de social mídia, porque as mídias sociais, as plataformas de conteúdo social passaram a ter uma grande importância na publicação de conteúdo, né? Hoje, em algumas equipes, esses são, inclusive, maioria.

E o programador de HTML, CSS e JavaScript, se assim poderia dizer, passou lá em 2010 a se chamar front-end, né? Por conta de reunir esses conjuntos de tecnologias, e essas tecnologias foram se aprimorando e crescendo, que hoje a gente chama até de engenheiro de front-end, dada a complexidade do ecossistema que a gente tem ali, de tantas opções para você desenvolver e aprimorar uma aplicação. Então, em termos de perspectiva, lá em 2010, o HTML5 trouxe, por exemplo, get user media, uma API para usar microfone do usuário, geolocalização, armazenamento de dados, novos elementos de formulário, elementos de semântica que permitiam os buscadores ter acesso mais rapidamente ao conteúdo importante de uma página, controle de mídia, de áudio e vídeo, permitia embutir no documento outros

formatos, como o SVG, FNL(F) e outras linguagens, e todo esse conjunto possibilitou o surgimento das Progressive Web Applications, que também o Diego citou no painel anterior, que facilitava com que você desenvolvesse... facilita com que você possa desenvolver uma aplicação que gradativamente você vai incrementando em recursos e a experiência do usuário.

SR. CLÉCIO BACHINI: Bom dia.

SRA. SELMA DE MORAIS: Bom dia.

SR. CLÉCIO BACHINI: Bom dia, Selminha. Bom dia, Thiago. Bom dia a todos os amigos aqui do... que estão aqui no chat. Todos os amigos que a gente está revendo aqui no Zoom também, muito bom estar com todos vocês aqui.

Eu vou complementar o Thiago, eu vou fazer um breve, breve histórico da minha experiência com o HTML. Eu comecei no HTML na versão 3.2 em 97. Então, para você ter ideia do que era naquela época: era boa prática você codificar tudo em maiúscula, por exemplo. Então, tinha umas coisas muito bizarras ali, era bem incipiente a tecnologia, mas, para mim, foi uma tecnologia que sempre me encantou, porque diferente do que eu fazia até aquela época, até aquele momento, todas as tecnologias maduras, elas eram pagas e precisavam de alguma licença para serem aplicadas, e o HTML e a Web veio como uma grande tecnologia aberta e gratuita, e isso era uma coisa surpreendente na época, e que me encantou, e que fez eu me tornar um grande defensor dessa tecnologia, né? Até que nos anos 2000, pelas limitações mesmo de desenvolvimento, tecnologias paralelas para a Web começaram a desenvolver, como o Flash, e o HTML acabou sendo colocado um pouco de lado, e veio o Flash, e você usava HTML como um container de Flash, assim, no desktop, o que era uma coisa terrível, que me doía muito aquilo. Você desenvolvia aplicações que eram muito inacessíveis, muito complexas, mas que eram aplicações multimídia e que o HTML não permitia fazer até aquele momento.

E daí, num certo momento, as pessoas começaram a discutir que o HTML deveria incorporar aquilo, né? E foi um momento muito bonito, né? Foi o momento do Tableless, né? Foi o momento ali que tinha o Diego e o Élcio levantado essa bola dos padrões Web, e teve um movimento muito forte em cima disso, que foi ali no final dos anos 2000, e que culminou com o primeiro Web BR, que foi ali em 2009, logo em seguida veio o pessoal que estava discutindo HTML5, que foi maravilhoso aquilo. E, logo em seguida, em 2010, teve esse grande curso de HTML5, que mudou a minha vida completamente. Foi uma mudança de paradigma. Conhecer as pessoas, estar ombro a ombro lutando com pessoas por padrões. E foi incrível aquele momento. Foi aquele momento que deu realmente um lampejo de que a coisa poderia virar o jogo e a gente poderia ter uma grande tecnologia aberta, semântica e acessível, para que todo mundo pudesse usar, né? E foi incrível.

E o que aconteceu logo em seguida, foi que daquilo ali começaram a surgir movimentos e eventos de front-end, de HTML, levando a palavra do HTML5, né? Eu sempre brincava com o Reinaldo assim, que a gente ia evangelizar a palavra do HTML5 nos lugares. Surgiu um movimento muito bacana. No ano seguinte a isso, teve uma forte resistência, óbvio que teve uma forte resistência. As pessoas tinham empregado muito dinheiro, muito tempo aprendendo Flash. Então, elas não queriam perder esse conhecimento. Então, isso daí foi... teve uma grande resistência. Essa resistência foi sendo vencida e o HTML foi se tornando maduro, o HTML foi se tornando maduro e o mercado começou a embarcar nessa ideia.

Teve o mobile first, então muita coisa responsiva e Web, mobile começou a surgir, o que foi muito interessante. E daí, logo em seguida, começou os movimentos de redes sociais, redes sociais se tornando muito, muito importante. E a gente começou a ver, assim, a gente teve, acho que até metade de 2004, 2005, a gente teve uma grande abertura, uma grande abertura semântica, códigos semanticamente bem feitos e uma Web muito rica, né? Só que, obviamente, complexa de se administrar. E por isso acabaram surgindo vários frameworks naquele momento, né? E o que aconteceu a partir dali foi um fechamento. Eu vejo que foi um novo fechamento da Web em cima dos frameworks, né? Que eu acredito... eu entendo a razão dos frameworks, utilizo no meu dia a dia. O que eu procuro fazer, falando até complementando um pouco o que o Diego falou na fala anterior, é que, assim, eu ainda acredito que semântica é importante. Eu ainda acredito que um código que gere... mesmo gerado, ele tem que ser gerado de uma maneira com qualidade, sabe? Então, eu sempre procuro escolher ferramentas que gerem isso na ponta, porque isso vai ser melhor para a acessibilidade, isso vai ser melhor para a CO, isso vai ser melhor para a experiência, sabe?

A gente... Acho que essa desconexão entre o código gerado, o front-end, e os princípios, o bê-á-bá do HTML, pode gerar coisas muito perigosas e que você não tenha controle, sabe? Que você não tenha acesso, e isso eu acho um pouco perigoso. Esse momento, de alguns anos para cá, em que você voltou a ter essa questão de ser tudo gerado, todo o Mark Up, todo o código de marcação ser gerado automaticamente, isso gera um pouco de falta de controle. Um pouco de falta de... A gente acaba perdendo a democratização da linguagem e do HTML, mas, lembrando, sou um romântico, estou falando sobre o romantismo da coisa. E eu procuro, nas soluções que eu desenvolvo com a equipe e tudo mais, tentar focar nos princípios ainda de Web Standard lá de trás, né?

Mas é isso. A gente está num momento que é aquele momento das discussões sobre coisas fechadas, né? Sobre a Web fechada, sobre redes sociais, sobre os códigos gerados que também são inacessíveis, né? E eu acredito que é essa uma discussão do momento, principalmente pela questão do futuro da Web.

SRA. SELMA DE MORAIS: Legal. Tenho mais uma pergunta para vocês, meninos. O que vocês acham que foram as principais mudanças de comportamento nesses últimos dez anos, né? Desde a realização lá do primeiro de HTML. O que mudou, principalmente relacionado a comportamento, de lá para cá?

SR. THIAGO PRADO: Bom, eu acho que em 2010, né? Naquele cenário, todo mundo queria estar presente na Web, né? Todo mundo queria ou precisava ter um site na Internet, todas as empresas precisavam marcar presença ali, e isso era um ponto, uma situação. Os aplicativos também começaram a surgir, né? E aí você tinha que estar presente nos dois lugares, né? Na Internet, digamos assim, na Web e no celular do usuário, em forma de aplicativo. Mas, fora isso, existiu uma mudança também de comportamento, né?

Então, em 2010 a gente tinha, se eu não me engano, pela pesquisa do TIC Domicílios, cerca de um terço dos domicílios com acesso à Internet. E de 2010 para cá eu acho que dobrou esse número, né? Para dois terços ou quase três quartos, aí. Então, a gente teve um maior acesso. A Internet se popularizou bastante. Muita parte desse acesso também veio no mobile, né? Que acabou tendo mais uso do que os computadores desktop e notebooks. Então, o comportamento do usuário mudou. Ele passou a acessar Internet a qualquer momento, né? De qualquer lugar, né? Ele passou a consumir informação.

Uma coisa interessante, que a gente pode notar também, é que em meados de 2010, quando o HTML5 trouxe novos recursos semânticos, o comportamento do usuário ainda era buscar... fazer uma busca no Google, e a partir dali chegar no site ou recurso que ele desejava. Esses recursos semânticos, trazidos não só pelo HTML5, mas por um outro conjunto de ferramentas, fez com que o Google modificasse a sua

ferramenta de busca, trazendo um resultado mais preciso para o usuário. Às vezes até já trazendo a resposta da pergunta que trazia, né? Então, uma série de tecnologias de Web semântica permitiram isso. Então, o comportamento do usuário ainda era esse. Fazer uma busca por um conteúdo e acessar aquele conteúdo no site que disponibilizava, né? E esse comportamento foi mudando, foi mudando.

Como o Clécio citou, hoje a gente já tem as mídias sociais, as redes sociais que oferecem o conteúdo para o usuário e tentam prender o usuário dentro daquela rede, para que ele não saia para buscar informação. Algumas empresas já não necessariamente querem ter um site ou um espaço reservado de publicação dos seus produtos e serviços, mas apenas oferecê-los dentro de uma rede social, o que traz para elas também um novo desafio, porque as redes sociais, as mídias se multiplicam, né? Cada vez surge uma diferente. Uma plataforma diferente. Então, a gente teve um pouco dessa mudança. E também o conteúdo, né? A gente tem, hoje, quando você faz uma busca no celular, você nem sempre acessa a página, né? Com o conteúdo HTML. Você acessa o conteúdo... um subconjunto do HTML num conteúdo que foi cacheado, né? Pelos servidores do Google, por exemplo, né? Então, eu acho que houve uma série de mudanças de comportamento, tanto do usuário, no uso da Internet, como também dos serviços que passaram a ser oferecidos na Web. Então, as ferramentas de busca, as redes sociais, as próprias empresas na forma de oferecer seus produtos, né? Agora, com a pandemia também, a gente teve uma outra mudança de comportamento das empresas. Precisava chegar mais rapidamente aos usuários onde quer que eles estivessem, aos clientes, né? Então, acho que toda essa mudança de comportamento ocorreu ali nos últimos dez anos e impactou o uso e a escolha das tecnologias de desenvolvimento.

SR. CLÉCIO BACHINI: É, a gente teve... eu acho que a principal mudança nesse tempo foi a migração para o mobile, para o comportamento do smartphone. A gente viu que lá em 2010 a gente praticamente não tinha isso, a gente tinha lá o iPhone, tinha o Android bem incipiente, mas não era um uso amplo do smartphone. O que aconteceu também, assim, com a questão das mídias sociais, houve um movimento das pessoas começarem a confundir redes fechadas com redes abertas. A gente... as pessoas começaram a entender que a Web ou a Internet, mais amplamente, era o que se via nas redes sociais. Então, empresas, por exemplo, deixaram de ter suas páginas por um tempo. Agora tem um movimento de volta, que eles entenderam, mas, por um tempo, empresas deixaram de ter suas páginas na Web aberta para ter páginas só no Facebook, por exemplo, né? Algo que atualmente já perceberam que justamente pela questão de CO e semântica, as pessoas não vão no Facebook buscar a sua empresa direto. Elas vão no Google, ou vão num mecanismo de busca buscar um produto ou serviço para chegar à sua empresa. E aí a gente vê como continua sendo importante a questão semântica, né? Porque isso que foi colocado com o HTML5, da implementação no código, de você explicar o que você está fazendo ali, colocar os seus dados e metadados, codificar o seu código ali, isso é importantíssimo para que os mecanismos de busca amplamente consigam entender a sua página, indexar a sua página e que você consiga atingir o seu objetivo como cliente.

Então, a gente viu alguns comportamentos que foi meio bumerangue, né? Pessoa saiu, foi para a rede social, depois ela percebeu que não conseguiu atingir o que ela esperava e teve o movimento de retorno à Web. Mas é muito engraçado como a gente evoluiu pouco na questão de explicar para as pessoas a importância da semântica nesse sentido, sabe? Até... esse ano menos, mas há uns dois, três anos, eu vi muita, muita página sendo gerada por framework de JavaScript em que o código todo era gerado só no momento da renderização da página, não tinha nenhum HTML por trás disso. Isso causava que simplesmente a página era pouco encontrada, e outra série de problemas, de acessibilidade nem se fala. Uma série de problemas. Assim, coisas que são facilmente resolvidas por uma boa estruturação, por uma boa arquitetura de informação e que, por causa do movimento de simplesmente pensar na geração de

código automático, você tem que ter uma série de subterfúgios para conseguir isso, e penduricalhos que você coloca ali no seu código ou no que está sendo desenvolvido, coisas que estão muito maduras no HTML e nos padrões Web hoje, né?

E uma coisa que é muito importante, e que eu acho positivo, que é o foco na experiência, e que eu acho que foi movido também... na Web isso foi movido pelo HTML5, de você poder oferecer uma melhor experiência de usuário para o seu cliente. Coisa que todo o arcabouço do HTML permite hoje de uma maneira muito madura, todas as APIs que estão ali de volta, as experiências de vídeo, a questão de você poder colocar legenda, de você sincronizar eventos na sua página, com... para criar uma experiência rica. Isso é uma coisa que foi um movimento muito positivo. Esse foco na experiência eu acho que é uma das coisas que mais enriqueceram a Web nesses dez anos.

SRA. SELMA DE MORAIS: Legal. Muita coisa mudou, né? Desde o primeiro curso realizado há dez anos, né? Eu acho que o HTML, ele se consolidou como um padrão universal, confiável, interoperável, né? Vejo que o W3C no Brasil também teve um papel bem, bem importante nessa questão do fomento e adoção dos padrões aqui no Brasil. E eu acho que o nosso desafio agora é pensar, nos unir e pensar na Web que a gente quer daqui para frente, né? Eu acho que esse é o próximo ponto que a gente precisa nos unir como comunidade novamente e pensar como que a gente pode melhorar, enfim, o que a gente quer.

Meninos. Eu vou passar agora as perguntas, tá? Do pessoal que está lá no YouTube. Só um minutinho aqui. O Danilo Camargo Bueno, ele pergunta: As APIs para Web complements, custom elements, showdown, HTML templates resolveram bem o problema de componentização hoje ou ainda não?

SR. THIAGO PRADO: Olá, Danilo, obrigado pela sua pergunta. Então, Danilo, essas APIs para Web complements, né? Elas... o objetivo delas é tentar resolver o problema de componentização. Mas como eu até coloquei no chat ali para você, não há uma implementação completa ainda dessas APIs pelos navegadores atuais, né? Além disso, alguns ainda estão em estágio final da especificação. Não é ainda uma recomendação. Entretanto, essas outras bibliotecas que a gente tem, React, Vue JS, elas usam, por trás dos panos, alguns desses recursos, ou algumas dessas lógicas das custom elements, showdown(F), templates para poder funcionar de forma adequada ou próxima ao desejado.

Então, eu diria para você que, em termos de padronização de standards, estamos caminhando para que um dia a gente chegue lá, mas ainda, se você pegar ao pé da letra só especificação do W3C para usar no browser, você ainda vai ter uma certa dificuldade, algumas coisas não vão funcionar, tá? Mas isso é normal, é natural. Assim como eu citei mais cedo, por exemplo, o [ininteligível] trouxe o care selector, que é embutido nele, na biblioteca, que mais tarde foi incorporado ao API que a gente usa hoje nos browsers. Então, uma série de outros recursos, eles acabam surgindo, às vezes, primeiro, sendo implementados em bibliotecas, e depois de um tempo a gente consegue... o W3C consegue padronizar e incorporar.

Acho que o caminho é por aí mesmo. Essas APIs devem se tornar mais sólidas, devem até facilitar, depois, que essas bibliotecas de componentes tenham um certo padrão de desenvolvimento, né? Para que também você consiga até interoperar, de repente, lá na frente no uso de uma ou de... no recurso de um ou de outro. Pelo menos essa é a minha visão para esse recurso.

SRA. SELMA DE MORAIS: Clécio, quer complementar alguma coisa?

SR. CLÉCIO BACHINI: Não, já falou tudo. Tem que complementar nada, não.

SRA. SELMA DE MORAIS: Agora a gente tem mais uma pergunta, que é do Sérgio Eduardo Antônio. Ele fala: Deficientes intelectuais fazem grande uso da Internet. Visualizam a possibilidade de incremento de parental control diretamente na configuração de APIs? O que vocês acham sobre isso?

SR. CLÉCIO BACHINI: Então, eu não sei se é questão de controles dos pais sobre o conteúdo estaria na camada da Web desenvolvimento, tá? Assim, os browsers já têm isso de alguma forma implementado, porque eu acho que é uma questão mais ampla do que a aplicação em si, tá? Talvez isso possa ser implementado como ferramental, mas sinceramente, eu não sei se isso é algo que necessariamente tem que estar dentro das bibliotecas, tá?

SRA. SELMA DE MORAIS: Beleza. A gente tem mais uma última pergunta, que eu acho que vai da interpretação de vocês, que o Jairo [ininteligível] fala: WordPress ou HTML puro?

SR. CLÉCIO BACHINI: Cara, então, olha só, eu sou usuário de WordPress e WordPress gera HTML. Como todas essas ferramentas geram, assim. E, de acordo como você programa, ele vai gerar HTML semântico, sabe? A questão é que você usa as ferramentas de CMS para fazer uma geração de código para você e controles que estão numa outra camada, tá? O HTML puro, você pode fazer tudo página a página, até fazer o seu protótipo em HTML puro para fazer os testes, né? Mas é impossível, impossível pode ser muito forte, mas é muito difícil de você gerenciar amplamente um site muito grande, um portal muito grande ou uma aplicação muito grande sem que você tenha uma ferramenta por trás para fazer esses controles, né? A gente está falando de camadas um pouco diferentes, entendeu?

Então, assim, o que você vai escolher para gerar seu o código, o que é importante que você saiba é que você entenda o que está sendo gerado, interprete e saiba a qualidade do que está sendo gerado e como isso vai ser interpretado para o seu objetivo final. E qual experiência que você está entregando para o seu cliente, para seu usuário, que experiência que você está entregando para seu usuário. A ferramenta, cara, ela tanto faz. Se você ficar preso à ferramenta que você vai usar, o foco é outro. O foco é qualidade e experiência. Ferramenta vai servir simplesmente para você... é uma escolha pessoal.

SR. THIAGO PRADO DE CAMPOS: Posso só complementar a fala do Clécio? Concordo com tudo que ele falou, né? E só para fazer uma analogia, depende da situação, né, Clécio? Por exemplo, você quer ir na padaria e ela fica na esquina da sua casa. Você vai a pé de chinelo ou você vai de carro? Ou vai pegar o ônibus para ir até a padaria? Então, depende da situação. Você vai fazer um hot site, uma página simples, página de apresentação, o bom e velho HTML, né? Simplesmente com um pouquinho de CSS e talvez um JavaScript resolve o problema. Agora, você precisa agilizar a publicação do conteúdo por outras pessoas, né? Você, então, pode recorrer ali, por exemplo, ao WordPress.

Só lembrando, como o Clécio falou, é bom você conhecer como essa ferramenta gera o conteúdo final que é entregue para o browser e também ficar de olho em questões de segurança, de atualização, né? Porque, no caso do WordPress, principalmente, por ser muito usado na web em geral, ele é alvo de diversos ataques, né? Para tentar tirar do ar, para tentar prejudicar o conteúdo de uma página que usa essa plataforma. Então, quando você usa uma ferramenta de terceiro, você tem que ficar atento a essas situações.

Mas a resposta é: depende, né? Então, você, quando você vai usar, escolher ferramentas, você escolhe com base no uso, né? E na mesma forma é o uso de HTML puro ou o WordPress para o site ou para a aplicação que você quer ofertar.

SRA. SELMA DE MORAIS: Legal. Meninos, muito obrigada a vocês. É sempre muito bom, sabe, trabalhar com vocês. E eu agradeço muito pela participação. Foi um excelente debate.

E agora eu queria chamar o Reinaldo Ferraz, que trabalha com a gente lá no Ceweb, e é especialista em desenvolvimento web do W3C e do Ceweb, para fazer uma homenagem para o Prof. Everaldo Bechara. Reij, é com você.

SR. REINALDO FERRAZ: Obrigado, Sel. Bom dia a todos, né? Primeiramente, também, mais uma vez, agradecer a todos pela participação nesse encontro, né?

Eu só queria lembrar um ponto rápido desse encontro. Que esse encontro, ele surgiu de uma ideia de celebrar esses dez anos do curso de HTML5 do W3C Brasil, porque ele foi um curso para gente poder capacitar multiplicadores, né? Então, quando a gente acompanha a carreira de praticamente todos que saíram desse curso, né? Muitos acabaram seguindo a carreira no HTML, outros buscaram outras frentes. Mas acho que é importante o que eles trouxeram para comunidade e o que eles puderam contribuir para a evolução dessa comunidade e desse padrão.

E é por isso que eu queria fazer uma... lembrar um pouco do Prof. Everaldo Bechara, né? O Prof. Everaldo Bechara, ele faleceu em 2013, mas ele foi um grande incentivador dos padrões web, né? Ele foi fundador do ILearn, né? E foi um dos primeiros filiados ao W3C internacional. Foi ele que fez a primeira tradução do documento de acessibilidade na web, né? As WCAG 2.0, em 2009. E essa tradução dele foi o que serviu de base para documentação do... para a primeira tradução autorizada que a gente tem do W3C, que é o WCAG 2.0, né? E ele também participou desde a primeira Web.br, falando sobre o ensino de padrões web, e participou também das iniciativas de acessibilidade que a gente tem feito. Então, essa daqui é uma breve lembrança do Prof. Everaldo Bechara, que é uma pessoa que participou muito com a comunidade, e eu queria deixar registrado esse momento. E como ele teve um envolvimento muito grande, especialmente em acessibilidade, agora eu já queria passar a palavra para a Amanda Marques, né? Que também trabalha aqui no Ceweb, para a gente começar o próximo painel, que é sobre o HTML5 e acessibilidade. Amanda, é com você.

SRA. AMANDA MARQUES: Oi, gente, bom dia. Antes de começar, eu queria me apresentar. Eu sou Amanda, eu trabalho no Ceweb.br e no W3C Brasil. Vou me descrever rapidinho, eu sou morena, tenho os cabelos na altura do peito, bem vermelhos, uso aparelho e uso óculos de grau transparente.

Eu vou fazer uma apresentação rápida dos dois painelistas que vão participar com a gente agora. A gente vai falar sobre acessibilidade e HTML5, qual a importância de construir um código bem estruturado e semântico, com recursos de acessibilidade desde o início do projeto. Para esse painel, eu vou chamar o Cesar Augusto Cusin. Ele é técnico em processamento de dados, graduado em Letras, mestre em ciência da computação e doutor em ciência da informação. Ele também é professor universitário da Unifap e da [ininteligível]. Ele é membro do GT de acessibilidade da W3C Brasil e atua como pesquisador de acessibilidade na web.

O outro painalista é o Laercio Sant'Anna, que é analista de sistemas e administrador de empresas. Ele trabalha desde 1988 na Prodam, sendo responsável pelas questões ligadas à acessibilidade. Oferece suporte a produtos específicos para pessoas com deficiência e acessibilidade à Internet e Intranet da Prodam e na Prefeitura de São Paulo. Ele também é membro da Comissão Permanente de Acessibilidade do município de São Paulo, foi participante das comissões da ABNT para criação das normas de acessibilidade para Internet, recursos de acessibilidade na televisão e dos caixas automáticos de bancos.

Também foi membro da Comissão De Unificação do Braille Para Informática nos Países de Língua Portuguesa.

Só para deixar combinado aqui, tanto com o Cesar quanto com o Laércio e também com vocês que estão aí no YouTube, todas as intervenções que eu tiver que fazer no painel, eu vou fazer com áudio aberto, vai ser por áudio e não vai ser por texto, assim como nos outros painéis foram. E, por favor, não se esqueçam de mandar perguntas de vocês no chat, que, no final, eles vão responder. Agora eu passo a palavra para o Cesar. Obrigada.

SR. CESAR CUSIN: Bom dia, Amanda. Bom dia a todos. Bom dia também aos PCDs, a comunidade surda, a comunidade cega. E eu queria, para a comunidade surda, que eu participo mais ativamente, deixar aqui o meu sinal. O meu sinal é uma letra C colocada ao lado esquerdo do meu rosto aqui, e fazendo um movimento com a mão, mostrando bem a letra C de Cesar. Foi um sinal que eu ganhei da comunidade surda, da qual eu faço parte na comunidade surda de Juazeiro do Norte, do Ceará.

Bom, esse painel sobre HTML5 e acessibilidade, ele se justifica de várias formas. Mas uma delas, e talvez a mais importante, é pelo censo do IBGE de 2010, que nós temos aí o número de 24,5% da população que é PCD. Então, nós temos números significativos. E mesmo que não fossem esses números, é uma preocupação que o HTML5 vem tendo, é uma bandeira minha de pesquisa desde mestrado e doutorado. E então esse painel é extremamente importante. E, só finalizando a minha apresentação, eu vou seguir os padrões que todo mundo fez, eu também sou moreno claro, tenho cabelos e olhos castanhos, pouco cabelo, para ser bem honesto, uso óculos de grau e a minha esposa me acha muito bonito, tá? É só para vocês saberem.

Bom, vamos lá. Fiquei muito feliz aqui com o convite, né? Acabei reencontrando amigos aqui. Eu lembro que quando eu fiz esse curso de HTML5 em 2010, ele foi ministrado pelo Diego e pelo Elcio, pessoas das quais eu tenho maior carinho. Não me esqueço deles. A gente se fala muito pouco, mas são pessoas que eu tenho o maior respeito. Me lembro do Tableless, como todo mundo falou. Adorei as falas do primeiro painel, do Pedro e da Ana, falando do bloco de notas, que tem que aprender mesmo [ininteligível], muito legal. O Thiago e o Clécio falando da importância da semântica, que tem tudo a ver aqui com acessibilidade. E a Selma, moderadora, fez uma fala ali, um momento, importantíssima, dizendo da importância do W3C Escritório Brasil para a nossa comunidade. O boom que isso deu, né? Então, alavancou demais. E o Reinaldo foi muito feliz em dizer que capacitou multiplicadores. Reinaldo, foi isso mesmo. Foi aquele curso que eu, eu já professor, concluindo meu doutorado na época, em 2010, fui multiplicador para os meus alunos. Eu tive um aluno que teve um projeto aprovado, também foi nesse curso de HTML5. Então, ele foi um divisor de águas para muita gente. Então, isso foi muito legal.

Mas vamos lá. Falando um pouquinho de acessibilidade dentro do HTML5, eu preparei aqui um histórico do que aconteceu nesses últimos anos. E obviamente que eu vou focar em acessibilidade. Então, vamos fazer um resgate rápido aqui. Se a gente pegar o HTML 4.01, que era uma recomendação de 99, o que tinha, o que falava um pouquinho de acessibilidade? A gente tinha, para começo, né? Algumas complicações, nós tínhamos três DTDs. Ou a gente fazia HTML strict ou transicional, frameset. Quem se lembra disso, e para usar os softwares leitores de tela era um inferno. Eu só não vou falar muito de leitores de tela, porque senão eu vou dar spoiler do meu colega Laercio, que vai falar após a minha aí, mas ele vai falar dessas dificuldades. Mas de acessibilidade que a gente tinha na época, que mais chamava atenção era os atributos titles, summary, longdesc. Era o que dava para ser feito e a gente lutava com isso, com as armas que a gente tinha. Depois veio o XHTML 1.0, que foi recomendação em 2000, tivemos o XHTML 2.0

em maio de 2005, que, na verdade, era só um working draft na época. Mas o que que tinha ali? Eles eram compatíveis com XML, facilmente visualizados, editados, validados, alguns atributos class e roll, nos elementos span e DIV, em conjunto com folhas de estilo, mas também a gente sofria muito ainda com acessibilidade.

Aí veio o curso em 2010, e o HTML5 acabou virando recomendação em 2014. Mas aí a gente viu aquele DTD que era gigante, três possibilidades em HTML 4.01, virando doctype HTML, ponto final, e resolvia o negócio. E veio forte, como bem dito lá pelo Elcio... Perdão, pelo Diego Eis e pelo Clécio, que eles falaram, era a semântica, era o forte do negócio, né? Durante o curso, a gente falou muito disso em 2010. As partes de seções, que você organizava o documento, né? O article, section, reader, [ininteligível]. Os próprios grupos de conteúdo, as tabelas, foi muito mais organizado, os formulários. Pô, o placeholder, que legal, na época, ajudava bastante. E, na verdade, foi um dos documentos que mais fortemente indicava o uso do WCAG 2.0 na época, que, na verdade, anda junto, né? Você ter um guia de disponibilização de conteúdo web acessível com o HTML5, junto com semântica e todas aquelas tags novas que ajudavam a tornar o mais compreensível, né? Para um software leitor de tela, por exemplo.

Depois veio, em 2017, uma recomendação do HTML5.1, que continuava reforçando a semântica e falava da criação de script, de APIs mais acessíveis, tanto para documentos estáticos ou aplicativos dinâmicos, então, veio ajudar mais um pouco. Em 2017, uma outra recomendação, que era do HTML5.2, que dava algumas definições, assim, o que eu estou pontuando, que dava algumas definições mais claras para os agentes do usuário, como tratar tudo isso e inclusive melhorar a interoperabilidade. Isso foi muito interessante. E nós estamos aí às voltas, desde 2018, um working draft do HTML5.3. Que está, por exemplo, dentre as outras discussões, está verificando o uso do atributo autocomplete, que não é permitido em formulários pelo input type hidden. Então, está uma discussão nisso, o pessoal está testando, e é para isso que serve um working draft, obviamente, né? E reforçando o uso do [ininteligível], como o Diego Eis já falou dentro do primeiro painel. Então, é óbvio que do HTML 4.01 para o 5.3, houve muitas mudanças, eu só estou destacando, e por questões até de limitações de tempo, os tópicos principais, o que chamou mais atenção dentro da acessibilidade em todo esse tempo.

E só para fechar a minha fala, com relação ao que a Selma falou e que o Reinaldo disse também, essa questão de multiplicadores, ou algumas perguntas dos painéis anteriores que disse: O que mudou de 2010 para cá? Meus amigos, se você pegar daquela nossa época em que a gente caçava - como o Pedro bem disse - revistas, ia na banca e corria atrás de um post novo do Diego e do Elcio no Tableless, depois desse evento, com as redes sociais e tudo mais, olha quanta coisa boa saiu. De lá, por exemplo, desse curso, saíram parcerias para cursos de pós-graduação com o Thiago da UTFPR, eu dentro da Unifap, fazendo parcerias de pós em desenvolvimento web. O próprio livro que nós escrevemos, eu e o Clécio, que já falou aqui para vocês, o Fábio, que vem depois da gente, o Fábio Flatschart. Pesquisas. Quantos orientandos eu tive de acessibilidade, quanta produção científica eu tive, quantos alunos eu capacitei voltados para acessibilidade. E isso me orgulha muito, né? Por isso que eu digo que esse curso foi um divisor de águas. Nós tivemos muitas coisas boas. Se vocês verem aí o decálogo da web. Vou citar três coisas só: Web para Todos, simples assim; Web organizada em padrões; e uma web acessível, está no decálogo. Há quanto tempo isso aí está lá? Então, basta a gente olhar para o que a gente tem. Sem falar também nas cartilhas de acessibilidade, os quatro fascículos, que estão numa ordem muito coerente, que eu recomendo para todo mundo. O fascículo 1, o que é acessibilidade. O fascículo 2, a legislação sobre isso, pertinente. O fascículo 4, como que eu torno um conteúdo acessível. Perdão, o 3, qual é o público alvo, que eu pulei, e o 4, como tornar esse conteúdo acessível.

Então, muita coisa foi produzida. Então, Reinaldo, se a preocupação era criar multiplicadores, durma tranquilo, isso aconteceu de fato. O GT de acessibilidade, fantástico, com trabalhos incríveis de traduções, prêmios de acessibilidade, as próprias normas que agora, inclusive, no GT, nós estamos trabalhando no Comitê Brasileiro de Acessibilidade, né? O CD 040, junto com a proposta 004, acessibilidade para inclusão digital. Então, tem muita coisa acontecendo para acessibilidade. E esse boom eu digo que foi graças à toda Internet, à forma como ela está sendo feita, à facilidade que a gente tem de se encontrar virtualmente e das coisas acontecerem mais rapidamente. Não vai ficar aquela história, aquele gap de um HTML 4.01 de 99 para um HTML5 de 2014. Esse intervalo não tem mais, haja vista que a gente já está no HTML5.3. Muita coisa boa para acessibilidade, muita coisa acontecendo rápido e isso é bom para inclusão.

Amanda, obrigadão. Estou aqui para as perguntas.

SRA. AMANDA MARQUES: Obrigada, Cesar, pela fala. Eu vou passar agora para o Laercio falar.

SR. LAERCIO SANT'ANNA: Olá, bom dia a todos. Me descrevendo, né? Eu sou um senhor branco, de cabelos grisalhos curtos, sou robusto e estou num ambiente que é um pequeno escritório. Ao meu lado dá para se perceber alguns móveis e alguns instrumentos musicais logo atrás de mim.

Bom, gente, falando de acessibilidade, eu gostaria de trazer um pouco do histórico da questão, principalmente dos leitores de tela, como já foi comentado aqui, né? Os leitores de tela que se tornam um grande desafio para a questão da acessibilidade. Porque nós precisamos transformar aquilo que está na tela em algo compreensível para um público que não está vendo o que tem na tela. Então, para uma pessoa leiga que esteja nos acompanhando nesse momento, leitor de telas nada mais é do que um software que interage com o que está aparecendo na tela e transforma isso em áudio, através de um sintetizador de voz ou de um dispositivo de linha braille, que são pequenos pontos que aparecem em uma plataforma e vão formando as linhas braille, para que uma pessoa, principalmente surdocega, consiga ler o que tem na tela.

Então, vou fazendo um retrocesso na história. Logo que surgiram os primeiros leitores de tela, o mundo era um pouco mais fácil para a pessoa cega, uma vez que a gente tinha uma tela bem definida, era uma matriz, 24 linhas por 80 colunas. Então, se eu dizia que algo está na linha 10, coluna 4, a pessoa com deficiência mentalizava essa posição e sabia exatamente onde estava aquilo que queria se transmitir como informação. Com o passar do tempo, o surgimento do Windows, nós passamos a trabalhar com telas com pixels, e aí as coisas passaram a ter tamanhos diferentes, posicionamentos diferentes, inclinações, por que não dizer, informações praticamente em 3D, e tudo isso passou a ser um grande desafio para um leitor de telas. Em contrapartida, ao mesmo tempo, as sínteses de voz que eram todas em inglês e com qualidade extremamente difíceis de serem compreendidas, porque eram extremamente robóticas, passam a ter uma qualidade muito superior. A gente ganha o que nós chamamos de natural voice, a ponto de hoje um sintetizador ser confundido com voz humana, haja vista a voz do Google, a voz que vem integrada na Alexa, na Siri. São vozes humanas que são trabalhadas com inteligência de junção de fonemas e produzem esse resultado que é muito semelhante à voz humana.

Percebam que eu separei leitor de telas com sintetizador de voz. Por que eu fiz isso? Isso é muito importante. Porque o leitor de telas é a inteligência que capta a informação que está na tela, analisa e coloca esse conteúdo de uma maneira, entre aspas, linear, para que ela possa ser verbalizada através de um sintetizador de voz. Eu gosto de brincar que, assim, imaginem que vocês estão diante de uma tela, de uma página da Internet e estão com o telefone no ouvido e vocês precisam contar para alguém que está do outro lado da linha, o que tem naquela página de Internet. Só que a outra pessoa só está ouvindo vocês

por telefone, não tem acesso à Internet. Se vocês não se organizarem e não passar essa informação de maneira bem estruturada, a pessoa do outro lado não entenderá o que tem na página. É exatamente esse o desafio do leitor de telas. O sintetizador de voz tem só por obrigação de ter a melhor qualidade de voz no idioma pretendido e uma dinâmica de fala que não dificulte a compreensão por quem está ouvindo.

Bom, antes do HTML5, os leitores de tela precisavam interpretar o HTML e tentar descobrir a melhor forma de disponibilizar aqueles conteúdos. E desenvolviam atalhos e recursos que uma pessoa com deficiência aprendia para conseguir navegar por aquele conteúdo. Então, uma página da Internet, ela é colocada de forma linear, colocando os conteúdos de cima para baixo, da esquerda para a direita, para que a pessoa consiga explorar aquela tela, aquela página como um todo. E com a chegada do HTML5, nós passamos a ter inúmeros recursos que já foram pensados para que um leitor de telas fosse capaz de, pegando essas informações, conseguir oferecer mais facilidades para as pessoas com deficiência. Então, por exemplo, quando você define uma página bem construída, com os títulos, as tabelas, as listas, elas são claras. E eu, como usuário de um leitor de telas, tenho recursos que eu domino, que eu aprendo, que fazem parte do meu leitor de telas, para que eu possa: Olha, eu quero agora navegar nessa página através dos títulos. Agora eu quero navegar na tabela. Eu quero andar em colunas. Eu quero andar em linhas. Tudo isso, os leitores de tela me oferecem, desde que a pessoa tenha usado corretamente os códigos, a questão da semântica e todas as tags em lugares adequados, eu consigo fazer uso de todos esses recursos. E o HTML5 trouxe inúmeros recursos que rapidamente os leitores de tela foram incorporando e aumentou imensamente a facilidade que eu passei a ter de integração para com essa página.

Então, antes, a gente tinha que fazer muito mais malabarismos, porque o HTML entregava menos para os leitores de tela. Depois do HTML5, nós conseguimos ter mais facilidades de explorar todo o conteúdo de uma página, porque o HTML5 ofereceu mais facilidades para que o leitor de telas pudesse usufruir de todos esses recursos e disponibilizar facilidades para eu navegar. Por que para eu navegar? Porque, na verdade, todos navegam por uma página. Você olha por um canto, você rola a tela, você quer olhar, por exemplo, o que está acontecendo só nos títulos. Seus olhos vão percebendo os títulos, porque eles estão ou em tamanho maior ou estão em destaque. Para um sintetizador de voz, ele falar a palavra título da página, se é uma fonte maior ou uma fonte menor, ele vai falar do mesmo jeito. Então, eu preciso de outros recursos para que aquilo dê destaque. Quando eu tenho, por exemplo, um alerta que eu fiz um preenchimento errado, aparece uma boxinha no meio da tela. Os sintetizadores... o HTML quando informa isso de maneira correta, o meu leitor de telas percebe que ali é um alerta, ele me joga o foco lá para dentro e impede que eu saia de lá enquanto eu não dê ok. Óbvio, esse impede é relativo. Eu tenho recursos para sair de lá, sim. Mas ele tenta me mostrar que, olha, nessa parte da página tem um alerta e esse alerta, ele é importante. Observe isso. Se o meu leitor não avisasse isso, eu não saberia que ali tinha um alerta e não seria informado daquela chamada de atenção que a página dele deu para aquela questão. Aí eu teria que dar ok para continuar. Então, para quem enxerga, os olhos foram rapidamente ali e já percebeu. Eu precisei de uma inteligência do meu leitor de telas para detectar a existência de um alerta e me levar o foco para lá, para que eu soubesse que preciso ser avisado ali. Isso é o papel do leitor de telas. O sintetizador só falou, de preferência, com uma voz feminina bem agradável, o conteúdo daquele alerta.

Então, a grande evolução dos últimos dez anos para mim é essa, justamente de toda essa questão semântica, de todos esses cuidados que foram implementados no HTML5, porque já se havia a consciência da importância de ser acessível. Já existia um público e esse público precisava de recursos para conseguir acessar um conteúdo em condições de igualdade com quem não tem nenhuma deficiência. Eu foquei na questão da deficiência visual, mas isso se aplica a todas as outras deficiências, como esses pequenos robzinhos que fazem Libras, que são colocados no canto da tela e que transformam, em tempo real, a

Libras para quem é surdo. As facilidades que se tem do reconhecimento de voz para que a pessoa, através da voz, consiga operar a navegação por uma página. Enfim, todos esses recursos, eles só aconteceram porque passamos a ter uma consciência maior da importância de ser acessível. E o W3C, ele tem um papel imprescindível nesse processo como um todo, nesse trabalho de tornar todos os conteúdos acessíveis. E a ponto de, no primeiro curso de HTML, eu ser convidado para fazer parte, como uma pessoa com deficiência, já pensando justamente nas questões da acessibilidade, em como que a gente poderia fazer com que ficasse garantido, dentro da codificação de uma página, os recursos, para que as pessoas com deficiência pudessem acessá-los. E é isso o que a gente tem visto. A cada vez que surgem novos leitores de tela, que é o que tem acontecido com o VoiceOver da Apple, o próprio TalkBack do Android, eles já nascem preocupados com a questão da acessibilidade e têm uma interação muito mais forte com esses recursos semânticos e as tags de acessibilidade do próprio HTML5.

Inclusive, atualmente, uma coisa que tem sido fortemente trabalhada, são as questões do uso de reconhecimento óptico de caracteres, que cada vez mais tem sido implementado para que, além de uma própria figura que alguém pode colocar uma tag para descrever, com uma descrição curta ou longa, uma imagem, que haja também um reconhecimento óptico de caracteres e inteligência artificial para que aquela imagem seja efetivamente descrita e uma pessoa possa usufruir daquele conteúdo. Isso está já no Facebook, está em outros recursos. O VoiceOver implementou isso na última, na versão 14, e agora vai ser possível se reconhecer o texto de uma imagem que é enviada pelo WhatsApp, por exemplo. Antes, eu precisava pegar aquela imagem e enviar para algum software que fizesse o trabalho de OCR. Agora isso já vai ficando cada vez mais automático, porque os próprios leitores de tela, na sua inteligência artificial, vão implementando essas facilidades.

Enfim, acho que um resumo rápido, pelo tempo que nós temos, seria isso. E eu acho que o grande mote é isso, é a mão dupla. Leitores de tela usufruindo das facilidades que os HTML5 implementaram, as pessoas conscientes da necessidade de tornar os conteúdos acessíveis, e esse tripé muda a vida de pessoas que, até então, tinham muita dificuldade de acessar um conteúdo e hoje podem interagir facilmente. Haja vista que eu estou diante de um computador, usando o Zoom, eu consigo ativar vídeo, ativar áudio, eu consigo, se fosse necessário compartilhar telas, graças à preocupação que o Zoom já teve desde o seu início de tornar os conteúdos acessíveis a quem precisa da acessibilidade. É isso. Agradeço ao convite do Reinaldo, do W3C, para poder participar aqui. E me sinto muito honrado em ter participado do primeiro curso, principalmente porque, com esse convite, com essa participação, o W3C estava deixando claro que a acessibilidade fazia, sim, parte dos planos de desenvolvimento do HTML5 e do W3C Brasil. É isso. Muito obrigado a todos.

SRA. AMANDA MARQUES: Obrigada, Laercio, pela fala. Obrigada, Cesar, também.

Eu vou ler agora algumas perguntas que a audiência mandou, tá bom? Vamos lá. O Irven e o Elcio perguntaram: Quem quiser começar a aprender HTML5 hoje, por onde começar? Como é que eles podem se especializar na área de acessibilidade?

SR. CESAR CUSIN: Bom, como professor, eu, hoje, graças a Deus, a gente tem muito material disponível na web que eu poderia indicar. Tem livros, inclusive do nosso colega que vai falar daqui a pouco, do Fábio Flatschart, tem o livro dele. Mas, assim, como eu sou da época dos dinossauros, que a gente usava o bloco de notas, que desenvolvia tudo na unha, eu recomendo que se siga os padrões, as especificações, as recomendações do W3C de HTML5, obviamente, e do WCAG, que já estamos na recomendação 2.1. Então, eu recomendo, como tornar esse conteúdo acessível, quais os cuidados que devo ter em termos de

marcação, de estrutura e como que eu torno um conteúdo disponível. Porque eu vou pegar um gancho da fala do Laercio aí, que falou: "Pô, a gente consegue fazer mais coisas". A ideia da acessibilidade é essa, é dar autonomia aos usuários, o que lhe é de direito. Ele tem que ter autonomia, para, como o Laércio disse, acessar, abrir um vídeo, compartilhar a tela, enfim, todo aquele trabalho que ele tinha com imagem, que ele falou, para um outro software tratar a imagem, hoje ele não tem mais. Então, é isso. A briga da acessibilidade é essa.

Então, se você está pensando em estudar acessibilidade, que essa foi a pergunta, eu recomendo, tem livros excelentes, mas também temos padrões e temos o WCAG, que eu acho um documento importantíssimo. Tem que passar por ali. Como que eu vou disponibilizar conteúdo acessível? Você tem que estar de olho no WCAG, em sua última versão.

SRA. AMANDA MARQUES: Laercio quer completar?

SR. LAERCIO SANT'ANNA: Acho que perfeito, não tenho a nada a acrescentar, não. Talvez...

[falas sobrepostas]

SRA. AMANDA MARQUES: Tá bom.

SR. LAERCIO SANT'ANNA: É claro que assim, talvez seja importante a pessoa realmente entender quais são necessidades que cada deficiência apresenta, e porque... saber para o que você está aprendendo, o que você está desenvolvendo, para que tipo de público e como é esse público. Isso sempre, com certeza, evita que se construa um Frankenstein.

SRA. AMANDA MARQUES: Tá bom. Tem mais uma pergunta que é: Existe uma forma... Quem fez foi o Irven, Tá? Existe uma forma de validar o código desenvolvido para leitores de tela? Pois, algumas vezes, nós, desenvolvedores, não temos a mesma experiência dos usuários que necessitam desses recursos. Qual a maneira mais recomendada de validar esse tipo de implementação?

SR. CESAR CUSIN: Eu não me lembro o nome de quem fez a pergunta, desculpa, mas...

SRA. AMANDA MARQUES: Irven.

SR. CESAR CUSIN: Irven, ótima a sua pergunta. Irven, existem os validadores de código para a parte de acessibilidade, sim. Só que, vamos lá, para a gente fazer a coisa certa, eu não posso disponibilizar um site ou dizer que ele está acessível porque ele passou por um validador e obteve 100%. Você tem que fazer testes com humanos. E quando o Laercio disse isso, eu achei muito importante. Ele está pendendo mais para a comunidade cega, porque, óbvio, é o que ele trabalha e, efetivamente, ele é cego. Mas nós temos os surdos, nós temos pessoas com baixa visão. Então nós temos que fazer testes de acessibilidade com pessoas. Nunca descarte sem fazer esses testes com humanos. Não adianta você fechar seu olho, que também eu acho uma tarefa legal, e tentar usar software leitor de tela para ver o impacto que isso causa. Isso é louvável, mostra que você está preocupado, mas você não fique só em validadores automáticos. Faça testes com humanos, convide usuários cegos, surdos, usuários videntes. Então, chame os usuários para o seu teste. O teste com humanos é importantíssimo para validar o seu trabalho em relação à acessibilidade.

SR. LAERCIO SANT'ANNA: Bom, complementando a fala do Cesar, né? É exatamente isso, é muito importante esse cuidado, porque às vezes, as pessoas falam assim: Ah, eu vou instalar um leitor de telas aqui no meu computador e eu testo. Olha, sem querer ofender, sem querer menosprezar, mas um leitor de

telas, ele chega a ter 2 mil comandos, atalhos, sabe? É uma coisa de louco. Por quê? Precisam de muitos atalhos para poder resolver tudo que precisa. E a gente, no dia a dia, vai se acostumando com tudo isso. Então, a pessoa que enxerga, por mais que ela tente usar um leitor de telas, sim, vai ter ideias em questões pontuais, mas nunca vai ter a dinâmica do uso de uma pessoa que é realmente usuária de um leitor de telas.

É o mesmo que acontece, os surdos reclamam bastante desses geradores de automático de sinais, de Libras - não sei se eu posso chamar assim - porque eles falam que tem problemas. Eu não consigo imaginar como são, porque eu não vivo a realidade deles. Mas eu tenho certeza que, obviamente, é melhor que nada, mas não é como um ser humano fazendo Libras para eles. Enfim, se você quer um site que realmente seja... que contemple todas as acessibilidades, é importante que você tenha todos os tipos de públicos que precisam daqueles sites sendo testados, porque a percepção e a qualidade será outra.

Para terem ideia, o município de São Paulo criou um selo de acessibilidade, principalmente com o objetivo de incentivar com que todas empresas e serviços públicos se tornem cada vez mais acessíveis. A gente tem uma primeira validação que é feita pelo Ases, o avaliador Ases, que ele faz toda essa análise, como o Cesar comentou. Mas, enquanto ela não passar pela CPA, que é a Comissão Permanente de Acessibilidade, e a gente fazer uma navegação cognitiva, uma validação através de uso de leitor de tela e de outros recursos, a gente não pode garantir que aquele site está aprovado para receber o selo.

Então é isso. É muito importante, sim, a validação, mas não existe nenhum software que substitua o teste humano. Seja ele de uma pessoa com deficiência visual, surda, analfabeta, idoso, porque cada um deles tem sempre sua peculiaridade.

SR. CESAR CUSIN: Se me for permitido, eu quero...

SRA. AMANDA MARQUES: Obrigada, pessoal. Desculpa. Pode falar.

SR. CESAR CUSIN: Só fazer uma correção. Quando me perguntaram sobre indicações de como começar a estudar HTML e acessibilidade, eu quero corrigir aqui e fazer justiça. Nós temos o recente lançamento do livro do Reinaldo, Acessibilidade na Web. Então, foi muito bem lembrado aqui. Reinaldo, desculpa, passou correndo aqui, mas a tempo eu consigo fazer. Indico também, como professor e como pesquisador na área, o livro do Reinaldo, Acessibilidade na Web. Obrigado.

SRA. AMANDA MARQUES: Obrigada, gente.

Eu queria poder fazer mais perguntas, mas a gente já está bem atrasado no cronograma, infelizmente. Quer dizer que foi bom, foram bons os debates, né? O pessoal está se engajando. Agradeço muito, Cesar e Laercio, pela participação. Agradeço muito o público que está no chat. Queria agradecer os nossos multiplicadores das iniciativas de acessibilidade. Sem vocês, assim, esse trabalho não estaria sendo disseminado por todo o Brasil. Agradeço muito ao Reinaldo por todo o seu trabalho desde o início, lá atrás, em levantar essa bandeira e fomentar o conteúdo aqui no Brasil.

E queria lembrar vocês que temos vários conteúdos de acessibilidade, inclusive traduzidos em português, tanto no site do Ceweb quanto no W3C Brasil. Nós temos os quatro fascículos de acessibilidade na web, nós temos o WCAG, e temos também o WAI-ARIA, que são materiais que podem fomentar vocês aí nos estudos de acessibilidade.

Agora, eu passo a palavra para a Bia, que vai moderar o último painel. Obrigada, gente.

SRA. BEATRIZ ROCHA: Oi, gente. Bom dia. Eu sou a Bia, como a Amanda disse. Vou me autodescrever aqui rapidinho. Eu sou negra, tenho cabelo cacheado e meu sinal para a comunidade surda é assim.

Só fazendo... acrescentando a Amanda também, o conteúdo está no acervo do Ceweb, recém-lançado também, né? Só fazendo o merchan também.

Nesse painel, nós vamos discutir um pouco sobre o futuro da HTML5, o que essa tecnologia nos trouxe e o que podemos esperar dela. Para isso, nós convidamos a Miriam Von Zuben, o Fábio Flatschart e o Elcio Ferreira. Sejam bem-vindos. Eu vou pedir para vocês se apresentarem rapidamente. E é um prazer tê-los aqui com a gente.

SRA. MIRIAM VON ZUBEN: Boa tarde a todos. Estão me ouvindo?

SRA. BEATRIZ ROCHA: Estamos.

SRA. MIRIAM VON ZUBEN: Estão? Meu nome é meu Miriam. Vou me descrever também. Eu tenho cabelos... acho que está meio castanho escuro agora com a pandemia, tenho cabelos cacheados, tenho a pele clara e uso óculos de grau. Eu sou analista de segurança, trabalho no Cert.br, que é o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, que é um dos departamentos do NIC.br. E agradeço muito o convite para participar. Quando o Reinaldo fez o convite, né? Tentei lembrar, nossa, dez anos atrás, né? Como a vida muda em dez anos, né? Quando eu fiz o curso, eu estive na primeira turma, para mim foi uma honra poder estar lá. Desde então, assim, eu não tenho trabalhado diretamente com desenvolvimento, com a parte de HTML, tenho trabalhado mais com a parte de estudar a experiência dos usuários. Então assim, como que os usuários estão nesse mundo todo, né? E coincide que, nessa época, em 2010, eu estava começando a escrever a cartilha de segurança do Cert.br. Então, é bem interessante que esses dez anos de evolução de HTML, também, para nós, é como se fosse dez anos de evolução de toda a parte do que os usuários estão percebendo de segurança e dos sistemas em geral, né?

Então, assim, até conto aqui com meus companheiros de painel, o Fábio e o Elcio, algumas perguntas mais relacionadas a desenvolvimento, eu vou passar direto para eles, porque a minha experiência é um pouco mais do outro lado. É isso.

SRA. BEATRIZ ROCHA: Seja bem-vinda. Elcio, você poderia se apresentar para a gente, rapidamente?

SR. ELCIO FERREIRA: Claro. Bom, meu nome é Elcio, né? Eu vou me descrever: eu sou um homem branco, de meia idade, tenho cabelos pretos, barba preta e uso óculos de grau. Eu trabalho com HTML desde 1997, acho, 96. Em 96 comecei a estudar, 97 comecei a trabalhar com isso. Sou apaixonado por HTML. A gente começou a construir sites usando CSS para layout em 97. Em 99, quando a gente se engajou no movimento via web standards, né? O pessoal nos Estados Unidos estava falando disso. E a gente começou a ensinar isso aqui, começou a ensinar as pessoas. O Diego foi meu aluno nessa época aí, né? Tive a sorte de tê-lo como aluno.

Eu sou um dos primeiros autores lá do Tableless. O Diego é quem começou a coisa toda, e eu estava lá junto com ele naquele começo. Eu tenho uma empresa desde 2006, que a gente trabalha desenvolvendo aplicações web acessíveis e ensinando as pessoas a fazer isso, né? E aí, em 2010, o Reinaldo cometeu a insanidade aí de me convidar para ser o professor lá do curso de HTML5 junto com o Diego. Foi muito importante para mim. Eu era, assim, sempre fui viciado nesse assunto, estava mergulhado tecnicamente no HTML, mas eu tive o desafio de dar aula para muita gente muito mais talentosa e inteligente do que eu, né? Quem está assistindo pelo YouTube aí viu os painéis anteriores, o pessoal falando aí.

Então, eu aprendi muito naquele curso. Alguns momentos do curso ali, que eu não estava dando aula, eu estava escutando os participantes ali, ouvindo e absorvendo o que as pessoas tinham a dizer. Então, eu não me considero um professor, eu fui mais um facilitador que deu a sorte de estar sentado ali para aprender com aquele pessoal todo.

SRA. BEATRIZ ROCHA: Obrigada. Fábio, você poderia também se apresentar?

SR. FÁBIO FLATSCHART: Olá, pessoal, bom dia. Muito prazer, para quem eu conheço, para quem eu não conheço ainda. Muitos de vocês já nos encontramos em outros lugares. Eu sou Fábio Flatschart. Eu sou careca, tenho barba branca, tenho 54 anos, né? E estou aqui no quarto do meu filho, cheio de games e pôsteres na parede, porque o meu quarto está em reforma. Mas fica simpático de qualquer maneira.

Bom, já trabalhei com webcomunicação há muitos anos, mas o meu gancho com HTML5 foi o seguinte. Por volta de, enfim, 2006, 2008, eu trabalhava como consultor da Adobe, né? E o Flash era a grande ferramenta do desenvolvimento. Então, como eu trabalhava para a Adobe, o Senac me convidou para escrever material didático. Então, Flash, ActionScript, Dreamweaver, enfim, softwares e as ferramentas Adobe todas. Aí, em 2009, o Senac fala assim para mim: "Fábio, vamos ter que mudar radicalmente o nosso portfólio aqui, porque"... Desculpa, gente. "O nosso portfólio, porque a gente vai ter que trabalhar com HTML5 agora, é um padrão, é uma tendência, né?". Então, assim, eu fui convidado para participar da segunda turma, né? Junto com o Diego, com Elcio, da HTML5. E a minha missão era trazer o HTML5 para o Senac, que tinha pouca, assim, vivência com esse assunto. Todos os outros cursos eram muito galgados em, como eu falei, no Flash, e ferramentas Adobe. Então, esse foi meu primeiro contato, né?

E desse contato saíram os frutos, né? Eu escrevi um livro de HTML5, que foi usado pelo Senac, muitos anos, né? Assim, eu acho que eu sou vitorioso nesse ponto, porque 6 mil livros, né? Assim, aqui no Brasil é uma coisa rara, né? Então, dentro do Senac foi um período muito bacana. Depois eu produzi junto com o Clécio e com o Cesar, que estava aqui agora há pouco, também um outro material sobre open web. Depois eu trabalhei alguns anos com editoras, né? Então, eu fui trazer o HTML5 para os formatos de ePUB, ePUB 3, né? Eu tenho um livro sobre isso também. Enfim, foi uma guinada na minha vida, o HTML5, tanto pessoalmente quanto profissionalmente, né? Para trazer sim. Eu não sou desenvolvedor, eu sou uma pessoa da área de artes, comunicação. Mas como eu sempre trabalhei com a parte acadêmica também, eu tomei esse rumo dentro do HTML5. É um prazer estar aqui ouvindo vocês novamente.

SRA. BEATRIZ ROCHA: Muito legal que vocês já passaram também um pouco da experiência de vocês com HTML5. E agora eu gostaria de saber como que vocês veem essa evolução tecnológica e o que vocês acham que a gente pode esperar do HTML5.

SR. ELCIO FERREIRA: Ah, legal. Eu queria falar nessa.

Bom, eu fiz uma lista aqui de coisas que a gente não podia usar, embora já fossem, de alguma maneira, ou parte da especificação, ou estavam em discussão lá em 2010. Mas que, por conta dos navegadores que a gente tinha na época, do padrão ser ainda todo muito novo, a gente ainda não podia usar ou podia usar para 40% do público, mas que hoje já são suportadas pela maioria dos navegadores. Quer dizer, os navegadores atuais, todos suportam. E aí você vai ter um pequeno percentual do público usando navegadores antigos, né?

Para começar, ECMAScript 6, tá? Então, quem é desenvolvedor aí, proxys, promises, arrow functions, funções assíncronas, tudo isso não era possível usar em 2010. A gente estudava esse assunto, achava muito legal e sonhava com o dia em que os navegadores iam implementar isso. Source set, lazy load, você

poder carregar páginas mais leves no celular e mais ricas em quem tem uma tela maior, né? Os tipos de campos de formulário. Então, formulário com input color, input [ininteligível], input date, progress. Então, você, em 2010, se você tinha que fazer, por exemplo, um usuário te fornecer uma data, você tinha que programar um jeito dele fornecer a data. Hoje você coloca lá um input type date, e se ele estiver no celular, vai aparecer um jeito bem confortável dele usar no celular, se ele estiver no navegador, vai aparecer um jeito bem confortável de ele usar no desktop. Então, funciona bastante bem. É padrão implementado na esmagadora maioria dos navegadores. E coisas que permitiram tornar a web muito mais rica, né? Como API para envio de notificações, WebGL, que permite você renderizar gráficos, gráfico 3D, Vibration API, permite você fazer o telefone vibrar, quando você recebe alguma coisa ali e tal. GetUserMedia, que é o que permite você capturar a câmera do usuário, o microfone dele, e que, junto com o webRTC, permite você fazer coisas como videoconferência no navegador, sem ter que instalar mais nada, sem ter que usar um outro aplicativo, nada disso. Beacon API, que serve para você detectar beacons, então você pode fazer geolocalização indoor, uma série de outras coisas, usando Beacon API. Web Audio API, serve para você gerar áudio no navegador do usuário. Você poderia fazer, por exemplo, um editor de áudio totalmente baseado em web. Index [ininteligível], que permite você gravar dados lá no navegador, ter um banco de dados rodando no navegador do usuário. E o service workers, os custom elements, essas coisas que permitem a gente transformar as páginas web, de fato, em aplicações.

Esses recursos todos, a gente junta eles todos, mas a possibilidade de uma aplicação navegar off-line, que também era muito incipiente ainda em 2010, e hoje já é uma realidade. A possibilidade de você oferecer a instalação de uma aplicação web, e você tem hoje plataforma, que é o que a gente chama de progressive web apps, de PWAs, que permite você construir páginas web, que, em sua maior parte, são tão ricas quanto um aplicativo instalado no celular ou uma aplicação instalada no seu computador.

Então, a web tem cada vez mais, ganhado recursos com HTML5, que fazem com que seja cada vez menos necessário para uma porção de casos de uso, você ter que desenvolver um aplicativo instalável, né? Você pode fazer isso via web. Isso é maravilhoso. Você faz uma vez só para todas as plataformas e todos os dispositivos, mantém uma versão só do seu aplicativo. Então, o HTML5 foi uma mudança de realidade, não só para web, mas para o desenvolvimento de software como um todo.

SRA. BEATRIZ ROCHA: Bom, nesse contexto, você poderia explicar um pouquinho para a gente o que é PWA?

SR. ELCIO FERREIRA: Quando o Steve Jobs lançou lá o primeiro iPhone, na apresentação, ele mostrou o iPhone, falou dos recursos que ele tinha: "Olha, tem uma calculadora, tem uma câmera, tem um telefone, tem um iPod aqui dentro e você pode desenvolver aplicações para ele, porque ele tem um Safari completamente funcional". O primeiro iPhone não tinha uma loja de aplicativos, a ideia original lá era: você vai fazer aplicações web. E a comunidade de desenvolvedores começou a protestar por conta disso. Por quê? Ah, numa aplicação web, eu não vou poder usar os recursos tão legais que vocês estão colocando nesse aparelho aí. Eu não vou poder usar a câmera, eu não vou poder fazer o aparelho vibrar, eu não vou poder fazer ligações telefônicas, eu só vou poder fazer coisas que as pessoas podem usar quando elas tiverem conexão. Então, no HTML5, o W3C incorporou recursos que resolveram esses problemas todos. Então, uma PWA é uma página web que usa recursos que podem torná-la tão rica quanto um aplicativo.

SRA. BEATRIZ ROCHA: Muito bem. Obrigada.

E eu também queria saber, nesse contexto, nessa evolução, o que essa evolução nos trouxe relacionado com segurança nas aplicações web?

SRA. MIRIAM VON ZUBEN: Um pouquinho? Está ouvindo? Essa eu posso tentar ajudar um pouquinho.

Eu acho que, contando com essa minha experiência, mais do lado do usuário, né? O que a gente via, em 2010, é que a gente tinha feito... tinha um trabalho muito grande de explicar para os usuários. Por exemplo, se você vai acessar um site, utilize sempre conexões seguras, e a gente tinha que explicar para ele o que era uma conexão segura. A gente tem visto com essa evolução, que hoje, tem tido um grande movimento. Eu até fiquei feliz numa das falas que alguém falou anteriormente de que nós estamos muito maduros em relação à segurança atualmente. Eu fico muito feliz de ouvir isso. Eu vejo que, sim, muita coisa evoluiu nesses últimos dez anos, né? Hoje, por exemplo, se você faz uma conexão não segura, aí sim que o browser te avisa, né? Então, assim, conexões seguras já passaram a ser padrão. Mas toda essa tecnologia, na verdade, sempre que você traz ferramentas novas, você acaba sempre abrindo um campo maior de formas de explorações, né?

Então, assim, infelizmente, a gente, acompanhando diversas estatísticas, a gente vê que aplicações web ainda são um vetor de ataque para muitas empresas, né? Quando eu falo de aplicações web, eu estou falando em geral, do servidor, da parte de desenvolvimento. Então, assim, a gente ainda vê diversas aplicações web sendo usadas como intermédio para outros tipos de ataque, né? Toda a parte de vazamento de dados hoje tem bastante coisa sendo explorada por aplicações web. Sistema sendo invadido para colocar páginas falsas de phishing, por exemplo. Então, assim, tem vários exemplos. Tem alguns malwares que exploram algumas características do próprio HTML5, né? Que é a parte como ele ofusca os dados, e assim, a forma como o usuário baixa, vai do lado do usuário. E aí esse malware é montado, isso aí acaba, algumas ferramentas de segurança não conseguem detectar isso. Então, assim, a gente tem recursos de HTML5 que acabam sendo explorados.

Então, assim, eu vejo que a gente está num momento em que, sim, realmente, tem muito recurso, tem muita coisa que está sendo desenvolvida, mas a gente está num cenário em que coisas antigas ainda precisam ser tratadas. Como, por exemplo, toda a parte de sistemas desatualizados, usuários com senha fraca. E daí, eu acho que o Fábio até pode ajudar depois, toda a parte de educação de usuário também, que ele tem trabalhado. Toda a parte de programação segura também. Eu acho que é uma área que precisa também ser trabalhada, né? Então, assim, eu vejo que área de segurança a gente tem bastante desafios ainda.

SRA. BEATRIZ ROCHA: Bom, já pegando o gancho do que você falou, né? Que o Fábio pode ajudar um pouco na área de educação. Nós ouvimos o Cesar e o Laercio falando sobre o aprendizado para quem quer conhecer sobre acessibilidade na web. E eu gostaria de saber também como que uma pessoa que está entrando agora na área e quer saber um pouco sobre HTML5, quer aprender um pouco sobre essa tecnologia, como que ele pode fazer, essa pessoa? Fábio.

SR. FÁBIO FLATSCHART: Legal, Beatriz. Bom, como vocês viram o Elcio e a Miriam falando, né? Como a gente tem uma capilaridade do HTML5 agora, né? Ele está... quando há dez anos atrás (sic) era uma linguagem que engatinhava, hoje ela está presente em todos os dispositivos, sistemas, plataformas, né? Então, no ponto de vista educacional, acho que nós também temos essa multidisciplinaridade. Então, eu vejo assim: quero aprender HTML5. Bom, eu vou dar um exemplo. Se eu sou um designer, eu vou trazer a experiência do HTML5, as boas práticas do HTML5, para aonde? Construção de wireframes, testes de usabilidade, prototipagem, né? User interface, user XP, ou seja, temas que tenham a ver com a interface. Então, eu já posso pensar num projeto comunicacional, num projeto de design, a partir das práticas do HTML5. Ou vamos pensar o seguinte. Eu sou profissional de comunicação, eu trabalho com rádio e TV,

jornalismo, mídias digitais, editoração. Então, eu vou pensar na produção de conteúdo com HTML5, né? Então, eu vou pensar em portabilidade, né? Pensar meu conteúdo, como as funcionalidades do HTML5, ele pode estar presente em todas as plataformas, né? Eu sou um profissional de marketing, eu vou pensar no HTML5 dentro do contexto do quê? De edges(F), né? De SO, né? Otimização de conteúdo.

Então, são várias portas de entrada do HTML5. Sem contar, lógico, da porta de entrada do desenvolvimento. Ou seja, quem vai ser mesmo um desenvolvedor de código, né? Que vai, enfim, trabalhar integração, acessibilidade. Então, assim, para quem está começando, assim, eu tenho visto assim, às vezes, nem todo mundo quer ser um desenvolvedor, mas é importante ter esse olhar do HTML5 nessas áreas todas. Eu acho que eu vou até dar um copy aqui, vou pôr aqui, depois, se vocês quiserem compartilhar, eu pus essas dicas aí. Porque eu acho que é importante. Eu vejo muito na minha experiência educacional, é o seguinte. Muitas pessoas procuram HTML5, e nem todos vão ser desenvolvedores, né? Vão ser programadores, né? Muitos procuram dentro desses vários contextos, design, comunicação, marketing, né? Então, ele hoje é multidisciplinar. Então, você tem que procurar um curso, ou uma escola, ou um livro, ou uma informação que te encaminhe para essa sua necessidade, né?

Agora, se você pensar em grandes temas do HTML5. Por exemplo, você pode pensar é importante estudar história e evolução, né? Ou seja, o HTML5 não foi uma tábua rasa, né? Ou seja, tivemos muitos antecedentes, né? Outras versões da linguagem. Então, importante para quem está estudando compreender a evolução de todas as versões, suas funcionalidades. Importante compreender o papel do HTML5 dentro das camadas front-end, o que é conteúdo, o que é formatação, o que é interatividade, comportamento, né? Assim, é difícil você, assim, achar um único foco, né?

Esse outro tema, eu acho muito importante, que é estudar como uma linguagem. Como se fosse, então, você tem a morfologia, o estudo das tags, das palavras, elementos. Você tem a sintaxe, né? Que são as regras de construção do código. E tem a semântica. Então, você pensar essas três subcamadas: morfologia da linguagem, a sintaxe e a semântica, né? A semântica é um tema que acho que todos os que já passaram por esse painel aqui, enfim, apontaram, como uma grande riqueza comunicacional no HTML5. Muitas outras pessoas podem procurar também o estudo do HTML5, por quê? Porque eles já trabalham com bibliotecas, com frameworks, com CMS, com APIs. Então, assim, você precisa ver qual é o seu foco dentro do HTML5, né? Independente disso, eu acho que é sempre bom pensar em padrões. E, a partir daí, você abre o seu leque para essas outras áreas. Desde que você tenha os fundamentos e os padrões como base. Não sei nem se eu falei demais, mas desculpa, gente.

SRA. BEATRIZ ROCHA: Não, está ótimo.

SR. FÁBIO FLATSCHART: É isso aí.

SRA. BEATRIZ ROCHA: Aproveitando que você falou sobre estudar a evolução da HTML5, da HTML, eu gostaria de aproveitar e perguntar para vocês: como que vocês imaginam que será a web daqui a dez anos?

SRA. MIRIAM VON ZUBEN: Eu vou começar. Eu acho que assim, acho que no mundo da tecnologia, é muito difícil a gente fazer qualquer tipo de previsão, né? Acho que é só para ver como era dez anos atrás, como está hoje, né? Difícil você prever. Eu costumo falar para os meus filhos que aquilo que eles vão trabalhar ou que eles vão ter como profissão provavelmente ainda não foi inventado, ou se foi inventado, a forma como isso vai ser feito vai ter muita novidade, né?

O que eu gostaria, na verdade, é de pensar que, independente da forma que vá acontecer daqui a dez anos, a gente não precise mais repetir problemas de segurança que a gente tem falado que então seriam 20 anos atrás, né? A gente não precisa mais falar toda essa parte de programação segura, assim, que a gente já vai ter o sonho de segurança por design, né? Isso daí já implementado. Que a gente possa, sim, aí focar em outras coisas, né? Então, esse, na verdade, seria o meu sonho para daqui dez anos.

SRA. BEATRIZ ROCHA: Obrigada. E vocês, meninos?

SR. ELCIO FERREIRA: Bom, eu imagino que a web vai, cada vez mais, ser o nosso sistema operacional. Quando eu penso nos últimos dez anos, eu penso como a conexão se tornou mais ubíqua, se tornou melhor, se tornou mais barata. Então, hoje a gente está conectado o tempo todo. Eu tenho quatro filhos, a gente está aqui em casa em isolamento, né? E de manhã, eu vejo todo mundo tendo aula, cada um num quarto, fazendo live ao mesmo tempo, eu fazendo live aqui. Eu tenho uma conexão doméstica, eu não pago uma fortuna de Internet, nada disso. E isso é muito bom. E a gente já, pelo menos uma parte da nossa população que vive em cidades já vive a situação privilegiada de quando falta Internet, parece que está faltando um serviço essencial, é como faltar eletricidade ou faltar água, sabe? Não é um extra.

Então, eu imagino que a gente vai, cada vez mais, não ter as coisas em nossos computadores e tê-las na web. E a gente vai, cada vez menos, depender de um dispositivo específico, né? Então, interface de navegação por voz, navegação na televisão, o celular no seu bolso, o seu desktop, se é que a gente ainda vai ter o modelo de ter um único desktop. Todos eles vão ser clientes da web. A web é o seu sistema operacional. A gente vai ter, cada vez menos, de chegar numa reunião e dizer: Puxa. Isso está no meu computador que ficou no escritório, né? Isso vai cada vez menos, acontecer.

SRA. BEATRIZ ROCHA: Fábio, você gostaria de acrescentar alguma coisa?

SR. FÁBIO FLATSCHART: Bem dito aí pelos colegas, né? Eu só gostaria de falar que, apesar de termos esse otimismo, assim, eu vou dizer, pelo menos da minha parte, uma certa frustração, né? O mercado editorial, eu achei que ele ainda tende muito a evoluir nesses próximos dez anos. Então, os formatos de livros digitais, de entretenimento digital, eu acho que ainda fazem pouco uso, né? Às vezes, as pessoas ficam muito presas ainda em aplicações fechadas, em conteúdos fechados, sendo que tem essa área gigante ainda de mídia, de mercado editorial, comunicação, que ainda poderia se beneficiar mais dessas múltiplas funcionalidades que a web hoje já permite, sem que você fique preso em uma plataforma, em uma ferramenta, em um sistema operacional. Ou seja, a web ainda pode ser o grande sistema, a grande plataforma, com esses novos entrantes aí da área de mídia, comunicação e principalmente o mercado editorial que torce para o formato ePUB3 decolar de uma vez, que ainda, na minha opinião, ele ainda está engatinhando. É isso.

SRA. BEATRIZ ROCHA: Obrigada, gente. Nós separamos três perguntas que fizeram no chat do YouTube. A primeira pergunta é do Sérgio Eduardo Antônio. Ele diz: Visando o incremento de segurança nas páginas da Internet, lembrando que a maioria dos usuários são leigos, pergunto: o que vocês... o que os novos padrões podem contribuir para o incremento da segurança?

SRA. MIRIAM VON ZUBEN: Bom, vou tentar ajudar aqui. É, não sei, pegando por base o que o próprio Elcio falou, quando a gente fala em PKA, é isso que você falou, né? A gente já vê que já tem toda a parte de, por exemplo, https por trás, né? Assim, então são ferramentas que estão surgindo já com uma visão de segurança já sendo implementada. Não sei se o Elcio pode me ajudar a complementar.

SR. ELCIO FERREIRA: É, sim. Eu acho que o W3C tem feito esse trabalho, sabe? Então, as especificações são revisadas de tempos em tempos e a implementação desses padrões nos navegadores vai mudando com o tempo. Então, recentemente, por exemplo, eu descobri que agora para você colocar um atributo download numa tag link lá no seu HTML, que é o que faz quando você clicar num link o navegador fazer download ao invés de simplesmente abrir a página. Para você fazer isso, você necessariamente precisa estar numa página sendo servida com https. E esse tipo de revisão acontece o tempo todo. Então, vamos lembrar que antes do HTML5, antes de os padrões web se tornarem superpopulares, existia um jeito de você criar uma janela chamada chromeless. Você abria um pop-up do navegador, uma janela, que ela não tinha a barra do Windows lá em cima, os botões de fechar, etc. Isso foi muito usado para ataques virtuais e aí isso foi eliminado do navegador. Então, os navegadores bloqueiam, por exemplo, requisição entre domínios diferentes, né? Então, boa parte dos ataques que a gente tem lá no OWASP top 10 são ataques que os navegadores tentam mitigar. As coisas seriam muito piores sem essa evolução do padrão. Então, isso sim, está acontecendo.

Acontece que as pessoas mal-intencionadas são bastante criativas, né? Então, o padrão vai sendo revisado de tempos em tempos, é natural, isso acontece mesmo, tá? Isso não substitui, acho que a Miriam falou muito bem sobre isso, não substitui uma cultura de segurança, claro. O navegador só está ali tentando evitar aquelas falhas que são de responsabilidade do próprio navegador. Mas, sim, você precisa continuar estudando, você precisa continuar construindo o seu código, e segurança tem a ver com o que você está construindo e não com que ferramentas você usa. Então, não adianta tentar: ah, vou implementar um padrão de segurança no HTML5. Não é assim que as coisas funcionam, tá? Você precisa, sim, estudar segurança. Você precisa, sim, incorporar isso no seu processo de desenvolvimento. Não tem jeito.

SRA. BEATRIZ ROCHA: É, nós temos mais uma pergunta do Sérgio Eduardo Antônio. Ele diz: Quais são os prós e contras de HTML e quais... e demais recursos [ininteligível]?

SR. ELCIO FERREIRA: Ah, legal. Essa aí, acho que eu posso responder, né?

Bom, você usar dart com flutter, ou você usar kotlin ou você usar swift, você vai gerar uma aplicação que é nativa. Então, a sua pergunta, na verdade, ela é mais ou menos assim: Quais são as vantagens de eu gerar aplicação nativa, compilada, para ser instalada ali naquela plataforma específica, ou de eu construir uma PWA, uma aplicação que é web, né?

Então, vamos lá. Vantagens do PWA. O PWA, ele é atualizado automaticamente. Você lançou versão nova, você pode forçar todos os seus usuários, quando acessarem, a estar simplesmente usando a versão nova. E eu não estou falando de mostrar uma caixinha para ele dizendo "você precisa atualizar essa aplicação para continuar usando, clique aqui para atualizar". Eu estou falando de simplesmente carregar a versão nova para ele. Então, essa é uma vantagem. Você constrói uma vez só para todas as plataformas. Essa é uma vantagem, uma vantagem muito legal.

Vantagem de você ter um código nativo, e aí, por exemplo, dart é um negócio apaixonante nisso. Você pode construir código para usar todas as APIs do sistema operacional em que você estiver trabalhando. Não, eu estou trabalhando com flutter no Android. Você vai poder fazer coisas ali que você não pode fazer numa aplicação web, como, por exemplo, ah, eu vou fazer uma aplicação aqui que tem que baixar gigabytes de dados para poder tocar vídeo em background. Você não vai conseguir fazer isso numa aplicação web. Pelo menos não vai por enquanto. Ou estou fazendo uma aplicação aqui que precisa monitorar o usuário em tempo real, mesmo quando essa aplicação não estiver aberta na tela, fazer um

envio da geolocalização dele em segundo plano, em tempo real, para um serviço qualquer. Você provavelmente vai precisar de uma aplicação nativa usando APIs do Android ali.

Então, ao fazer uma aplicação que é web, você se limita às APIs que existem para web. Se a aplicação que você vai construir pode ser limitada a essas APIs, performance em JavaScript hoje é muito difícil de ser um problema, tá? Embora, é claro que você não vai comparar a performance entre o nativo e o JavaScript, mas para 99% dos casos de uso, eu vou preferir a aplicação web. Exceto para esses casos em que eu realmente preciso usar algo que eu preciso ter a aplicação instalada lá.

SRA. BEATRIZ ROCHA: E agora nós temos mais uma pergunta, a última. É a pergunta do [ininteligível] Faria. Ele diz: Vocês acreditam que o desenvolvedor front-end atualmente deve possuir algum conhecimento sobre design, UX e prototipagem? E que esse pode se tornar um diferencial no mercado? Ou o desenvolvedor deveria focar apenas nos aspectos técnicos e implementação do software?

SR. FÁBIO FLATSCHART: Eu posso falar. Se alguém quiser complementar depois, né? Esse é um foco bem específico aí. Veja bem, no mercado, vocês vão ter várias especificações. Vocês têm o front-end designer, o front-end developer, front-end engineer. Certo? Então, são habilidades e conhecimentos que se mesclam, né? Independente de que área você aja, se você é da área de design ou de desenvolvimento, eu acho que hoje em dia você não pode pensar mais em caixinhas separadas. Então, todo projeto já deve nascer holístico, semântico, acessível, com boas práticas de experiência do usuário. Então, tanto da parte de quem escreve o código, tem que conhecer essas habilidades, quanto as equipes de projeto, design e prototipagem saber, pelo menos, como esse código pode ser implementado. Então, eu acredito sim, na integração das áreas, né?

Há uns anos atrás (sic) estava na moda, não sei se ainda, o termo devigner, que é uma mistura de developer com designer. Então, se escreve devigner, né? Ou devigner. Então, eu acho que isso é bem apropriado nesse contexto, você ter essa integração do pensar desenvolvimento e do pensar interface, experiência do usuário. Então, o mercado recebe muito bem pessoas que têm essas duas expertises, o olhar sobre o código e o olhar sobre UX, interface, direção de arte. Eu acho que é uma necessidade hoje do cliente e de quem contrata esse serviço. Minha opinião.

SR. ELCIO FERREIRA: É, eu gostaria de complementar. Achei fantástica a resposta do Fábio. É uma escolha pessoal, né? Tem gente que escolhe ser generalista e tem gente que escolhe ser especialista, e o mercado precisa das duas pessoas. Ele precisa de gente que tenha visão de tudo, que saiba fazer um pouco de tudo para coordenar projetos, para ajudar o time a se comunicar, para determinadas funções que realmente precisam mexer em tudo. E precisa de especialistas, né? Que é aquele projeto que é muito difícil, que precisa ter um especialista em cada área que sabe muito bem aquilo ali. Mas eu acho que um conselho bom é: se você escolher ser um especialista, você precisa saber muito da sua especialidade e um pouquinho do resto tudo. Você não pode ser 100% especialista, você tem que ter uma visão do todo.

Então, talvez um bom caminho seja você estudar um pouquinho de UX. Acho que todo mundo que trabalha com web, desde o cara que está fazendo só back-end, só construindo o banco de dados lá, tem que saber um pouquinho de UX. Porque ele tem que saber onde o trabalho dele vai impactar no final. Então, você estudar um pouquinho vai te ajudar a tomar uma decisão, se você quer ser um generalista ou um especialista, se você quer se tornar um devigner, ou se você quer ser um front-end engineer, um especialista nisso. Saber um pouquinho de tudo não vai fazer mal.

SRA. BEATRIZ ROCHA: É verdade.

SRA. MIRIAM VON ZUBEN: Vai muito...

SRA. BEATRIZ ROCHA: Pode falar.

SRA. MIRIAM VON ZUBEN: Vai muito do perfil de cada um, né? Porque informação, hoje em dia, é o que mais tem, né? Então assim, você pode até ficar perdido nesse mar de informações. Então, assim, acho que vai do perfil de cada um. Mas para qualquer um que esteja nessa área de tecnologia é estudar, estudar, estudar o tempo inteiro, né? Não tem como a gente ficar parado.

SRA. BEATRIZ ROCHA: Muito bem. Obrigada. Eu creio que vocês acrescentaram muito, não só ao pessoal que está assistindo, mas para mim também. Foi muito bom ouvir experiência de vocês, o que vocês esperam da HTML para o futuro também.

E agora, eu gostaria de chamar o Reinaldo para encerrar esse debate, esse painel. E, obrigada, galera. E é isso.

SR. REINALDO FERRAZ: Obrigado, Bia. Poxa, é até difícil fazer um encerramento, né? Acho que depois de tantas conversas, de tantas falas, né? Eu que trabalhei já tanto tempo também... trabalho há tanto tempo com web, com HTML, ouvir a fala desses especialistas que a gente... alguns a gente conheceu no curso, outros já acompanhavam o trabalho já há muito tempo. Então, foi muito bom, acho que foi um evento você muito bacana e eu acho que talvez a gente não precise esperar mais dez anos para fazer o próximo, para falar sobre as evoluções tecnológicas, né? Do jeito que a coisa evolui tão rápido, talvez, em pouco tempo a gente consegue fazer isso.

Mas, para encerrar, eu queria só fazer alguns agradecimentos, né? Acho que primeiramente agradecer toda a equipe de comunicação, evento, de sistemas do NIC.br, que conseguiu montar toda essa estrutura, por trás de tudo isso daqui, tem um monte de gente trabalhando para fazer isso aqui acontecer redondinho, né? Agradecer também ao pessoal, os intérpretes de Libras e da estenotipia também que garantem a acessibilidade aqui do evento, né? Agradecer toda a equipe do Ceweb também, que batalhou bastante para poder montar essa atividade e pensar em temas. E agradecer, assim, principalmente aos participantes desse painel, que é só uma parte dessa turma que participou desse curso de HTML5, né? Eu acompanhei pelo YouTube, tem muita gente que estava no curso e que também estava assistindo, né? A audiência... foi uma grande audiência. Acho que também, principalmente, por essas pessoas que participaram.

Então, eu queria agradecer a vocês que estão participando aqui, a vocês que participaram daquele curso há dez anos atrás (sic). E para quem está aqui fazendo... lutando para que essa web continue sendo essa tecnologia aberta e que a gente possa, como foi falado nesses painéis, que a gente possa pensar a web como a grande plataforma de desenvolvimento dessa nossa... de um ambiente tecnológico, né? Que a web saiu do computador, ela saiu da tela de um computador e hoje ela está em praticamente todos os dispositivos, né? Então, eu agradeço muito aqui, a vocês que estão participando desse evento e, bom, o Vagner entrou aqui. Quer falar alguma coisa?

SR. VAGNER DINIZ: Não.

SR. REINALDO FERRAZ: Ah, tá. Só para a gente...

SR. VAGNER DINIZ: Só para começar a organizar a homenagem ao Reinaldo, não é isso que está a demanda no chat?

SR. REINALDO FERRAZ: Ah.

SR. VAGNER DINIZ: Parabéns, Reinaldo, pelo evento. Parabéns pela montagem e agregar toda a galera aí. E toda a equipe do Ceweb também, né? Meus parabéns pelo evento. Obrigado à infraestrutura do NIC, a equipe de comunicação, e a todos os palestrantes, também quero agradecer muitíssimo, valeu muito.