Minha rotina matinal inclui ler as últimas notícias, verificar meus e-mails e redes sociais, ouvir podcasts enquanto corro em minha esteira, consultar minha agenda, o trânsito e finalmente chamar um serviço de carona para ir até o escritório.

Tudo isto parece comum não fosse o fato de eu ser cego e só ter acesso a este tipo de recursos graças a tecnologias assistivas como o leitor de telas que utilizo em meu computador e smartphone.

Não estou sozinho. Acredita-se que quase 1 bilhão de pessoas ao redor do mundo possui algum tipo de deficiencia seja ela física, sensorial ou cognitiva.

Mas qual o papel de nós desenvolvedores nesta história?

As plataformas móveis (iOS e Android) oferecem em sua API base recursos que permitem com que tecnologias assistivas tais como leitores de tela, aparelhos auditivos, ampliadores de tela e esquema de cores especiais apoiem o usuário na execução de suas atividades.

Normalmente estas APIs passam despercebidas pois quando utilizamos controles padrão da plataforma não temos de nos preocupar com elas já que acessibilidade é parte do guia de interface de cada plataforma.

Mas existem propriedades especiais que podemos utilizar para tornar a experiência ainda mais agradável, principalmente em controles alternativos ou na criação de novos tipos de interação.

O Xamarin.Forms oferece uma camada de abstração sobre as APIs nativas e é sobre elas que falarei neste artigo, mas maiores informações podem ser encontradas na documentação oferecida pela Apple e pela Google.

## AutomationProperties

AutomationProperties é uma propriedade anexada que permite definirmos como um controle é anunciado pelas tecnologias assistivas.

Através de suas propriedades podemos definir sua visibilidade, identificador e texto de ajuda tornando ainda mais claro como funciona a interação com este controle.

Suas principais propriedades são:

* IsInAccessibleTree
* Name
* HelpText
* LabeledBy

### IsInAccessibleTree

Esta propriedade diz se um controle deve ou não ser percebido pela tecnologia assistiva.

Quando o seu valor está definido para verdadeiro a tecnologia assistiva passa a anunciar este controle e permite sua interação com o mesmo. Os controles mais comuns já vem com este valor definido como verdadeiro por padrão como por exemplo Label, Entry, Button, etc.

Já controles como BoxView ou Image não são interativos por padrão, e tem o valor definido para falso.

Mas caso queiramos interagir com estes elementos podemos alterar o valor para verdadeiro e fornecendo as demais propriedades tornamos este controle acessível a todos.

### Name

A propriedade AutomationProperties.Name permite personalizar o nome do controle para a tecnologia assistiva. Esta é a primeira informação anunciada e deve identificar o contexto deste controle para o usuário.

Em uma página de Login podemos ter três controles: Um entry para o nome de usuário, um entry para a senha e um botão para efetuar login.

O botão recebe automaticamente como Name o valor “Entrar” que é o valor da sua propriedade Text. Já no caso dos dois Entries devemos fornecer esta propriedade como “Usuário” e “Senha” para que o usuário de tecnologia assistiva saiba do que se trata.

### HelpText

A propriedade HelpText nos permite dar maiores instruções ao usuário de como ele deve interagir com o controle.

No caso de uma página onde pedimos o número de Seguro Social (SSN) do usuário, podemos ter a informação de que é necessário informar apenas os números e não os separadores.

Em outras situações podemos pedir para o usuário entrar com os dados em um formato específico como data, valores monetários entre outros. A propriedade HelpText também serve para darmos maior contexto quando a interação não é tão óbvia como por exemplo com o uso de gestos como Swipe.

### LabeledBy

Esta propriedade permite que utilizemos um outro controle, normalmente um Label, para descrever o nosso controle.

Imagine que tenhamos um Image e logo abaixo alguma informação sobre a mesma. Podemos utilizar a propriedade LabeledBy para que a tecnologia assistiva use o texto da Label para identificar o elemento.

## Resumindo

As plataformas móveis evoluíram muito no que tange a acessibilidade, mas ainda existe uma longa jornada já que ser acessível não é apenas utilizar estas propriedades mas sim tornar o aplicativo usável por todos removendo toda e qualquer barreira física, sensorial ou cognitiva.

Por fim deixo aqui algumas recomendações:

* Apesar da WCAG 2.1 ter trazido considerações importantes sobre dispositivos móveis ela ainda tem um foco muito grande em Web por isto é mais recomendado se basear nos guias de interação de cada plataforma;
* Temos poucas ferramentas de validação de acessibilidade para mobile, mas elas são mais uma orientação ao desenvolvedor do que uma garantia de acessibilidade ao usuário final. Testes manuais, preferencialmente por pessoas com deficiência comprovadamente trazem melhores resultados;
* Introduza a acessibilidade em todo o ciclo de vida do seu produto, desde a concepção da funcionalidade até a sua entrega em produção. Da mesma forma que o processo de qualidade de software, quanto mais tarde introduzimos os requisitos de acessibilidade mais caro e mais difícil sua implementação;

Alexandre Costa é desenvolvedor com deficiencia visual apaixonado por tecnologia e evangelista da acessibilidade e inclusão. Microsoft MVP em Development Technologies, TDC Rock Star e Chapter Lead na ArcTouch Mobile & Connected Experiences.