

---

# Estratègies de desenvolupament

---

PID\_00245729

Jordi Rovira Piñol

---

Temps mínim de dedicació recomanat: 3 hores

---





# Índex

<b>1. Els factors d'una estratègia de desenvolupament.....</b>	<b>5</b>
1.1. Identificació del context tècnic d'un producte digital .....	5
1.2. Optimització de les característiques tècniques .....	6
1.2.1. Mides de pantalla .....	6
1.2.2. Model d'interacció .....	8
1.2.3. Especificitats tècniques .....	8
1.3. Consistència entre dispositius .....	9
<b>2. Tipus d'estratègies de desenvolupament.....</b>	<b>10</b>
2.1. Llocs web mòbils .....	10
2.2. Aplicacions mòbils .....	11
2.2.1. Aplicacions mòbils natives .....	12
2.2.2. Aplicacions mòbils híbrides .....	12
2.3. Aplicacions web .....	13
2.3.1. Disseny fix o estàtic .....	13
2.3.2. Disseny líquid/fluid .....	13
2.3.3. Disseny adaptable .....	14
2.3.4. Disseny <i>responsive</i> .....	15
<b>3. L'aplicació del disseny <i>responsive</i>.....</b>	<b>16</b>
3.1. Conceptes clau del disseny <i>responsive</i> .....	16
3.1.1. Retícules fluides .....	17
3.1.2. Punts d'interrupció .....	17
3.1.3. <i>Media queries</i> .....	19
3.2. El flux de treball <i>responsive</i> .....	22
<b>4. Elecció d'una estratègia de desenvolupament.....</b>	<b>27</b>
4.1. Aspectes que cal tenir en compte .....	27
4.2. Per on començar? .....	29
<b>Bibliografia.....</b>	<b>33</b>



## 1. Els factors d'una estratègia de desenvolupament

Les tècniques i eines utilitzades en el desenvolupament de productes digitals han experimentat una evolució notable en els últims anys. En un breu espai de temps hem vist com les metodologies que acompanyaven determinades tecnologies anaven caient en desús suplantades per procediments nous. Un canvi impulsat per la recerca de respostes als reptes emergents d'un sector altament dinàmic. En un escenari tan volàtil, per a garantir un procés de disseny fluid i adequat a les condicions particulars d'ús, resulta clau per al dissenyador d'experiències digitals prestar atenció al conjunt de factors globals que afecten el desenvolupament del projecte.

### 1.1. Identificació del context tècnic d'un producte digital

Considerar l'estratègia de desenvolupament abans de començar el disseny d'un producte és clau per a analitzar les condicions tècniques amb les quals un producte digital serà construït. Aquestes condicions afectaran l'experiència d'usuari en gran mesura, i per tant, no poden ser obviades. Hi ha bàsicament quatre elements tècnics que cal tenir en compte, les decisions entorn als quals obriran possibilitats o bé afegiran limitacions als recursos de disseny que poden ser implementats:

- L'entorn d'accés (navegador o aplicació mòbil).
- El llenguatge de codi utilitzat.
- La multiplicitat i característiques del contingut.
- La modalitat de descàrrega.

En l'actualitat destaquen tres tipologies de desenvolupament: els **llocs web mòbils** (*m-dot sites*), el **desenvolupament web** i les **aplicacions mòbils** (*apps*) natives o híbrides. Aquestes tres àrees cobreixen (en totes les seves combinacions) les necessitats dels ecosistemes formats pels dispositius més habituals: telèfon, tauleta i ordinador.

D'altra banda, si bé alguns dels dispositius portables<sup>1</sup> existents al mercat ja funcionen també a través de botigues d'aplicacions mòbils natives (i, per tant, poden formar part de les estratègies del grup anterior), la major part d'experiències destinades a dispositius fora del grup nucli requereixen encara plantejaments tècnics fets a mida.

<sup>(1)</sup> wearables

## 1.2. Optimització de les característiques tècniques

Els patrons d'ús multidispositiu com ara la complementarietat, la coherència o la consistència, entre d'altres, generen sinergies entre dispositius i posen de relleu reptes tècnics que afecten de manera directa el disseny de l'experiència.

Així doncs, hem d'observar de manera global les característiques tècniques del conjunt de dispositius per als quals volem crear el producte per a generar una experiència que tregui el màxim partit d'aquestes condicions. Alhora, hem de buscar la millor resposta a les possibles limitacions o incompatibilitats del context específic de l'experiència.

Les diferències entre els diversos sistemes operatius, la varietat de modalitats d'interacció o els requisits tècnics de cada tipus de dispositiu, entre altres circumstàncies, són variables que afectaran l'estratègia de desenvolupament.

Elements com el llenguatge de codi utilitzat, la plataforma de desenvolupament escollida, o fins i tot la metodologia, poden tenir un impacte en el disseny del producte que limiti l'aplicació de determinades funcionalitats o redueixi el marge de personalització de la interfície d'usuari.

Professionals del sector de la usabilitat com Karen McGrane han posat de relleu en les seves investigacions dels últims anys el fet que els usuaris esperen trobar-se el mateix contingut i la mateixa funcionalitat durant la seva interacció amb productes i serveis digitals, independentment del dispositiu d'accés i del context en què es trobin. Per aquest motiu, les velles estratègies basades en la creació de funcionalitats i/o continguts diferents per a cada tipus de dispositiu han caigut ràpidament en desús i han donat lloc a experiències multidispositiu més coherents, que augmenten les possibilitats de satisfer els objectius de l'usuari.

«El tipus de dispositiu i les seves característiques tècniques són, per ara, les úniques variables d'informació fiables que podem obtenir del dispositiu, sent aquestes variables insuficients per a justificar la necessitat de diferenciar l'experiència oferta en cada dispositiu.»

Karen McGrane, «Content in a Zombie Apocalypse» (2014).

En conseqüència, la mida de la pantalla, la modalitat d'interacció i les especificacions tècniques (elements intrínsecs del dispositiu) són els factors clau a tenir en compte a l'hora d'optimitzar el desenvolupament tècnic d'un producte digital multidispositiu.

### 1.2.1. Mides de pantalla

Al mercat actual disposem d'una àmplia gamma de dispositius amb una gran varietat de mides de pantalla, factor que introdueix complexitat al plantejament d'un disseny consistent entre dispositius. A més, al marge de les mides

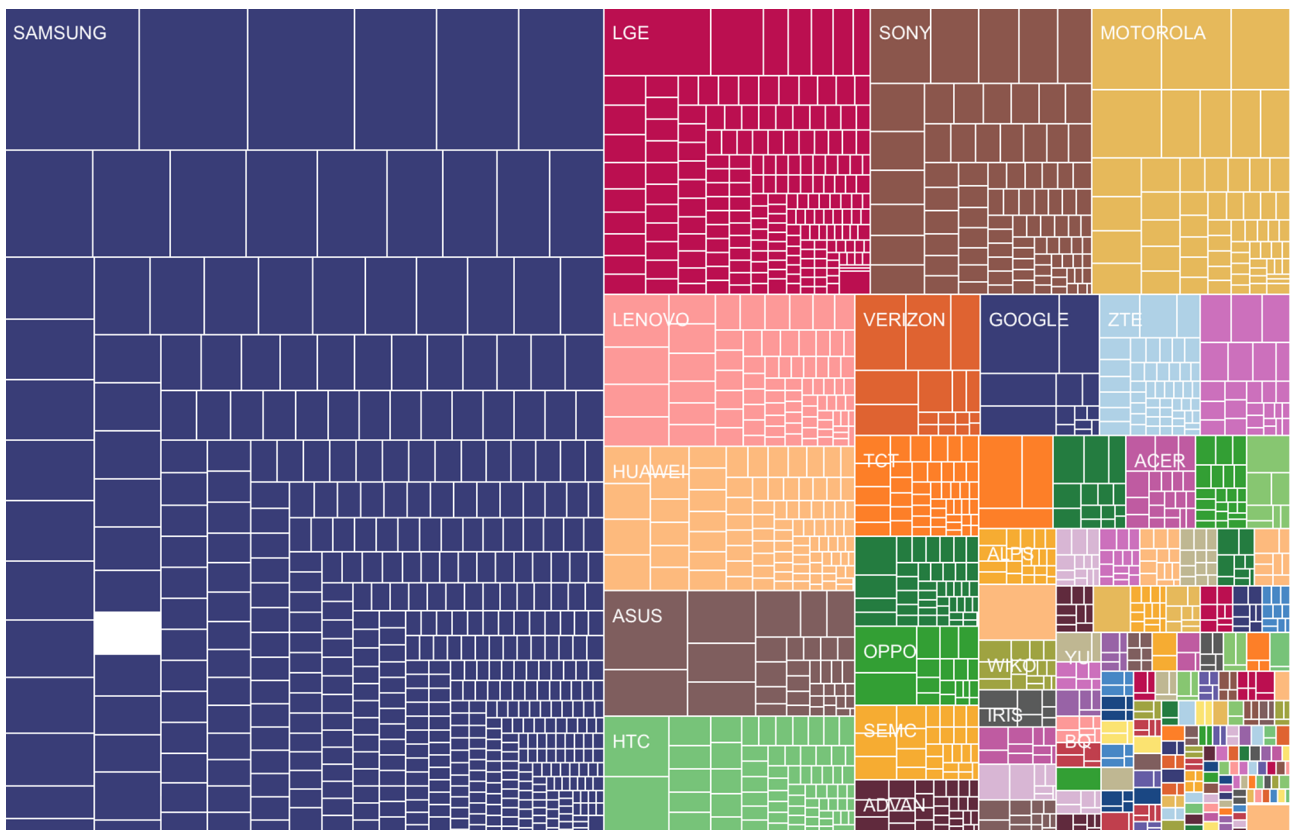
#### Vegeu també

Els patrons d'ús multidispositiu es tracten en l'apartat 3 del mòdul didàctic «Experiència d'usuari i estratègies de disseny» d'aquesta assignatura.

de la pantalla, hem de considerar també aquells dispositius que permeten l'ús d'un navegador. En aquests casos, la possibilitat de canviar la mida del navegador introdueix encara més complexitat pel que fa a mides i versions del contingut.

En el gràfic següent, d'Open Signal, veiem la fragmentació de mercat tan sols per als dispositius Android, i ens transmet una idea clara de la complexitat que suposa el disseny d'interfícies optimitzades per l'enorme dispersió de dispositius del mercat actual.

Fragmentació de marques i mides de pantalla en dispositius Android



Font: thenextweb.com

Les mides de pantalla dels dispositius mòbils d'Apple o Microsoft, per contra, estan bastant més unificades. En qualsevol cas, la fragmentació de mides de pantalla té, com hem reiterat, un fort impacte en el disseny i desenvolupament de productes optimitzats per al seu ús en multidispositius. Si ampliem aquesta anàlisi a altres tipus de dispositius menys estandarditzats com els portables, televisors connectats o l'Internet de les coses en general, la influència del factor de la fragmentació en el disseny d'experiències és encara més evident.

### 1.2.2. Model d'interacció

En el desenvolupament d'una experiència d'usuari multidispositiu s'han de tenir en compte les possibilitats d'interacció de tots els dispositius de l'ecosistema i planejar l'accés al contingut en funció de les modalitats d'interacció disponibles.

Cada tipus o família de dispositius combina diferents modalitats d'interacció. Per exemple, els telèfons iPhone ofereixen versatilitat en la combinació de l'ús gestual de la pantalla tàctil amb les ordres de veu i el botó físic «Home» com a drecera per a la pantalla principal. En canvi, altres dispositius, com la majoria dels ordinadors de sobretaula, no disposen de pantalles tàctils i vehiculen la interacció amb perifèrics com ratolins, teclats, llapis o tauletes gràfiques, entre d'altres.

A continuació enumerem les principals formes d'interacció actuals:

- Botons físics.
- Pantalla tàctil i gestos.
- Reconeixement de veu.
- Elements accessoris de maquinari: ratolins, teclats, llapis, etc.

#### Vegeu també

Les modalitats d'interacció que tenen una relació directa amb la interfície d'usuari s'estudien en el mòdul didàctic «Interfícies i interacció» d'aquesta assignatura.

### 1.2.3. Especificitats tècniques

L'observació de les particularitats de cada dispositiu és crucial per a integrar-les de forma apropiada a l'ús del producte i treure'n així el màxim partit. A tall d'exemple, un rellotge intel·ligent i un televisor connectat presenten característiques tècniques molt diferents que complementen i/o ofereixen continuïtat a la resta de dispositius de l'ecosistema. A més de la pantalla, els sensors són els elements diferencials en un dispositiu connectat, entre els quals es troben:

- GPS.
- Acceleròmetre.
- Giròmetre.
- Etc.



### **1.3. Consistència entre dispositius**

Al marge de l'estratègia de desenvolupament triada, hem de mantenir la consistència en l'experiència prevista per a tots els dispositius per a intentar aportar qualitat i professionalitat al producte i evitar que l'usuari tingui una vivència fragmentada i inconnexa. En aquest sentit, hi ha tres àrees a les quals cal prestar atenció: funcionalitat, contingut i llenguatge visual.

#### **1) Funcionalitat**

És recomanable que les funcionalitats principals del producte es mantinguin en tots els dispositius, i funcionin de la mateixa manera o de la manera més semblant possible. El disseny ha d'aportar flexibilitat al producte de manera que cobreixi les particularitats de cada dispositiu sense generar friccions o diferències notables en el seu funcionament.

#### **2) Contingut**

L'elaboració d'una arquitectura de la informació comuna per a tot l'ecosistema (o el més semblant possible) facilita a l'usuari reconèixer el mateix contingut i reprendre'n el consum amb comoditat.

#### **3) Llenguatge visual**

La coherència en el llenguatge visual entre dispositius ha de fer possible que l'usuari reconegui fàcilment el producte, la seva estructura i el seu funcionament. La utilització d'una quadrícula base permetrà optimitzar i igualar variables com la grandària dels elements en pantalla (imatges, botons, etc.), així com la mida de la font o l'espaiat entre tots aquests elements.

## 2. Tipus d'estratègies de desenvolupament

Com hem dit, en l'actualitat destaquen tres tipus d'estratègies de desenvolupament: els llocs web mòbils, les aplicacions mòbils i el desenvolupament web. A continuació tractarem aquests tres tipus d'estratègies.

### 2.1. Llocs web mòbils

L'estratègia dels llocs web mòbils<sup>2</sup> té com a objectiu aconseguir llocs web<sup>3</sup> que siguin cent per cent *mobile-friendly*, és a dir, que el seu funcionament estigui optimitzat per a aquest tipus concret de dispositiu.

<sup>(2)</sup> *m-dot sites*

<sup>(3)</sup> *websites*

S'accedeix a les webs mòbils des d'un navegador mòbil i des d'un enllaç d'Internet URL. El contingut d'aquesta web és únic, i ha de ser mantingut de manera individual. Sovint resulta difícil diferenciar a nivell visual una web optimitzada per a mòbil d'una aplicació mòbil híbrida.

Les webs mòbils estan basades en tecnologies web com HTML i CSS, i la seva producció segueix els principis del desenvolupament web. No depenen de botigues d'aplicacions<sup>4</sup> i es poden actualitzar en qualsevol moment. El seu comportament pot ser sovint impredecible atesa la gran varietat de dispositius existents i que el marge de coordinació amb les funcionalitats natives del dispositiu és curt.

<sup>(4)</sup> *appstores*

#### Lectura recomanada

El portal de Google dedicat a *webmasters* ofereix una guia completa destinada al desenvolupament de llocs web mòbils *mobile friendly*:

**Google.** «Mobile friendly websites» (accessible en línia).

#### HTML i CSS

##### HTML

Sigla en anglès d'*hypertext markup language* (llenguatge de marques d'hipertext). Fa referència al llenguatge de marcatge per a l'elaboració de pàgines web. Aquest llenguatge defineix el contingut d'una pàgina web (text, imatges o vídeos, entre altres). És l'estàndard que s'ha imposat en la visualització de pàgines web i és el que tots els navegadors actuals han adoptat (Font: Wikipedia).

##### CSS

Sigla en anglès de *cascading stylesheets* (fulls d'estil en cascada). És un llenguatge de disseny gràfic per a definir i crear la presentació de documents estructurats, escrits generalment en HTML o altres llenguatges de marcatge. CSS està dissenyat principalment per a marcar la separació del contingut del document i la forma de presentació d'aquest en característiques com ara les capes (*layouts*), els colors i les fonts. El seu ús proveeix més flexibilitat i control en l'especificació de característiques presentacionals d'un document (font: Wikipedia).

Actualment, el sector digital professional considera en general aquesta estratègia en desús i poc adequada a causa de la dificultat de garantir que el lloc web desenvolupat en el present continuï estant optimitzat per a futurs dispositius.

Hi ha algunes excepcions en què aquest tipus de desenvolupament segueix considerant-se una opció convenient, per exemple, els projectes en què hi hagi limitacions en un CMS<sup>5</sup> ja existent, i per descomptat, en aquells casos en què aquest enfocament sigui requerit de manera específica pel client.

<sup>(5)</sup> *Content management system*

## 2.2. Aplicacions mòbils

Les aplicacions mòbils<sup>6</sup> són productes fets a mida, adaptats al sistema operatiu del dispositiu i que permeten un major grau de personalització, de manera que generen experiències en què l'atenció dels usuaris està centrada exclusivament en el producte digital. En el desenvolupament d'aplicacions mòbils hem de tenir en compte les versions anteriors de l'*app*, així com la compatibilitat amb dispositius i/o sistemes operatius anteriors.

<sup>(6)</sup> *apps*

Cada sistema operatiu (iOS, Android, Windows Phone, etc.) disposa de les seves pròpies GUI<sup>7</sup>, a les quals els usuaris ja hi estan acostumats. Seguir les directrius d'aquesta interfície és recomanable per a aconseguir una major integració de l'aplicació mòbil en els patrons d'interacció habituals de l'usuari amb el dispositiu.

<sup>(7)</sup> GUI és la sigla anglesa per a *interfície gràfica d'usuari*.

Hi ha tecnologies de desenvolupament com Xamarin o React Native, entre altres, que permeten l'ús del codi multiplataforma, amb el consegüent estalvi de temps i recursos que això comporta.

Hi ha dos tipus d'aplicacions mòbils, les natives i les híbrides. Per a determinar quin tipus d'aplicació mòbil és la més recomanable, s'han de valorar les variables següents:

- Existència de requeriments específics per part de l'usuari o client.
- Ús de funcionalitats natives del dispositiu.
- Necessitat d'accedir al contingut fora de línia<sup>8</sup>
- Existència de limitacions pressupostàries.
- Exigència de mesurar el trànsit d'entrada al producte<sup>9</sup> (la monitorització de dades és més senzilla en el cas de les aplicacions mòbils natives).

<sup>(8)</sup> *offline*.

<sup>(9)</sup> *direct traffic*

### 2.2.1. Aplicacions mòbils natives

Mitjançant el desenvolupament d'aplicacions mòbils natives s'aconsegueix una experiència amb un grau d'optimització màxim. Amb aquest tipus de desenvolupament, l'accés a les funcionalitats natives del dispositiu, així com a les API<sup>10</sup> és molt més senzill, si bé requereix l'ús del codi específic de cada plataforma o sistema operatiu.

<sup>(10)</sup>API és la sigla anglesa per *interfície de programació d'aplicacions*.

Una aplicació mòbil nativa pot funcionar sense una connexió activa a Internet, ja que les dades són emmagatzemades de manera local al dispositiu. Les aplicacions mòbils natives sovint ofereixen una millor experiència, amb una major integració de la funcionalitat del dispositiu, i són tècnicament menys restrictives. La seva distribució és més fàcil, ja que es fa a través de les botigues d'aplicacions. Tanmateix, aquest fet restringeix la introducció d'actualitzacions i la publicació de noves versions de l'aplicació mòbil a la botiga d'aplicacions.

El desenvolupament d'aquest tipus d'aplicacions mòbils pot ser car, ja que depèn dels llenguatges de programació que cada plataforma admet. Cada plataforma està basada en tecnologies i marcs de desenvolupament<sup>11</sup> diferents. Per a optimitzar-ne l'ús i l'experiència s'han de tenir en compte els patrons d'interacció, gestos i guies visuals de cada plataforma. Instagram, Facebook, Evernote o Twitter són exemples d'aplicacions mòbils natives amb desenvolupaments optimitzats per a cada plataforma o sistema.

<sup>(11)</sup>*frameworks*

### 2.2.2. Aplicacions mòbils híbrides

Una aplicació híbrida és una barreja entre una aplicació web i una aplicació mòbil nativa. Es tracta d'una alternativa de desenvolupament raonable per a petits-mitjans projectes multiplataforma. Utilitza tecnologies web, això vol dir que és pràcticament una aplicació web que es distribueix a través de les botigues d'aplicacions.

Quan es desenvolupa de forma híbrida, l'aplicació mòbil s'insereix en un contenidor<sup>12</sup> mitjançant l'ús d'HTML5 i Javascript amb l'objectiu de poder utilitzar les funcionalitats natives del dispositiu.

<sup>(12)</sup>*wrapper*

Aquesta modalitat implica el manteniment de menys versions, si bé la reproducció d'interaccions similars a les d'una aplicació mòbil nativa pot comportar més temps de desenvolupament. Aquest enfocament permet crear aplicacions unificades, que són similars a les aplicacions mòbils natives, per a totes les plataformes. No obstant això, no és recomanable tractar d'imitar el funcionament d'una aplicació mòbil nativa mitjançant tecnologies web, ja que això sol generar expectatives que, si no es compleixen, poden conduir fàcilment a una experiència d'usuari pobre.

## 2.3. Aplicacions web

Amb els anys, el disseny web ha anat evolucionant i adquirint complexitat al mateix temps que l'oferta de dispositius ha anat creixent. Des dels dissenys HTML estàtics dels inicis d'Internet fins a l'actual disseny *responsive*, les tècniques de disseny i desenvolupament de pàgines web han anat incorporant fluïdesa i adaptabilitat, seguint les demandes de la tecnologia.

Un bon disseny web és aquell que considera dins del seu desenvolupament tant els elements bàsics del disseny gràfic (la diagramació, el color, els gràfics i les fonts) com els fonaments tècnics (estructura, compatibilitat, funcionalitat i interactivitat), tots dos necessaris per a crear un impacte visual al costat d'una experiència d'usuari òptima.

Aportar el grau de flexibilitat i modularitat que permeti optimitzar l'aspecte i comportament del contingut a cada una de les mides de pantalla seria el model ideal, i l'RWD<sup>13</sup> és la tècnica que avui en dia cobriria les necessitats del major nombre d'escenaris i dispositius.

No obstant això, el cost i la complexitat d'aquest tipus de disseny comporta que sovint s'opti per la combinació de diverses tècniques. És el cas, per exemple, de projectes ja existents que requereixen una actualització i/o migració de dades progressiva. A continuació es descriuen quatre dels principals enfocaments que se segueixen en el disseny web.

### 2.3.1. Disseny fix o estàtic

Un disseny estàtic o d'amplada fixa utilitza una mida de pàgina predefinida que no canvia basada en l'ample del navegador. Aquest és el sistema mitjançant el qual es desenvolupaven les pàgines web tradicionalment fins que es va introduir el concepte de disseny *responsive*.

Per a webs desenvolupades seguint aquest mètode, cada dispositiu mostrarà el disseny fix de forma diferent, amb la qual cosa l'aparença final és en certa manera imprevisible. Aquest tipus de disseny s'evita en productes de nou desenvolupament ja que la seva adopció implica la necessitat de desenvolupar versions separades per a cada dispositiu per tal de garantir un visionat correcte.

### 2.3.2. Disseny líquid/fluid

En aquest tipus de disseny, s'optimitza la posició i aspecte del contingut en funció de l'ample de pàgina del navegador d'accés. Per a poder mantenir el posicionament dels elements de pàgina coherent entre dispositius, es fa ús d'unitats relatives (percentatges) en lloc d'unitats fixes (píxels), d'aquesta manera la posició del contingut sempre està en funció de l'entorn de visionat.

#### Lectura complementària

Per a més informació sobre les característiques que ha de complir un bon disseny web, es pot consultar l'enllaç següent:

«Disseny web» (accessible en línia), *Wikipedia*.

<sup>(13)</sup>RWD és la sigla anglesa de *responsive web design*.

Si bé aquesta tècnica no requereix tanta feina com un disseny *responsive*, cal anar amb compte amb els escenaris extrems. El visionat en finestres molt estretes (navegadors reduïts) pot aixafar els continguts generant un efecte visual poc recomanable, mentre que el visionat en finestres exageradament amples (navegador eixamplat) provocaran l'efecte contrari, amb línies de text massa llargues.

### 2.3.3. Disseny adaptable

Una web amb disseny adaptable mostrarà pàgines diferents depenent del tipus de dispositiu des del qual es visiti la pàgina. Així doncs, la pàgina per a un telèfon mòbil serà totalment diferent i separada de la que es veurà des d'un ordinador personal.

El major avantatge de les pàgines adaptables és el grau de personalització per a cada dispositiu i la capacitat de carregar només el contingut específic per al dispositiu d'accés, de manera que s'elimina la càrrega d'altres continguts que no siguin consultats en el mateix moment. No obstant això, el manteniment de diferents versions d'una mateixa web incrementa notablement el volum de treball.

A les pàgines adaptables s'usen fulls d'estil CSS combinats amb ordres de codi (o *media queries*) per a detectar l'ample del navegador i alterar el disseny mostrat en concordança. Aquestes *media queries* poden utilitzar unitats fixes (píxels) o relatives (percentatges) a diferència del disseny fix, oferint així múltiples opcions de visualització. Els estàndards actuals del sector recomanen migrar a unitats relatives per a aportar una major flexibilitat.

A tall d'exemple, un disseny de pàgina adaptable podria contenir *media queries* amb instruccions del tipus: «Si el navegador té 500 píxels d'ample, estableix la mida del contenidor principal a 400 píxels d'amplada. Si el navegador és de 1.000 píxels d'amplada, estableix la mida del contenidor principal a 960 píxels d'ample», etc.

Més enllà dels amples de pàgina, els *media queries* poden dictar també el canvi de lloc d'elements en la pàgina o la seva supressió. Per exemple, un disseny de dues columnes podria convertir-se en una única columna si el navegador és massa estret.

Els dissenys adaptables requereixen en general menys temps de desenvolupament que els dissenys *responsive*, i són d'ajuda en aquells casos en què el presupost és un factor determinant. Aquesta tècnica s'utilitza sovint com a solució temporal per a actualitzar antics dissenys fixos de manera que puguin admetre dispositius mòbils.

#### Media queries

Per a més informació sobre les *media queries*, es pot consultar l'enllaç següent:  
«CSS media queries» (accessible en línia)

Un disseny adaptable dóna més control sobre el disseny ja que el nombre d'estats a la pàgina és limitat. En general és una solució més pràctica, més barata en implementació i més fàcil de testejar que el disseny *responsive*. Es considera la versió barata del Responsive Web Design ja que els resultats encara mantenen la qualitat suficient requerida en aquells projectes on no hi ha molts recursos.

### 2.3.4. Disseny *responsive*

El disseny *responsive* és una estratègia de disseny i desenvolupament web en què el contingut es reorganitza en funció de la mida de la pantalla. Els llocs *responsive* carreguen tot el codi cada vegada, al marge del dispositiu, de manera que el temps de càrrega pot augmentar especialment en dispositius mòbils amb connexions lentes. Una manera eficient de comprovar si un lloc és *responsive* és canviant la mida del navegador i observar si el contingut es reorganitza o no.

Una pàgina web *responsive* fa ús d'unitats relatives i *media queries*, combinant els principis de disseny líquid i adaptatiu. Així doncs, a mesura que el navegador augmenta o disminueix en ample, el disseny respon de manera flexible tal com ho faria un disseny líquid, amb la diferència que si el navegador va més enllà de certs amplex fixats mitjançant *media queries*, el disseny canviaria de manera més dramàtica reposicionant determinats elements per a acomodar-los a aquests nous amplex de pàgina.

Segons la popularment acceptada estratègia *mobile first* de Luke Wroblewski, és recomanable que els dissenys *responsive* es desenvolupin començant per la versió mòbil, de manera que es crea el disseny mòbil primer i a mesura que el navegador s'expandeix a altres dispositius més amples (tauleta, ordinador), el dissenyador treballa en la redistribució dels elements. Aquest procediment tendeix a contribuir a la creació de millors experiències, ja que obliga el dissenyador a depurar l'experiència i optimitzar la presentació i interacció juntament amb el contingut. A nivell visual, a més, resulta més simple expandir la distribució d'elements en pantalla a formats més grans (amb més espai), que intentar encaixar una gran quantitat d'elements en un espai reduït.

Per al dissenyador, la tasca de crear webs *responsives* implica algunes dificultats afegides, a més dels reptes tècnics inherents. Un disseny *responsive* pot generar fàcilment centenars d'estats diferents, la qual cosa dificulta predir com es veurà el disseny final. De la mateixa manera, testejar un disseny *responsive* resulta molt més complicat i imprecís.

#### Lectura complementària

Per a més informació sobre l'estratègia *mobile first* de Luke Wroblewski, es pot consultar l'obra següent:

**Luke Wroblewski (2011).** «Mobile first» (accessible en línia). *A book apart*.

### 3. L'aplicació del disseny *responsive*

El terme *responsive web design* va ser encunyat per Ethan Marcotte i és la pràctica d'utilitzar retícules fluïdes, imatges flexibles, punts d'interrupció<sup>14</sup> i *media queries* per a optimitzar una pàgina web de manera progressiva per a diferents contextos de visualització. Com ja hem assenyalat, es tracta d'una estratègia de disseny i desenvolupament web per a atendre l'enorme fragmentació de dispositius presents en l'ecosistema.

(14) *breakpoints*

Marcotte va ser pioner en el sector a l'hora de definir un marc de treball amb el qual crear productes digitals que poguessin donar resposta a necessitats canviants. En l'actualitat, el desenvolupament web *responsive* és l'enfocament més recomanat i usat, ja que ofereix una major cobertura de dispositius mantenint la consistència en l'experiència.

Tal com hem vist, el desenvolupament d'un lloc web *responsive* no és una tasca fàcil ja que cal considerar i testejar múltiples dispositius i mides de pantalla. En els últims temps han sorgit marcs de desenvolupament (*responsive frameworks*) que utilitzen CSS i faciliten el treball en gran mesura. Alguns d'aquests marcs de desenvolupament, com Bootstrap o Foundation, requereixen un bon coneixement d'HTML5 i CSS.

Per ara no hi ha un consens en el sector pel que fa a la nomenclatura en espanyol dels termes *adaptive* i *responsive* (el terme *responsive* sovint es tradueix com *adaptable*) i el seu ús erroni pot generar certa confusió. En el context d'aquesta assignatura, ens cenyirem a les descripcions aportades amb anterioritat en aquests materials. El dissenyador Sandijs Ruluks publica al seu popular blog Froot, un article, «9 Basic principles of responsive web design», que és de gran ajuda a l'hora de desxifrar les diferències i complementaritats entre les dues tècniques de desenvolupament.

#### Lectura recomanada

L'article de Sandijs Ruluks sobre les diferències i complementaritats entre les dues tècniques de desenvolupament es pot consultar a:

**Sandijs Ruluks** (2014). «9 Basic principles of responsive web design» (accessible en línia). *Froot - Open design blog*.

#### 3.1. Conceptes clau del disseny *responsive*

L'ús del marc *responsive* implica en si mateix un replantejament de la forma com abordem el disseny. La multiplicitat de formats, mides i dispositius per als quals aquesta tècnica aporta solucions requereix una preparació prèvia dels escenaris per als quals anem a dissenyar. El coneixement de la terminologia i els conceptes bàsics del *responsive web design* és bàsic per a dur a terme aquesta preparació, que ens ha de permetre dissenyar de manera ordenada i efectiva.



Entendre la importància de l'ús d'una retícula fluida i/o modular i la implementació de punts d'interrupció (i les seves sinergies amb les *media queries*) permet als dissenyadors construir un sistema gràfic sòlid i, al seu torn, facilita la coordinació del seu treball amb els desenvolupadors.

### 3.1.1. Retícules fluides

En el disseny *responsive*, el contingut es disposa en una retícula en columnes delimitades mitjançant punts d'interrupció que determinaran quan s'han de mostrar els estils CSS predefinitos.

Exemple de retícula flexible multicolumna



Font: Harry Roberts @csswizardry

La retícula ajuda a definir els mòduls per pàgina necessaris en el producte. Les columnes poden ser fixes, flexibles o combinades, encara que es recomana l'ús de dissenys líquids on sigui possible, i sempre evitant unitats fixes.

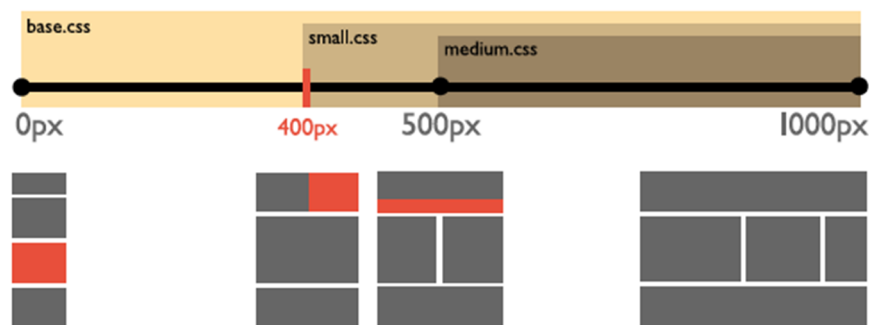
Es tracta de dotar el disseny d'elasticitat per a satisfer qualsevol mida, dispositiu o mitjà. Aquí és on les retícules fluides entren en acció, amb columnes que s'expandeixen i es contrauen segons les necessitats de l'usuari.

### 3.1.2. Punts d'interrupció

Els punts d'interrupció, punts de ruptura o *breakpoints* són les coordenades o mesures de pantalla en què el disseny del lloc web respondrà i canviarà per a proporcionar a l'usuari el contingut en la millor disposició possible per a consumir la informació.

La imatge següent exemplifica la correspondència entre els diferents fulls d'estil associats als diferents punts d'interrupció definits per a un producte web. Veiem com a mesura que l'ample del navegador es redueix, el contingut es recol·loca a la pàgina seguint les instruccions d'un full d'estil CSS o un altre.

Gràfic de punts d'interrupció mínims amb la indicació de l'estil CSS aplicat



Font: Stephen Hay, *Responsive Design Workflow*.

Hi ha diverses tàctiques per a decidir on col·locar els punts d'interrupció en un disseny *responsive*. Una manera de definir els punts d'interrupció és observar on els elements del disseny es divideixen de forma natural i ubicar-los en concordança.

Hi ha la vella creença que diu que s'han de basar en les mides de pantalla dels dispositius més comuns, però s'ha demostrat que això no és escalable amb l'actual diversificació de dispositius en el mercat. El concepte de «mides comunes» com a tal ja no existeix. L'oferta de dispositius sempre està canviant, de manera que els valors d'avui poden ser discutibles d'aquí a un any. La web és inherentment fluida, per tant, és tasca del dissenyador crear interfícies que es vegin i funcionin bé en qualsevol escenari. A més, l'ús de mides fixes basades en dispositius concrets generen resultats inesperats en ser visualitzats en dispositius amb dimensions diferents, com ara espaiats massa grans o massa estrets, que afecten la qualitat i l'aspecte final del producte.

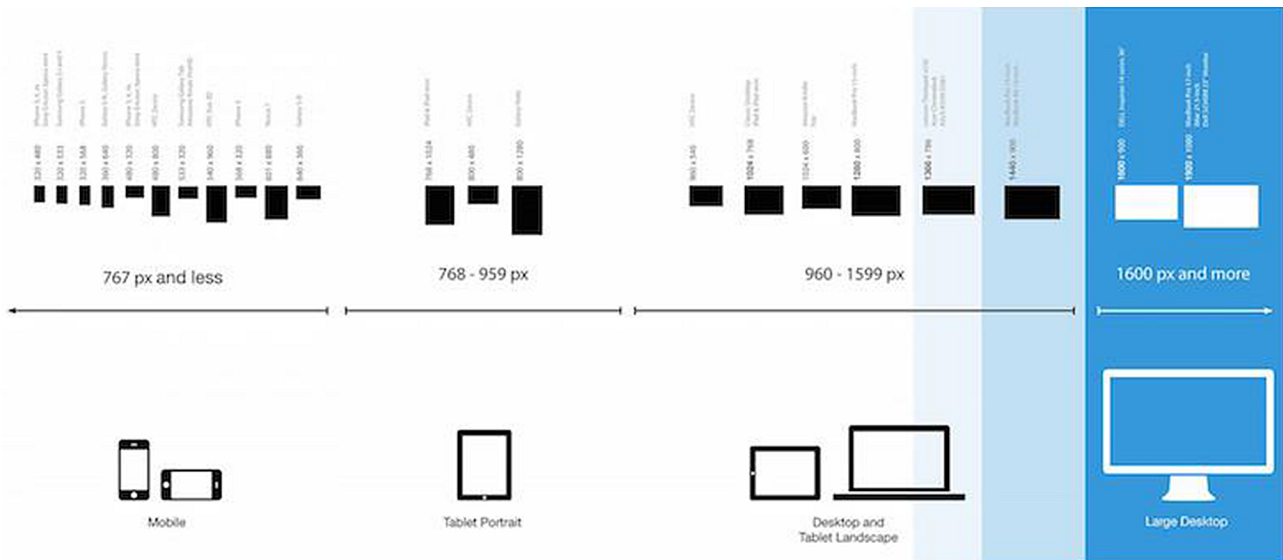
Referent a això, Stephen Hay, autor del llibre *Responsive design workflow*, proposa pensar en termes genèrics i centrar les nostres decisions en tipus o formats de dispositiu (telèfons, tauletes, ordinadors, televisors, consoles de videojocs), en lloc de marques o mides de pantalla concretes.

#### Lectura recomanada

Stephen Hay. *Responsive design workflow*.

Seguint la proposta de Stephen Hay, el gràfic següent d'Invision Lab cobreix els rangs de mides de pantalla dels quatre dispositius nucli, que ofereixen un marc genèric d'aplicació de punts d'interrupció.

Taula resum dels punts d'interrupció segons el tipus de dispositiu



Font: Johannes Hall / Invision Lab

### 3.1.3. Media queries

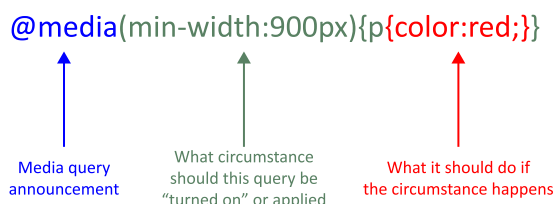
En desenvolupament web, una *media query* és una sèrie de condicions que podem aplicar als fulls d'estil CSS per a adaptar la presentació del contingut a determinades característiques del dispositiu. Aquestes característiques poden ser la mida o resolució de la pantalla (per exemple, un telèfon intel·ligent<sup>15</sup> o una pantalla d'alta definició), o bé la presència de característiques d'accessibilitat com el braille. L'ús de *media queries* es va convertir en un estàndard recomanat per la W3C el 2012 i és un principi bàsic de la tecnologia de disseny *responsive*.

<sup>(15)</sup> *smartphone*

En disseny *responsive*, les *media queries* s'encarreguen de detectar variables com ara l'amplada o l'alçària de la pantalla i controlar l'ús de grups d'estils CSS associant a cada estil la resolució i mida de pantalla adequades per a evitar així la necessitat de canviar el contingut cada vegada.

A la figura següent podem observar-ne l'estructura, formada per una declaració de codi que la identifica com a *media query*, seguida d'una circumstància (en aquest cas, la mesura limita de l'amplada de pàgina a 900 píxels) i seguida d'una variable d'aplicació.

Estructura sintàctica d'una *media query*



Font: Patrick Sexton varyy.com

**Alguns exemples visuals**

La pàgina web Media Queries (accessible en línia) publica una àmplia galeria d'exemples visuals molt recomanable per a consultar l'ús de punts d'interrupció i *media queries*, i comprovar com s'aconsegueix presentar diferents elements en una pàgina de manera distinta, o sacrificar-los en cas de mides de pantalla reduïdes.

### **Exemple d'aplicació de *media queries***

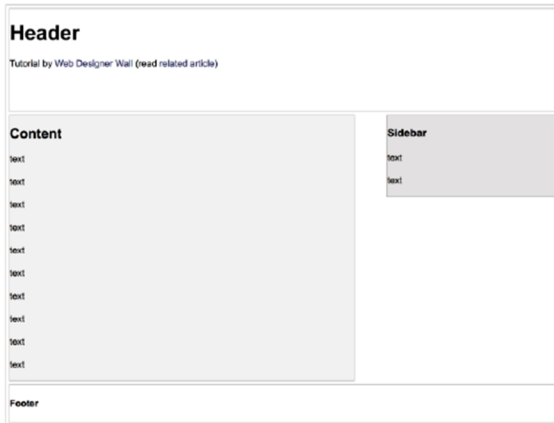
En l'exemple següent veiem l'ús de *media queries* per a determinar tres circumstàncies (tres possibles mides de dispositiu) en la presentació d'una pàgina HTML simple formada per un contenidor principal amb una capçalera (*header*), una àrea de contingut (*content*), una barra lateral (*sidebar*) i un peu de pàgina (*footer*) al seu interior.

El desenvolupador d'aquest exemple crea tres punts d'interrupció, fixats a 980 píxels, 700 píxels i 480 píxels respectivament, i defineix unes condicions específiques per a cada un. Per al primer punt d'interrupció defineix que el contingut i la barra lateral han d'ocupar el 65 % i el 30 % de l'ample de pàgina respectivament, que ocupen una posició flotant alineada a esquerra i dreta en el contenidor principal.

Per al segon punt d'interrupció, anul·la la propietat de flotació i defineix l'ample d'aquests dos elements a «automàtic», de manera que ocuparan tot l'espai horitzontal, i en conseqüència, es col·loquen un damunt de l'altre de forma automàtica. Finalment, en el tercer punt d'interrupció s'anul·la la visualització de la barra lateral (per falta d'espai) i es canvia l'alçària de la capçalera a «automàtic» (ja que la reducció d'espai horitzontal força l'ús d'una doble línia de text) i es redueix la mida de la lletra en concordança.

Exemple d'aplicació de *media queries*

a.

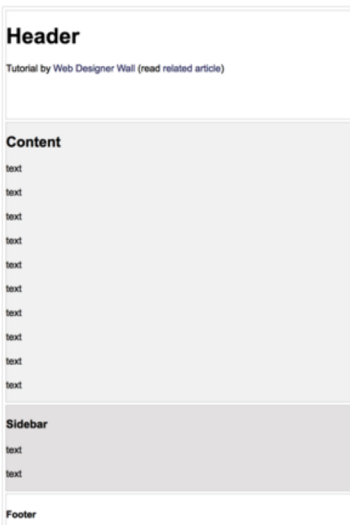


```

/* for 980px or less */
@media screen and (max-width: 980px) {
  #pagewrap {
    width: 94%;
  }
  #content {
    width: 65%;
  }
  #sidebar {
    width: 30%;
  }
}

```

b.



```

/* for 700px or less */
@media screen and (max-width: 700px) {
  #content {
    width: auto;
    float: none;
  }
  #sidebar {
    width: auto;
    float: none;
  }
}

```

c.



```

/* for 480px or less */
@media screen and (max-width: 480px) {
  #header {
    height: auto;
  }
  #h1 {
    font-size: 24px;
  }
  #sidebar {
    display: none;
  }
}

```

### 3.2. El flux de treball *responsive*

Quan ens trobem davant d'un projecte *responsive*, hi ha una sèrie de passos comuns el compliment dels quals facilitarà la realització d'un disseny coherent, que són:

1) **Establir la retícula de disseny i els punts d'interrupció per a diferents dispositius.** La utilització d'una retícula ajuda a mantenir una estructura coherent i facilita la distribució dels mòduls de contingut corresponents a cada pàgina.

#### Exemple d'establiment d'una retícula

Al lloc web següent es recorre a una retícula de vuit columnes que es redueix proporcionalment (8/6/4/2) a mesura que l'ample de pantalla disminueix. El manteniment d'un disseny modular que s'ajusta a aquesta retícula permet recol·locar els elements amb facilitat mantenint una composició consistent.

Exemple d'ús d'una retícula flexible en un desenvolupament multidispositiu



Font: Igentics.com

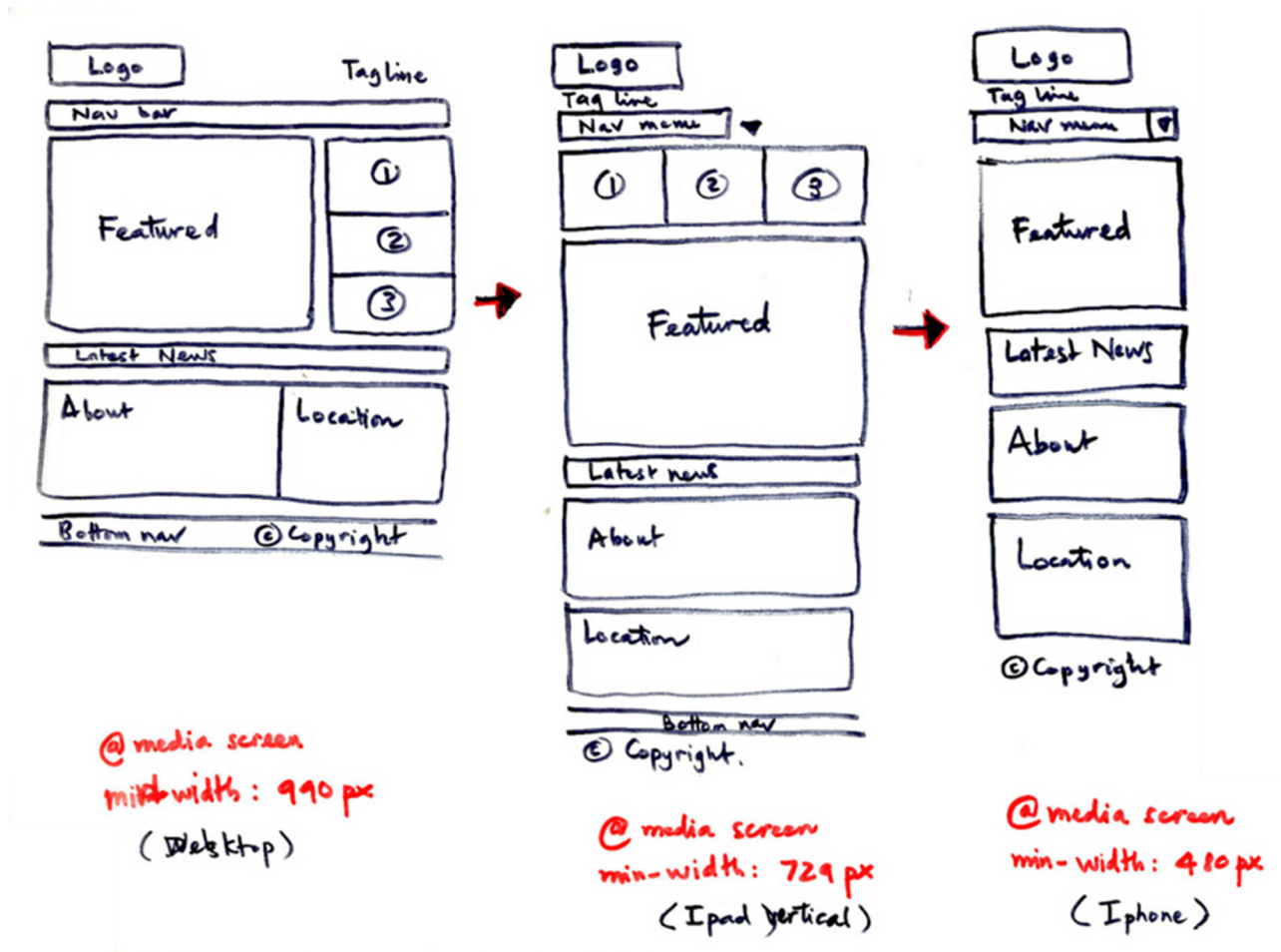
2) **Adoptar i aplicar una estratègia de contingut**<sup>16</sup>. Observar els dispositius i orientacions que els requisits del projecte necessita cobrir i traçar una estratègia de distribució dels mòduls de contingut per al major nombre d'escenaris possible.

<sup>(16)</sup> *content stacking strategy*

#### Exemple d'estratègia de contingut

En l'exemple següent veiem un esbós de treball en què es pot observar una estratègia de contingut molt clara per a tres formats de dispositiu (ordinador, tauleta i telèfon). Els tres blocs de contingut, numerats de l'1 al 3, canvien la seva ubicació i orientació en el format tauleta, mentre que desapareixen per complet al telèfon. La resta d'elements (capçalera, elements de navegació o els apartats etiquetats com «About» i «Location» adapten la seva amplada a la pàgina i passen de mostrar-se en paral·lel a aparèixer de forma apilada.

## Esbossos de distribució de contingut



Font: Accademia Cappiello.

3) **Coreografiar el contingut.** El contingut ha de ser «coreografiat» (ordenat seqüencialment amb un criteri lògic) per tal de mantenir el missatge coherent, independentment del dispositiu des del qual es vegi. Quan distribuïm el contingut de les nostres pàgines, hem de pensar en el contingut que es mostrarà en primer lloc, segon, tercer i successius, d'acord amb la mida dels diferents dispositius. El contingut de la part superior esquerra de la pàgina sempre és el primer que l'usuari detecta i la resta de continguts s'ordena a partir d'aquest punt. Analitzar l'estructura del producte i prioritzar els elements de navegació en la jerarquia de continguts és essencial per a una comprensió de la interacció per part de l'usuari. Els dissenyadors d'experiència sovint complementen els seus dissenys i esquemes de pàgina<sup>17</sup> amb anotacions que reflecteixen les seves intencions pel que fa al comportament del contingut.

<sup>(17)</sup> wireframes

#### Exemple d'esquema de pàgina multidispositiu amb anotacions

En el cas de l'exemple següent es matisen les diferències entre l'aspecte i distribució del contingut en dos formats diferents. En aquest exemple, l'autor indica que en el format per un telèfon intel·ligent les opcions de navegació es mantindran amagades en un menú desplegable, mentre que en el format tauleta, aquestes estaran visibles en tot moment. De la mateixa manera, els blocs modulars s'apilaran per al primer dispositiu, i per al segon es mostraran en una retícula de tres columnes.

Esquemes de pàgina multidispositiu amb anotacions



Font: Felix Wolfstrom.

4) Definir la càrrega narrativa de cada pàgina. No es tracta només de jugar amb els blocs de contingut i les columnes. La quantitat d'informació de la pàgina, així com el lloc que cada bloc de contingut ocupi condicionen

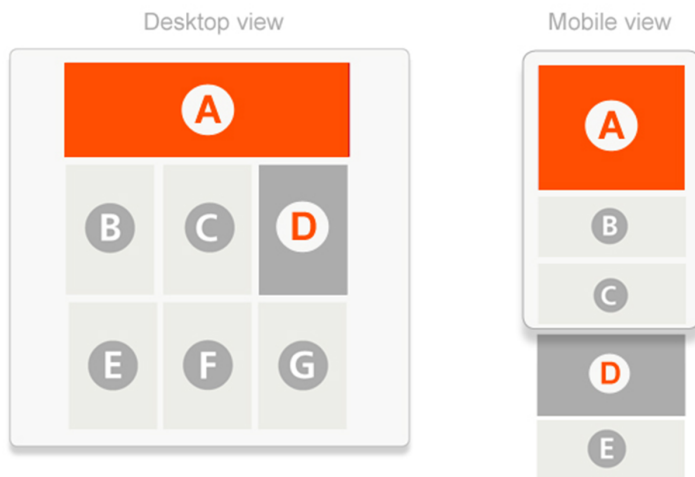


l'experiència de l'usuari i la seva comprensió i interacció amb el contingut. Ens hem d'assegurar que cada pàgina comunica de forma independent el missatge desitjat i compleix amb la funció esperada.

### Exemples de càrrega narrativa

A la imatge següent veiem un esquema de distribució de mòduls de contingut en què la seqüència de lectura difereix entre dispositius, de manera que (presumiblement per motius estratègics) el contingut visible per defecte en pantalla no és el mateix per als dos dispositius.

Distribució de blocs de contingut en dos dispositius diferents



Font: WebStudio 316.

**5) Establir un punt d'inici.** Quan es tracta de prioritzar el contingut i la interacció de l'usuari amb aquest, hem de començar pels dispositius més petits i escalar cap als més grans. Tal com hem vist, aquesta és l'estratègia coneguda com a *mobile first*.

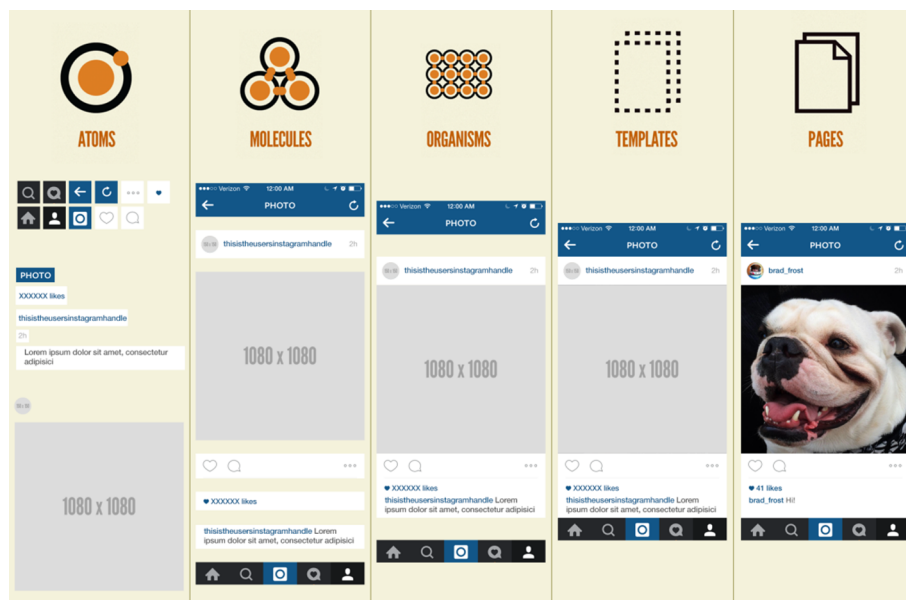
Ana Dahlstrom, dissenyadora UX i autora especialitzada en temes de disseny multipantalla, resumeix en aquest «decàleg del disseny *responsive*» aspectes importants que cal tenir en compte i que complementen el flux de treball descrit anteriorment:

- 1) Definir l'audiència, objectius i necessitats (tant dels usuaris com del client o negoci).
- 2) Identificar les principals històries d'usuari.
- 3) Identificar les pàgines clau del producte.
- 4) Identificar el contingut i els requeriments funcionals de cadascuna de les pàgines clau.
- 5) Identificar mòduls de contingut comuns a totes les pàgines (per exemple, els elements de navegació).

- 6) Llistar i ordenar amb criteris de prioritats el contingut de cada pàgina.
- 7) Identificar possibles variacions en els diferents mòduls de contingut (per exemple, la reducció de la quantitat de text en dispositius més petits o la necessitat d'usar diferents formes de navegació).
- 8) Distribuir els mòduls de contingut a les pàgines clau i treballar l'estratègia de contingut per als diferents dispositius.
- 9) Definir plantilles i patrons per als diferents mòduls de contingut (establir totes les diferents formes de visualitzar el contingut tenint en compte les variacions observades anteriorment).
- 10) Ajustar el disseny de cada pàgina mitjançant testeigs amb dispositius fins a aconseguir un resultat satisfactori.

En l'ecosistema de dispositius multipantalla actual és clau considerar la importància de l'ús de sistemes modulars com a mesura per a garantir un fàcil manteniment i escalabilitat de dissenys. En aquest sentit, Brad Frost, en el seu llibre *Atomic design*, ofereix una proposta complementària que consisteix en una metodologia de treball amb sistemes gràfics que descomponen conceptualment els elements de la pàgina en «molècules» i «àtoms». L'ús sistemàtic d'aquesta tècnica té com a objectiu millorar l'eficiència i durabilitat del treball de disseny.

Exemple de la metodologia *Atomic design* emprada a l'aplicació mòbil nativa per a Instagram



Font: Brad Frost.

#### Lectura recomanada

El llibre de Brad Frost *Atomic design* es pot consultar a l'enllaç següent:

**Brad Frost (2016).** *Atomic design* (accessible en línia).

## 4. Elecció d'una estratègia de desenvolupament

L'estratègia de desenvolupament tindrà un impacte directe tant en el disseny que el dissenyador durà a terme (ja que limitarà en major o menor mesura les possibilitats gràfiques i creatives) com en el seu flux de treball (ja que dictarà el dispositiu o plataforma pel qual s'haurà de començar a treballar). Per tant, és primordial valorar detingudament tots els aspectes del projecte per tal de garantir que les decisions tècniques contribueixen a la consecució dels objectius globals de l'experiència d'usuari.

### 4.1. Aspectes que cal tenir en compte

Wolfram Nagel, autor del llibre *Multi screen UX design*, publica una interessant reflexió al voltant de la presa de decisions sobre l'estratègia de desenvolupament més adequada:

«La decisió sobre quina és la millor solució per a dispositius mòbils, si una aplicació nativa o bé una aplicació web, s'ha de prendre observant cada cas per separat. El sistema de desenvolupament depèn de cada projecte en particular, considerant en cada cas si un disseny *responsive* o bé una aplicació mòbil s'adapten millor als objectius, necessitats i recursos del projecte.

Un lloc web optimitzat per a mòbils arriba a un públic objectiu més ampli. Potencialment es pot mostrar en tots els navegadors mòbils. És més fàcil de trobar a través de motors de cerca, sovint més barats de produir, i no tenen dependències temporals ja que no existeix un cicle de publicació com el de l'*appstore*.»

D'altra banda, quan s'utilitzen aplicacions natives cal decidir per quines plataformes i en quins llenguatges de programació es desenvoluparan. Això depèn en gran mesura del públic objectiu i les funcionalitats del producte. La solució òptima (aplicació nativa, híbrida o *web app*, un lloc web mòbil o una pàgina web amb desenvolupament *responsive*) dependrà de diferents factors com ara la finalitat del producte, el públic objectiu i els requeriments tècnics del projecte.

Si el producte serà utilitzat amb poca freqüència i conté informació relativament simple, una web optimitzada per a mòbil segurament serà suficient. Els usuaris no voldran descarregar-se una aplicació mòbil només per a això.

En canvi, si el producte serà utilitzat amb més freqüència, s'hauria d'avaluar si l'aplicació mòbil requerirà funcionament fora de línia, si ha de generar beneficis (això és normalment més simple a través de botigues d'aplicacions), si serà indexada en cercadors o botigues d'aplicacions, si farà ús de les funcionalitats del dispositiu (càmera, localització, acceleròmetre), i com tots aquests elements s'integren en l'experiència d'usuari.

A tall de recapitulació, incloem uns quadres resum de les característiques de les diferents estratègies amb els elements diferencials i els principals avantatges i inconvenients.

### M-dot (webs específiques per a mòbil)

- 100% *mobile-friendly*, funcionament optimitzat per a dispositius mòbils.
- Accés a través d'un navegador mòbil mitjançant un enllaç URL.
- Arxius separats amb manteniment d'un lloc web dedicat.
- Contingut diferent que ha de ser mantingut de forma separada.
- Tecnologies HTML i CSS.
- No dependent de botigues d'aplicacions.
- Fàcils de mantenir i actualitzar.
- Estratègia amb ús cada vegada més limitat.

### Apps

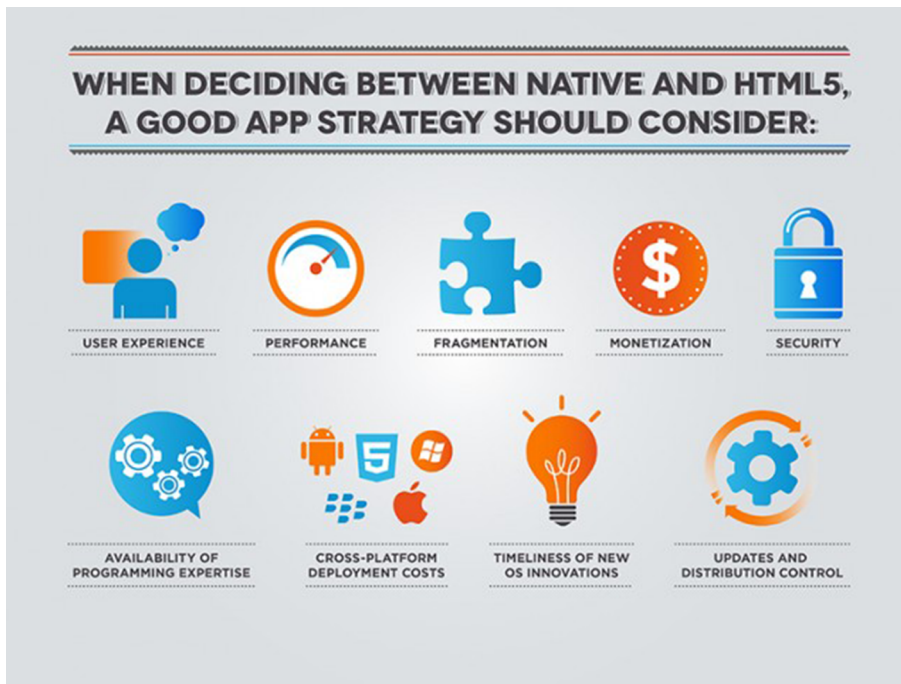
Natives	Híbrides
<ul style="list-style-type: none"> <li>• Grau d'optimització màxim.</li> <li>• Accés a les funcionalitats natives del dispositiu.</li> <li>• Requereix l'ús d'un codi específic a cada plataforma o sistema operatiu.</li> <li>• Pot funcionar sense una connexió activa a Internet.</li> <li>• Millor experiència, amb una major integració de la funcionalitat del dispositiu.</li> <li>• Distribució a través de les botigues d'aplicacions.</li> <li>• El desenvolupament pot ser més car.</li> <li>• Cada plataforma està basada en tecnologies i <i>frameworks</i> diferents.</li> </ul>	<ul style="list-style-type: none"> <li>• Barreja entre una aplicació web i una aplicació nativa.</li> <li>• Adequada per a petits-mitjans projectes multiplataforma.</li> <li>• Es distribueix a través de les botigues d'aplicacions.</li> <li>• Inserida en un contenidor (<i>wrapper</i>) mitjançant l'ús d'HTML5 i Javascript.</li> <li>• Manteniment de menys versions.</li> <li>• Limitacions en la reproducció d'interaccions com les pròpies d'una aplicació mòbil nativa.</li> <li>• Més temps de desenvolupament.</li> </ul>

### Aplicacions web

Fix / Estàtic	Líquid / Fluid	Adaptable	Responsive
Requereix un o més llocs web. Independència de la mida de pantalla o dispositiu. No canvia el contingut. Desenvolupament més econòmic. Més fàcil de testejar. Més control sobre el resultat.	Requereix una sola pàgina web. Basat en la mida de la pantalla. No canvia el contingut. Desenvolupament més econòmic. Més fàcil de testejar. Menys control sobre el resultat.	Requereix diverses webs. Basat en tipus de dispositius. Canvia el contingut. Desenvolupament més econòmic. Més fàcil de testejar. Menys control sobre el resultat.	Requereix una sola pàgina web. Basat en la mida de la pantalla No canvia el contingut (pot variar seguint <i>media queries</i> ). Desenvolupament més car. Més difícil de testejar. Més control sobre el resultat.

Així mateix, en el gràfic següent de l'agència Justcreative es resumeixen les principals variables que cal tenir en compte a l'hora d'analitzar quina estratègia és la més adequada per al desenvolupament d'una aplicació mòbil.

Factors a tenir en compte quan es decideix una estratègia de desenvolupament per a una aplicació mòbil



Font: justcreative.com

L'empresa Axis41, per la seva banda, afegeix en una taula de continguts altres variables que s'han de tenir en compte a l'hora de triar una estratègia de desenvolupament multidispositiu, com ara l'optimització del producte per a la seva cerca i posicionament SEO.

#### 4.2. Per on començar?

El principi «code once, deploy everywhere»<sup>18</sup> és un enfocament àmpliament acceptat en el sector professional. No obstant això, s'han d'analitzar els comportaments i patrons d'ús en les diferents plataformes existents per tal de detectar les necessitats específiques. A tall d'exemple, la integració de tecnologies natives del telèfon requereix sovint l'ús de llenguatges de codi específics que complementin les mancances del codi multiplataforma.

Pel que fa al dilema de per quin dispositiu hauríem de començar, ens trobem amb dues teories contraposades que defensen començar o bé pels dispositius més petits (*mobile first*), o bé pels més grans (*desktop first*).

Com és lògic, des de la perspectiva del disseny aquesta decisió afecta la part visual de l'experiència. Tal com hem vist anteriorment, ens resultarà més fàcil començar per la distribució d'elements de la pàgina en els dispositius més petits (mòbil) i escalar el disseny a les mides més grans. D'acord amb les tendències actuals del sector professional, aquesta sembla ser en els últims anys l'aproximació més comuna.

#### Lectura recomanada

La taula de continguts elaborada per l'empresa Axis41 es pot consultar a l'enllaç següent:

Axis41. «Multi-screen strategy guidelines» (accessible en línia).

<sup>(18)</sup>«Programa una vegada i desenvolupa arreu»

Des de la perspectiva del desenvolupament, en canvi, el dispositiu pel qual comencem implica observar les limitacions i possibilitats dels navegadors per a assegurar el funcionament correcte de la web. En el cas del *mobile first* o *progressive enhancement*<sup>(19)</sup> es desenvolupa una versió bàsica de la web que és completament operativa, amb la possibilitat d'anar afegint millores que assegurin un funcionament correcte en els últims navegadors, tal com mostra la il·lustració següent.

(19) millora progressiva

Desenvolupament per passos del concepte *progressive enhancement*



Font: adroll.com

En el cas de *desktop first* o *graceful degradation*<sup>(20)</sup>, es desenvolupa la web optimitzada per als últims navegadors, amb la possibilitat que funcioni en navegadors antics. La imatge següent ajuda a il·lustrar aquest concepte.

(20) degradació correcta

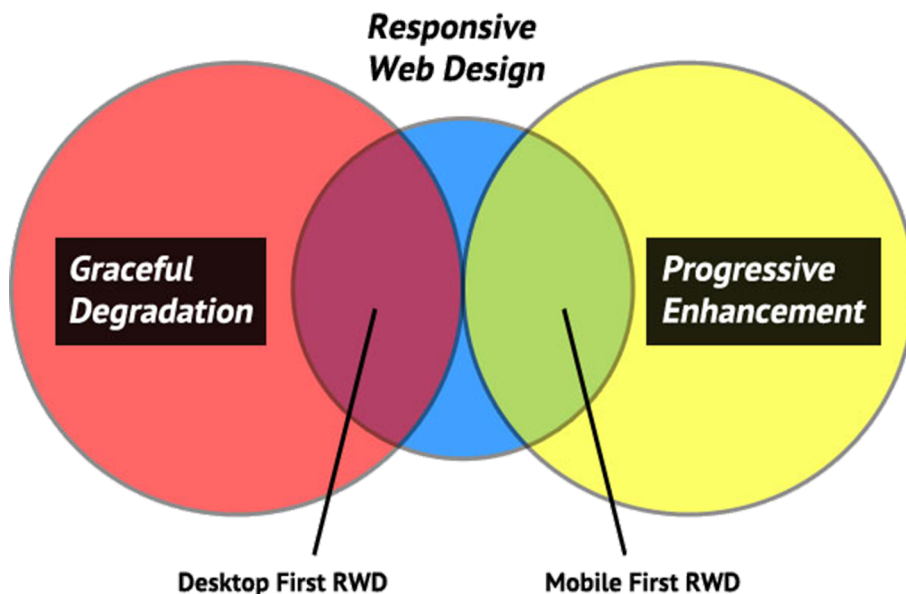
Desenvolupament per passos del concepte *graceful degradation*



Font: adroll.com

El disseny *responsive* aporta solucions a tots dos escenaris creant un marc de desenvolupament flexible i escalable cap als dos extrems. En el gràfic que apareix a continuació se situen les dues idees mitjançant el concepte de *responsive web design*.

*Graceful degradation vs. progressive enhancement*



Font: brettjankord.com

Adolfo Sanz de Diego ha escrit un article que ofereix una completa perspectiva sobre els diferents criteris tècnics que cal tenir en compte a l'hora de valorar com s'ha de començar un projecte.

Des de la perspectiva del contingut, Karen McGrane, especialista en contingut i desenvolupament *responsive*, afirma que en el context actual, la solució de més valor que una empresa pot adoptar a l'hora de desenvolupar un projecte multidispositiu comporta:

- Apostar per un disseny *responsive* sòlid.
- Traçar una estratègia de contingut que redueixi el contingut a l'essencial.
- Oferir la mateixa informació a tots els usuaris.
- Oferir solucions específiques per a cada dispositiu (*adaptive*) quan sigui necessari, però sempre emmarcades en una base *responsive*.

#### Lectura recomanada

L'article d'Adolfo Sanz de Diego sobre els diferents criteris tècnics a tenir en compte es pot consultar a l'enllaç següent:

**Adolfo Sanz de Diego** (2014). *Responsive web design* (accessible en línia).





## Bibliografia

- Ad Roll** (2014). «Unlocking mobile ROI» (accessible en línia), a LinkedIn.
- Avery, Justin**. «Defining breakpoints» (accessible en línia). *Responsivedesign.is*.
- Cyr, Karly** (2016). «M.dot vs. responsive vs. progressive: What's the right solution for your company?» (accessible en línia). *Mobify*.
- Dahlstrom, Anna** (2013). «Designing for multiple devices» (accessible en línia), a LinkedIn.
- Frost, Brad** (2013). «7 habits of highly effective media queries» (accessible en línia). *bradfrost.com*.
- Frost, Brad** (2016). *Atomic design* (accessible en línia).
- Hall, Johanness** (2016). «Large desktop experience responsive design» (accessible en línia). *Inblog*. *invisionapp.com*.
- Hay, Stephen** (2013). «Designing for breakpoints» (accessible en línia). *A list apart*.
- Jacob, Sherice** (2013). «16 top tools for responsive web design» (accessible en línia). *The dailyegg*.
- Marcotte, Ethan** (2010). «Responsive web design». (accessible en línia). *A list apart*.
- Nagel, Wolfram** (2014). «Designing for Multiscreen—Challenges are chances! (6/6)» (accessible en línia). *Medium*.
- Pettit, Nick** (2014). «Which Layout? Static, Liquid, Adaptive, or Responsive» (accessible en línia). *Treehouse.com*.
- Roberts, Harry**. «Cswizardy-grids» (accessible en línia).
- Ruluks, Sandijs** (2014). «9 Basic principles of responsive web design» (accessible en línia). *Froont - Open design blog*.
- Transport for London** (2014). *Transport for London design style guide* (accessible en línia).
- Wroblewski, Luke** (2009). «Mobile first» (accessible en línia). *lukew.com*.
- Zimmer, Cody** (2014). «The different types of responsive design» (accessible en línia). *Duda blog*.

### Recursos

- Interface sketch (accessible en línia).
- Adaptive images (accessible en línia).
- Responsive layouts, responsively wireframed (accessible en línia).
- Patterns and resources for creating responsive web experiences (accessible en línia).

