



# Baliabide mugikorak garatzeko gida teknikoa

*Estandarren dokumentua*

**2012**

Berrikuspena	Data
1	2015eko otsaila
2	2015eko azaroa

## Aurkibidea

1	Sarrera .....	3
2	Baliabide mugikorren garapenaren gaineko oharra.....	4
3	Aplikazio motak .....	6
3.1	MWA (Mobile Web Application): Web-aplikazio mugikorra.....	6
3.2	Native RMA (Native Resident Mobile Application): Berezko aplikazio mugikor egoiliarra.....	7
3.3	Hybrid RMA (Hybrid Resident Mobile Application): Aplikazio mugikor egoiliar hibridoia.....	8
4	Garapen-ingurunea .....	9
4.1	MWA (Mobile Web Application) .....	9
4.2	Native RMA (Native Resident Mobile Application).....	9
4.3	Hybrid RMA (Hybrid Resident Mobile Application) .....	9
5	Bertsioak egitea .....	11
5.1	MWA (Mobile Web application) .....	11
5.3	Native RMA (Native Resident Mobile Application).....	11
5.2	Hybrid RMA (Hybrid Resident Mobile Application) .....	12
6	Ezartzeko eta argitaratzeko prozedura .....	15
7	Garapen mugikorreko jardunbide egokiak .....	16
8	Aplikazioa argitaratzeko eman beharreko datuak.....	20

# 1 Sarrera

Eusko Jaurlaritzaren Sare Korporatiboan baliabide mugikorrek garatzeari begira kontuan hartu beharreko jarraibide eta gomendio teknikoak jaso dira agiri honetan.

Ondorengo ataletan daude deskribatuta baliabide mugikor motak eta mota bat edo bestea aukeratzeko kontuan hartu beharreko irizpenak, hala nola zer-nola egingo den aplikazioaren entrega Sare Korporatiboko bertsioen kudeaketa sisteman (Subversion [SVN]).

## 2 Baliabide mugikorren garapenaren gaineko oharrak

Baliabide mugikorrek garatzean, aplikazioak bete beharreko eskakizunen arabera alderdiak hartu behar dira kontuan:

### Gailuaren berezko gaitasunak:

Gailuaren gaitasunak erabiltzea.

- **Gailuaren hardwarea:** azelerometroa, giroskopia, kamera, GPSa, komunikazioak (*wifi, nfc, bluetooth*), grafikoen 3Dko azelerazioa erabiltzea eta gailuaren memoriaren edo kanpo memoria hedatuaren edukiera.
- **Beste aplikazio batzuekin bateratzea:** gailuaren egoera, kontaktuen agenda, egutegia, kokapenari loturiko argazkiak, aplikazioen artean edukiak partekatzea.
- **Interfazearen kontrolak:** gailuaren sistema eragileak zehazten ditu erabiltzaileak nabigatzean erabili beharreko kontrolak.

### Garapeneko kostuak:

- **Portabilitatea:** baliabide mugikor baten gailu mugikorren plataforma-kopuru handiago baterako portabilitateak garapen- eta mantentze-kostu handiagoak dakartza. Kostu horiek murrizteko, baliabide mugikorrek plataforma erabilienetara mugatuko dira, baita korporatiboki kontuan hartu beharrekoetara ere. Portabilitateak garapenean duen inpaktuan eragiten dute honako hauek:
  - ✓ **Kode-berrerabilera:** aplikazio bat hainbat plataforma mugikorretara hedatzeko aukera. Kode-berrerabilera txikiak plataforma guztietan berariazko aldaketa asko eta asko egin behar izatea eragingo du, garapen-kostuak handiagotuz.
  - ✓ **Tresna-berrerabilera:** garapen-tresna multzo berdina berrerabiltzeko aukera ematen du, baliabide mugikorrek helburuko plataformetan ezartzeko.
  - ✓ **Plataformen euskarria:** baliabide mugikorra argitaratuko den plataforma mugikorren kopurua.

Ondorengo koadroan dago azalduta baliabide mugikor motak garapen-proiektu baten ezaugarriei zer-nola eragiten dien:

	Webgunea	Hibridoa	Berezkoa
Kostua	arrazoizkoa	arrazoizkoa	garestia
Denbora-epea	laburra	laburra	luzea
Portabilitatea	handia	handia	nulua
Errendimendua	ertaina	berezko errendimendua (behar izanez gero)	oso bizkorra
Berezko-funtzioak	ez	guztiak	guztiak
Aplikazioa <i>store</i> -etan banatzea	ez	bai	bai
Hedagarria	ez	bai	bai

## 3 Aplikazio motak

Aplikazio eta arkitektura mugikorrek garatzeko ezarritako betekizunetan oinarrituta, aplikazioak honako mota hauetan sailka daitezke:

### 3.1 MWA (Mobile Web Application): Web-aplikazio mugikorra

Nabigatzailearen bat duen edozein gailutan, izan finkoa edo mugikorra, bistaratu eta erabil daitezkeen web-aplikazioak dira. Aplikazioa ez da gailuan instalatuko, informazioaren aurkezpena eta formatua interfazearen neurrien arabera bereiz ditzakeen gune baten bidez eskuratuko baita.

Esaterako, mahai gaineko ordenagailu batean aurkeztekoa bada, 10 zutabeko taula agertuko da. Alabaina, *tablet*an aurkeztekoa bada, 10 zutabe etzan edo 5 bertikal agertuko dira. Bestalde, gailua telefono mugikorra bada, informazioa erregistro-mailan aurkeztuko da.

Mota horretako aplikazioetan garapena sistema eragiletik eta gailuaren gaitasunetatik bereizten da, honakoetan oinarrituta:

- HTML5, CSS3 eta Javascript.
- Gailuaren araberrako diseinua (*responsive design*).
- *Media queries*, bereizmenaren arabera sortzeko interfazea.
- *Single Page Applications*, aplikazioa orrialde bakarrean bistaratzeko, edukiak dinamikoki gehitu eta kenduz, eta erabiltzailearen esperientzia arinduz.
- *Frameworks javascript* aurkezpen-geruzan (jQueryMobile, Sencha, AngularJS, Backbone, Ember).

Aplikazio horien arkitektura web-aplikazio tradizionalen antzekoa da. Honako hauek izan ohi ditu:

- APIen, web-zerbitzuen eta restFUL zerbitzuen bidez kontsumitzen diren negozio-zerbitzuen multzoa, kudeatu beharreko informazioa eta datuak eskaintzen dituen.
- Bistaratzeko-pantailaren bereizmenara egokitzen den edukiaren aurkezpen-geruza.

Normalean, mota horretako garapenetara jo ohi da, web-aplikazio bat gailu mugikor batera epe laburrean edo kostu mugatuekin egokitu nahi denean edo aplikazio baten modulu bat gailu mugikorretan kontsumitzea nahi denean.

## 3.2 Native RMA (Native Resident Mobile Application):

### Berezko aplikazio mugikor egoiliarra

Gailuan instalatutako aplikazioak dira, plataforma jakin batentzat berariaz garatu direnak, dagozkien *Software Development Kit*-aren (SDK) arabera.

Plataforma mugikor, dagozkien SDK-k barne, eta programazio- lengoia hedatuenak honako hauek dira:

- Android: Android Development Tool, Java lengoia
- Apple iOS: XCode, Swift lengoia eta Objective C
- Microsoft Windows Phone: Windows Phone SDK, .Net lengoia
- Blackberry: Blackberry RIM, Java lengoia

Aplikazio horiek honako ezaugarriak dituzte:

- Helburuko plataformaren berariazko garapena edukitzea.
- Plataforma guztiek bere programazio- lengoia edukitzea.
- Fabrikatzaileak gomendatutako garapen-tresnen multzoa edukitzea.
- Gailuaren berezko gaitasunak baliatzea (kamera, gps-a, azelerometroa, etab.).

Garapen-kostuen ikuspegitik, aukera hori garestiagoa da; garapena osoa izan behar da eta aplikazioa argitaratuko den helburuko plataforma guztiek berariazko baliabideak behar dituzte.

### 3.3 Hybrid RMA (Hybrid Resident Mobile Application): Aplikazio mugikor egoiliar hibridoa

Aplikazio mugikor egoiliar hibridoak gailu mugikorrean instalatu eta lengoia eta eredu propioen bidez garatzen dira, gailu mugikorraren berariazko gaitasunetara sarbidea emanez.

Ezaugarriak:

- Web- lengoia batean oinarritutako garapena (C#, Java, Javascript, C++, etab.), dagokion sistema eragilearen (Windows Phone, Android, iOS, Blackberry OS, etab.) lengoian jasota eratuko dena.
- Berezko gaitasunak erabiltzea, plataformaren berariazko *plugin*-ak sartuta.
- Tresnak (phonegap, cordova, xamarin, appcelerator, kony, etab.).



## 4 Garapen-ingurunea

Jarraian, labur azalduko dugu aplikazio moten garapen-ingurunea.

### 4.1 MWA (Mobile Web Application)

Mota horretako garapenetan web-aplikazioak garatzeko erabilitako ohiko ingurunea erabiliko da.

Mota horretako aplikazioen garapena **UDA**ren bidez gauzatu daiteke, alderdi zerbitzu-emailerari dagokionez. Interfazeari dagokionez, berriz, honako jarraibide hauek hartu behar dira kontuan:

- **Single Page Application (SPA)**: erabiltzailearen ekintzari erantzuteko beharrezko baliabideak orri bakarrean agertzen diren eta interakzioa orri bakar horretan gauzatzen den aplikazioak.
- **Responsive Design**: gailuaren araberako diseinua, interfazea eta aplikazioarekin elkarrengaitza egokitzen direna aplikazioarekin elkarrengaitzako erabilitako gailura (mahai gaineko pantaila saguarekin eta teklatuarekin; *tableta* teklatuarekin; *tableta* edo mugikorra teklatu birtualarekin).

Gainera, mota horretako garapenetan *framework*-ak erabil daitezke aurkezpen-geruzan; besteak beste, jQuery Mobile, Angular JS, backbone, bootstrap, moustache eta ionic.

### 4.2 Native RMA (Native Resident Mobile Application)

Berezko aplikazio bat garatzean, garapena baliabide mugikorra argitaratuko den plataformei dagokien SDK bidez egin beharko da.

### 4.3 Hybrid RMA (Hybrid Resident Mobile Application)

Aplikazio hibrido bat Cordova/Phonegap bidez garatuz gero, balizko bi lan-fluxu bereizi behar dira.

## **Cross-platform development**

Cordova/Phonegap tresnak erabiliko dira plataformen *build*-ak sortzeko.

*Build*-ak egin eta plataformak kudeatzeko, dagozkien SDK-k instalatu beharko dira.

Honako hau da plataformen euskarria:

- iOS (Mac)
- Amazon Fire OS (Mac, Linux, Windows)
- Android (Mac, Linux, Windows)
- BlackBerry 10 (Mac, Linux, Windows)
- Windows Phone 7 (Windows)
- Windows Phone 8 (Windows)
- Windows 8 (Windows)
- Firefox OS (Mac, Linux, Windows)

## **Platform-centered development**

Aplikazioaren *build*-ak sortu nahi diren plataformei dagozkien SDK-k erabiliko dira.

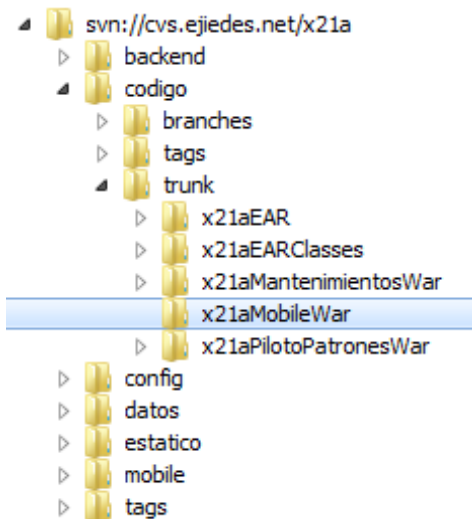
## 5 Bertsioak egitea

Atal honetan proiektuek SVN-en izango duten egituraren xehetasunak eskainiko dira. Errespetatu beharreko karpeta-egitura ezarri beharreko aplikazio motaren arabera izan beharko da.

### 5.1 MWA (Mobile Web application)

Mota horretako garapenetan web-aplikazioak garatzeko erabilitako ohiko egitura erabiliko da.

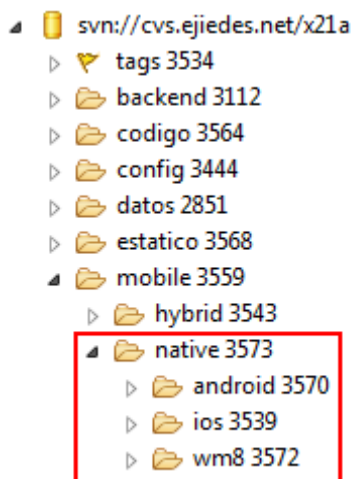
Aplikazioak hainbat web-modulu izango ditu, ezarritako funtzio-bereizketa gorabehera, eta horietako bat edo gehiago gailu mugikorren nabigatzaileen bidez aurkezteko modukoa izango da.



### 5.3 Native RMA (Native Resident Mobile Application)

Aplikazioa garatu nahi dugun plataforma guztien berezko aplikazioak garatzea erabaki bada, dagokion karpeta sartu beharko da plataformari loturiko iturburu-kodea.

- android: Android-en SDK-rekin egindako garapenari dagozkion iturburuak.
- ios: iOS-en SDK-rekin egindako garapenari dagozkion iturburuak.
- wp8: Windows Phone 8-ren SDK-rekin egindako garapenari dagozkion iturburuak.
- Beste plataforma batzuetarako garatuz gero, bertsio bana garatu beharko da.

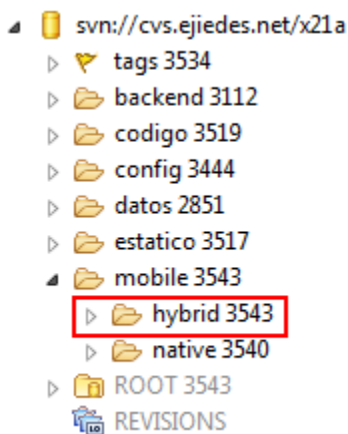


## 5.2 Hybrid RMA (Hybrid Resident Mobile Application)

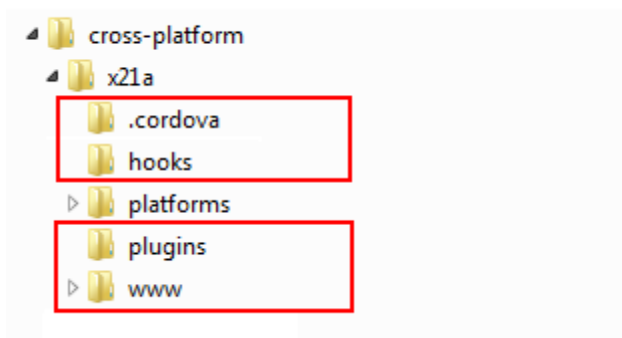
Cordova/Phonegap bidez aplikazio hibrido bat garatzean, bi lan-fluxu bereizten dira aplikazioa garatzeko. Bat edo bestea hautatzearen mende egongo da aplikazioaren bertsioa egingo den SDK-ren egitura.

### Cross-platform development

*Cross-platform workflow* bidez garatutako aplikazioen bertsioak egingo dira SVN biltegiko *mobile/hybrid* direktorioaren barnean.

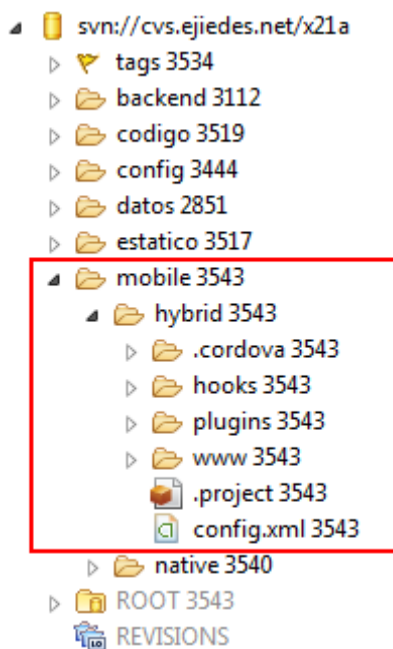


Garapena SVNra igotzean, *cross-platform* lan-fluxuarekin lan eginez gero, honako direktorio hauek igo beharko lirateke:



*Platforms* direktorioaren bertsiorik ez da egin beharko, cordova CLIren *build* atazen bidez sortzen baitira.

Biltegiaren amaierako itxurako honakoa izango da:

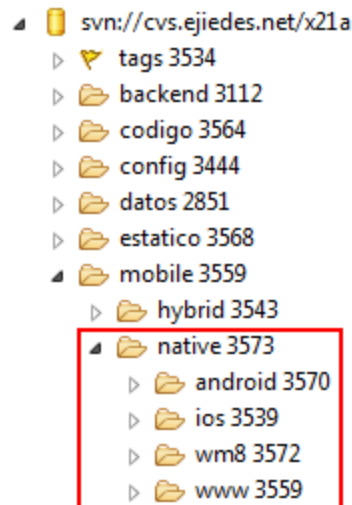


### Platform-centered development

Garapena Phonegap/Cordova bidez egin nahi bada, baina plataforma bakoitzaren SDK erabilia, fitxategiak SDK-ren honako direktorioetara igo beharko dira:

- *www*: plataforma bakoitzari dagokion *web view*-ean erabili beharreko web-baliabide komunak kokatuta dauden direktorio komuna.
- *android*: Android-en SDK-rekin egindako garapenari dagozkion iturburuak.
- *ios*: iOS-en SDK-rekin egindako garapenari dagozkion iturburuak.
- *wp8*: Windows Phone 8-ren SDK-rekin egindako garapenari dagozkion iturburuak.

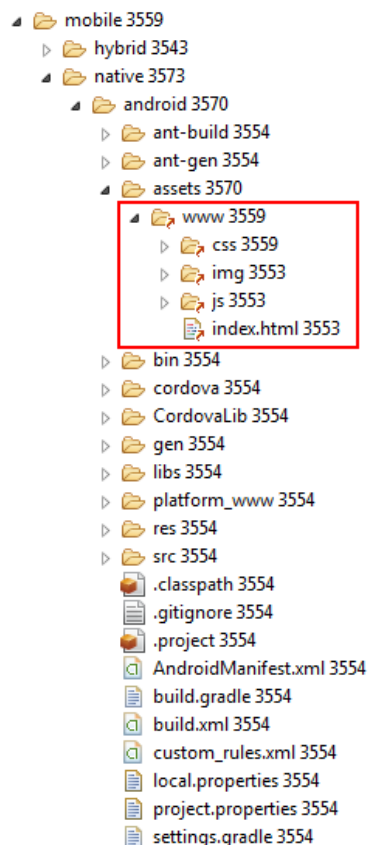
- Beste plataforma batzuetarako garatuz gero, bertsio bana garatu beharko da.



www direktorioaren edukiaren gaineko garapen komuna SDK guztietatik gauzatzeko, SVNren *Externals* funtzionaltasuna erabiliko da. Horrela, plataforma bakoitzaren berariazko proiektuen eta www direktorio komunaren arteko mendekotasuna ezarriko da.

## Android

*svn:externals* bidez, www direktorio komunarekiko mendekotasuna ezarriko da android/assets direktorioaren barnean:



## 6 Ezartzeko eta argitaratzeko prozedura

Eusko Jaurlaritzaren Sare Korporatiboan garapen mugikorrek ezartzeko eta argitaratzeko prozeduraren protagonistak garapen mugikorra garatzeko taldea, aldaketak kudeatzeko taldea eta laguntza zerbitzua dira.

Aplikazio mugikorrek argitaratzeko prozedura ohiko aplikazioak argitaratzeko prozeduraren antzekoa izango da.

Lehen fasean, honakoa eskatu beharko du aplikazio/baliabide mugikorrek:

### 1. Aplikazio-kodea eta SVN eskaera

RFC bat eskatu behar da, aplikazio-kodea eskuratzeko. Eskaeran adierazi beharko da aplikazio mugikor bat dela eta zein dendatan (*store*) argitaratuko den.

Subversion-en esku hartzeko eskaera ondorengo kasuetan egin beharko da:

- Ekimena berria denean eta aplikazio-kodea berria bada, SVN biltegia sortzeko.
- Ekimena berria denean eta aplikazio-kodea existitzen bada dagoeneko, SVN biltegia aldatzeko (betiere 5. atalean zehaztutako *mobile* egitura errespetatuz).

### 2. Argitalpen eskaera

Aplikazioa argitaratuko den dendatan (*store*) hedatzeko eginkizuna eskatu beharko da.

Aplikazio mugikorraren kodea lekualdatze gunean utzi beharko da:

```
/traspaso/aplic/[código_aplicación]/
```

### Argitaratzea

Aplikazio mugikor bat argitaratzeko beharrezko datuak desberdinak dira, ostatu hartuta dagoen *Denda* gorabehera. Datu horiek «*Aldaketak Kudeatzeko Zerbitzua*»ri helarazi beharko zaizkio, dagoen *Dendan* behar bezala argitaratzeko.

Aplikazio bat aplikazio-dendetan argitaratzeak jende asko uxatu dezake; arrazoi asko dagoelako edozein aplikaziori uko egiteko. Hori ahal beste ekiditeko, plataformen banaketa-gida irakurtzea komeni da.

- [Apple App Distribution Guide](#)
- [Android Google Play Guide](#) eta [Android Launch Checklist](#)
- [Windows Phone Distribution](#)

## 7 Garapen mugikorreko jardunbide egokiak

Hona hemen garapen mugikorra gauzatzean kontuan hartu beharreko jarraibideen eta oharren sorta bat.

### Aplikazioaren tamaina

Aplikazioak behar duen espazioa kontuan hartu behar da:

- Dendatik (*store*) deskargatzean, gailuan instalatu beharreko aplikazioa baldin bada.
- Bere baliabideak baliatzeko espazioa behar badu.

### Diseinu-gidak

Gailu batean instalatu behar den aplikazio bat garatzean, nahitaez kontsultatu behar dira dagozkion plataformen diseinu-gidak.

- [iOS Developer Library](#)
- [iOS Human Interface Guidelines](#)
- [Android Design](#)
- [Windows Design](#)

### Berezko itxura lortzeko tipografiak

Aplikazio hibridoen kasuan, plataformaren tipografia bereizgarria erabiltzeak berezko itxura ematen die eta erabiltzaileei ezaguna egingo zaizkie. Zorionez, aski erraza da letra mota horiek aplikazio batean txertatzea. Hona hemen horretarako gidak eta erabilitako letra motak.

- [iOS Fonts](#)
- [Android Fonts](#)
- [Windows Phone Fonts](#)



## Erabiltzailearen interfazea gailua da

Aplikazioa errendatzea bizkortzeko, erabiltzailearen interfazea zerbitzariaren geruzan sortzea saihestu behar da.

Helburua bezeroan erabiltzaile-interfazea HTML, JavaScript eta CSS bidez sortzea da, zerbitzariari eskatutako informazio dinamikoan oinarrituta.

Bezeroaren eta zerbitzariaren arteko komunikazioa REST zerbitzuen bidez egingo da. Erantzuna gorabehera, erabiltzailearen interfazea aldatuko zaio bezeroari, JavaScript erabiliz.

## Single page applications

Aplikazio mugikorrek diseinatu eta garatzean, ezinbestean eskaini behar zaio erabiltzaileari erabilera-esperientzia arina, bezeroaren interakzioari aplikazioak ahalik eta bizkorren erantzutea xede dela.

Helburu hori lortzeko estrategia, ahal dela, aplikazioaren pantaila guztiak jasota egongo diren orrialde bakarra sortzea da. Hala, pantailen arteko trantsizioak arinagoak eta bizkorragoak izango dira, ez baita beharrezkoa erakutsiko den hurrengo orrialdea eskatzea.

## Sarerako sarbidea mugatzea

Aplikazio mugikorren erabilera geldotzen duen faktore nagusietako bat aplikazio-zerbitzariei eskaerak egin osteko itxaronaldia da.

Urruneko sarbide hori beharrezkoa da kasu gehienetan, erabiltzaileari erakutsi beharreko informazioa lortzeko. Hala ere, *cache*-an gehiago edo gutxiago sar daitekeen edukia hautemateko aukera dago.

- **Eduki estatikoa:** estatikotzat jotako edukia, inoiz aldatuko ez dena, gailuan biltegitatu beharko da. Hala, zerbitzariari alperreko eskaerak egitea ekiditen da.
- **Eduki dinamikoa:** zenbaitetan, ez da beharrezkoa eduki dinamikoa denbora errealean eguneratzea. Esaterako, edukia pantailara lehenengoz sartzean kargatuko da edo denbora jakin batean ostean eguneratuko da. Horretarako, toki-biltegitatzeko estrategiak jarrai daitezke, informazioari aldi batean eusteko.
  - *LocalStorage*.
  - Fitxategien sistema.
  - SQL tokiko datu-basea.

## Ez blokeatu erabiltzailearen interfazea

Erabiltzailearen interfazea blokeatzea saihestu behar da aplikazio-zerbitzariari eskaerak egitean. Era berean, ez da geroratu behar interfazea erakustea, behin datuak jasota. Hori saihesteko, informazioa eguneratzeko prozesua abian dela adierazteko elementu grafiko bizidunak erabili daitezke.

Hala, aplikazioaren erabilera arinagoa da eta erabiltzailearen interfazeko blokeoak, erabilera itsusten dutenak, saihesten dira.

## Berezko hardware-azelerazioa erabiltzea trantsizioetan

Aplikazioek arinago eta bizkorrago gauza ditzaten pantailen arteko trantsizioak eta nabigazioak, gailuen hardwarea azeleratzeko gaitasunak baliatzen dituzten CSS motako trantsizioek eskainitako funtzionaltasun berriak erabili beharko dira.

Egun, gailu gehienek dauzkate hardwareak azeleratutako gaitasun grafikoak. Horrek ikusizko esperientzia eraginkor eta atseginagoa dakarkio erabiltzaileari.

Adibidez, 3Dko trantsizioak CSS-etan, pantailen arteko nabigazioan.

## Irudiak optimizatzea

Gailu mugikorretan irudiak erabiltzea modu askotara optimiza daiteke:

- **CSS Sprite Sheets erabiltzea:** irudiak banako fitxategi asko eta askoren bidez sartu ordez, komeni da CSS Sprites sortzea. Hala, fitxategi bakarra deskargatu daiteke eta CSS-ren bidez erakutsi beharreko irudia zehazten da.



- **Irudi-tamaina egokia erabiltzea:** irudiak HTMLen bertan eskalan jartzea ekidin behar da. Komenigarriagoa da zenbait irudi-tamaina erabiltzea, gailu desberdinei egokituta, pisu handiagoko irudi bat erabili eta gailu bakoitzean eskalan jartzea baino.

## JavaScript kodea optimizatzea

JavaScript kodea idazteko garaian oso garrantzitsua da erabilera eta ezarpena optimizatzea. Garrantzi handia du *framework*-ari edo erabilitako javascript liburutegiei dagokien kodea optimizatzeko gomendioei jaramon egiteak.

## 8 Aplikazioa argitaratzeko eman beharreko datuak

Mugikorrentzako aplikazioak argitaratzeko eman beharreko datuak ezberdinak izan daitezke dendaren arabera. Aipatutako datuak «Aldaketen kudeaketa» Zerbitzuari eman behar dizkiogu dagokigun dendan argitalpena modu egokian egin ahal izateko.

Ondoren, Apple, Google eta Microsoft dendetan beharrezkoak diren datuak zehaztuko ditugu:

Denda	Apple	Google	Microsoft
			
<b>App izena</b>	Bai	Bai (30 karaktere, gehienez)	Bai
<b>Titulua</b>	-	Bai (30 karaktere, gehienez)	-
<b>Deskripzioa</b>	Hizkuntza bakoitzeko deskripzio paragrafoak idaztea gomendatzen dizugu	Hizkuntza bakoitzeko titulu/deskripzio paragrafoak idaztea gomendatzen dizugu (4.000 karaktere, gehienez)	Hizkuntza bakoitzeko deskripzio paragrafoak idaztea gomendatzen dizugu
<b>Pantaila-kaptura</b>	Ondorengo ezaugarriak dituzten gailuentzat gutxienez: 3,5 hazbeteko, 4 hazbeteko eta iPad gailuentzat.	2 eta 8 kaptura artean. 24 biteko JPEG edo PNG formatuko artxiboak. Argazki bakoitzak 320 pixeleko luzera minimoa eta 3.840 pixeleko luzera maximoa izango ditu.	768 x 1280 px bereizmena
<b>Ikonoa</b>	Bai	32 biteko PNG formatuko artxiboak; eta 512 x 512	99 x 99 px
<b>Taxonomia</b>	Kategoria nagusia, Bigarren mailako kategoria eta Gako-hitzak	Aplikazio mota, Kategoria eta Edukiaren sailkapena	Kategoria nagusia, Bigarren mailako kategoria eta Gako-hitzak
<b>Webgunea</b>	Lehenetsita <a href="http://www.euskadi.eus">http://www.euskadi.eus</a>		
<b>Posta elektronikoa</b>	-	Lehenetsita <a href="mailto:ejgv.apps@gmail.com">ejgv.apps@gmail.com</a>	-