

# Razvoj hibridnih mobilnih aplikacija pomoću PhoneGap platforme

Igor Dujlović

Elektrotehnički fakultet Banja Luka  
Banja Luka, BiH  
dujlovic@gmail.com

Zoran Đurić

Elektrotehnički fakultet Banja Luka  
Banja Luka, BiH  
zoran.djuric@etfbl.net

**Sadržaj** – U ovom radu su prikazani teoretski principi native, web i hibridnog načina razvoja mobilnih aplikacija, kao i mogućnost primjene svakog načina razvoja. Posebno je razmotrena mogućnost primjene razvoja hibridnih aplikacija pomoću HTML5 i povezanih tehnologija uz upotrebu PhoneGap platforme.

**Ključne riječi:** HTML 5, PhoneGap, mobilne aplikacije, hibridne mobilne aplikacije, cross-platform aplikacije

## I. UVOD

Prije početka razvoja mobilne aplikacije koja treba da se izvršava na različitim mobilnim platformama, neophodno je definisati njene funkcionalnosti, jer postoje razlike u dostupnosti određenih funkcionalnosti na različitim platformama. Veoma često se u aplikacijama koriste neke opšte funkcionalnosti koje su dostupne na svim platformama. Svaka platforma pruža određene funkcionalnosti i uglavnom nije moguće koristiti istu aplikaciju na različitim platformama. Upravo veliki broj različitih platformi i veliki broj različitih tehnologija za razvoj mobilnih aplikacija dovode do toga da razvoj aplikacija za različite platforme zahtjeva više vremena, pa se samim tim i troškovi razvoja povećavaju. Osim toga, u slučaju izmjene dijela aplikacije potrebno je izvršiti identične ili slične promjene u izvornom kodu svih varijanti aplikacije pisanih za svaku pojedinačnu platformu. Dakle, jedna aplikacija ima više različitih varijanti izvornog koda za različite platforme, pisane u različitim programskim jezicima. Svi ovi problemi su doveli do ideje da se aplikacije razvijaju na univerzalan način, pomoću jedne tehnologije, a da se uz minimalna prilagođavanja mogu izvršavati na više platformi. Realizacijom ove ideje smanjuju se troškovi razvoja što u poslovnom smislu dovodi do većeg finansijskog „uspjeha“ aplikacije. U tehničkom smislu sam razvoj se pojednostavljuje, a promjene u funkcionalnostima aplikacije postaju jednostavnije, jer se jedan izvorni kod koristi za sve platforme. U nastavku rada će biti prikazani načini razvoja aplikacija, kao i detaljniji prikaz mogućnosti razvoja aplikacija za više platformi pomoću HTML5 i PhoneGap tehnologija.

## II. VRSTE MOBILNIH APLIKACIJA PREMA NAČINU RAZVOJA

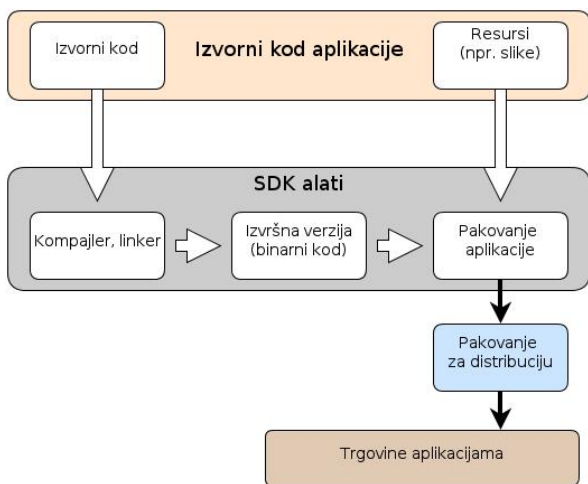
Aplikacije mogu biti *native*, *web*, ili hibridne, u zavisnosti od toga koji pristup za izradu je izabran. Izbor određenog pristupa nije jednostavan i zavisi od više parametara koji se

moraju razmotriti prije početka razvoja, a odnose se na tehničke i ekonomske aspekte, kao i na ograničenja pri razvoju. Tehnički aspekti se odnose na funkcionalnosti uređaja koje se mogu koristiti u aplikaciji, neophodne performanse aplikacije, kao i dizajn. Ekonomski aspekti su određeni vremenom i troškovima razvoja, kao i mogućnošću korištenja iste aplikacije na većem broju platformi.

### A. Native aplikacije

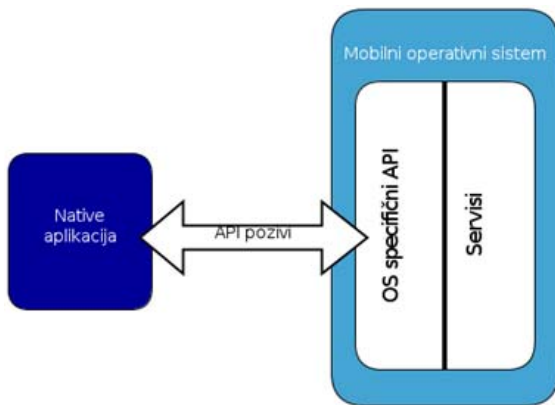
Native aplikacije razvijaju se za upotrebu na određenoj platformi. Kada se aplikacija instalira na uređaj, ona komunicira sa operativnim sistemom uređaja, a može da koristi sve funkcionalnosti uređaja, jer se njen razvoj bazira na kompletnom API-ju (*Application Programming Interface*). Osim toga, ovakve aplikacije koriste ugrađene grafičke komponente i imaju dizajn koji je karakterističan za platformu na kojoj se aplikacija izvršava. Aplikacije se preuzimaju i smještaju na fajl sistem mobilnog uređaja, a njihova distribucija obično se vrši preko „trgovina“ aplikacijama za određenu platformu.

Razvoj native aplikacija se vrši pomoću softverskih alata za izabranu platformu. Kod aplikacije se piše u jednom od programskih jezika koji su predviđeni za razvoj aplikacija na datoj platformi. Napisani izvorni kod se kompajlira i povezuje sa potrebnim bibliotekama pomoću SDK (*Software Development Kit*) alata koji dostavlja proizvođač platforme. Izvršnoj verziji programa dodaju se resursi koji se koriste u aplikaciji, i tako zapakovana aplikacija postavlja se na „trgovinu“ aplikacijama [1]. *Native* aplikaciju moguće je razviti i upotrebom alata za razvoj aplikacija za više platformi. Pri tome, moguće je koristiti skup zajedničkih funkcionalnosti svih izabranih platformi, ili se aplikacija razvija posebno za svaku platformu nakon čega se vrši kompajliranje, tipično HTML5 i JavaScript koda aplikacije u *native* format. Razvoj *native* aplikacija prikazan je na slici 1.



Slika 1. Razvoj native aplikacija [2]

Native aplikacije se izvršavaju pod kontrolom operativnog sistema uređaja, bez bilo kakvog dodatnog kontejnera ili sloja. Pristup funkcionalnostima uređaja se vrši pomoću API-ja operativnog sistema (slika 2).



Slika 2. Izvršavanje native aplikacije [2]

Prednosti *native* aplikacija ogledaju se u mogućnosti upotrebe svih funkcionalnosti koje obezbeđuje operativni sistem, boljim performansama kod zahtjevnijih aplikacija i standardizovanom izgledu aplikacija na određenoj platformi [3].

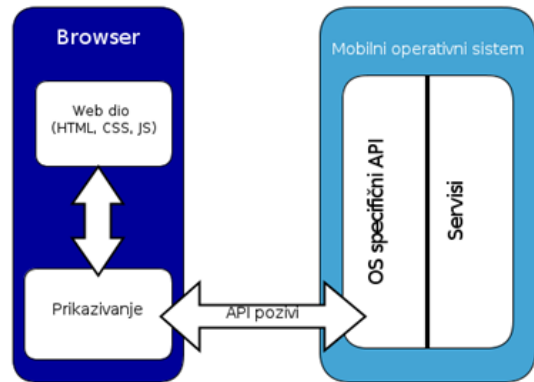
Osnovni nedostatak *native* aplikacije jeste mogućnost njenog korištenja na samo jednoj platformi [3].

### B. Web aplikacije

Web aplikacije zasnovane su na web tehnologijama: HTML5, CSS i JavaScript, uz mogućnost korištenja dodatnih biblioteka i softverskih okruženja kao što je jQuery. Aplikacije se mogu instalirati na uređaj, ali to nije obavezno. Pokretanje web aplikacija vrši se unosom URL adrese, aktiviranjem *hyperlink*-a ili pokretanjem prečice.

Za razliku od *native* aplikacija, web aplikacije ne izvršava direktno operativni sistem uređaja, već se ove aplikacije

izvršavaju u okviru *browser*-a, kao što je prikazano na slici 3 [1].



Slika 3. Izvršavanje web aplikacija [2]

Zbog toga web aplikacija nema pristup cijelom API-ju operativnog sistema, već samo onom dijelu koji je dostupan iz *browser*-a, koji je i sam dio API-ja. Kao što je prikazano na slici 3, web aplikacija se izvršava u dodatnom modulu koji može ostvariti API pozive prema operativnom sistemu.

Osnovna prednost web aplikacija jeste njena prenosivost - jednom napisana aplikacija može se izvršavati na svim platformama. Kako se za razvoj web aplikacija koriste HTML5, CSS i JavaScript, nije potrebno učiti tehnologije koje se koriste za razvoj na određenoj platformi. Ovo dovodi do uštede resursa i vremena. Pored toga, web aplikacije nemaju *update*, a uvijek se koristi aktuelna verzija [3].

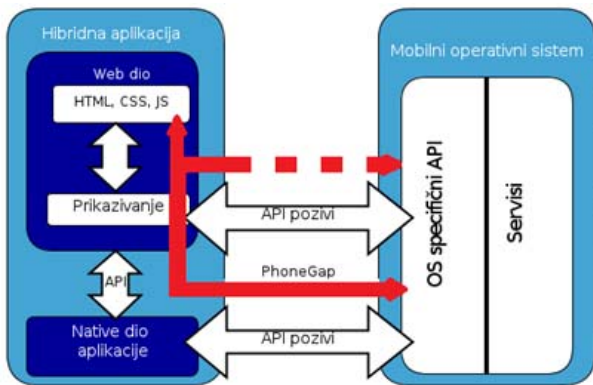
Nedostaci web aplikacija su nešto lošije performanse pri izvršavanju u odnosu na *native* aplikacije, iako je to u posljednje vrijeme sve manje izraženo zbog poboljšanja komponenti samih uređaja, upotrebe JIT (*Just-In-Time*) kompajlera i novih standarda kao što je WebGL (*Web Graphics Library*) za razvoj igrica. Najveći nedostatak su ograničenja u funkcionalnostima koje se mogu realizovati, zbog toga što se ne može pristupiti cijelom API-ju operativnog sistema. Pored toga, web aplikacije se ne objavljuju na trgovinama aplikacija [3].

### C. Hibridne aplikacije

Kombinacijom pristupa razvoju *native* i web aplikacija nastale su hibridne aplikacije. Kod ovakvog pristupa dio aplikacije se razvija kao i u slučaju web aplikacije - u HTML5, CSS i JavaScript-u, dok se pristup API-ju operativnog sistema vrši pomoću dodatnih *native* dijelova, koji su prilagođeni da se mogu koristiti pomoću *web* tehnologija. *Native* dijelovi se mogu samostalno razvijati, a mogu se koristiti gotova rješenja kao što je PhoneGap. PhoneGap i slične softverske platforme vrše apstrakciju *native* servisa, inače nedostupnih web aplikacijama, na uniforman način, i tako ih čine dostupnim na identičan način, na različitim mobilnim operativnim sistemima.

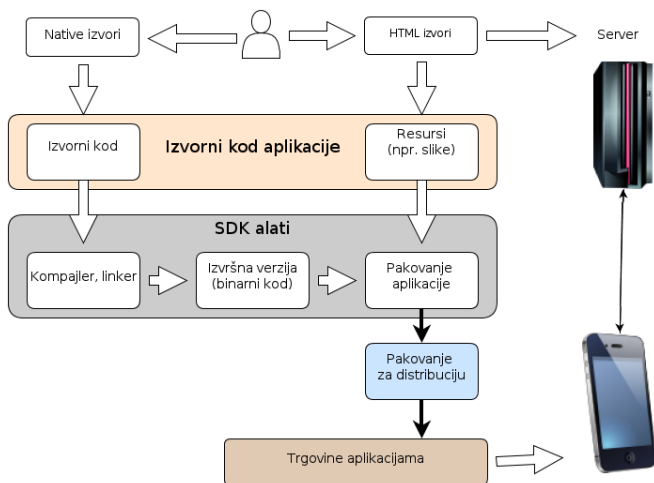
Dio hibridne aplikacije koji se razvija pomoću web tehnologija funkcioniše na isti način kao i kod web aplikacija - pokreće se i izvršava pomoću *browser*-a i ima ograničen pristup API-ju operativnog sistema. *Native* dio aplikacije ima potpun pristup API-ju operativnog sistema, i povezuje se sa

dijelom za obradu web komponenti aplikacije (*browser*-om). Na ovaj način aplikacija ostvaruje *native* funkcionalnosti, što je i prikazano na slici 4 [1].



Slika 4. Izvršavanje hibridnih aplikacija [2]

Na slici 5 prikazan je proces razvoja hibridne aplikacije. Kao što se vidi, on je veoma sličan procesu razvoja *native* aplikacija, uz dodatak da se dio aplikacije razvijen pomoću web tehnologija dodaje u resurse aplikacije, a može se preuzimati i sa servera [1]. Ukoliko je dio aplikacije dostupan preko servera, aplikacija gubi osobinu da je dostupna ukoliko nije povezana na Internet, ali se olakšava promjena dijelova aplikacije.



Slika 5. Razvoj hibridnih aplikacija [2]

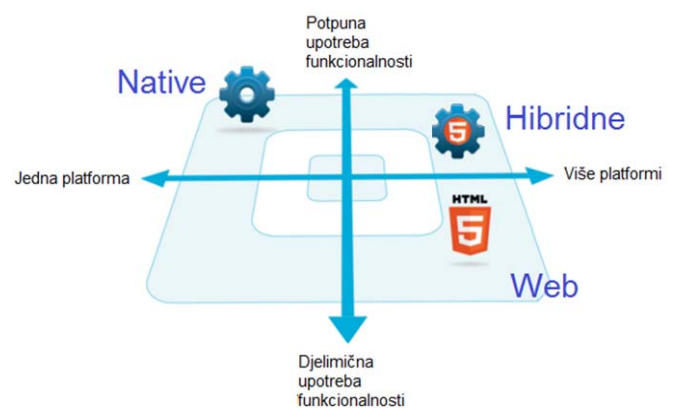
Prednosti razvoja hibridnih aplikacija su mogućnost korištenja cijelog API-ja operativnog sistema, razvijanje aplikacije na uniforman način (pomoću web tehnologija) i mogućnost distribucije na više platformi [3].

Nedostaci se ogledaju u tome što se moraju poznavati tehnologije za razvoj *native* dijelova koje su različite za svaku platformu, ali samo u slučaju kada se ne koristi softversko okruženje kao što je PhoneGap. Bitno je napomenuti da su moguća i ograničenja u pristupu funkcijama uređaja, ukoliko one nisu podržane u *native* dijelu aplikacije [3].

### III. IZBOR VRSTE APLIKACIJE

Svaki od navedenih pristupa za razvoj aplikacija za mobilne uređaje ima svoje prednosti i nedostatke, a izbor određenog pristupa zavisi od namjene aplikacije i funkcionalnosti koje ona treba da ima. *Native* aplikacije se obično koriste za razvoj aplikacija za jednu određenu platformu, sa velikim zahtjevima u pogledu korisničkog interfejsa i upotrebom *native* funkcionalnosti. Web aplikacije se često koriste za razvoj tzv. „probnih“ aplikacija i aplikacija koje se izvršavaju na većem broju platformi, kojima nisu potrebne specifične funkcionalnosti mobilnih uređaja, kojima performanse nisu od ključnog značaja, a potrebna im je stalna veza sa Internetom [1] [3] [4].

Na slici 6 je prikazan odnos prethodno navedenih vrsta mobilnih aplikacija u zavisnosti od broja platformi na kojima se aplikacija može izvršavati, i mogućnosti upotrebe funkcionalnosti uređaja.



Slika 6. Poređenje vrsta mobilnih aplikacija [4]

Sa slike 6 se vidi da hibridne aplikacije omogućavaju upotrebu jedne aplikacije na više platformi uz implementaciju velikog broja funkcionalnosti platforme, za koju se aplikacija razvija. Na ovaj način se postiže upotreba jednog izvornog koda za izradu aplikacija koje rade na različitim platformama bez većih izmjena a sam razvoj je brži i jeftiniji. Zbog toga će u nastavku rada biti razmotren razvoj hibridnih aplikacija pomoću HTML 5 i povezanih tehnologija uz upotrebu PhoneGap softverskog okruženja.

### IV. RAZVOJ HIBRIDNIH APLIKACIJA POMOĆU HTML 5 I PHONEGAP-A

Kao što je navedeno u prethodnoj sekciji, hibridne aplikacije predstavljaju kompromis u pogledu upotrebe jednog izvornog koda za razvoj aplikacije koja će se izvršavati na različitim platformama, a da pri tome aplikacija može da ostvari upotrebu velikog broja *native* funkcionalnosti. Kada se hibridna aplikacija razvija pomoću standardnih web tehnologija i PhoneGap platforme aplikacija se sastoji iz dva dijela. Prvi dio se razvija pomoću HTML5, CSS3 i JavaScript-a, a služi za realizaciju korisničkog interfejsa aplikacije, poslovne logike i pozive PhoneGap funkcionalnosti. Osim navedenih tehnologija

često se koristi i jQuery Mobile<sup>1</sup>. Drugi dio aplikacije je PhoneGap, koji omogućava korištenje funkcionalnosti mobilnog uređaja. Struktura aplikacije realizovane pomoću navedenih tehnologija prikazana je na slici 7.



Slika 7. Struktura hibridne aplikacije

HTML5 je opisni jezik koji se koristi za razvoj web stranica i mobilnih aplikacija, pomoću koga se definiše izgled stranice, raspored komponenti i sadržaj koji se prezentuje. Od posebnog interesa za razvoj mobilnih aplikacija su geolokacijski API, keširanje aplikacija, čuvanje lokalnih podataka kao i upotreba multimedijalnih elemenata. Uz HTML5 i povezane tehnologije moguće je koristiti i dodatne alate kao što su jQuery Mobile, Sencha Touch<sup>2</sup> (prvo HTML5 softversko okruženje za razvoj mobilnih aplikacija), Worklight<sup>3</sup> i Google Web Toolking<sup>4</sup>.

CSS3 se koristi za opis izgleda i uređenje elemenata na HTML stranici. Predstavlja proširenje prethodne verzija CSS-a i podržava sve atribute iz starijih verzija. CSS3 omogućava zamjenu kompleksnijih JavaScript animacija, transformacija i vizuelnih efekata novim jednostavnijim elementima.

JavaScript je besplatan i platformski neutralan skriptni programski jezik, namijenjen prvenstveno programiranju na klijentskoj strani. JavaScript je zasnovan na ECMA-262 standardu. Napisani kod se interpretira, a definiše se u <script> tagovima na proizvoljnom mjestu unutar HTML dokumenta, ili se izdvaja u posebne fajlove. U mobilnim aplikacijama se tipično koristi za dinamičku promjenu sadržaja stranice, implementaciju dijela poslovne logike aplikacije, komunikaciju sa drugim servisima i pozive PhoneGap funkcionalnosti.

## V. PHONEGAP

PhoneGap je *open-source* softversko okruženje za razvoj aplikacija za mobilne uređaje na različitim platformama koristeći HTML5, CSS i JavaScript. Aplikacije razvijene pomoću PhoneGap-a su hibridne, izvršavaju se u okviru *webview*<sup>5</sup>-a, a imaju pristup API-ju operativnog sistema i uređaja.

Nastao je u avgustu 2008. godine a prvi put je predstavljen na iPhoneDevCamp-u u San Francisku. PhoneGap je razvijen u kompaniji Nitobi Software koju je 4. oktobra 2011. godine kupila kompanija Adobe, i od tada je PhoneGap dio *Apache*

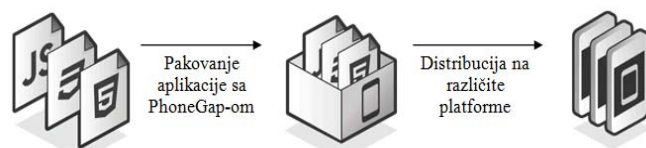
*Software Foundation* pod nazivom *Apache Cordova*. Zaštićen je Apache 2.0 licencom [5].

Pomoću PhoneGap-a mogu se razvijati aplikacije za najpopularnije platforme: iOS, Android, BlackBerry, Windows Phone, Palm WebOS, Bada, Symbian i Tizen [6]. Upotrebom PhoneGap-a moguće je razviti aplikacije bez poznavanja načina razvoja aplikacija na ciljnoj platformi i programskih jezika koji se za to koriste. Aplikacije se pišu jednom, a distribuiraju na sve platforme bez značajnijih izmjena. Ovakav pristup skraćuje vrijeme potrebno da se aplikacija razvije i smanjuje troškove razvoja.

Aplikacije se razvijaju pomoću tehnologija koje se primjenjuju za razvoj web aplikacija: HTML5, CSS i JavaScript. Zbog toga nije potrebno poznavati programske jezike koji se koriste za razvoj *native* aplikacija kao što su Java, Objective-C ili drugi. Kako bi se omogućila upotreba funkcionalnosti mobilnih uređaja, PhoneGap posjeduje univerzalni API za pristup funkcionalnostima uređaja koji se može koristiti na svim podržanim platformama.

PhoneGap je kontejner u kome se nalaze odgovarajuće biblioteke i xml fajlovi koji omogućavaju izvršenje aplikacija na uređaju i upotrebu funkcionalnosti uređaja. Ukoliko se ne koriste funkcionalnosti mobilnog uređaja, napisanu aplikaciju je moguće otvoriti pomoću običnog *browser*-a i koristiti kao običnu web aplikaciju. Ukoliko se u aplikaciji koriste funkcionalnosti iz PhoneGap-ovog API-ja, napisana aplikacija postaje hibridna aplikacija, i pozivi API-ja su jedina razlika u odnosu na razvoj web aplikacija. PhoneGap-ov API nudi interfejs napisan u JavaScript-u. Dakle, razvoj korisničkih interfejsa (ekrana) i dodavanje funkcionalnosti aplikacije je moguće uraditi u proizvoljnom HTML editoru.

Da bi se omogućilo izvršenje aplikacije na mobilnom uređaju, neophodno je zapakovati urađene HTML5 stranice, sa odgovarajućim CSS i JavaScript fajlovima i PhoneGap bibliotekama za platformu na kojoj se aplikacija treba izvršavati. Nakon toga, pravi se izvršna verzija aplikacije koja se instalira na uređaj. Osnovne faze izrade mobilne aplikacije pomoću PhoneGap-a su prikazane na slici 8.



Slika 8. Proces razvoja aplikacija pomoću PhoneGap-a

HTML5 i JavaScript ne mogu direktno pristupiti API-ju uređaja pa je neophodno koristiti dodatne biblioteke koje to omogućavaju. PhoneGap obezbeđuje pristup API-ju uređaja pomoću JavaScript interfejsa kojem se pristupa iz programskog koda aplikacije. Ovaj interfejs poziva *native* realizaciju određene funkcionalnosti. *Native* realizacije se nalaze u PhoneGap biblioteci za svaku pojedinačnu platformu, i nije potrebno vršiti izmjene ovih biblioteka da bi se koristile na različitim platformama. U PhoneGap-u je podržan veliki broj funkcionalnosti, ali one nisu jednako implementirane na svim platformama, prvenstveno zbog specifičnosti određenih

<sup>1</sup> <http://jquerymobile.com/>

<sup>2</sup> <http://www.sencha.com>

<sup>3</sup> <http://www-01.ibm.com/software/mobile-solutions/worklight/>

<sup>4</sup> <https://developers.google.com/web-toolkit/>

<sup>5</sup> Dio mobilnog operativnog sistema koji omogućava prikaz web sadržaja

platformi i činjenice da neke funkcionalnosti nisu podržane na samim uređajima. Za korištenje funkcionalnosti API-ja neophodno je definisati dozvole za svaku platformu. Osim toga, implementacija određenih funkcionalnosti se može razlikovati u korištenju određenih metoda ili parametara [7].

Kako bi se omogućila veza između aplikacije i određene funkcionalnosti platforme na kojoj se izvršava aplikacija, a koja nije dio PhoneGap API-ja, moguće je koristiti *plugin-e*. *Plugin-i* se sastoje iz interfejsa koji je napisan u JavaScript-u (koji se koristi na svim platformama) i *native* implementacije funkcionalnosti za svaku pojedinačnu platformu [7]. Na ovaj način su implementirane i sve Cordova funkcionalnosti. Poziv funkcionalnosti koja je realizovana *plugin-om* se vrši preko JavaScript interfejsa, koji komunicira sa *native* implementacijom preko *cordova.exec* funkcije. Da bi se razvili *plugin-i* neophodno je poznavati tehnologije za razvoj funkcionalnosti za svaku platformu što smanjuje univerzalnost primjene web tehnologija za razvoj mobilnih aplikacija.

Razvoj hibridne aplikacije, koja kao *native* dio koristi PhoneGap vrši se na dva načina [8]. Prvi način je razvoj aplikacija pomoću SDK za svaku od platformi za koje se razvija aplikacija. To podrazumijeva da se na računar, na kome se razvijaju aplikacije, instalira SDK za određenu platformu i preporučeno razvojno okruženje kako bi se mogao kreirati odgovarajući projekat, na način kako se prave *native* aplikacije. Zatim se, uz promjenu određenih parametara dodaju PhoneGap biblioteke, HTML5, CSS i JavaScript fajlovi. Izvršna verzija aplikacije se dobija pomoću SDK alata. Bitno je napomenuti da je podešavanje projekta različito za svaku platformu. Drugi način pakovanja aplikacija je pomoću PhoneGap-ovog servisa u oblaku (eng. *cloud*) kome se dostavljaju HTML5, CSS i JavaScript fajlovi, i na osnovu njih se, u oblaku, pakuje aplikacija za željenu platformu. Ovaj servis se zove PhoneGap Build<sup>6</sup>. Postoji nekoliko načina korištenja ovog servisa, a jedan od njih je besplatan.

## VI. ZAKLJUČAK

Razvoj aplikacija na PhoneGap platformi omogućava da se aplikacija razvije za relativno kratko vrijeme upotrebom standardizovanih tehnologija, pri čemu se isti izvorni kod može upotrebiti za pakovanje i distribuciju aplikacije na više različitih platformi. Osim toga, ovako dobijen izvorni kod aplikacije je jednostavan za održavanje i prilagođavanje. Aplikacije razvijene pomoću PhoneGap-a mogu koristiti veliki broj *native* funkcionalnosti uređaja. Upotrebom HTML5 i CSS3, moguće je napraviti potpuno funkcionalne aplikacije sa „bogatim“ grafičkim interfejsom, pri čemu je proces dizajniranja aplikacija veoma jednostavan. Pri dizajniranju je moguće koristiti programe za grafičku izradu HTML stranica. Bitno je napomenuti i da je PhoneGap-ov API veoma dobro dokumentovan, pri čemu su navedene preporuke i ograničenja

<sup>6</sup> <https://build.phonegap.com/>

u korištenju svake funkcionalnosti za svaku podržanu platformu [9].

Ipak, proces razvoja i distribucije aplikacija na različite platforme, kao i korištenja funkcionalnosti samih platformi nije tako idealan kako se čini. Aplikacije koje se razvijaju za izvršavanje na više platformi često izgledaju isto na svim platformama što znači da nisu „domaće“ ni na jednoj platformi, pa je potreban dodatan rad programera kako bi se aplikacija vizuelno prilagodila određenoj platformi. Zbog činjenice da veliki broj HTML5 funkcionalnosti nije podržan na svim platformama, nije moguće u potpunosti iskoristiti sve mogućnosti za razvoj, a opcije koje su iskorištene u aplikaciji namijenjenoj za jednu platformu ne moraju biti podržane na nekoj drugoj platformi. Osim toga, funkcionalnosti koje se nalaze u PhoneGap-u nisu podržane na svim platformama, a za neke funkcionalnosti je potrebno uraditi dodatna podešavanja da bi se mogle koristiti. Za aplikacije kod kojih su veoma važne performanse izvršavanja, preporučuje se *native* način razvoja [9].

Na kraju, bitno je još jednom naglasiti da svaki od prikazanih načina razvoja mobilnih aplikacija ima određene prednosti i nedostatke, te da ne postoji najbolji način razvoja. Isto tako, ukoliko se nastavi sa ubrzanim razvojem tehnologija koje se koriste pri razvoju hibridnih aplikacija, ovaj način razvoja mobilnih aplikacija bi mogao postati dominantan.

## LITERATURA

- [1] “Native, web or hybrid mobile-app development”, IBM Corporation, april 2012.
- [2] <http://www.slideshare.net/tessierv/ibm-worklight-nativeweborhybridmobileappdevelopment>, posjećivano novembra 2012.
- [3] [https://blogs.oracle.com/fusionmiddleware/entry/developer\\_s\\_corner\\_developi\\_mobile](https://blogs.oracle.com/fusionmiddleware/entry/developer_s_corner_developi_mobile), posjećivano novembra 2012.
- [4] [http://wiki.developerforce.com/page/Native,\\_HTML5,\\_or\\_Hybrid:\\_Understanding\\_Your\\_Mobile\\_Application\\_Development\\_Options](http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options), posjećivano novembra 2012.
- [5] John M. Wargo, “PhoneGap Essentials”, Addison-Wesley Professional, 2012.
- [6] <http://phonegap.com>, posjećivano novembra 2012.
- [7] <http://docs.phonegap.com>, posjećivano novembra 2012.
- [8] Thomas Myer, “Beginning PhoneGap”, John Wiley & Sons, Inc., 2012.
- [9] Igor Dujlović, “Razvoj HTML 5 mobilnih aplikacija korištenjem PhoneGap platforme”, diplomski rad, 2012.

## ABSTRACT

This paper presents theoretical principles of development of native, web and hybrid mobile applications. Especially, the possibility of hybrid mobile applications development using HTML5 and related technologies with usage of PhoneGap platform is considered.

## DEVELOPMENT OF HYBRID MOBILE APPLICATIONS USING PHONEGAP PLATFORM

Igor Dujlović, Zoran Đurić