Validation of implicit expectations on mobile applications

Natalia Andriano [1], Mauricio Silclir[1], Diego Rubio[1]

[1](nandriano, 47920, drubio) @frc.utn.edu.ar
Laboratorio de Investigación y Desarrollo en Ingeniería y Calidad del Software, Universidad Tecnológica Nacional – Facultad Regional Córdoba,
Ciudad Universitaria, Maestro M. Lopez esq. Cruz Roja s/n, 5000, Córdoba, Argentina

Abstract. The increasing amount of active software users together with the emergence of free access to multi-screen devices (being mobile phone the leaders in this area), have started to cause a growing phenomenon on the user expectations for every software product (no matter what it is about). Identifying these expectations will benefit the development of products (for example, in estimating more accurately the size and cost of a product), while will improve the reception of it by end users. This research verifies and validates a list of implicit items, deemed common to many products, against the top 10 worldwide most used apps.

1. Introduction

The increasing amount of active software users, together with the emergence of free access to multi-screen devices (being mobile phone the leaders in this area), have started to cause a growing phenomenon on the user expectations for every software product.

These new users' era generates new software development challenges: the identification of users' implicit expectations is one of them [1] [2]. As stated in [3] a preliminary list of implicit expectations was created and validated. This was achieved through bibliographical research on multiscreen development and then validated during interviews designed for assessing this list into real multiscreen product. The objective of the list is to be an additional input for new projects when defining their scope based on what current multiscreen users expects.

As a second phase for the previous research, further research was conducted to identify which are the most used apps in the world taking into account the two main mobile platforms. After building a ranked list of the top 10 most used apps, acceptance criteria and test cases were defined for each implicit expectation, so every top 10 app could be objectively evaluated against those implicit expectations.

The aim of this paper is to validate the list of "implicit expectations" identified during phase 1 [3] by evaluating the top 10 mobile applications and analyze the level of coverage they have over them. The objective of doing so, is to provide a wider validation of the identified implicit requirements not only for local apps but to most worldwide used mobile apps.

It is assumed that if implicit expectations are supported by these apps, therefore these expectations are also part of the characteristics end users look for in mobile applications.

2. State of the art

Over the years and since its inception, the software industry has evolved along with new available technologies and devices. A significant amount of great ideas and available knowledge have been identified, with focus on how to effectively develop software, from traditional structured programming [4] to organized work through the best modern development practices, such as agile methodologies [5], iterative models [6], and software development managed by risk and quality [7]. This evolution on the engineering practices has occurred in product design, processes and tools that support their creation [8] [9] [10].

Recently, there has been a new technology paradigm shift that introduces new challenges for software development. As Business Insider mentions in its study [11], mobile devices and tablets sales have exceeded computers' numbers. Likewise, Gartner, in its 2018 report [12] shows that the number of mobile devices sold is increasing, with a prediction of growth in the industry. In addition, a significant percentage of people is using "the mobile web" from the comfort of their home, as reported in [13] [14] [15] [16] [17] [18] [19]

Thus, user experience offered by each product starts to have more relevance, and because of this, different development methods and strategies have begun to raise new concepts and technologies. Many companies and organizations have responded to this movement by creating mobile tailored versions of their websites. However, the sheer number of devices makes it difficult to design for all the available screen sizes.

There is, therefore, a need for websites to be adaptive and accessible regardless of the device used [20]. One of the responses to this requirement is a concept known as "responsive web design" (RWD), introduced by Ethan Marcotte [21]. This new approach works with the web page rendering logic, to adapt the way it looks according to the device and screen size in which it's being shown. Recent work also considers the viewer proximity as part of the viewing context as an extension for RWD. This concept has become, in short time, a common method to create websites that automatically adapt the layout to different screen sizes. The number of mobile devices with web access is constantly increasing and screen sizes are becoming both smaller and larger at the same time [21].

One of the critiques brought up is the poor performance and long load times in responsive websites [22] [23]. Both Google [24] and Yahoo [25] mentioned that, in order to avoid a dramatic increase in load time, it is important to reduce the number of requests to the server.. Joel Nandorf [26] has identified the following techniques to cope with poor performance:

Image Optimizer, Responsive Images, Optimizing JavaScript and CSS, Lazy Loading. Gomez [27] reports that 60% of web users expect a website to load on a mobile phone in 3 seconds or less and that 74% are only willing to wait 5 seconds or less before leaving the site. Moreover, half of the mobile web users are unlikely to return to a website that they had trouble accessing to.

Besides the ability of an application to be responsive to the screen size, there is also a diversification in the way and the contexts in which these mobile devices are used. Nicola Thompson [28] and Luke Wroblewski [29] comment on the diverse use of different devices and the number of users for each of them. They also mention a great growth in the use of these devices when doing almost any activity (shopping, watching TV, shopping, etc.). Nowadays, users not only expect an app to run on every screen, but also for each screen to present specific functionalities more suitable of being executed in that particular screen. In Kaavya Seethamraju [30] words, we are facing the need of having mobile skills to grant users the information requested where and when they request it. This behavior is described by Christophe Stoll [31] into a set of characteristics (coherence, synchronization, screen sharing, device shifting, complementary and simultaneity) that led to the evolution of different multi-screen development patterns,.

Similarly, Michal Levin [32] proposes 3 terms to take into account for multi-screen applications development:

1) Consistent experience: the experience is replicated in different devices in terms of content, flow, structure and main behavior, some adjustments are made to accommodate to the specific characteristics of the devices (mainly the size of the screen and the interaction model);

2) Continuous experience: the experience moves between devices, through the same action or following a sequence of actions and

3) Complementary experience: the devices can be completed between them (information / functionality) creating a new type of experience as a connected group, there are two types of relationships: collaboration and control.

Main platforms usually define a set of guidelines that help developers to create applications that are consistent for each of the platforms. For iOS[1], there exist a set of guidelines for creating human interfaces that are suitable for this Apple[2] platform [33], and that will ensure compatibility and better usability in its users community. In the case of Android[3], Google[4] has developed a design language known as “Material Design”. Material

[1] https://www.apple.com/ios/

[2] https://www.apple.com/

[3] https://www.android.com/

[4] https://www.google.com/

Design is a visual language that synthesizes the classic principles of good design with the innovation of technology and science [34].

In this context, new software development challenges start to raise, where software development companies and engineers need to understand not only the explicit requirements pointed out by customers and end users, but also their implicit expectations that this growing base of users have. Companies also need to respond to this expectation in a cost-effective manner to ensure quality of experience, understanding and developing methods, tools and processes capable of achieving such objectives. [3] explore and present a list of base implicit expectations that should be considered in the development of every new modern application.

3. Methods and Materials

As part of a much broader investigation conducted by LIDICALSO[5] this paper addresses the second phase of the "SIUTNCO0004865 - Characterization of engineering methods, tools and resources for the development of complexity-growing modern software" research project.

The first phase was focused on the definition of a list of implicit expectations for modern software, from a business perspective. Those results were documented and published in the report Implicit expectations on multiscreen software products [3]

This second phase is to validate the list of implicit expectations resulting from phase 1. In order to do so, the following steps were completed:

1. Bibliographic research on the top ranked mobile applications worldwide
2. Identification and definition of an analysis/assessment criteria to evaluate the apps found against the implicit expectations defined in [3].
3. Validation of the approach for evaluating applications, by running a pilot over one of the top ranked applications determined by step 1 above.
4. Identification of improvements to the analysis/assessment criteria and incorporation of those improvements to the analysis
5. Complete assessment of the list of implicit expectations defined in [3], for the top 10 ranked mobile applications

[5] Laboratorio de Investigacion y Desarrollo en Ingenieria y Calidad de Software,» [En línea]. Available: http://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/

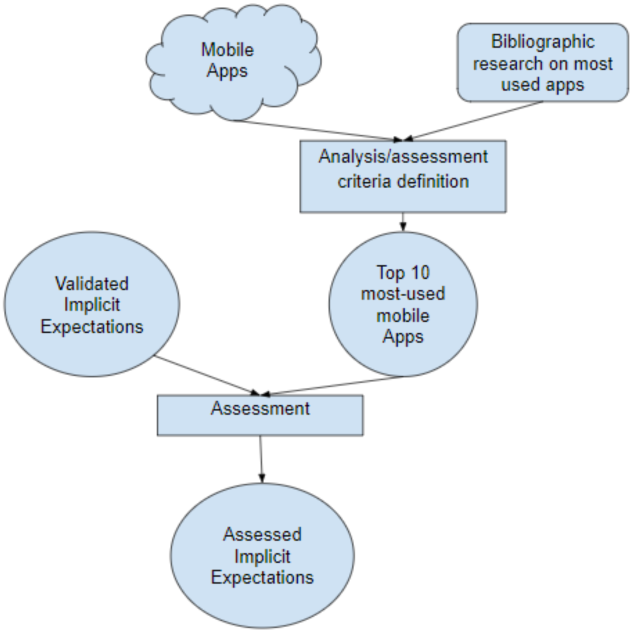
Mobile Apps
Bibliographic research on most used apps
Analysis/assessment criteria definition
Validated Implicit Expectations
Top 10 most-used mobile Apps
Assessment
Assessed Implicit Expectations

Fig. 1. Second phase flow.

Raw data is published and can be found in LIDICALSO **[35]**

Step 1: Bibliographic research on the top ranked mobile apps worldwide

As a starting point, bibliographic research has been conducted in order to identify which are the top 10 ranked mobile applications worldwide. The following considerations were made during this research:

- **Publication's time frame**: only publications made from 2018 and 2019 were taken into account. The reason for this limitation is that the intention of the current research is to identify the *current* top 10 mobile apps, and as technology keeps changing, considering older publications may lead to inaccurate results.
- **App's ranking**: not all publications showed the same range for specifying the ranking (some of them stated the 10 most used app, some of them the 15 most used apps, and so on). **Publication's privilege**: it was also taken into account the prestige (amount of readers and publications) of the publications under evaluation. A weight has been defined in order to differentiate if the publication has been done

by a known author or not, and this weighted value was included in the normalized ranking

  - Personal blog or website = 10
  - Specialized magazines and sites = 4
  - Researches and important reports (e.g. Gartner) = 1

A total of 20 publications have been analyzed and a list of the top 10 mobile apps was created. As mentioned before ranking for the apps was maintained and publication ́s privilege was assessed. After that, app's weight was calculated as follows:

Normalized app's weight = app's ranking * publication's privilege

With this formula, the closer the value is to 1, the better the app's normalized weight (better ranked and published in sources with more reputation).

For each publication, the following information was gathered:

- Publication source, title and privilege
- Publication date
- Platform considered in the publication (e.g. Android vs iOS)
- Global market considered (i.e. worldwide comparison or focused on a certain region)
- Category (e.g. any app, or focused on certain domains, like games or finance)

And then, the actual apps' weight determinations:

Table 1. App's weight calculation process example.

Source	Publication date	Publication privilege	App	Rank	Normalized app weight
The Ultimate List of Most Used Apps In The World 2019 [36]	Mar-19	10	**Instagram**	4	40
Top 10 Most Popular Apps 2018 [37]	Nov-18	4	**Spotify**	8	32
Apple reveals the most popular iPhone apps of 2018 [37]	Dec-18	1	**Facebook**	5	5

After all app's weight has been calculated, the 10 top most used were obtained, using the average of the "normalized apps weight".

Table 2. Top 10 mobile applications worldwide

App	Weight
Waze	8
Yahoo Mail	56
Facebook	88
WeChat	100[6]
Uber	114
Facebook Messenger	123
Google Chrome	133
Spotify	134
Snapchat	136
Instagram	250

[6] Instagram; Snapchat; Spotify; Google Chrome; Facebook Messenger; Ubber; WeChat; Facebook; Yahoo mail

Step 2: Identification and definition of an analysis/assessment criteria to evaluate the apps found

Main input to this step is the list of implicit expectations generated on previous phase [3]. The outcome of that phase was a verified list of implicit expectations every application must comply with.

As implicit expectations were written as high level requirements, it was very important to define acceptance criteria for each one of them so the assessment can be done in an objective way. For this purpose, a QA expert identified the acceptance criteria for each implicit expectation.

> **Acceptance criteria definition:** The acceptance criteria (AC) of a user story consists of a set of test scenarios that are to be met, for the user story to be assessed as complete. Acceptance criteria are highly important and required in Agile, because the criteria indicate what exactly the product owner (PO) expects and what the Scrum team needs to accomplish. Test scenarios are captured as part of acceptance criteria, signifying the behavior of a feature after it is implemented [38].

With the acceptance criteria, it was possible to define in an objective and verifiable manner the implicit expectations so as to reduce ambiguity and to provide the assessor concrete information on what and how to test each implicit expectations for each app.

Definition of the acceptance criteria for each requirement

This section presents that description for each of the expectations that will be validated later on.

Table 3. Acceptance criteria definition example

Expectations	Acceptance criteria
A.1 The application layout shall adapt to multiple device screens	1. All main screens of the app are properly displayed in a web browser of a desktop computer 1.1. When resizing the screen, the layout is updated to fit all elements in the new size of the window 2. All screens of the app are properly displayed in a web browser of a smartphone or tablet, by either responding to the new size of the screen, or by automatically adapting the content (type and amount) to the new device. 2.1 It's mobile and tablet friendly 3. When accessing the website (in case of web apps), it keeps the same URL than when opening the same website from a mobile device. i.e. No "mobile" word as part of the URL
A.3 The application shall work in all the main platforms	The application is available in the 3 main platforms • iOS • Android • Web

Step 3: Validation of the approach for evaluating applications, by running a pilot over one of the top ranked applications determined by step 1 above

Before starting the assessment for the complete list of the most used apps, a pilot was run over Facebook [39] and Instagram [40]. In order to perform the assessment a score was defined:

- Score 0 = Expectations does not apply to the product
- Score 1 = Expectations applies to the product, but it’s not being tracked as part of the product’s scope
- Score 2 = Expectations has been identified for the product, and its part of its current scope

The applications were assessed taking into account each implicit requirement and a score was assigned to each one of them. The following table shows an example of some of the verified implicit expectations for Facebook [39] and Instagram [40]:

Table 4. Facebook and Instagram verified implicit expectations example.

Implicit expectation	Facebook	Instagram	Comments
A.1 The application layout shall adapt to multiple device screens	2	2	IPHONE 7 plus: a) User is able to log in --> Ok b) User is able to perform a simple search --> Ok search done by person's name c) User is able to perform a CRUD of a particular entity --> Ok create a new user, update user data, delete user d) Verify that layout has adapted to the screen. .--> Ok

As shown in the picture comments were also added to each assessment in order to provide more insight / information on what was considered when setting the score.

Step 4: Identification of improvements to the analysis/assessment criteria and incorporation of those improvements to the analysis

After executing the pilot, analysis was performed to identify inconsistencies and discrepancies in the way the implicit expectation was verified. In order to reduce the ambiguity even more on what and how to perform the assessment, it was decided that formal test cases needed to be defined.

> **Software testing** A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item [41]

An experienced QA engineer created test cases to verify each one of the identified acceptance criteria for the implicit expectations. A peer review [42] was performed to assure that both assessors understood what and how to assess the apps. The following steps were executed:

1. Test case creation based on validated acceptance criteria for each implicit requirement
    a. Input and expected output were identified
    b. Steps were defined
2. Test case review conducted using peer review method
3. Test cycle defined
    a. Test cycle created by each platform
4. Test cycle execution done
5. Test results analysis conducted.

All test cases were defined as manual black-box functional testing [43] [44] [45].

Step 5: Complete assessment for the top 10 ranked mobile applications

The assessment for the 10 most used apps worldwide was made. Applications were installed in several devices (both iOS and Android [46]) and predefined test cases were executed. Also bibliographic evidence for specific features was registered.

The following mobiles were used to complete the assessment: iPhone 7plus, Samsung A20, Motorola G5 Plus.

Table 5. Top 10 mobile applications worldwide – Expectations score

App	Weight	Expectations score
Waze	8	21
Yahoo Mail	56	23
Facebook	88	24
WeChat	100[7]	21
Uber	114	24
Facebook Messenger	123	23
Google Chrome	133	26
Spotify	134	27
Snapchat	136	24
Instagram	250	23

[7] Instagram; Snapchat; Spotify; Google Chrome; Facebook Messenger; Ubber; WeChat; Facebook; Yahoo mail

Results

Taking the validated list of implicit expectations from phase 1 [3] and the identification of the 10 most used apps in the world in 2019 as main inputs, acceptance criteria was defined for each item and test cases were executed to validate if those implicit expectations are addressed in those apps.

The score for each requirement on each of the apps was set as follows:

2 points → the requirement is completely fulfilled by the app

1 points → there is a partial implementation on the requirement, for the app

0 points → the app doesn't implement the requirement

The following table shows the final score for each requirement, sorted by total score obtained (a value between 0 – the implicit expectation doesn't apply to any of the apps evaluated - and 20 – the implicit expectation fully applies to all of the apps evaluated), being 20 the highest score to be obtained:

Table 5. Implicit expectation evaluation into the 10 most used moble apps.

Implicit expectation	Total Score
A.2 The application shall adapt images that are being shown depending on the size of the screen of the current device	20
A.3 The application shall work in all the main platforms	20
G.1. The application shall protect user's data.	20
G.2. The application shall maintain user's data backup (persistency)	20
H.1 The application shall be updatable	20
A.1 The application layout shall adapt to multiple device screens	19
A.5 The application shall recommend the download of the mobile application	19
D.1 The application shall be self-explained, intuitive enough to allow the user to understand its features and how to use it in a first glance	19
E.1 The application shall be accessed within 2 seconds from the user's request	19
A.7 The application shall provide the user the ability to analyze data through different filters and queries	16
F.2 The battery utilization of the application shall not exceed a predefined value of the total battery consumption of the device	12
C.1 The application shall be accessible to all users, regardless any kind of disability, such as visual impairment, auditory impairment, motor or dexterity impairment	11
D.2 The application shall follow the standard graphic and usability design patterns for each software platform	11
F.1 The memory consumption of the application shall not exceed a predefined value of the total available memory of the device	11

Fourteen (14) implicit expectations were verified and validated.

Based on the assessment it was found that a subset of 5 implicit expectations were fully implemented in all 10 evaluated apps.

Subset 1: items fully implemented in all evaluated apps

- A.2 The application shall adapt images that are being shown depending on the size of the screen of the current device
- A.3 The application shall work in all the main platforms
- G.1. The application shall protect user's data.
- G.2. The application shall maintain user's data backup (persistency)
- H.1 The application shall be updatable

4 out of 14 implicit expectations were at least partially implemented in all 10 evaluated apps.

Subset 2: items implemented at least partially in all evaluated apps

- A.1 The application layout shall adapt to multiple device screens
- A.5 The application shall recommend the download of the mobile application
- D.1 The application shall be self-explained, intuitive enough to allow the user to understand its features and how to use it in a first glance
- E.1 The application shall be accessed within 2 seconds from the user's request

Finally, in the last 5 items there were cases where the expectation was not implemented in some of the evaluated apps.

Subset 3: items only implemented (full or partial) in some of the apps

- A.7 The application shall provide the user the ability to analyze data through different filters and queries
- C.1 The application shall be accessible to all users, regardless any kind of disability, such as visual impairment, auditory impairment, motor or dexterity impairment
- D.2 The application shall follow the standard graphic and usability design patterns for each software platform
- F.1 The memory consumption of the application shall not exceed a predefined value of the total available memory of the device
- F.2 The battery utilization of the application shall not exceed a predefined value of the total battery consumption of the device

Conclusions

There is a huge increment on the usage of modern-multi screen applications, most of them mobile, among current users. Those users take as granted that when installing a mobile application there are some functionality that will be there, independently from the type of application being installed.

Based on this context, a list of implicit expectations was identified, validated and then verified. As part of the assessment a top 10 worldwide most used mobile apps was generated. In order to generate that list bibliographic research and some criteria for ranking the apps all was created.

From the analysis, it's possible to conclude that all implicit expectations listed in the "Subset 1: items fully implemented in all evaluated apps" category must be part of the requirements for every new mobile application to be developed.

The same conclusion can be made for the items in "Subset 2: items implemented at least partially in all evaluated apps", since all those implicit expectations are at least partially implemented in all the evaluated mobile apps. Hence, they must be part of the initial requirements for every new mobile app to be developed.

Finally, there is a list of implicit expectations, listed in "Subset 3: items only implemented (full or partial) in some of the apps", that, even though they're not dimmed mandatories, it's possible to state that they should be listed at least as nice-to-have items for every new mobile application to be developed, since those items are part of most than half of the top 10 mobile apps evaluated.

Future works

As part of the new software development challenges stated previously, there are a few new lines of research that could be followed.

There were some areas of expectations that seemed to be more applicable to some products than others. Generating some sort of taxonomy of product types with the subset of implicit expectations that apply to each type of product could be generated, so as to make it easier to apply the list to each particular project.

Based on a list of expectations, there would be room to research on how to actually implement each of the items in it, looking for reusable solutions that could be applied to all new products, so as to shorten the time to market and reduce costs. This would include characterizing methods, tools and resources that help reduce development costs and at the same time, complying with minimum software engineering and user's quality level.

Finally, it will be important to understand what are the implications / impacts on the users when an implicit expectation is not present in the apps they are installing

Bibliography

[1] N. Wolfram., «Designing for Multiscreen — Think Multiscreen,» [En línea]. Available: https://medium.com/@wolframnagel/title-designing-for-multiscreen-92d14a67749.

[2] N. Wolfram, «Multiscreen UX Design 1st Edition,» ISBN: 9780128027509. Paperback ISBN: 9780128027295, 2016.

[3] M. S. D. R. Natalia Andriano, «Implicit expectations on multiscreen software products.,» de *CONAIISI ISSN 2347-0372*, 2018 .

[4] E. W. D. C. A. R. H. Dahl, Structured Programming Academic Press, 1972.

[5] Ambler, Agile Modeling Best Practices, 2013.

[6] M. Rouse, «Iterative Development,» 2008. [En línea]. Available: https://searchsoftwarequality.techtarget.com/definition/iterative-development.

[7] SPEM, «Software & Systems Process Engineering Meta?Model Specification,» 2008. [En línea]. Available: https://www.omg.org/spec/SPEM/About-SPEM/..

[8] J. Humble, «Continuous Delivery,» 2017. [En línea]. Available: http://continuousdelivery.com/.

[9] C. Huang, «Software and Systems Traceability ISBN 978447122395,» 2012.

[10] J. Humble y D. Farley, «Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation,» Addison? Wesley Signature Series (Fowler)) ISBN?13: 978?0321601919 ISBN?10:0321601912.

[11] B. Insider, «Business Insider,» 2014. [En línea]. Available: https://www.businessinsider.com/what-the-internet-of-things-will-need-to-really-work-2014-5.

[12] Gartner, «Gartner Top 10 Strategic Technology Trends for 2018,» 2018. [En línea]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/.

[13] T. F. Cast, «The Fone Cast,» 2017. [En línea]. Available: https://thefonecast.com/News/EntryId/3602/Mobile-shopping-is-popular-when-watching-TV-says-Orange-UK-research.

[14] L. Wroblewski, «When & Where Are People Using Mobile Devices?.,» Febrero 2011. [En línea]. Available: https://www.lukew.com/ff/entry.asp?1263.

[15] C. S. Aryal., «Design Principles for Responsive Web,» Diciembre 2014. [En línea]. Available: http://www.theseus.fi/bitstream/handle/10024/85995/The%20Design%20Principles%20for%20Responsive%20Web%20Final%20version%203.pdf;jsessionid=.

[16] D. Raggett., «Responsive Design.,» 2013.. [En línea]. Available: https://www.w3.org/2013/Talks/responsive-design.pdf.

[17] E. Enge., «Mobile vs Desktop Traffic in 2019.,» April 2019. [En línea]. Available: https://www.stonetemple.com/mobile-vs-desktop-usage-study/.

[18] C. Petrov., «52 Mobile vs. Desktop Usage Statistics For 2019 [Mobile's Overtaking!].,» March 2019. [En línea]. Available: https://techjury.net/stats-about/mobile-vs-desktop-usage/.

[19] Statista, «Percentage of all global web pages served to mobile phones from 2009 to 2018,» 2019. [En línea]. Available: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/.

[20] J. Nandorf, «Responsive Web Design - Evaluation of Techniques to Optimize Load Time,» [En línea]. Available: https://pdfs.semanticscholar.org/3050/a79337cd73a501cb390f905dde9bcc97dbbf.pdf.

[21] E. Marcotte. [En línea]. Available: https://alistapart.com/article/responsive-web-design/.

[22] J. Grigsby, «CSS Media Query for Mobile is Fool's Gold,» 2010. [En línea]. Available: https://cloudfour.com/thinks/css-media-query-for-mobile-is-fools-gold/.

[23] G. Podjarny, «Responsive and fast,» [En línea]. Available: https://www.akamai.com/us/en/multimedia/documents/secure/responsive-and-fast-implementing-high-performance-responsive-design-white-paper.pdf.

[24] Google, «About google,» [En línea]. Available: https://about.google/.

[25] Yahoo, «Best Practices for Speeding up Your Website.,» [En línea]. Available: http://developer.yahoo.com/performance/rules.html. [Último acceso: Retrieved May 5, 2013,].

[26] J. Nandorf., «Responsive Web Design,» 2013. [En línea]. Available: https://pdfs.semanticscholar.org/3050/a79337cd73a501cb390f905dde9bcc97dbbf.pdf.

[27] Gomez, «What Users Want from Mobile.,» [En línea]. Available: http://www.gomez.com/wpcontent/downloads/19986_WhatMobileUsersWant_Wp.pdf. [Último acceso: Retrieved April 24,].

[28] N. Thompson, «How mobile-ready are corporate websites?,» Marzo 2012. [En línea]. Available: https://www.slideshare.net/WebMgrs/how-mobileready-are-corporate-websites .

[29] L. Wroblewski, «When & Where Are People Using Mobile Devices?,» February 2011. [En línea]. Available: https://www.lukew.com/ff/entry.asp?1263 .

[30] «Offline access to information anywhere, anytime,» [En línea]. Available: https://www.innovapptive.com/blog/offline-access-to-information-anywhere-anytime.

[31] C. Stoll, «Multiscreen Patterns.,» Mayo 2011. [En línea]. Available: http://previous.precious-forever.com/2011/05/26/patterns-for-multiscreen-strategies/.

[32] L. Michal, «Designing Multi-Device Experiences,» 2017.

[33] Apple, «Human Interface Guidelines,» 2019. [En línea]. Available: https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/.

[34] Google, «Material Design,» [En línea]. Available: https://material.io/design/introduction/.

[35] L. D. I. Y. D. E. I. Y. C. D. SOFTWARE, «Validation of implicit expectations on mobile applications,» 2020. [En línea]. Available: https://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/?pIs=1685.

[36] A. GuptaLast, «Most used apps in the world,» December 2019. [En línea]. Available: https://blog.sagipl.com/most-used-apps/.

[37] A. SINGH, «Top 10 Most Popular Applications in 2020,» 30 May 20119. [En línea]. Available: https://www.netsolutions.com/insights/top-10-most-popular-apps-2018/.

[38] P. M. Pujar, «Acceptance Criteria-Driven Development.,» [En línea]. Available: https://www.scrumalliance.org/community/articles/2014/march/acceptance-criteria-driven-development.

[39] «Facebook,» [En línea]. Available: https://www.facebook.com/workplace/partners#integrations.

[40] «Instagram,» [En línea]. Available: https://www.instagram.com/.

[41] S. 1.-1. -. I. G. f. S. V. a. V. Plans.

[42] S. Alliance, «Jingjie Wang Reviews and Agile,» 2013. [En línea]. Available: https://www.scrumalliance.org/community/articles/2013/march/reviews-and-agile.

[43] «Types Of Software Testing: Different Testing Types With Details,» 1 08 2020. [En línea]. Available: https://www.softwaretestinghelp.com/types-of-software-testing/.

[44] «What is BLACK Box Testing? Techniques, Example & Types,» 2020. [En línea]. Available: https://www.guru99.com/black-box-testing.html.

[45] «What is Functional Testing? Types & Examples (Complete Tutorial),» 2020. [En línea]. Available: https://www.guru99.com/functional-testing.html#:~:text=FUNCTIONAL%20TESTING%20is%20a%20type,output%20against%20the%20Functional%20requirements..

[46] Android, «Android Platform,» [En línea]. Available: https://www.android.com/.

[47] F. Nah, «A Study on Tolerable Waiting Time: How Long Are Web Users Willing to Wait?,» December 2003. [En línea]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1751&context=amcis2003.