

## Design of Platform to Support Workflow Continuity in Multi-Device Applications

Oscar Chacón-Vázquez<sup>1</sup>, Luis G. Montané-Jiménez<sup>1,\*</sup>, Carlos Alberto Ochoa-Rivera<sup>1</sup>, Betania Hernández-Ocaña<sup>2</sup>

<sup>1</sup>FEI, Universidad Veracruzana, Xalapa, 91020, México

<sup>2</sup>Universidad Juárez Autónoma de Tabasco, Villahermosa, 86040, México

### ARTICLE INFO

Article history:

Received: 25 December, 2020

Accepted: 31 January, 2021

Online: 16 February, 2021

Keywords:

Continuity

Platform

Multi-device

### ABSTRACT

Nowadays, the Internet has become an indispensable tool for the realization and continuity of activity at a different time, place, and technological context (e.g., mobile, pc, tablet), so that interaction techniques through the use of multi-device support have become of great interest. From this perspective, continuity in interactions is an essential concept in the face of changes in context environments where an interaction develops. There are works related to the continuity and support of multi-device environments through software platforms that are useful to improve continuity support; however, reducing steps to resume an activity on a different device is an aspect that needs to be studied in greater detail. In support of the above, this paper presents an exploratory study that shows that continuity is a useful feature for users; however, there are still aspects that need to be studied. Therefore, in this paper, we propose a platform to implement continuity in a workflow that reduces the steps necessary to continue and resume activity in a different device context and a case study which serves as a method to evaluate the platform proposal and the models on which it is based.

## 1. Introduction

The Internet has become one of the main tools for carrying out different activities in various sectors, for example, work, entertainment and communication [1]. The evolution of the Internet has had effects in many areas of information and communication technologies such as computers [2] that make use of it, and the paradigms and models of software development [3].

The evolution of computers towards small devices with considerable computing capacity and intelligence, have brought about changes in the design and interaction paradigms seeking a better integration in our environments [4]. The constant evolution of computers towards small devices with considerable computing capacity and intelligence, have brought about changes in the design and interaction paradigms seeking a better integration in our environments [5]. In general, these developments use tools that make it easier to carry out software projects.

Interaction techniques through the use of platforms with support for multi-device environments have become one of the fields of greatest interest thanks to the great adoption of the internet and the evolution of information and communication technologies, taking advantage of its advantages [6]. The tools that have emerged derived from these techniques are broad and useful, focusing on different areas of interaction such as the use

of single sign-on systems, the use of multiple devices to interact with the same system at the same time, and the ability to continue certain activities with a long duration across different contexts [2].

In multi-device interactions, an activity can be carried out through different devices and in other contexts; this is much more complex to keep a user-focused on an activity and recover the activity's status was left. For example, a student who writes a document on a computer at his school may interrupt his work and continue it later from a different computer at home. This activity implies having to transfer the information from one computer to another and recovering the last state (e.g., page of a document where the student was, available tools, and information on the clipboard).

Continuity has become an essential concept in recent years for commercial information systems. We can find these systems such as Netflix, Amazon Video, and Spotify, which allow through a login and constant synchronization between devices to maintain a state between them and allowing the user to resume their activity (playback of some content multimedia) in a different context. Similarly occurs with Apple operating systems, which, in the case of system-specific applications, allow resuming an activity between devices, omitting several additional steps to do so.

Continuity has been one response to the interruptions brought by changes in context, reducing the number of steps necessary to

\*Corresponding Author: Luis G. Montané-Jiménez, Email: [lmontane@uv.mx](mailto:lmontane@uv.mx)

continue an activity; however, its implementation is complex due to multiple aspects [4], [6]. Therefore, we propose a conceptual platform that helps implement continuity in a software project, and it helps reduce the number of steps necessary to continue an activity when changing context. This proposed platform allows programmers and designers to focus their efforts completely on said design and reducing the workload to implement continuity. It also reduces the users' cognitive load in applications complex by quickly migrating an activity between different devices, reducing interruptions in the workflow.

This paper is structured as follows. Section 2 reviews the terms related to continuity in the workflow, and Section 3 presents a comparison of the work related to continuity and workflow. Section 4 presents a study of continuity in commercial applications. Section 5 presents a conceptual model of continuity in software projects. Section 6 presents a case study where the platform was implemented for testing. Finally, Section 7 presents the conclusions and future work.

## 2. Continuity in workflow

### 2.1. Workflow

The term workflow is a concept applicable to all areas where the human being develops. Specifically, information systems refer to a series of tasks that must be followed and completed so that a user can obtain a result (e.g., complete an activity). In [7], the authors refer to the workflow simply as flow. The workflow tends to be confused with the activity itself. This activity is the aim pursued by the user. For example, when someone to read a book, the workflow is necessary to complete it: i) read chapter 1, ii) read chapter 2, among others. The workflow has the characteristics shown in Table 1.

### 2.2. Interaction between devices

Adopting multiple devices in people's daily lives creates an environment where information is easily shared between them [8]. Currently, users are not limited to owning only a computer and a mobile phone, systems that in most cases are isolated from each other. In these environments, a user can have multiple computers that are used in different places and with other devices such as tablets or Smart TVs that have tools and applications that have the ability to interact and communicate with each other to offer the user a user interface that is seen as a whole, regardless of the equipment or device where users perform an activity.

Interaction between devices is an emerging branch of Human-Computer Interaction (IHC) focused on the way in which users interact between the different devices to which they have access, as well as the way in which applications take advantage of these forms interaction to offer much more dynamic user experiences [9]. There are two types of interaction between devices, i) interaction with multiple devices and ii) interaction across multiple devices [6]. The i) interaction with multiple devices is based on the ability of each of the devices to operate a system or application at the same time [10], allowing to create an environment where each device takes a specific role within the application and thereby better distributing the options and capabilities of the system. It focuses especially on collaborative environments where each person can take a specific role within the application, adapting the interface to each role and thereby

avoiding overloading the interface with unnecessary elements. On the other hand, ii) interaction through multiple devices focuses especially on monitoring activities that were started on a certain device, and due their duration, it may need to be interrupted and continued on a different device [7], facilitating the change of context for the user using concepts such as continuity and workflow to provide better user experiences.

Table 1: Workflow characteristics

Characteristic	Description
Divisible in tasks	The activities can be divided into more specific and shorter tasks, the completion of these tasks has the consequence of approaching the end of the activity.
Duration	It is an important factor in the definition of an activity and its flow, it defines the time that is invested in carrying out the activity and generally greatly affects the methods and moments of interaction.
Focus	It refers to the number of people involved in the activity; it affects the way in which the activities are carried out. It is possible to catalog the activity flows in two: one user and multi-user.

### 2.3. Continuity

The continuity term is defined as the ability to perform an activity on devices used sequentially and without presenting major interruptions in their workflow (context changes) [6]. It is a capacity and characteristic that we can find present in many commercial systems and applications and that take advantage of the benefits of the Internet and the capabilities of the devices with which we interact to reduce the steps necessary to continue an activity between them.

The experiences that implement continuity have 6 main characteristics that can be seen as objectives to be achieved [6]: i) Privacy, provide the user with the assurance that the data is safeguarded. ii) Appropriation, operate inherently in conjunction with the user and their work context. iii) Personalization, provide users with the possibility to choose which and when the continuity functions are activated. iv) Awareness, ability to "infer" the next steps that the user will perform. v) Inclusion, broad compatibility with current and future devices of the user, increasing interaction possibilities. Finally, vi) Troubleshooting, offering the ability to solve problems arising from the use of continuity to the user.

In a specific context in which a user interacts with a system that implements continuity and requires interrupting their work and continuing it in a different context, it is possible to distinguish two types of continuity [7]: monoactivity, and sequential activities. In monoactivity, continuity refers to activities such as: reading a book, watching a movie, or writing a text that may require a long time to complete. The advent of technology and the great expansion of mobile devices have contributed significantly to users changing how they performed these actions [2]. These activities can be carried out in short periods, where the context of use and the time available have an essential effect on the

interaction. Continuity in one-time activities focuses on this type of interaction, trying to minimize the necessary steps so that a user who left an activity pending completion can resume it when he deems it appropriate.

Continuity in sequential activities refers to when the user faces activities whose time and necessary effort allow them to be divided into the performance of sub-activities [7]. The continuity in these types of activities does not depend on being able to carry out the same activity through different contexts, but rather on maintaining a state between all the sub-activities, allowing the user to minimize the mental processes necessary to continue working. The design of these interactions is not based on its duration but rather on the place where a sub-activity occupies the general activity. In general, these types of interaction focus on offering the user information related to the last time they interacted with the system rather than minimizing the steps necessary to continue with the activities.

### 3. Continuity related work analysis

Multi-device environments and the accelerated evolution of information and communication technologies have led to the development of a great variety of works with multi-device and continuity implications; each of them has its characteristics and qualities. Among the works carried out, the most noteworthy include frameworks, development tools, and evaluations of applications that implement continuity.

#### 3.1. Frameworks

The frameworks are a recurring theme in computer science because they allow us to establish future work bases and propose different approaches to solve a problem. In the multi-device and continuity field, there are several works where proposals for theoretical frameworks are presented. Recent works conceptually propose a framework for co-located devices in the same environment distributing the interface of an application through the users' roles when using the interface simultaneously [10].

In [11] propose a conceptual framework where for each device a role must be established and used in some way within the adaptation process, continuity is included at the conceptual level. More recent works introduce the concept of continuity as in "Design a framework to support the development of smart cross-device applications" [9], where a framework is proposed which provides guidelines and tools to distribute interfaces through nearby devices. Continuity is introduced as a state maintenance that improves the user experience.

A recurring theme in framework development is maintaining state through different methods. In [12,13] present frameworks that allow data to be automatically migrated between applications. However, they do not have current tools that maintain the state completely. On the other hand, [14] propose a tool to migrate data through interaction with Kinect to migrate data during a work session with several devices easily.

#### 3.2. Development tools

Application development is a complex task that uses many tools to facilitate the process. In the field of continuity and multi-device interaction, work has also been carried out based on tools' development.

In [15] present a studio-type tool focused on the design of multi-device interfaces, and focused on its interactivity. The tool focuses on designing web-based interfaces that can be distributed and function in a coordinated manner between various devices; it also allows defining contexts for the devices to adapt according to them.

Several tools focus on data migration; among them, we can find tools that allow data to be migrated between "neighboring" applications of a mobile device [16]. In [2] propose a tool that allows for migrating information searches between different devices while maintaining history. On the other hand, there are works such as present in [17–19] that allow for migrating the state through different triggers such as biometric sensors and JavaScript methods. Finally, in [20] present a tool that allows data to be migrated through NFC tags, mainly focused on web searches.

Other authors have focused on the development of tools that support the internal procedures of the applications. In [5], the author propose a middleware type tool to implement a single sign-on between devices; continuity is a crucial part of the tool since, through this sign-on, the state can be transferred between devices.

#### 3.3. Evaluations to applications

Continuity is a tool that has been implemented in both software projects and projects for the commercial field, verifying the effectiveness of applications that implement continuity. A multimedia application prototype to continue multimedia consumption through different devices was developed in [21]. This prototyping's main objective is to demonstrate the interaction techniques' effectiveness between the most used devices. Through the study and the prototype, it was possible to demonstrate that it developed software with a user-centered approach is necessary. However, there are still processes that can be implemented to improve interaction.

On the other hand, a conceptual model is presented in [22] where long-term searches on the web are supported across different devices; their approach is not just based on allowing long-term searches across different devices. It also provides intelligence by offering suggestions based on search patterns. The study carried out on the model showed significant differences in user interaction, depending on the context and type of device used.

In [6], the author evaluated one of the most representative applications of continuity in a commercial context: *Apple Continuity*. The study carried out contrasts the characteristics of Apple's service against user comments; the results indicate that continuity is accepted and desired by users in general. However, adapting to these new interaction methods can become complex, so the focus on user-based development is important to avoid such complexity. Further, [23] present an evaluation of automatic code generation to implement continuity with a strong emphasis on commercial applications.

Finally, there are work that present the applications of a method developed by authors to evaluate the migratory aspect of multi-device application such as presented by [24], [25].

#### 3.4. Continuity related work comparative

By reviewing the work related to continuity and multi-device applications, Table 2 is presented, which shows that several

elements come into play when continuity is implemented in devices and systems.

The concept of multi-devices is applied in many ways throughout the different jobs, allowing the creation of different forms of interaction in the systems. It is also proven that it is a key point to give way to the continuity of activities in applications.

Table 2: Comparative of papers related to continuity

Paper	Year	Type	Emphasis	Multi-device	Continuity	
					Explicit	Implicit
Mobile to TV migration	2008	Tool	Backend	Yes	Yes	Yes
Remote AppBus	2009	Tool	Backend	Yes	Yes	No
Flexible framework	2009	Framework	Backend	Yes	No	Yes
JavaScript state	2011	Tool	Frontend	No	No	Yes
Multimedia prototype	2011	Evaluation	Frontend	Yes	Yes	Yes
Deepshot	2011	Framework	Backend	Yes	Yes	Yes
Single session start	2012	Tool	Backend	Yes	No	Yes
Virtual Browser	2012	Tool	Backend	No	Yes	No
Taskshadow-W	2013	Tool	Backend	No	Yes	Yes
Communication evaluation	2013	Evaluation	Backend	Yes	Yes	Yes
Interface IDE	2014	Tool	Frontend	No	No	No
XDKinect	2014	Framework	Backend	Yes	Yes	Yes
Search model for continuity	2015	Evaluation	Backend	Yes	Yes	No
Code generator	2015	Evaluation	Backend	Yes	Yes	No
Apple Continuity	2016	Evaluation	Frontend	Yes	Yes	Depends
CustomConsole	2016	Framework	Backend	Yes	No	No
Moving context kit	2017	Framework	Backend	Yes	Yes	No
Yanux framework	2018	Framework	Frontend	Yes	No	Yes
CORMORANT	2019	Tool	Backend	Yes	Yes	Yes
Model 4C's	2019	Framework	Backend	Yes	No	Yes

In the same way, works have been created that support the implementation of a part of continuity in applications. The main problem found in them is the poor compatibility with applications due to their focus on a specific aspect makes it difficult for continuity to be implemented by developers except in the aspects for which these tools were developed. It is also noteworthy that, in recent years, jobs such as [9–11] have focused on the creation of frameworks both at a conceptual and technical level for the creation of tools with interactions in continuity, showing a huge advantage over applications because they allow developers to expand the range of end uses that may have an application.

Based on the information presented in Table 2 and the reviewed works, we can see that, although explicit continuity and implicit continuity (called by many authors as maintenance of a state) are characteristics that do not always go hand in hand. The results, especially from those studies focused on evaluations of

continuity implementations, suggest maintaining a state in applications that implement continuity is a very desirable feature.

With the analysis of the related works, it can be affirmed that one of the ways to implement continuity in the applications is through a framework or technological platform that provides developers with the necessary tools to create ecosystems with continuity characteristics. Although forays into these types of work have previously been made, there are aspects that remain to be covered, such as compatibility with modern platforms and with future platforms and the generalization of activities.

One of the most desirable features in continuity is maintaining an application state, which, in the long run, reduces the steps to resume an activity. This aspect is a topic little explored by researchers.

#### 4. Exploratory study

In support of the points identified in the review of works, an exploratory study was designed to identify points of improvement in the user experience in multi-device applications, with a specific focus on applications that implement one or more aspects of continuity. The methods and materials used for this study are described below.

##### 4.1. Experiment design

For this study, an exploratory experiment was designed and executed using the Google Forms tool; a questionnaire was constructed with a total of 20 items or questions that encompassed four different categories. Users were chosen randomly, regardless of their age, gender, or degree of academic studies. A total of 30 people were chosen for the application of the study.

The categories considered for the exploratory study are the following: i) general user data for classification and future consideration during data analysis; ii) continuity analysis in multi-device applications focused on music streaming; iii) continuity analysis in multi-device video streaming applications; and iv) continuity analysis in multi-device applications for document editing.

##### 4.2. Evaluation tools

For the study, a questionnaire with 20 items was designed, the first four dedicated to categorical information from the user; the rest are described in Table 3. For its execution, the Google Docs tool was used; specifically, the Google Forms application allowed the questionnaire to be distributed quickly and effectively to the respondents. Another reason for its choice is that it provides a graphical analysis of the results and avoids the results' repetition by monitoring the respondents with their email.

Table 3: Questions applied in the exploratory study

ID	Question	Detail
5	How often do you use these devices to play music?	Question on Likert 5 points scale



6	It would be useful if the application automatically changes places to play your music on the most suitable device	Question on Likert 5 points scale
7	Would you like the application to be displayed as you interrupted it on another device when changing devices?	Question on Likert 5 points scale
8	What would you expect your player app to do when you want to switch devices?	Multiple option question
9	How often do you use these devices to play video?	Question on Likert 5 points scale
10	Of the following characteristics of an application for video playback, which of them are most important to you	Multiple option question
11	Is it useful to you that a video player application allows you to continue a movie / series right where you left off?	Question on Likert 5 points scale
12	You would like a video application to allow you to continue with a movie / series on another device only by turning it on and opening the application	Question on Likert 5 points scale
13	When it comes to playback across multiple devices, what would you improve on a video app?	Open question
14	How often do you use these devices to work on a document?	Question on Likert 5 points scale
15	Have you ever used the "pick up where you left off" feature in Microsoft Office	Multiple option question
16	How useful is this feature for you when switching from one workplace to another or between different devices?	Question on Likert 5 points scale
17	How much would you like Microsoft Office to allow you to continue a document without having to indicate it using the "continue where you left off" function?	Question on Likert 5 points scale
18	Would it be useful for you if, in addition to taking you to the section where you worked in the document, also the settings, clipboard and tools kept their last state?	Question on Likert 5 points scale
19	Would you like that when you open the Microsoft application (Office, Excel, PowerPoint) it would automatically open the last document you had open?	Question on Likert 5 points scale
20	When it comes to writing documents, what would you expect from an application when switching between contexts or devices?	Open question

#### 4.3. Process

The participants' selection was carried out randomly, a short introduction to the subject of the study was given, and the instructions for filling out the tool; their participation was completely voluntary.

The tool was sent by electronic means to the users chosen for the test who had the approximate time of 1 day to answer them without affecting their working hours or daily activities. Finally, after a week, the data obtained through the platform was collected for subsequent analysis.

#### 4.4. Results

Regarding music playback applications, it is notable that one of the characteristics most sought after by users is the "intelligence" of the application to play on the most suitable device available.

The results translate into that it would be instrumental for most users if they are playing music, and this automatically chooses the most appropriate device to play on. For example, a user who plays content on their cell phone, and when they get to Your home could automatically transfer this playback to your TV or computer, without the need for intermediate steps.

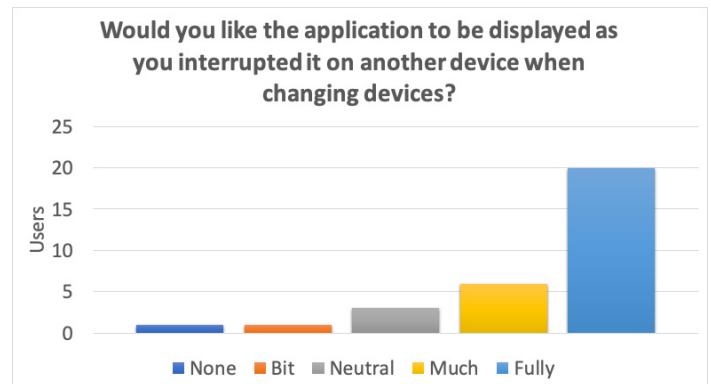


Figure 1: Opportunities of continuity in music streaming apps

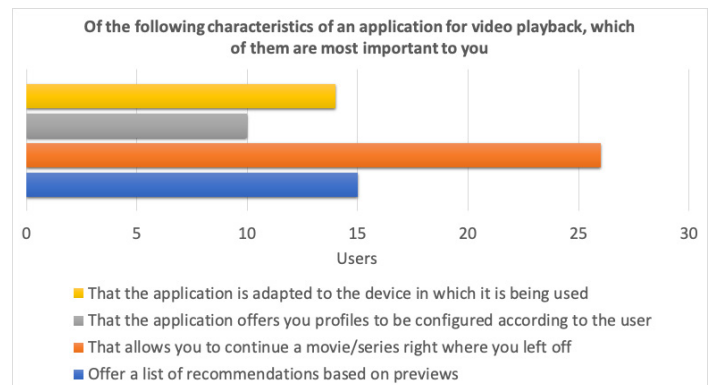


Figure 2: Continuity in video streaming apps

In addition, an aspect that music applications lack and that users expect from applications is the maintenance of a state, that is, that the application remembers where it was stopped the last time it was used, as can be seen in the second graph of Figure 1.

Regarding video applications, continuity has a greater presence in the application processes and turns out to be one of the characteristics that users like the most, as shown in Figure 2. In the same way as playback applications, music users are interested in the possibility of maintaining the state of the content reproduction when exchanging between devices. However, the approach applied in implicit continuity is not as relevant for users as seen in the second and third graphs of the survey conducted. Finally, for productivity applications, an analysis of the continuity implementation was carried out for an office suite widely used in the world and whose continuity implementation is explicit. Each time the user starts the application and opens a document you have been working on recently, you are presented with a button to go directly to the place where you last edited the document.

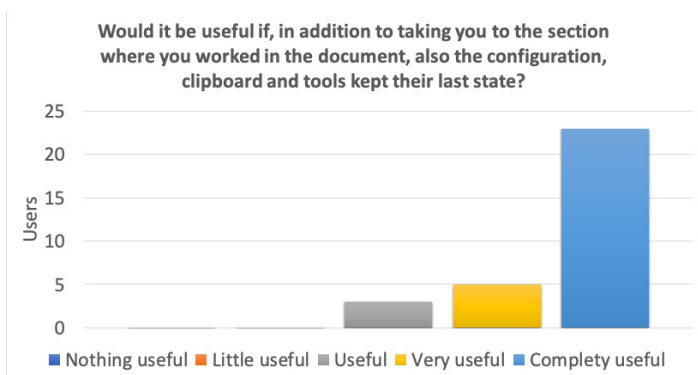


Figure 3: Continuity in document edition apps

In this type of application, two important aspects can be distinguished with regard to continuity. An important aspect mentioned by users is that the application directly remembers the status of the application with the documents they are working on, without the need for this function to be explicitly activated by the users. On the other hand, users would also find it useful to retrieve the configuration of tools for editing documents such as clipboards, temporary settings of the application, and additional tools such as a calculator or a browser; the results of the latter can see in Figure 3. With what was previously described, some users differ in the case that the application implements implicit continuity. It could be noted in the review of the comments made by the users; not all agree that the last document worked should be opened.

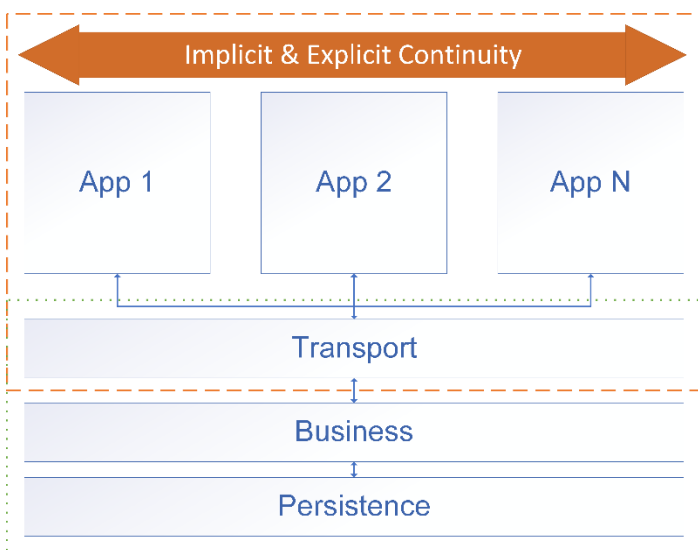


Figure 4: General model of the platform for implementation of continuity

**5. Conceptual model for continuity implementation in software projects**

Based on the models proposed by [5,9–11,15] and in the results of the continuity study carried out, a technological platform proposal was made at a conceptual level for the implementation of continuity in the activity flows shown in Figure 4 and detailed in its components in Figure 5. This model proposes an architecture for a platform where applications compatible with different devices can be easily implemented, and one of its goals is to allow its users to continue their task in a simple way when faced with a context change. The objective and the structure of the model for

the platform allows the reduction of steps necessary to continue an activity between several devices, that is, to facilitate the implementation of implicit continuity in several applications.

In Figure 4, the general behavior of the platform can be observed. Through a set of layers, the necessary support is provided to create multi-device environments, taking advantage of the possibilities of explicit continuity and, above all, of implicit continuity, regardless of the type and number of applications being developed.

The persistence layer focuses on the storage of information corresponding to the application domain and the context where it operates (for example, the type of device used and the environment where it is done), typically composed of three elements, i) the application's usage status data, that is, configuration and usage information, this data contains information about how the last device was used ii) the device status data that considers device-specific details such as type of last device or the app version used which helps to adapt the interface of next used device and iii) the domain data referring to the application's own information (documents, photos, among others) which is used to present the data that were working, is based on unstructured language for the storage of information. This layer also contains the basic operational engine, which provides essential functions (create, update, delete, and read) over the data storage in order to operate with the basic application information.

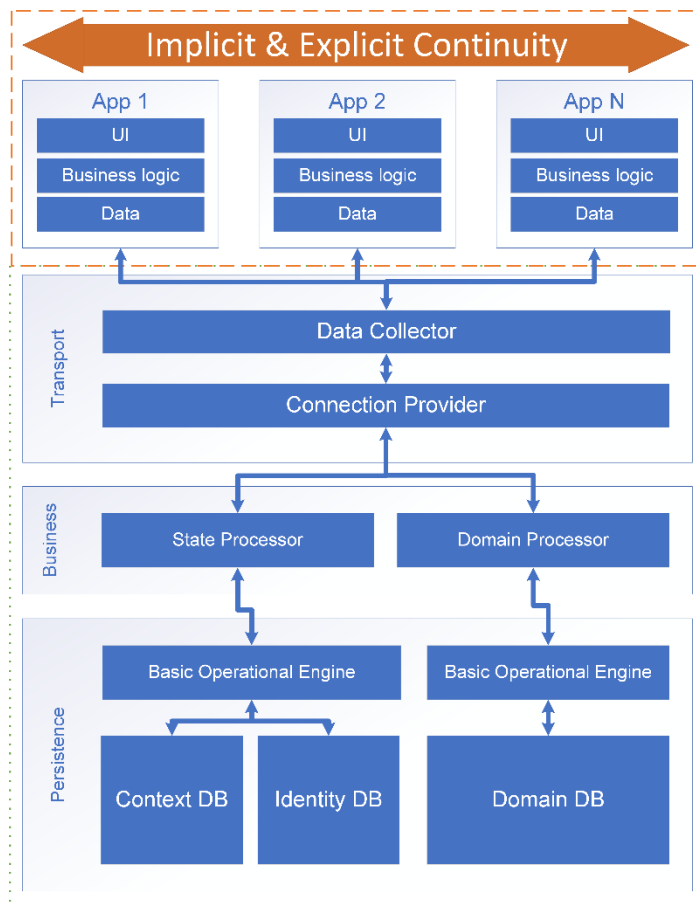


Figure 5: Components of the technological platform for the implementation of continuity in software projects

Table 4: Description of the conceptual model of a technological platform for the implementation of continuity in workflow

Characteristic	Description
Context DB	Storage component that allows to save the information of the context of use of the devices, it is a long-term data storage software, it contains elements that function as tools for the selection of device sequence and the configuration of interfaces
Identity DB	Storage component that stores information corresponding to users and that is used as a starting point to create state representations, this database must contain data from the user that help to identify the information that is stored in the context and domain database, its purpose is to maintain control over the information that the platform operates
Domain DB	Storage component focused on the operational information of the applications, it contains representations of the information that the user enters or obtains from the application.
Basic Operational Engine	It is a logical component that provides access to the basic functions on the storage components, the proposed implementation design defines an engine for domain data and another for identification and context information in order to reduce workloads over other components.
State Processor	Logical component that allows composing and decomposing the state representations and the information obtained from the operational engines and sending that information to the next layer depending on whether it is retrieved or entered into the platform, its purpose is to offer an interpreter for both directions of the platform.
Domain Processor	Logical component that provides functionality on domain information and that must establish the data transformation bases for the changes that are necessary between the jumps of multiple applications. This element works as a translator when it is necessary. It must also be responsible for communicating the domain data to the upper and lower layers in order to reduce the information contained by state representations
Connection provider	High-level component that functions as a data input to the lower layers, receives the information from the data collector and is responsible for passing it to one of the two processors of the business layer. This component is proposed to function as an API in order to be able to be integrated into software projects without having to modify the base structure of the project
Data Collector	Its a high-level component that is proposed to be responsible for direct communication with the final applications, it must be in charge of obtaining the necessary information to create the representation of the state that will be sent to the connection provider and also of presenting the information when a representation is retrieved from the platform towards a final device, It is proposed that this element be built as libraries for the specific programming languages of each final application
App 1,2... N	These are the final applications where the platform can be implemented, which can go from 1 to N, each one can have its own logic processes, information storage media and graphic interface.

The persistence layer communicates with the business layer, which comprises two processing services (state and domain logic). These processing elements are responsible for making the necessary changes to the data; being the most important elements since a large part of the adaptation depends on them. Finally, in the highest part, the transport layer is composed of two elements (data collector and connection provider). These elements are

those that are exposed to the developers and are integrated into the final applications, built in the form of APIs, extracting the necessary information from the applications and sending them to the processing models so that the rest of the platform works. It is also responsible for making the necessary changes to the application interfaces. It is through these two elements that applications reduce the steps necessary to resume activities. These elements have to be able to scale to serve as many final applications as a program has been developed.

In order to support the understanding of this platform, a diagram was developed with the components of the model and which is presented in Figure 5, as well as the description of these components which is found in Table 4.

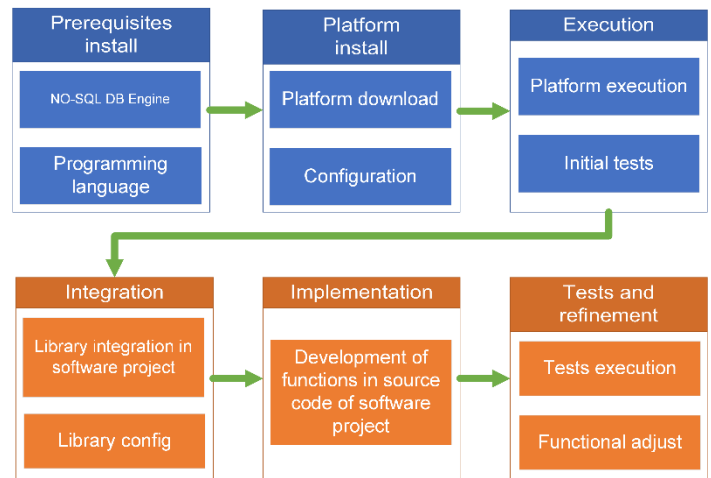


Figure 6: Platform implementation roadmap

In Figure 6, the platform implementation is shown; in the upper part the work stages are shown directly on the platform which are: i) pre-requisite installation in which the execution environment of the platform through the configuration of tools for database management and the programming language used by the platform, ii) installation of the platform that consists of downloading the platform code and its configuration with the database management tool, iii) Execution which consists of the start-up and rapid tests of the correct operation of the platform.

At the bottom of the implementation sheet, the steps after the installation of the platform are shown, which apply directly to the software project; these steps are: iv) Integration where the libraries that are added to the software projects allow communication to the platform and its configuration, v) Implementation where the views and functions of the software are developed, and it is indicated where the platform libraries will work and vi) Testing and refinement where the tests should be carried out on each final application and adjusting functions in case these changes are necessary.

## 6. Case Study

The platform is an intermediate software that communicates with the different components of other applications; due to the nature of this type of software. It is impossible to carry out functional and use tests directly on its construction, so the software is implemented in a Case study where both the function and the user experience when using continuity could be evaluated.

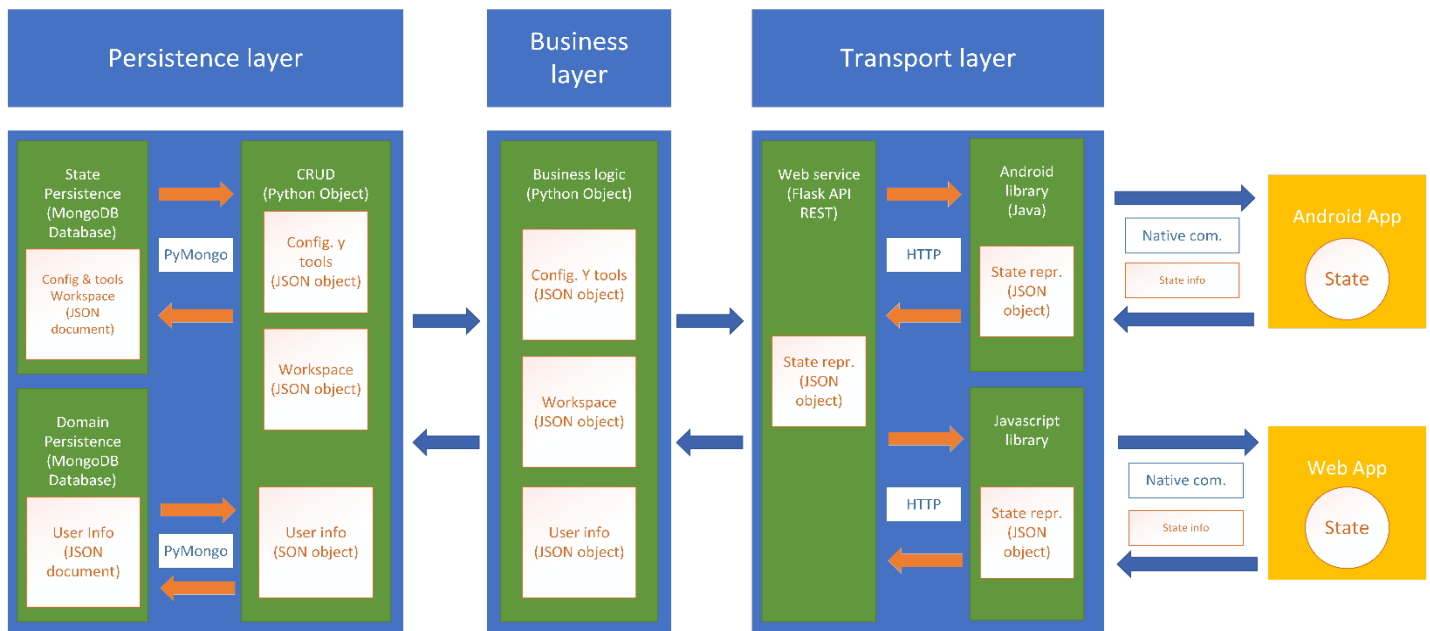


Figure 7: Implementation architecture for study case

The selected case study is an application that allows you to create exams according to a catalog of questions previously evaluated and approved. The process of editing an exam can be a complex task for those who take it, and because of this, in many cases, it is necessary to move it between one device and another.

The case study has three elements: a direct administration system that allows the question catalog to be generated, edited, and debugged. A web application is attached to the administration system to generate the exams from any browser. Finally, an Android application that also allows generating the exams.

The structure of the selected application makes it perfect to test the proposed solution because it's possible to change between two different types of apps and two different contexts, also, the selected applications can be executed in many modern mobile devices and even in any device that can accessing internet through a web browser.

On the other hand, the limitations of this case study are that it does not allow the analysis of an interaction that is too deep, such as using a text editor, in addition to the fact that it is an application in its final stages of development, so an error that is not related to the use of continuity and biases the study in a certain way.

Two libraries were built that allowed to integrate the platform in the case study, the first in JavaScript for the web sections and the second in Java-Android for the android section of the case study. Figure 7 shows in detail the structure and elements that make up the platform in its implementation for the case study.

The persistence layer is implemented through two databases that represent the storage components, one for the domain and one for the state. These connect a first layer of python that offers basic functions on databases (basic functional engines). At the second level there is a business layer that implements methods and processes to arm and disarm the state representations and finally the transport layer is responsible for implementing a RESTful-type web service and the two libraries mentioned above, while the REST service receives and sends the information when the

libraries so require, the libraries control directly and in a practically automatic way the events in the applications of the case study that save and retrieve the information.

## 7. Conclusions and future work

The development of computer systems has changed over time, adapting to the different tools that arise daily. The notable increase in the use of mobile devices is one of the main reasons that have led to the emergence of new interaction and development paradigms.

One of the main topics that researchers and the market related to systems development have focused on is multi-device interaction. Continuity arises as part of the multiple ways in which a multi-device interaction occurs. The possibilities that continuity offers to improve the different ways in which a user interacts with a system are vast, yet there is still a long way to go.

One of the problems that researchers currently focus on solving through the use of continuity is the minimization of interruptions in the flow of an activity, something that could be seen through the review of works related to this research and the exploratory study carried out is a feature that users are beginning to expect from the systems.

The design of a platform to implement this type of interaction is complex due to the components that are required for correct operation, as well as the very logical structure that is required to maintain the state of the applications between the changes made by the user.

The results obtained once the implementation was carried out in the case study were favorable, the platform facilitated the continuity implementation in the case study and also allowed to reduce the necessary steps to resume an activity, which facilitated context changes for users.

In contrast to other works that were reviewed during the writing of this document, we can see that a platform that provides full support for continuity has a deeper impact than applications such



as [2,4,5,12–14], whose emphasis is strongly focused on the login and the migration of user data over the migration of operation data.

On the other hand, a better performance can also be noted than the proposals made by authors such as [3,9,11,16,18], where migration Information is a present feature, but small details such as the configuration or the last view requested are not considered, forcing the user to have to remember this when resuming an activity. It is still necessary to make improvements with respect to the platform's ability to adapt based on usage preferences and configurations that affect continuity, as well as to carry out tests that guarantee its effectiveness, with continuity being a useful tool in different contexts of use. However, it still requires a long and dedicated development process.

Finally, as part of the future work, it is expected to carry out a series of tests through the case study in order to validate the functionality and design of the Platform.

### Conflict of Interest

The authors declare no conflict of interest.

### Acknowledgment

The authors thank the reviewers of this article for their helpful comments. The corresponding author also thanks to CONACYT-México for the grant No. 715387 used for his postgraduate studies.

### References

- [1] L. Tan, N. Wang, "Future Internet: The Internet of Things," in *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering*, Proceedings, 2010, doi:10.1109/ICACTE.2010.5579543.
- [2] B. Cheng, "Virtual browser for enabling multi-device web applications," *Proceedings of the Workshop on Multi-Device App Middleware - Multi-Device '12*, 1–6, 2012, doi:10.1145/2405172.2405175.
- [3] C. Ye, J. Wei, H. Zhong, T. Huang, "Middleware Support for Internetware : A Service Perspective," *Systems Research*, 2010, doi:10.1002/ardp.201300158.
- [4] H. Kim, E.A. Lee, "Authentication and Authorization for the Internet of Things," *IT Professional*, **19**(5), 27–33, 2017, doi:10.1039/b904090k.
- [5] P.A. Cabarcos, F. Almenares, R. Sánchez Guerrero, A. Marin, D. Díaz-Sánchez, "Multi-device Single Sign-on for cloud service continuity," *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, 644–645, 2012, doi:10.1109/ICCE.2012.6162011.
- [6] D. Raptis, J. Kjeldskov, M.B. Skov, "Continuity in Multi- - Device Interaction: An Online Study," *Proceedings of the 9th Nordic Conference on Human-Computer Interaction - NordiCHI '16*, 2016, doi:10.1145/2971485.2971533.
- [7] M. Levin, *Designing Multi-Device Experiences*, First, O'Reilly Media, Inc, 2014.
- [8] J. Schöning, H. Reiterer, C. Holz, J. Vermeulen, N. Marquardt, C. Klokmoose, S. Houben, "Opportunities and challenges for cross-device interactions in the wild," *Interactions*, **24**(5), 58–63, 2017, doi:10.1145/3121348.
- [9] A. Pedro, C. Nuno, N. Rui, "Designing a framework to support the development of smart cross-device applications," *Mum18 Rw*, 367–374, 2018.
- [10] A. Marzo, "CustomConsole : A Framework for Supporting Cross-device Videogames," 1952–1958, 2016.
- [11] K. O'Leary, T. Dong, J.K. Haines, M. Gilbert, E.F. Churchill, J. Nichols, "The Moving Context Kit: Designing for Context Shifts in Multi-Device Experiences," *Proceedings of the ACM Conference on Designing Interactive Systems (DIS 2017)*, 309–320, 2017, doi:cshg.
- [12] M. Barisch, J. Kögel, S. Meier, "A flexible framework for complete session mobility and its implementation," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **5733 LNCS**, 188–198, 2009, doi:10.1007/978-3-642-03700-9\_20.
- [13] T.H. Chang, Y. Li, "Deep shot: A framework for migrating tasks across devices using mobile phone cameras," in *Conference on Human Factors in Computing Systems - Proceedings*, 2011, doi:10.1145/1978942.1979257.
- [14] M. Nebeling, E. Teunissen, M. Husmann, M.C. Norrie, "XDKinect: Development framework for cross-device interaction using kinect," in *EICS 2014 - Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2014, doi:10.1145/2607023.2607024.
- [15] M. Nebeling, T. Mints, M. Husmann, M. Norrie, "Interactive development of cross-device user interfaces," *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*, 2793–2802, 2014, doi:10.1145/2556288.2556980.
- [16] N. Narasimhan, C. Janssen, M. Pearce, "Remote AppBus - Enabling seamless access to short term memory on mobile devices," 2009 6th IEEE Consumer Communications and Networking Conference, CCNC 2009, 1–5, 2009, doi:10.1109/CCNC.2009.4784773.
- [17] D. Hintze, J. Kepler, "CORMORANT: Ubiquitous Risk-Aware Multi-Modal Biometric Authentication across Mobile Devices," **3**(3), 2019.
- [18] F. Bellucci, G. Ghiani, F. Paternò, C. Santoro, "Engineering JavaScript state persistence of web applications migrating across multiple devices," in *Proceedings of the 2011 SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS 2011, 2011, doi:10.1145/1996461.1996502.
- [19] F. Paternò, C. Santoro, A. Scordia, "User interface migration between mobile devices and digital TV," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **5247 LNCS**, 287–292, 2008, doi:10.1007/978-3-540-85992-5-28.
- [20] S. Li, G. Pan, Y. Xu, Z. Ye, L. Chen, "Taskshadow-W: NFC-triggered migration of web browsing across personal devices," in *UbiComp 2013 Adjunct - Adjunct Publication of the 2013 ACM Conference on Ubiquitous Computing*, 2013, doi:10.1145/2494091.2494119.
- [21] R. Bernhaupt, M. Abdellatif, T. Mirlacher, "Cross-device continuous media consumption," 217, 2011, doi:10.1145/1941007.1941046.
- [22] S. Han, Y. Zhen, H. Daqing, "Understanding and supporting Cross-Device Web Search For Exploratory Task with Mobile Touch Interactions," *ACM Transactions on Information Systems*, **33**(4), 2015.
- [23] E. Umuhoza, H. Ed-Douibi, M. Brambilla, J. Cabot, A. Bongio, "Automatic code generation for cross-platform, multi-device mobile apps: Some reflections from an industrial experience," in *MobileDeLi 2015 - Proceedings of the 3rd International Workshop on Mobile Development Lifecycle*, 2015, doi:10.1145/2846661.2846666.
- [24] R. De A. Maués, S.D. Junqueira Barbosa, "Cross-communicability: Evaluating the meta-communication of cross-platform applications," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **8119 LNCS(PART 3)**, 241–258, 2013, doi:10.1007/978-3-642-40477-1\_15.
- [25] T. Dong, E.F. Churchill, J. Nichols, "Understanding the Challenges of Designing and Developing Multi-Device Experiences," *Designing Interactive Systems*, 62–72, 2016, doi:10.1145/2901790.2901851.