UADY

UNIVERSIDAD
AUTÓNOMA
DE YUCATÁN

Software Engineering

-Software Engineering Fundamentals-

Second Increment

Osiris Team Members:
Fernan Enrique Cetina Escalante
Jorge Teodoro Dawn Rodriguez
Rodrigo Alejandro Castrejón Cervantes
Cinthia January Huchin Pedrero
Ricardo Reyes Balam Cupul

Mérida, Yucatán. January 9th, 2021.

Functional requirements and not functional requirements:

- The system shall provide a short introductory tutorial that teaches users its main features: add, edit and remove tasks, update progress, and change notification frequency.
    - It should be accessible via a separate tab at any time.
- The user shall be able to see a list of all of their current active tasks. It is very important to display due date and progress status.
    - The tasks should be organized by their due date, nearest one first.
- The user shall be able to add, edit or remove a task from their task list.
    - These actions should appear in the list view and should take one click to use them.
    - Our system should not save any identifiable information from users, these being: name, username, and email.
- The user shall be able to set a due date when creating a task and also be able to modify it [due date] via the button containing the date.
    - The due date should be set using a date picker.
    - The default date is the next day in which the task is created.
- The system should show a menu to help users reassign a task when a user misses its due date.
    - A calendar should be displaced to pick a new date.
    - The system should allow us to remove or reschedule the task.
- The system shall display the current progress of each task and allow the user to update said progress.
    - The system should allow for progress to be reverted in case of a revision or an accidental update.
    - The ‘progress update’ action should be accessible directly on the task list, right next to each task.
    - It should only take one click to update the progress.
- When the user has active tasks, the system shall keep pushing notifications saying “You have pending duties” until they complete them.
    - By default, the system should only send one notification per day (10:00 hrs).
    - The user shall be able to set the notification frequency.
- Each task shall allow users to append ‘update notes’ to them in order to keep better progress.
    - The notes should be displayed alongside its corresponding task in the task list.
    - Notes should be autosaved.

Use cases

The main actor for every use case is the user; the functionalities listed must be implemented and as an exception we will not describe the way the system processes the information.

Actions the user must be able to do:

- Set the notification frequency.
    - Flow of actions:

        - Notification icon is clicked.
        - A menu with notification rates and arrival hours is displayed.
        - User clicks the options or types it's custom settings.
    - Exception: If the user does not set a notification frequency the system should only send one notification per day at 10:00 hrs.
- Add tasks.
    - Flow of actions:
        - Landing page is seen
        - 'add a task' button is clicked
        - A menu with empty text field is displayed
        - The task is written
        - When 'enter' is pressed the task autosaves
        - The task is displayed as a list within the landing page
    - Exception:
        - An empty task cannot be listed
- Within the display of the tasks users can edit the task, add notes, set due dates and edit the progress tracker for each task.
    - General precondition: A task has been created

  Edit Task:
    - Flow of actions:
        - The user clicks on the edit button
        - The system will display a text field
        - The user re-writes the title of the task on the text field
        - The user clicks on the save button inside the text field
        - The system displays the task with the new title
    - Exception:
        - When the due date passes and progress tracker is not at 100%, a menu that displays a calendar in which the user can reschedule the task shall pop-up.
        - If the user does not click on the save button after writing the new title, the system will not save the new title and keep displaying the current title.
        - If the user does not want a title to be saved he can just click outside the text field.

  Notes:
    - Flow of actions:
        - The user click on the notes button
        - The system will display a text field
        - The user write his note
        - The user clicks on the save button inside the text field
        - The system closes the text field saving the new notes
    - Exception:

        - If the user does not click on the save button after writing his notes, the system will lit up the save button. If the user clicks out again, the system will not save the new changes.

Edit due dates:

- Flow of actions:
    - The user clicks on the date button
    - The system will drop down calendar
    - The user can change the due date
    - The system will save the new due date
    - The system closes the calendar after the user clicks
- Exception:
    - If the user does not add a new date since the creation of the task, the system will show as default the next immediate day.

Progress tracker:

- Flow of actions:
    - The user slides the slider
- Precondition:
    - The progress tracker is set to 0

- The user can visit the 'Tutorial'.
    - Flow of actions:
        - The landing page is seen
        - The 'tutorial' button is clicked
        - A web browser tab is open with a YouTube video

Modeling

We used figma.com for the Wireframe('mock up') modeling. Here we represented the UI and saved all traceability aspects to ensure high quality on static and some dynamic testing.
It can be seen here: https://bit.ly/3rizTGe
The main goal of Team Osiris is to provide itself or another team of software development enough tools to guide their own project with the specified requirements. As such it is not displayed: Architectural, Data Types and Procedural design, because it is not intended to show that level of specificity, each team can decide what fits best on future technologies.

Implementation

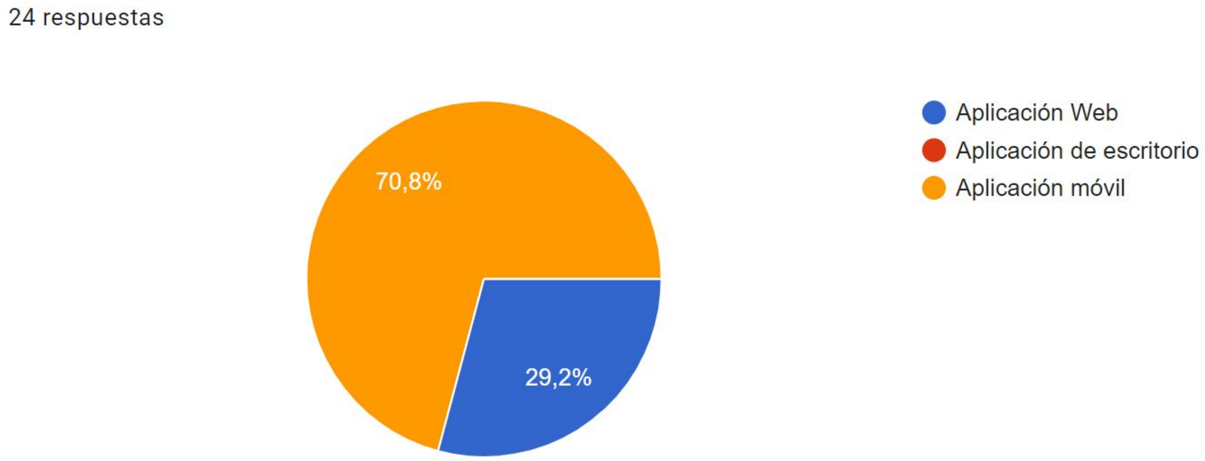
For the MVP of the second increment the development team makes the commitment of releasing functional wireframes (previews and templates) as the Implementation artifacts for the reasons mentioned prior (Modeling section). But if everything is finished on time, the mentor gives approval and users give positive feedback, a coding phase may take place (so more testing and artifacts will be presented and indexed). With that consideration we present a viable plan for such events:

For the platform of Osiris we presented the customers with a new survey, in which 24 participants were registered and decided 'Mobile App' for the platform; but for this early stage of Osiris, ‘WebApp’ can be really helpful to make instant updates and guide the UI and back/front end for the Mobile App, that's why we will make functional prototypes of the product on Web but on further stages it will be deployed con Android & IOS.

For the implementation phase we considered generating the functional code with the JavaScript framework React, we use this framework for the functionalities and facilities that it provides. In this way we focus on developing prototypes of functional interfaces for the users and start evaluating how the user would interact with our system. The implementation follows a continuous integration methodology where we create the root component and each member works with separate components and merges them to the system, all this process is made in our GitHub repository.

For considerations of the schedule we aren’t making the commitment of releasing the code for this increment, just in case that we finish the MVP of the code generation it would be presented in the final delivery.

(Evidence and results of the survey):



Translation: 24 answers. WebApp(blue); Mobile App(yellow); Desktop App (red)

Testing

For the project, three types of tests were carried out. The first one being inspection, where we, with the help of our mentor, decided if each requirement is properly represented in the design. The second test was the ‘acceptance testing’, where we showed 6 random students our wireframe design and later asked them relevant questions that were intended to help the development of a new version of the project. The final type of testing was the ‘five seconds test’, where information was gathered and imported into a spreadsheet, this is intended to provide feedback of the UI design.

Mentor feedback:

During a meeting with our mentor, we reviewed in detail the prototype that we designed. Some of the observations that our mentor gave us focused on the organization of the UI components. He also recommended using other tools for the layout.

However, further analysis of the prototype shows that it meets the basic functions set out in the requirements document. Since it is only a prototype, it limits the visualization of all the functions that the system must fulfill, and only focuses on the structure of the elements and keeps the style, in terms of colors and format, simple.

Acceptance testing (Q/A):

For this section, we performed a survey on 6 software engineering students. According to the results, we can conclude that indeed the prototype accurately represents the basic features that had been established. However, the opinions of users also revealed a disagreement with the design that was presented. Some users said that the colors used are distracting or confusing. They also indicated that some features could be implemented, such as being able to customize the schedule, allowing tasks to be grouped, adding some kind of visual effect to the icons.

The most notable suggestions were the desire to modify the colors used, enlarge the frames, remove the pencil icons, and adding a menu that pops up when clicked that groups the delete icon, edit notes and edit tasks.

https://docs.google.com/spreadsheets/d/1HYlgbaf8r4m12fkXYYfUy2Lo22G3ugskVvfwrTlHq3w/edit?usp=sharing

For the second wireframe attempt we took into account the observations that users shared in the previous survey and suggestions that the mentor told us during the meeting. The sketches showed a more detailed structure of the interacting elements in the wireframe, with a simpler design, no required functions were deleted, we rearranged text boxes to improve visualization. To validate whether or not our wireframe meets the needs of users, we surveyed 6 other software engineering undergraduate students for our acceptance test. The users showed a really high rate of approval in almost all aspects of the design of the wireframe. Moreover, to know their insights on the UI design(for further analysis), a sample was also tested and they said the colors are attractive, the spaces are great and the interface is intuitive.

Finally, among the observations we had: fonts should be improved and the background image can be more attractive. One respondent said he would like the settings feature to be added because he thinks it would be very important. In conclusion, we can say that the results are satisfactory.

https://docs.google.com/forms/d/e/1FAIpQLScR5dnDwhTXXQBdk4yoT26gmSLmNdWuQ-jcqrxZhoLgUszomg/viewform?usp=sf_link

5 seconds test (Q/A):

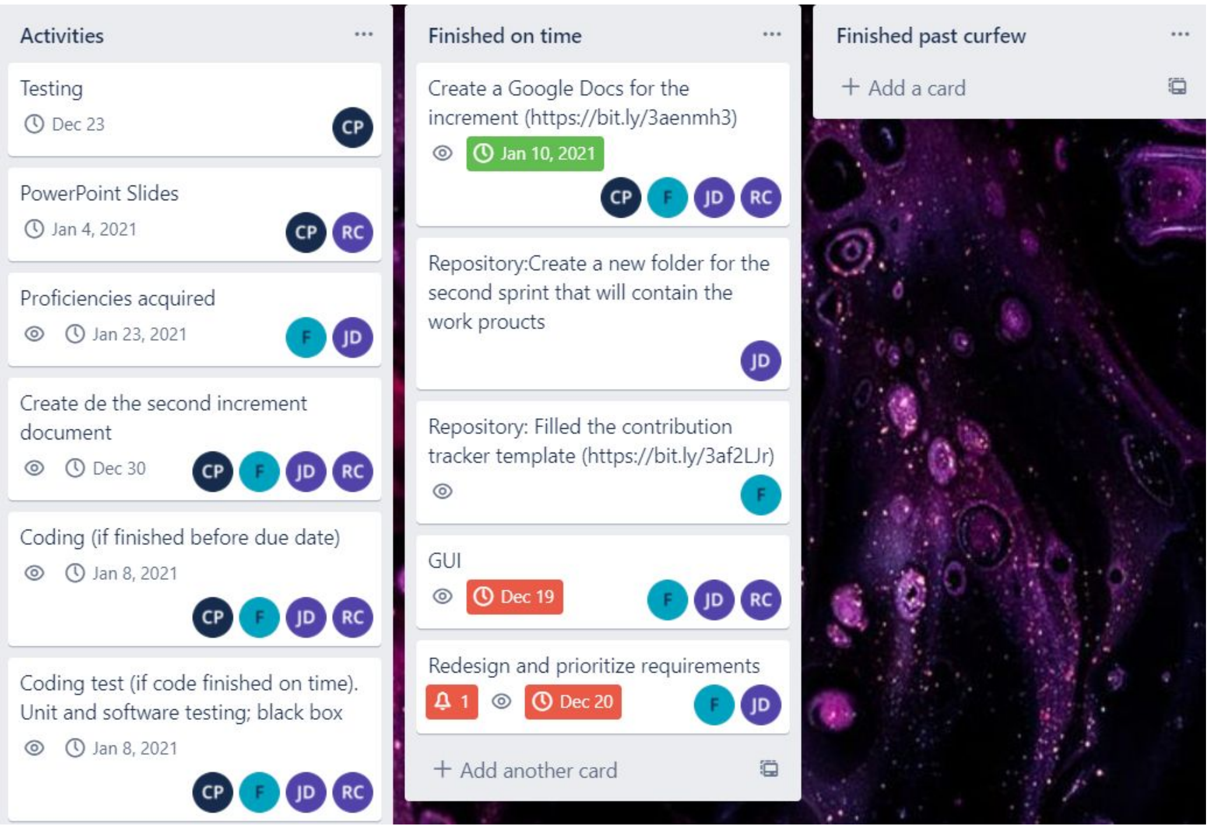
The information collected in the “five seconds” test reflected the same disagreement regarding style and color combination. The testers shared their opinions about the design of the webapp, some of them could not identify which components were supposed to be buttons. "the tutorial icon does not look like an icon, I thought it was part of the design", one said. It is important to note that when we explained each section to them, they understood the functionality better (this could indicate that we are on the right track).

These tests were carried out through the Teams platform where some of the sessions were recorded. The objective of our prototype was explained to the testers, they were also informed that the sessions were being recorded, and at the end, we thanked them for their collaboration and their time.

As for the second ‘five seconds test’ the users surveyed said they are satisfied with the information that is handled in wireframe, the design is intuitive, the interface performs the right functions, but it can be further improved. They also said that for the sample they would like the task blocks to be bigger, the help icon is too big, the logo takes up too much space and the background image should be changed.

https://docs.google.com/forms/d/e/1FAIpQLSdqMp484hZ_zyKqoxCFN3Hjc3HiW7E4TuBCJB4vIatirSWrAQ/viewform?usp=sf_link

Project Planning / Teamwork:



Activities

Testing
Dec 23
CP

PowerPoint Slides
Jan 4, 2021
CP RC

Proficiencies acquired
Jan 23, 2021
F JD

Create de the second increment document
Dec 30
CP F JD RC

Coding (if finished before due date)
Jan 8, 2021
CP F JD RC

Coding test (if code finished on time). Unit and software testing; black box
Jan 8, 2021
CP F JD RC

Finished on time

Create a Google Docs for the increment (https://bit.ly/3aenmh3)
Jan 10, 2021
CP F JD RC

Repository:Create a new folder for the second sprint that will contain the work proucts
JD

Repository: Filled the contribution tracker template (https://bit.ly/3af2LJr)
F

GUI
Dec 19
F JD RC

Redesign and prioritize requirements
1 Dec 20
F JD

+ Add another card

Finished past curfew

+ Add a card

We used Trello for the assignment of the duties, the accountables(Fernan Cetina[F], Jorge Rodríguez[JD], Rodrigo Castrejón[RC] y Cinthia Huchin[CP]) and due dates. The artifacts with accountables are:

1. The UI model (F, JD, RC)
2. This document (Team Osiris)
3. PPT presentation (CP, RC)
4. The contribution tracker (F, JD)
5. Github binnacles and project advances (Team Osiris)

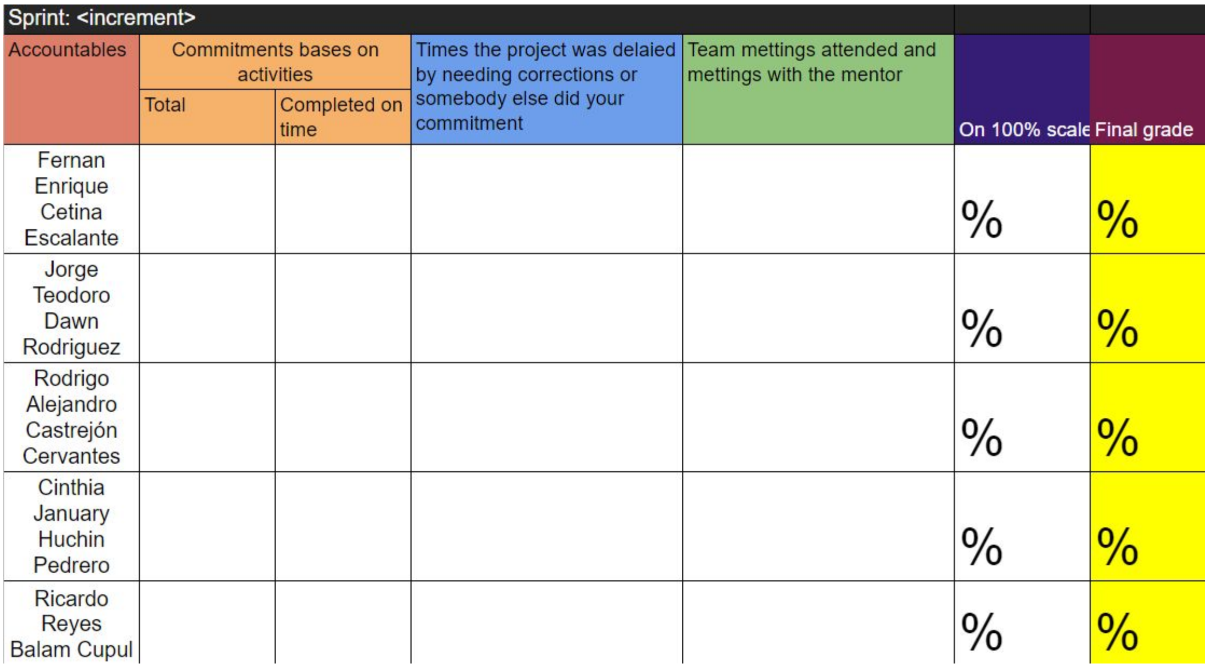
It can also be seen that a Coding phase with testing is planned but not mandatory for the MVP of the increment.

Furthermore we have 3 lists according to the First Increment document methodology (activities, and if 'finished on time' and 'finished past curfew').

For the team dynamic and development process it is used 'Scrum Methodology' and can be reviewed at length on the First Increment Document available in the Repository.

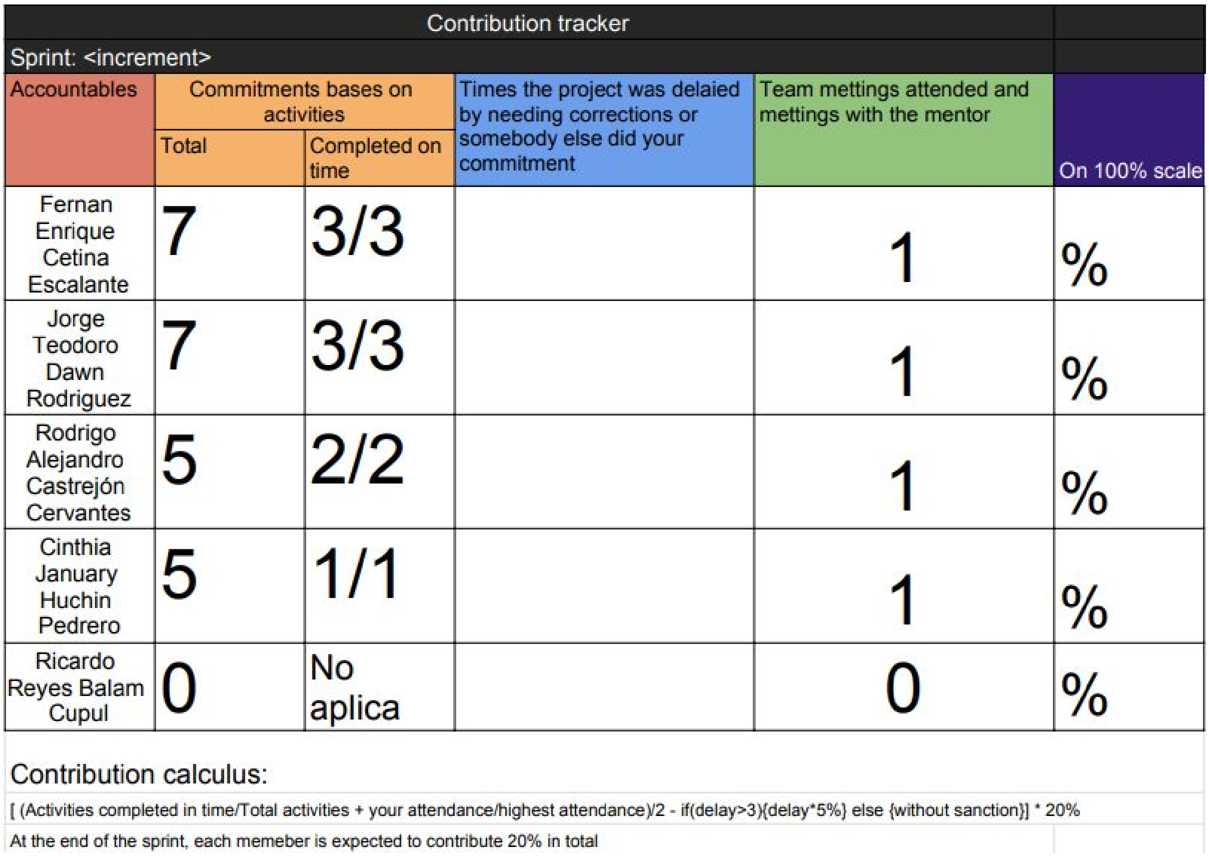
Repository: https://github.com/FernanCetinaE/TeamOsiris

As for the format of the contribution tracker we implemented this scheme:



Sprint: <increment>						
Accountables	Commitments bases on activities		Times the project was delaied by needing corrections or somebody else did your commitment	Team mettings attended and mettings with the mentor	On 100% scale	Final grade
	Total	Completed on time				
Fernan Enrique Cetina Escalante					%	%
Jorge Teodoro Dawn Rodriguez					%	%
Rodrigo Alejandro Castrejón Cervantes					%	%
Cinthia January Huchin Pedrero					%	%
Ricardo Reyes Balam Cupul					%	%

And we used it as follows:



Contribution tracker					
Sprint: <increment>					
Accountables	Commitments bases on activities		Times the project was delaied by needing corrections or somebody else did your commitment	Team mettings attended and mettings with the mentor	On 100% scale
	Total	Completed on time			
Fernan Enrique Cetina Escalante	7	3/3		1	%
Jorge Teodoro Dawn Rodriguez	7	3/3		1	%
Rodrigo Alejandro Castrejón Cervantes	5	2/2		1	%
Cinthia January Huchin Pedrero	5	1/1		1	%
Ricardo Reyes Balam Cupul	0	No aplica		0	%

Contribution calculus:

[(Activities completed in time/Total activities + your attendance/highest attendance)/2 - if(delay>3){delay*5%} else {without sanction}] * 20%

At the end of the sprint, each memeber is expected to contribute 20% in total

Subject Proficiencies:

Generic proficiencies:

- Works with ICT in his/her professional interventions and in his/her private life in a suitable and responsible way.
- Works with others multi, inter and transdisciplinary environments in a cooperative way.
- Takes decisions in his/her professional and private practice in a responsible manner.
- New: Manage the knowledge in his/her professional and private practice in a responsible manner.

Specific proficiencies:

- Identifies the concepts linked to the phases of requirements, design, development, testing and maintenance, according to the recognized organisms of the discipline.
- Identifies human factors immersed in Software Development that contributes to the success of the Software project.
- Uses software engineering terminology properly in its professional interventions.
- New: Establishes the differences between the professional profiles in ICT.

We consider that every proficiency listed is developed in complete way, to demonstrate such statement we tried to ensure that in every part of the process a proficiency was met and that's how we actively seeked to cover them all on a balanced way, although some were more prominent than others apparently, we don't consider that to be the case on a personal level.

Demonstration of how the general proficiencies are met by specific activities of the development process:

- Works with ICT in his/her professional interventions and in his/her private life in a suitable and responsible way.
    - First increment: Throughout the whole project different tasks like “Work process” or “Meetings” require a different ICT tool, like meetings that are more accessible via TEAMS, Excel tracks better a checklist of contributions or Trello lets us visualize in a graphic way the pending activities. And for the future, more specific software like the ones used for modeling will also be tested to fit our needs and this abstraction thinking of deciding which one suits best is carried to our professional use of tools but also evolves our personal needs.

    - Second increment: like in the previews increment we used tools like Trello where dates were fixed, commitments made and were not allowed to change; for mentor feedback we used Power Point presentations and Teams and for the Wireframe tool we worked with Figma and tested if it fitted our needs and any slate not created from the team received credits.

- Takes decisions in his/her professional and private practice in a responsible manner.
    - First increment: At the beginning of the increment each member makes a commitment based on their time and abilities ("Work process"), so being self aware takes great relevance to make a responsible decision that affects your work and others. As joined tasks develop, the sub-teams are able to negotiate what is in the best interest of the daily scrum(e.g. in "Subject proficiency" Teodoro works on the repository management and Fernan on quality assurance) and in this sense, commiting to specific actions helps you develop a better decision taking mind.

    - Second increment: For this increment we did our best for having a well-organized list of activities so that we could be prepared for each one and advance using our most developed abilities. We took important decisions as a team like prioritize the design and testing phases. For each of them we achieve a more detailed knowledge of the phases in which we were working and we engage ourselves to finish our activities on time, being this an fundamental goal for considering the code generation. In this case we Fernan, Rodrigo and Teodoro integrated their knowledge and research to conceive the design of the UI and Cinthia working in the collection of the information for the user acceptance in order to improve our UI design.

Demonstration of how the specific proficiencies are met by specific activities of the development process:

- Identifies the concepts linked to the phases of requirements, design, development, testing and maintenance, according to the recognized organisms of the discipline.

    - First increment: All the phases(based on SWEBOK by IEEE) were considered in the creation of the timeline and have a determinate place in the organization of the documentation. We also considered making generic templates that can be filled according to the needments of the later phases. Activities like "Timeline" and "Scrum organization" are an example.

    - Second increment: as a direct proof of this proficiency the Modeling subtitle can demonstrate it and through team meetings and meetings with the mentor, such concepts were implemented.

- Identifies human factors immersed in Software Development that contributes to the success of the Software project.

    - First increment: We considered dividing most of the activities and hash out the possible contribution of each member of the team depending on its abilities. We also prioritized the communication in the meetings and tried to be the most attentive as possible, we always maintained a respectful approach in the activities and valued the contributions of the different stakeholders. On the other hand in the early activity “Product definition” we asked users to give us their needs and for the next increment a new contact with them will occur.
    - Second increment: once again we had active dialogue within the team through meetings and by giving positive feedback on individual work; and we had even more interactions with users than in the previous increment via interviews and surveys.

- Uses software engineering terminology properly in its professional interventions.
    - First increment:Since our project was conceived and during “Product definition” and “Timeline” we adjust a holistic view of the development process. We take into account activities as: requirements eliciting, approving requirements, communication, scrum framework. In a broader perspective, we try to evaluate the feasibility of our project in question as “How are we going to develop this product?”, “Does there already exist another product that provides this service?”.

    - Second increment: in the Trello screenshot, Github repository and along the meetings such terminology is actively used, like mentioning when mentioning the important distinction between the capacities of a code focused engineer and professional profiles such as QA and UX.

- New: Establishes the differences between the professional profiles in ICT.
    - Second increment: as the development of the increment evolved we noticed that complex task as quality assurance, design and implementation (functional code) really require specified professionals with experience in order to ensure that users get the best possible product; we worked our way out by improvising but dedicated professionals in areas like Coding, Quality and UX are key to the success of a project.

Additional proficiencies

According to the didactic planning of the subject, there are considered determined proficiencies for each unit, we try to include them by default but we also consider other proficicies that were involved in our process and were not in the planning of these units. We considered these proficiencies to be our additional proficiencies.

Generic proficiencies

- Works with others in multi, inter and transdisciplinary environments in a cooperative way.
    - First increment: Some tasks like writing this document(from “Application definition” to “subject proficiency”) requires a different depth of knowledge, some team members are more skillful in english than others and we take advantage of that; in this way, tasks like creating a repository, the project vision, ways to organize and track progress and so on, are led by the ones with more experience that share their insights and helps the overall achievement of new concepts.

    - Second increment: in this increment members like Rodrigo knew how to use Figma and Teodoro knew UML so the rest of the team got mini classes on how to use them and in consequence the team work improved.

- New: Manage the knowledge in his/her professional and private practice in a responsible manner.
    - Second increment: During the second increment process we needed to learn new abilities and adapt us to the requirements for the development phases involved; in order to to reach our goals we had to be self-aware of our edges and organized our time in order to have enough time for managing our lack of knowledge and our personal lives, specially in this times of the year where we are with our loved ones. Throughout this increment, we spend part of our time learning how to use design tools and trying to use them in the best way. We also had to analyse the feasibility of generating the code in order to recognize what knowledge we have in this area.

Specific proficiencies

In the case of the specific proficiencies we considered them in an holistic view and we try that each proficiency is involved in our process.