MUSIC LIBRARY

TABLE OF CONTENTS

1. General Analysis.
   1. Structure or Interface Analysis.
   2. Cases and Classes Diagram
2. Project requirements.
3. Project planning.
   1. Project organization.
   2. Priority of each task.
   3. List Task.
   4. Total approximate project estimate.
   5. Project schedule.
4. Incident management.
5. Irrigation Management.
6. Management, control and quality metrics.
7. Sprint development.
8. GIT documentation.
9. Tools used in the project.

MUSIC LIBRARY

1. General Analysis.
   1. Structure or Interface Analysis.

The web must be development to see in different views for all technologies dispositive. It will be a one page where can see all songs search list

* *“Search.*

*The web must be dynamic and the actions will be carried out without reloading the page.*

*The search engine must implement the following options:*

* *Search songs by name.*
* *Search artists by name.*
* *Search albums by name.*
* *Search music videos by name.*

*In addition, the following additional optional parameters must be applicable to the search:*

* *Filter by country.*

*You should consult a specific third-party API to obtain the necessary information to apply the filter.*

*Ex:* [*https://api.printful.com/countries*](https://api.printful.com/countries)

* *Filter by explicit content “Yes” or “No” options.*
* *Filter by result limit Range from 1 to 200.*

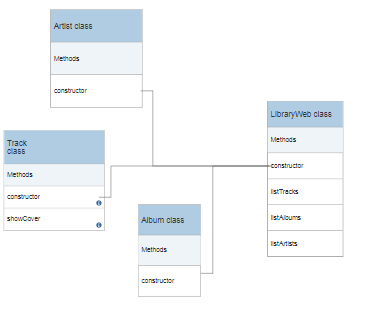
*To make the API call from JavaScript you will have to first understand what JSONP is and when to use it.*

* *Results.*

*Once these requirements have been analyzed, you should consider the following points when displaying the results:*

*Depending on the search performed, you should be able to see for each item found the following information at least:*

* *For a song:*
* *Cover.*
* *Name of the song.*
* *Artist name.*
* *Album name.*
* *Song Price.*
* *Release date.*
* *Song length.*
* *Musical genre.*
* *Audio sample of the song.*
* *Song Link in iTunes.*
* *For an artist:*
* *Name.*
* *Musical genre.*
* *Artist link on iTunes.*
* *For an album:*
* *Cover.*
* *Album name.*
* *Artist name.*
* *Album price.*
* *Number of songs.*
* *Release date.*
* *Musical genre.*
* *For a music video:*
* *Cover.*
* *Name of the song.*
* *Artist name.*
* *Song Price.*
* *Release date.*
* *Song length.*
* *Musical genre.*
* *Clip video sample.*
* *Link of the music video on iTunes.”*
  1. Cases and Classes Diagram



1. Project requirements.

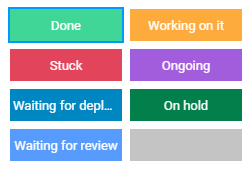
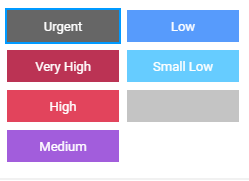
* “Create a clear and orderly directory structure.
* Use jQuery for the entire project development.
* Implement all functionalities on the same screen dynamically.
* Make use of third-party APIs to solve the project.
* Additional search parameters will be optional when performed, may or may not be applied individually (visual example in the wireframe).
* The web will have to be responsive and compatible with the main browsers in the market.
* The search must be carried out each time the input changes, that is, each time a character is entered, the page will show the content found.
* Any item found by a search can be added to favorites and will be displayed in the favorites section (example in the wireframe).
* Items previously found may be deleted from favorites.
* Favorite items should be stored in localStorage.
* All comments included in the code must be written in English.
* Use the camelCase code style.
* In the case of using HTML, never use online styles.
* In the case of using different programming languages ​​always define the implementation in separate terms.
* It is recommended to divide the tasks into several subtasks so that in this way you can associate each particular step of the construction with a specific commit.
* For the project documentation a PDF version is required within the repository.
* You should try as much as possible that the commits and the planned tasks are the same.
* Delete unused files.
* You must create a correctly documented README.md file in the root directory of the project (see guidelines in Resources).”

1. Project Planning.
   1. Project organization.

*“Next you will have to create a document that specifically exposes how the current project is organized. It is important that it be improved throughout the life of the project. The document must include at least:*

* *Requirements documentation*
* *List of tasks to be performed*
* *Priority of each task*
* *Title and description of each of them*
* *Difficulty level*
* *Estimated time for each task*
* *Record of incidents that were detected during project execution*
* *Schedule or Calendar of the project. (Choose the one you consider most chosen given the size of the project)*
* *Quality metrics*
* *Risk documentation in the event that these exist*
* *Documentation about the git workflow you are going to use*
* *Documentation about the tools used in the project*
* *Record of lessons learned”*
  1. Priority of each task

A total of 8 SPRINTS of the project were defined with each of its tasks, assigning the priority of execution and the state of the task.

8 scenarios with their respective colors were proposed for the status of each task:

* *Done* (Light Green), for task have been finished
* *Working on it* (Orange), tasks are underway and almost finished.
* *Stuck* (Red), tasks are stagnant for some reason and could not be finished.
* *Waiting for review* (Blue), tasks are waiting for a review and approval.
* *Waiting for deployment* (Dark Blue), tasks are pending for status update and repository update.
* *Ongoing* (Purple), tasks are on development.
* *On hold* (Dark Green), tasks are waiting for something resource or search information or depend on another tasks to start.
* (Gray), tasks without assign.

6 scenarios with their respective colors were proposed for assign the tasks priority. The priority upper level is *High* (Red) and *Small Low* (Light Blue) is the priority lower level. For tasks with null scenario (it doesn’t have priority) it will be assigned with *Nothing* (Gris). The *Urgent (Dark Gray)* scenario was established for tasks with extreme and excessive importance.

* 1. List Task.



* 1. Total approximate project estimate.

Using a tool for project management, called monday.com. Tasks are reflected within sprints or stages. Adding the hours each task for each sprint we have that the project will be executed in 32.8 hours in 6 days, it will be 5.6 hours average per day for the project deadline.



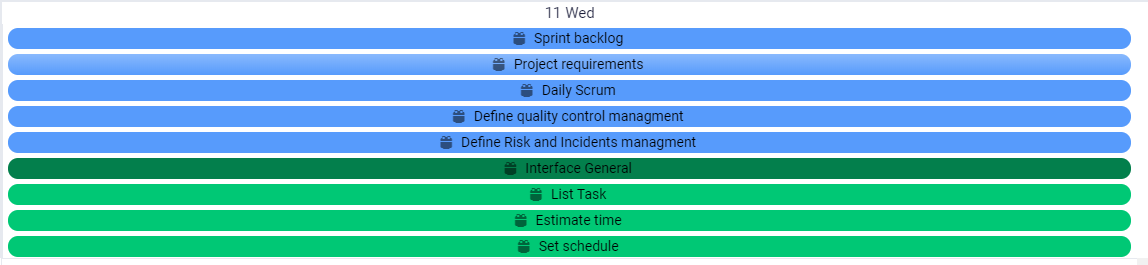
The first sprint called General Analysis were divided in two main sub-sprints: Structure analysis and Project Planning.



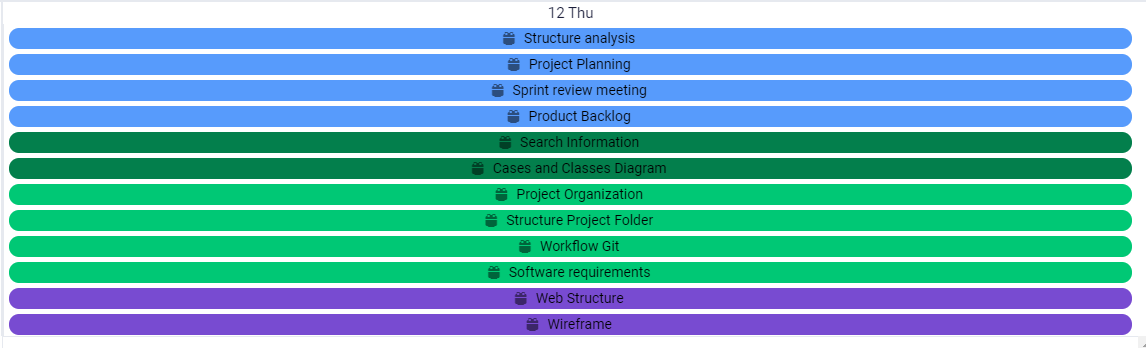
* 1. Project schedule.

According to the project schedule, the tasks to be executed for the work days in each of them are ordered by priorities to prepare the project calendar with their daily activities.

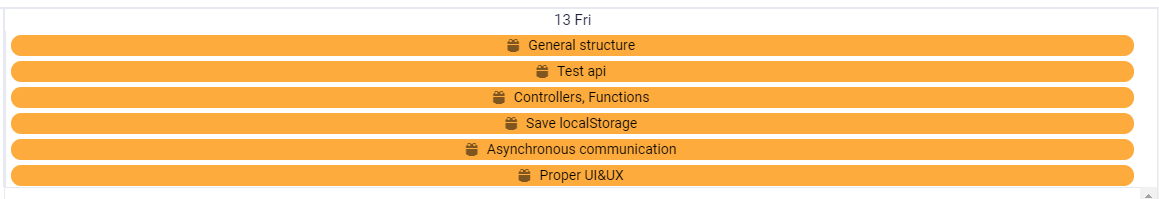
On December 11, only tasks belonging to the General Analysis sprint will be executed. Include the tasks belonging to the Structure Analysis will be started.



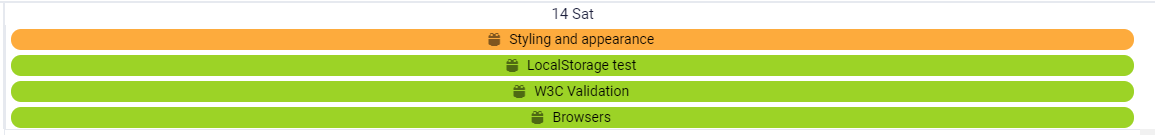
On December 12, the pending tasks of General Analysis will be executed and accomplished this day and also the structure analysis sub-sprint and planning tasks too. Two technical design sprint tasks will be developed this day.



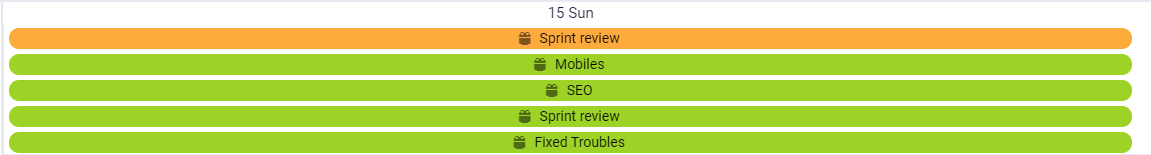
On December 13, only tasks belonging to the development sprint will be executed.



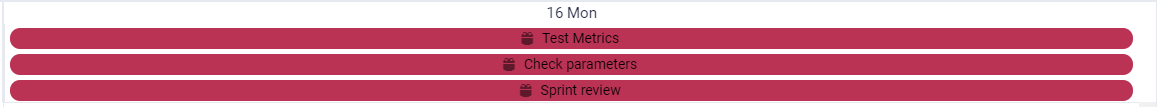
On December 14, the pending tasks of the development will be executed, in which they will be finishing pending details about the layout and styling appearance. First tasks belonging to the sprints test will also be executed too.

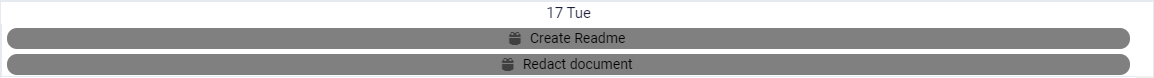


On December 15, the final sprint development revisions are planned and it will be finished all site test too. When the test are ready finished, the troubles are going to be fixed.

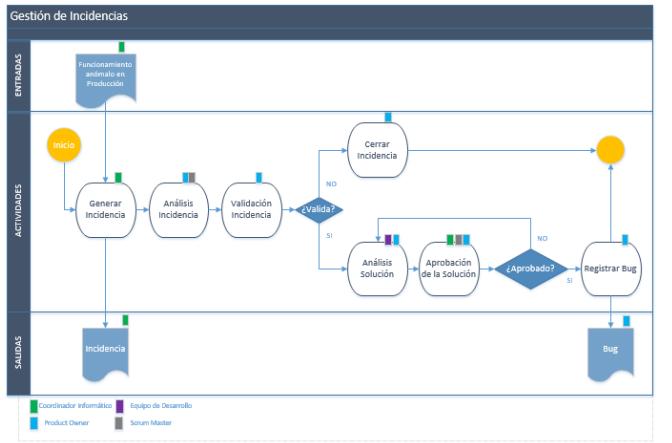


On December 16, the quality sprint will be development. In the end a sprint review it will have been done.



On December 17, the final revisions to the project are planned and the tasks belonging to the sprint control - follow-up will be executed, to manage unforeseen events and unresolved incidents, to finalize the writing of the deliverable document. 

1. Incident management

The procedure to manage the incidents presented in the project is proposed.

The incidents will be recorded in the updates of each task of the support tool for project management used in its development. This section will record the incident detected, if validated, the solution found for this solution and if it was correctly solved. The incidents will be reviewed again in the partial and final meetings of each sprint, to review the pending incidents and generate actions so that they are not recurring.

1. Irrigation Management.

A procedure is proposed to manage the possible risks that may arise in the project. Based on the SCRUM methodology, we have to:

*“Project Risk management includes the processes to carry out risk management planning, as well as identification, analysis, response planning. To increase the probability and impact of negative project events. It is important periodically to manage the possible risks that affect the project, both directly and indirectly. Therefore, the risk management process must be carried out prior to scope management, taking advantage of the scope control and risk control reports obtained at the end of each iteration.*

*Once identified, they perform an enumerated analysis of each of them, in order to be able to propose actions in this regard.”*

“Prepare a series of proposals for reaction to risks, which depending on the severity of the risks, or the probability of their occurrence, may be of the following type:

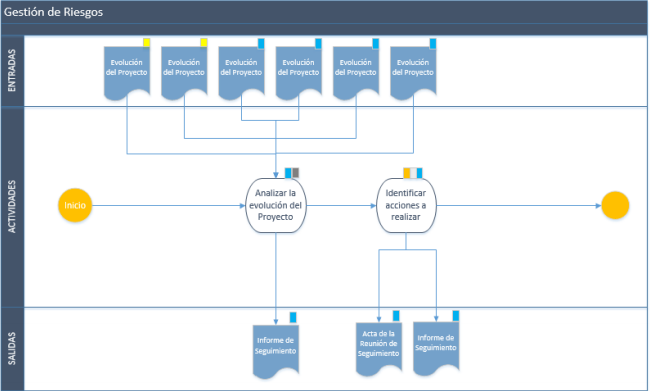
a.) Evasive: To be able to perform some action that allows us to definitely avoid the risk

b.) Mitigating: If you cannot avoid the risk, try to minimize the impact

c.) Transferred: Obtain through any agreement or action that the risk is assumed by third parties

d.) Acceptance: Sometimes it is not possible to perform any of the above actions and the risk must be assumed by absorbing the impact in the best possible way

e.) In the risk response planning, the need to initiate the response plans is evaluated, so that they can be included in the scope management.”



In case of detecting any risk in the execution of any task, they will be recorded in the updates of the same in the project management support tool. The management procedure will be carried out as a high priority. In case of not detecting any risk, the possible risks that may exist in the partial and final reviews of each sprint will be analyzed.

1. Management, control and quality metrics.

Taking a guide to control the Software Quality and the satisfaction it has with the customer is necessary, define the parameters, indicators or measurement criteria. According to the previous idea, it can be said that a software has certain measurable indices that will be the basis for determining quality, control parameters and productivity improvement. Therefore, when selecting quality indices, the control process must be established. It is suggested to perform the following steps:

1. Define the software to be controlled: classification by type, scope, complexity, etc., in accordance with the standards established for software development.

2. Select a measure that can be applied to the control object. For each kind of software it is necessary to define the indicators and their magnitudes.

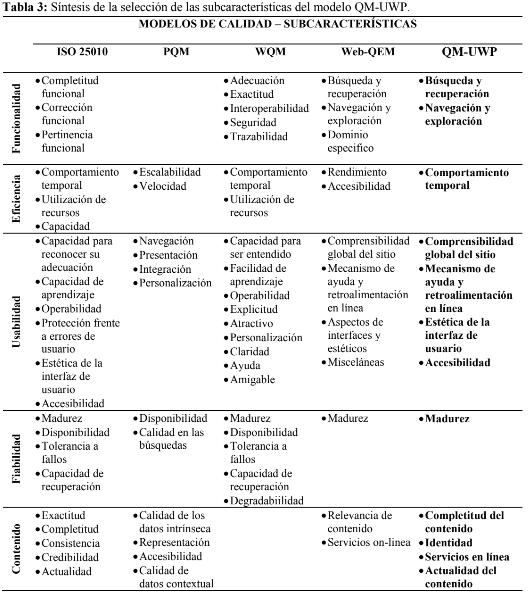
3. Create or determine the valuation methods of the indicators: manual methods such as questionnaires or standard surveys for the measurement of expert criteria and automated tools to measure the calculation criteria.

4. Define the organizational regulations to carry out the control: who participates in quality control, when it is carried out, what documents should be reviewed and prepared, etc.

A functional test report is presented where the minimum requirements of the portal for development tasks are verified.



The quality parameters to be measured are: 1) Functionality, 2) Reliability, 3) Ease of Use, 4) Efficiency, 5) Ease of Maintenance and 6) Portability.



1. Sprint development.

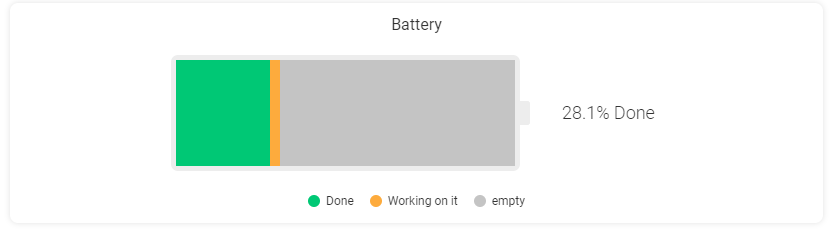
In the reviews of the sprints, the retrospectives of the incidents presented, the possible risks that are detected and the knowledge or lessons learned in the completed tasks are carried out. The results obtained in the partial reviews of each Sprint are presented below.

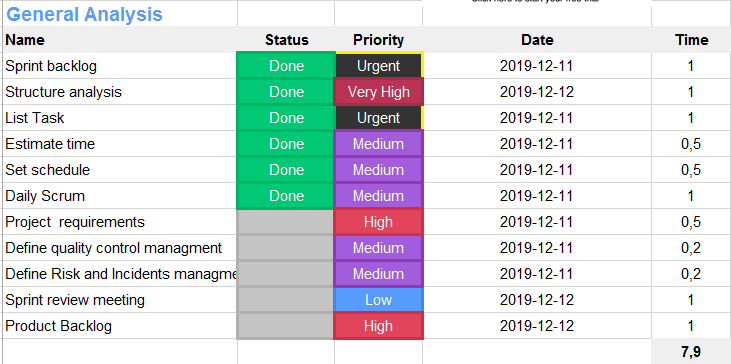
* Revisión Final del Sprint 1: Al definir los SPRINTS y las tareas se ejecutaron de acuerdo a la planeación establecida. Se agregó una tarea llamada análisis general, en el que se planifican las actividades que se realizarán en la etapa de desarrollo.

INCIDENTS

* Import SASS with Web pack.
* Consume a countries third-party API. It was necessary consume a different countries API example.

The panorama of the current sprint in which several tasks were completed. The current development of the project is 28% and that of the current sprint is as follows:





CONCLUSIÓN:

It’s more easy use web-pack for the Project with ES6.

The main idea to use JavaScript for this project is the Asynchronous communication with thirds APIs.

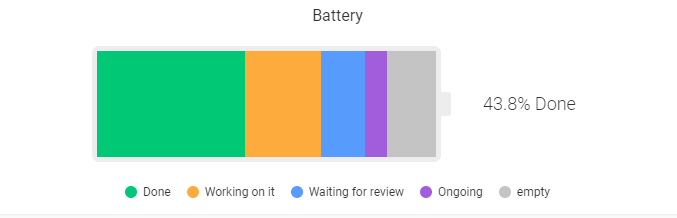
With a JavaScript request we’re using inside callbacks.

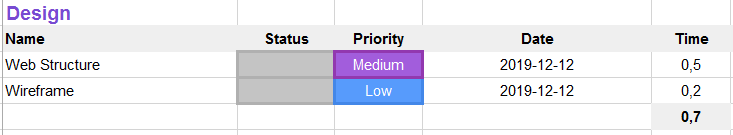
* Final Review of Sprint 2: The technical design of the website was proposed, taking advantage of the WIREFRAME provided in the requirements. Resources were dedicated to the search for information resources, and images of the project, the user interface. The sprint was executed one hour less than the planned one.

INCIDENT

* + It’s necessary design several web parts site to complete de functionally and activities the library musical.

The technical design sprint panorama, the planned tasks were completed in their entirety and in less time compared to the planned one. This allowed more time to be devoted to the “Information search” task of the previous sprint, in order to minimize the risks analyzed and establish in a more concise way the activities that belong to each of the tasks of the development stage. The current development of the project is 43.8% and that of the current sprint is as follows:





CONCLUSION:

Facilitating a previous design in the project requirements minimizes the hours allocated to the planning of the physical structure that a website must have. However, the technical design of what was needed was carried out.

* Final review of Sprint 3: The development time in tasks and activities in the development stage was longer than planned, with 11 hours of difference. The proposed tasks responded to the fulfillment of all the requirements, in a longer period of time. Therefore, more resources were allocated to update the activities that are within the established tasks. Although it detected the risk in the previous phases, it could not be minimized to the maximum.

The Github platform was integrated with Monday.com, where any commit made to the repository will automatically create updates on the relevant task.

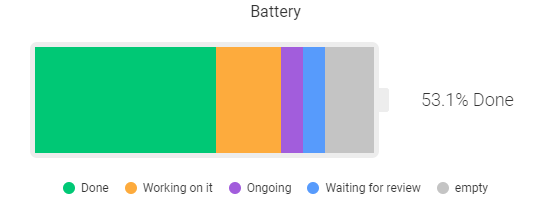
INCIDENT:

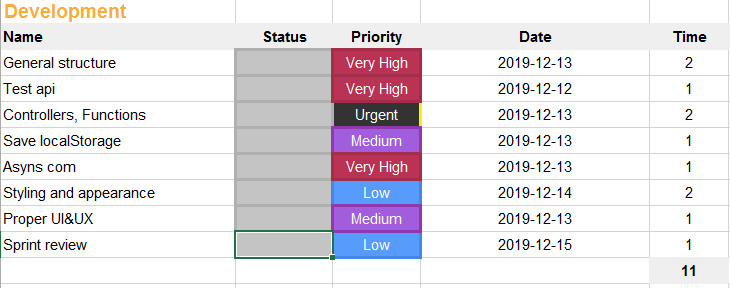
* + Learn to use SASS.
  + Add media breakpoints in SASS.
  + Unknown files structure about SASS.

RIESGOS

* + Unknown knowledge of the SASS can lead to delivery delays.

In order to minimize risks, greater resources are devoted to the current sprint. However, a large part of the tasks are awaiting revision and the development time is longer than the planned time. There is a difference in the hours of execution in the tasks carried out almost double with respect to the planned time. The current development of the project is 53.1% and that of the current sprint is as follows:





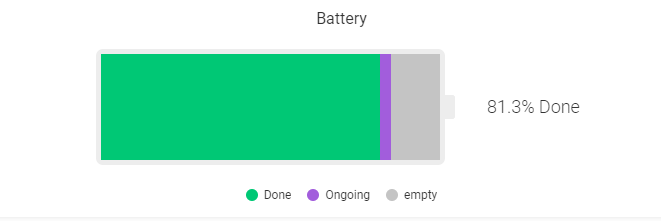
* Final review of Sprint 4:

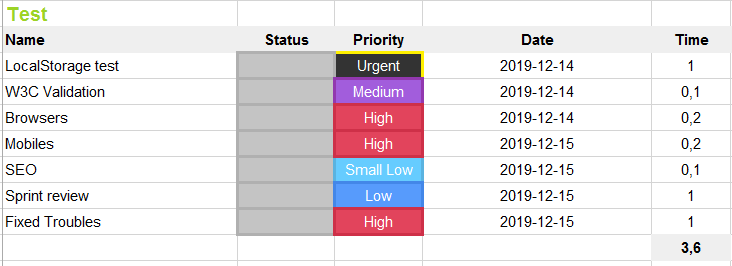
Again the tests are performed, in browsers and on mobile devices to meet the requirement of being responsive. For this, the template designed to verify the characteristics and leave a record of what is observed in the tests is used. You can see the results of the tests performed in Annex 2. Functional tests. The task that was not necessary to execute, the validation of the code in W3C, was raised. However, the time that had been allocated for its execution is not significant for the rest of the tasks. The formatting of the entire portal code is made to facilitate its compression.

CONCLUSION

Performing tests at the time of development prevents processes from being returned to earlier stages to correct them. Therefore the validations in browsers and mobile devices were developed successfully and passed the content tests. However, the functionality tests were not entirely assertive, for the reasons mentioned above.

The project is ready for its quality and monitoring phase. The current development of the project is 81% and that of the current sprint is as follows:

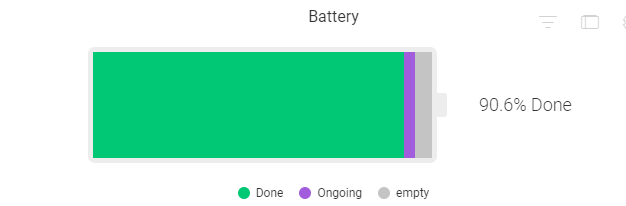


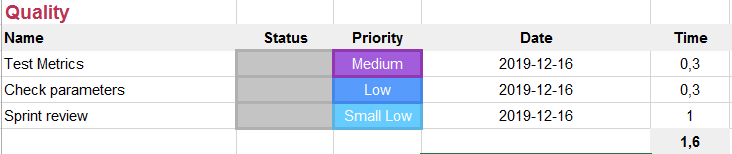


* Revisión final del Sprint 5:

Audits are performed in the CHROME browser inspector to the portal, in which several anomalies are detected. These anomalies will be corrected in the next sprint. Audits can be seen in Annex 3. Audits to evaluate quality metrics.

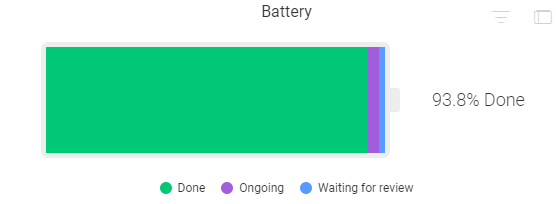
The quality indicators evaluated in several aspects with good results. The best practice metric was the lowest with 85%. The total execution of the sprint was according to plan and the percentage of the project is 90.6% and the current sprint is:

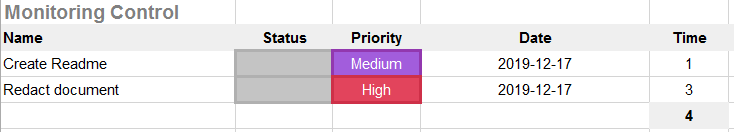




* Gestión de improvistos:

Las incidencias detectadas en las anteriores revisiones que no fueron solucionadas, se solucionan en esta sección. Dentro de esta sección se dedican recursos para finalizar la redacción del documento para presentar.





1. GIT documentation.

The GIT repository is located at: https://github.com/diegosilva91/Music-Library and was cloned into the computer to create the corresponding static portal files. The master BRANCH was used to carry out all the loads with their respective commit.

1. Tools used in the project.

Project management was carried out with Monday.com, to plan tasks, sprints, calendar and schedule. The link for more detailed planning is available at https://assembler-company.monday.com/boards/361790718/.

To design the class diagram, a http://www.jrromero.net/tools/jsUML2 was used

ANEXO 1. PRUEBAS FUNCIONALES



ANEXO 2. AUDITORÍAS PARA EVALUAR MÉTRICAS DE CALIDAD.

