Plan del Proyecto

Proyecto Artemis

Versión 1.0

# Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 05/11/2020 | 1.0 | Propuesta del Plan para el Desarrollo del proyecto | Analista |
| 02/12/2020 | 2.0 | Documentación de GitHub y modificación del flujo de artefactos | Analista |
|  |  |  |  |
|  |  |  |  |

# 

# Otros Documentos

|  |  |
| --- | --- |
| **Previos a este** | **Siguientes a este** |
| Documento de Negocio | Cronograma del Proyecto |

# 

# Tabla de Contenidos

[**Historial de Revisiones**](#_7mjm6ch5i7f5) **2**

[**Otros Documentos**](#_l1pvnlkb538x) **2**

[**Tabla de Contenidos**](#_tb80powsq4d3) **3**

[**Plan de Proyecto**](#_n422xb7a5slc) **5**

[Introducción](#_uj9jdigibew2) **5**

[Propósito](#_pu65a0uzk8wp) 5

[La metodología RUP](#_vd2m48tk2ant) 5

[Roles](#_826xmmeqv255) 6

[**Organización**](#_w6fhf7amp85e) **6**

[Fases, Actividades y tareas](#_vnt6654lp9bl) 6

[Secuencia de Artefactos](#_l21cthcpksy) 7

[Repositorio GitHub](#_6jkbhky668h6) 11

[**Estándar de Artefactos**](#_hqy22mjtk4f2) **11**

[Artefactos Word](#_fl4y19vfi39k) 11

[Artefactos Excel](#_tks53lvu8jq8) 12

[Artefactos de Código Fuente](#_j1s1yanhgrlh) 12

[Artefactos de Aplicación desplegada](#_xdavnv7kx98b) 12

[Commits](#_4hkc35ypcv7a) 12

[Tipo](#_tydvrp91uw2k) 13

[Ámbito:](#_litnz9cffvi) 13

[Descripción](#_nn86bdw4cr84) 14

[Cuerpo](#_cuw60x19ygug) 14

[Nota de Pie](#_nj66zr31s693) 14

[**Seguimiento del Proyecto**](#_tan435zb750k) **14**

# Plan de Proyecto

# Introducción

El presente documento tiene como objetivo en conjunto con el Cronograma del proyecto presentar y detallar los procesos y actividades a realizarse por el equipo para la creación de la plataforma Artemis para la asignatura de Gestión de la Configuración de Software.

El proyecto está basado en la metodologìa Rational Unified Process, la cual se seguirá para abordar todas las fases del proyecto y se seguirán los roles establecidos a esta misma.

Como parte de este modelo se producirán artefactos o entregables, estos también están detallados en el cronograma del proyecto, los cuales ayudarán a completar el producto final de software. Este documento es uno de ellos y se tomará como artefacto de la fase de Inicio.

## Propósito

El propósito del presente documento es proporcionar una guía para el equipo que ayude al desarrollo y control de la misma.

## La metodología RUP

RUP es un marco de proyectos de desarrollo de software basado en componentes en el que cada componente puede verse como un componente de software siendo el conjunto de ellos el producto final de software. Se encuentra en el conjunto de metodologías denominadas iterativas- incrementales, esto quiere decir que en cada iteración se va a ir presentando mejoras de los módulos a desarrollar. Además esta metodología está dirigida por el Modelo de Casos de Uso.

Y cuenta con cuatro fases: Iniciación, Planificación, Desarrollo y Transición, además de un conjunto de actividades de soporte.

## Roles

* **Jefe del Proyecto:** Diego Torres
* **Analista:** Daniel Cifuentes
* **Diseñador de Software:** Edward Ramos
* **Diseñador UX/UI:** Sebastian Vargas
* **DBA:** Mauricio Linares
* **Desarrolladores Front-End:** Arian Zambrano y Sebastian Vargas
* **Desarrolladores Back-End:** Diego Torres y Mauricio Linares
* **Tester:** Juan Perez

# Organización

## Fases, Actividades y tareas

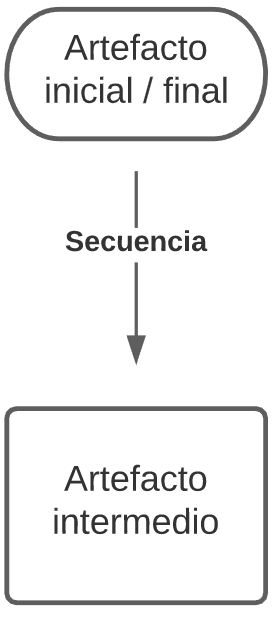
Como bien se ha dicho, la metodología RUP cuenta con 4 fases pero en este proyecto se han renombrado con el fin de ser más precisos y entendibles por el equipo. Cada fase comprenderá un conjunto de actividades y cada actividad contendrá tareas.

|  |
| --- |
| **Fase 1** |
| Actividad 1 |
| *Tarea 1* |
| *Tarea 2* |
| Actividad 2 |
| Actividad 3 |
| Hito #1: dd/mm/aaaa |

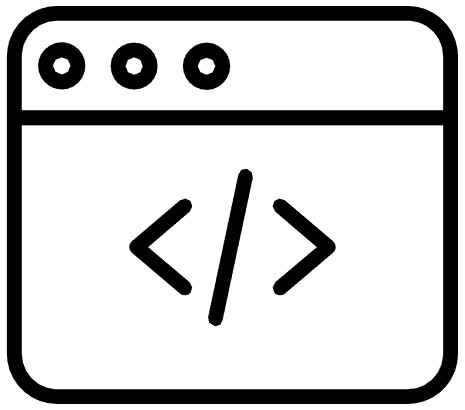
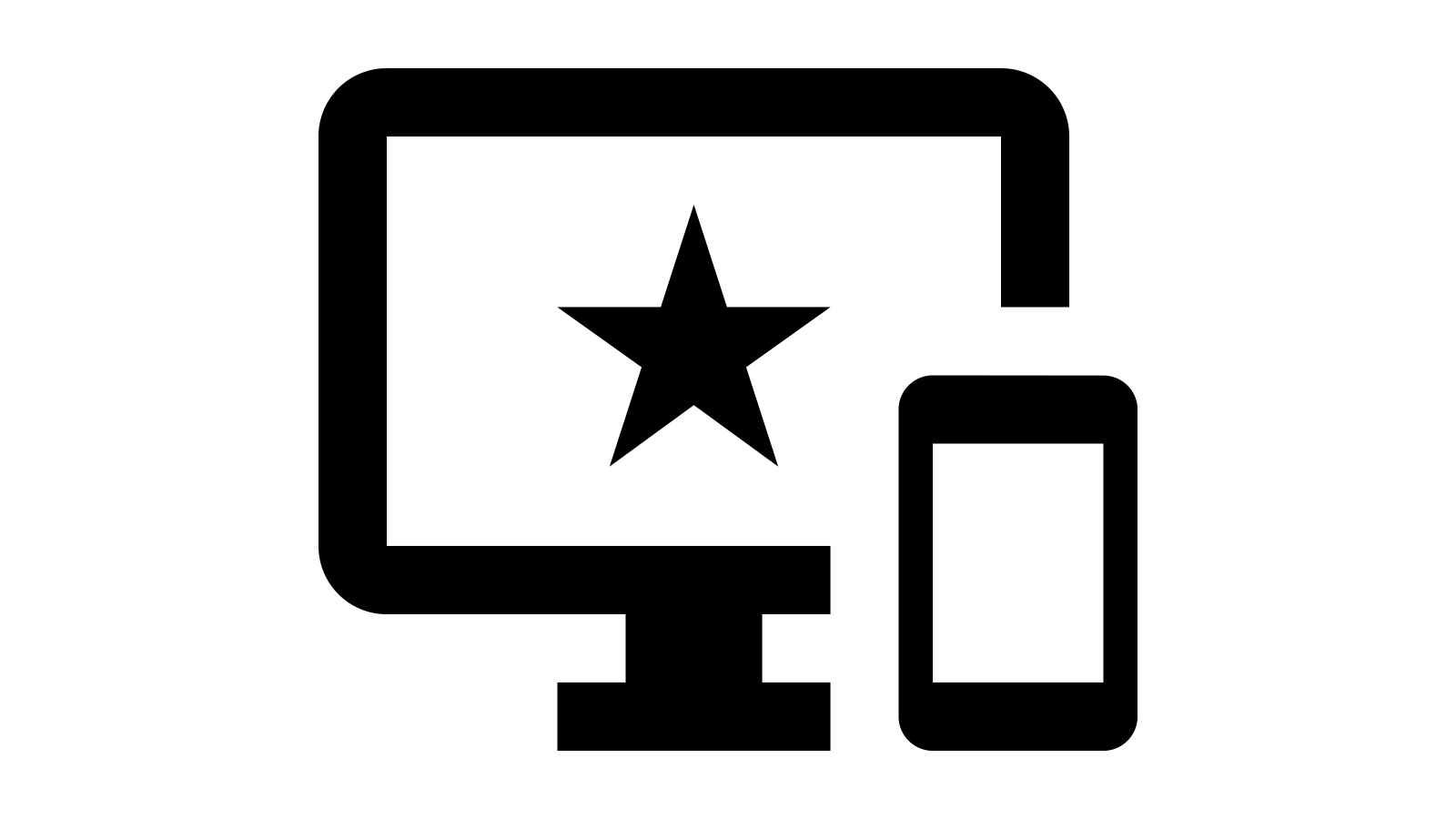
Al finalizar cada actividad se obtendrá un Artefacto (Entregable) del proyecto y al finalizar una fase se marcará un hito como terminado.

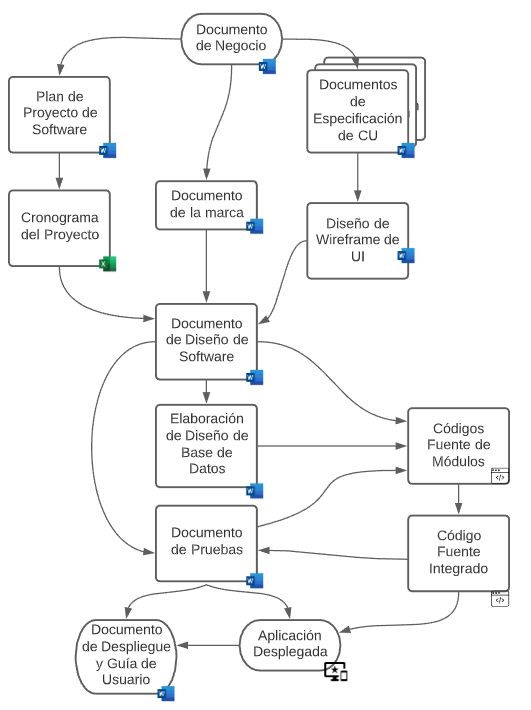
## Secuencia de Artefactos

Para tener una mejor lectura de la documentación y una guía para conocer la dependencia entre los artefactos, se elaboró el siguiente diagrama el cual representará la Secuencia de Artefactos según la siguiente nomenclatura.



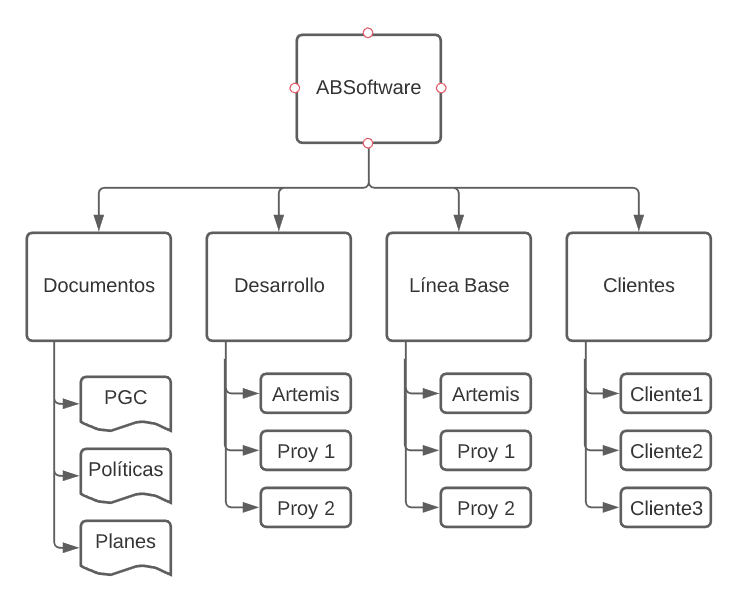
Además el tipo de entregable será especificado en la esquina inferior del artefacto.

Word (), Excel (), Código (), Ejecutable ().



## Repositorio GitHub

El repositorio de GitHub tiene por nombre ABSoftware. Este tiene la siguiente distribución:



* **Documentos:** Cuenta con los documentos pertenecientes a la empresa
* **Desarrollo:** Dentro tiene carpetas por cada proyecto en Desarrollo
* **Línea Base:** Contiene la línea base de cada proyecto
* **Clientes:** Contiene a los entregables para cada cliente de la empresa.

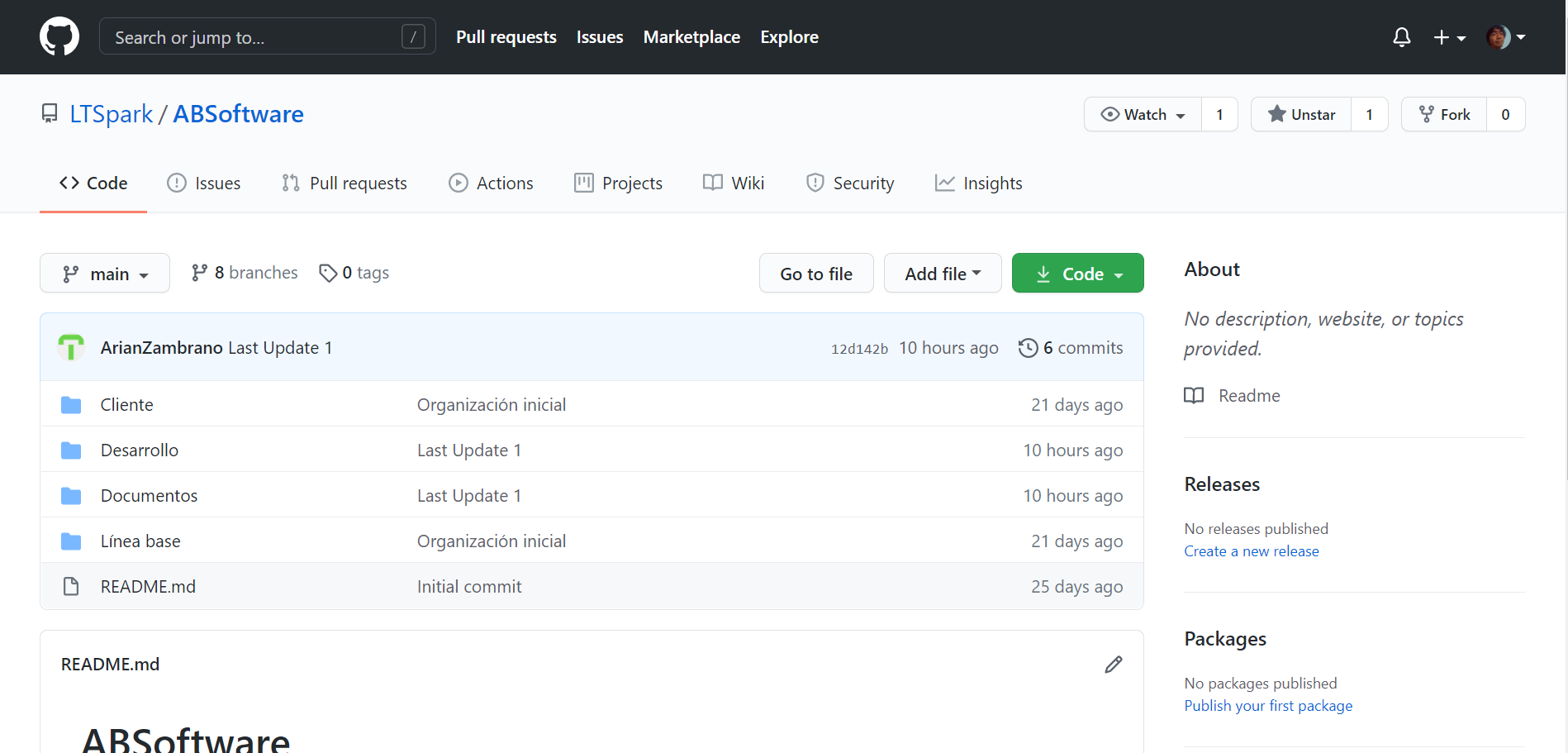


Foto captura al 03/12/20.

# Estándar de Artefactos

## Artefactos Word

Los artefactos Word tendrán una cabecera que se encontrará en cada página del artefacto. Tendrá forma de tabla y describirá los siguientes campos.

|  |  |  |
| --- | --- | --- |
| Proyecto *Nombre\_Proyecto* | Versión | Fase |
| Nombre\_Documento | Fecha dd/mm/aaaa | Nombre\_fase |

En la primera página tendrá como carátula la inclusión del nombre del Artefacto, el del Proyecto y la versión de este, los tres puntos con alineación a la derecha y tamaño de fuente 48.

En la segunda página tendrá el Historial de versiones que incluirá la fecha, versión, descripción y autor de la versión, y la sección de Otros Documentos la cual guiará al lector según la Secuencia de Artefactos.

En la tercera página se tendrá el contenido del documento en forma de índice.

Para el contenido preferiblemente usar:

|  |  |  |
| --- | --- | --- |
| **Texto** | **Tipo de Fuente** | **Tamaño de Fuente** |
| Título | Oswald | 48 |
| Título nivel 1 | Oswald | 20 |
| Título nivel 2 | Oswald | 16 |
| Título nivel 3 | Barlow | 14 |
| Título nivel 4 | Barlow | 12 |
| Texto normal | Barlow | 11 |

En el pié de página se encontrará los permisos del documento (Confidencial o Público) alineado a la izquierda y la paginación con el formato de “Página #” alineado a la derecha.

## Artefactos Excel

Estos documentos no tienen un estándar tan marcado. Lo único resaltante es que deben de contener explícitos los campos de:

* Título del documento
* Versión
* Encargado del documento
* Fase
* Zona de Otros documentos (previos y siguientes)

## Artefactos de Código Fuente

Estos artefactos contendrán solamente los siguientes campos en líneas comentadas al inicio del código:

* Versión
* Encargado del código
* Último en modificar
* Descripción de lo que hace dicho código

## Artefactos de Aplicación desplegada

Sin ningún estándar en específico.

## Commits

Se usará una adaptación del estándar Conventional Commits para los commits subidos al Repositorio GitHub.  
 (Más info: https://www.conventionalcommits.org/en/v1.0.0/)

Estos tienen la siguiente estructura:

<tipo>(ámbito opcional): <descripción>

-espacio en blanco-

[cuerpo opcional]

-espacio en blanco-

[nota de pie opcional]

### Tipo

Los principales tipos de commits seràn los siguientes

|  |  |
| --- | --- |
| feat | Una nueva característica agregada. Por ejemplo, un botón añadido, login añadido, etc.. |
| fix | Una corrección de una característica. Por ejemplo, corrección del buscador, corrección en la BD, etc. |
| perf | Cambia la experiencia de usuario. Por ejemplo, cambio de lugar de un botón, cambio de procedimientos para realizar cierta acción, etc. |
| style | Cambios en el estilo del código pero que no agrega ninguna funcionalidad. Por ejemplo, indentación, comentarios, etc. |
| refactor | Cambios en la organización del código. Por ejemplo, modularización de ciertas funciones. |
| doc | Commits donde solo se modifica la documentación del proyecto. |
| chore | Cambios en la metadata, releases y otros cambios grandes del proyecto. Por ejemplo, cambiar una librería, integrar varios módulos. |
| org | Cambios en la organización del repositorio |

En caso de hacer varios tipos de cambios en el commit, colocar el de mayor relevancia y en el cuerpo del commit colocar lo demás.

### Ámbito:

Campo opcional en paréntesis, sirve para precisar qué aspecto ha sido modificado en el commit. De preferencia mantener una cantidad baja de ámbitos para evitar ambigüedades.

* Cuentas: login, logout, manejo de cuentas, etc.
* Interacción: artista-artista (seguir, dejar de seguir, colaborar, ver perfil), artista-obra (buscar, comentar, likear), etc.
* Obras: subir, eliminar, editar, historial de versiones, colaboración, etc.
* Otros: estadísticas y demás.

### Descripción

Breve descripción del commit, es obligatorio.

### Cuerpo

Descripción más extensa del commit, es opcional.

### Nota de Pie

Notas extra, son opcionales.

En caso de realizar un cambio que rompa la compatibilidad de alguna API, colocar “BREAKING CHANGE” en la nota de pie además de una breve descripción de este evento.

# Seguimiento del Proyecto

En esta sección se anotarán los eventos ocasionados en el desarrollo del proyecto:

* 11/11/2020: Por temas de tiempo se decidió especificar todos los casos de uso, pero especificar solamente algunos requisitos funcionales, estos serían los fundamentales a implementar.