

Gestor de Pedidos (GP)

Trezza, Nicolás – Lucero, José – Olmos, Martin – Benítez Haugg, Griselda

Licenciatura en Sistemas – Departamento de Desarrollo Productivo y Tecnológico Universidad Nacional de Lanús
ntrezzan@gmail.com – jose.maximiliano.lucero@gmail.com – olmos.martin.1992@gmail.com –
griselda_gsis@hotmail.com

Extracto—En este documento constarán las actividades realizadas por el equipo de desarrollo para la entrega pautada de un primer hito. El mismo constara del análisis preliminar(Abstracto)

I. INTRODUCCION

La incorporación de éste proyecto surge a partir de un pedido de Campus Virtual de la Universidad Nacional de Lanús (UNLa), que tendran una pagina web de gestor de pedidos para la Universidad de Tecnología Nacional (UTN). Para que pueda matricular, crear aulas, brindar capacitaciones, solicitar formularios y actualizar virtualizaciones, y por último habra una parte de estadísticas que podran ver ciertos usuarios.

Tendremos un sitio web, habra cinco áreas, la primera area será crear usuario para que pueda matricularse, es un formulario de gestión de usuarios en las aulas virtuales. La segunda area será aulas, esto va a contemplar los pasos administrativos para la gestión de un espacio curricular en una aula virtual, es decir, crear un aula, reutilizar o eliminar. La tercera area será capacitación que brindara capacitación constante a la comunidad educativa, como tutoría o microtalleres. La cuarta area sera virtualización es un formularios para la solicitud y actualización de virtualización de unidades curriculares. Y por último la quinta area sera estadísticas que solamente podran ver lo que tienen permiso de extraer información del sistemas. Todas estas información se enviara a la base de datos y dependiendo el area, un correo electronico formato pdf y una copia al usuario.

Este sistema propone la solución a todo lo mencionado en forma automatica.

II. PROCESO DE SELECCIÓN DE UN MCVS

A. Identificar los posibles MCVS:

Es importante conocer las principales características de cada ciclo de vida para poder seleccionar la correcta para nuestro diseño.

Las principales diferencias entre estos modelos de ciclo de vida es:

- El alcance del ciclo dependiendo de hasta donde llegue el proyecto correspondiente, hasta su retirada del mercado.

- Las características de las fases en que dividen el ciclo de vida, esto puede depender del cada proyecto a que se refiere.
- La estructura de la sucesión de las fases que puede ser lineal, con prototipado o en espiral por ejemplo:

Ciclo de vida en Cascada

Este ciclo de vida lo utilizan mucho, por ser el más sencillo. La actividad de global de proyecto consiste en descomponer en fases que muestra en manera lineal, es decir, cada una de esas se realiza una sola vez tras la anterior y antes de la siguiente. Con el ciclo lineal es facil dividir tareas entre equipos y prevenir los tiempos obviamente sumando los de cada fases.

Ciclo de vida en Prototipado

Si no se conoce exactamente cómo de desarrollar un determinado producto o cuáles son las especificaciones iniciales para hacer un prototipo o sea, un producto parcial (no hace falta que contenga funciones suficientemente probadas) y provisional (no se va a fabricar realmente para clientes, por lo que tiene menos restricciones de costo y/o prestaciones). Este tipo de procedimiento es muy utilizado en desarrollo avanzado.

La diferencia entre el ciclo de vida cascada, puede decirse que el ciclo de vida con prototipado repite fases de definición, diseño y construcción dos veces: para el prototipo y para el producto real.

Ciclo de vida en espiral

El esquema de este ciclo de vida muestra para estos casos por un bucle en espiral, donde los cuadrantes son fases de especificación, diseño, realización y evaluación. En cada vuelta el producto llega aproximarse al final deseado hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

B. Seleccionar un modelo para el proyecto

A veces ocurre en desarrollos de productos con innovaciones importantes, o cuando se prevé la utilización de tecnologías nuevas o pocos probadas.

Si no se conoce exactamente cómo de desarrollar un determinado producto o cuáles son las especificaciones iniciales para hacer un prototipo o sea, un producto parcial (no hace falta que contenga funciones suficientemente probadas) y

provisional (no se va a fabricar realmente para clientes, por lo que tiene menos restricciones de costo y/o prestaciones). Este tipo de procedimiento es muy utilizado en desarrollo avanzado.

La experiencia del desarrollo y su evaluación deben permitir la definición de las especificaciones mas completas y seguras para producto definitivo.

La diferencia entre el ciclo de vida cascada, puede decirse que el ciclo de vida con prototipado repite fases de definición, diseño y construcción dos veces: para el prototipo y para el producto real.

Debido a que no se conoce exactamente cómo desarrollar o cuáles son las especificaciones de forma precisa que suele recurrir a definir especificaciones iniciales para hacer un prototipado, es decir, un producto o provisional. Se elige seguir este modelo de ciclo de vida prototipado. Puede comprenderse las fases de este ciclo de vida en la imagen 1.

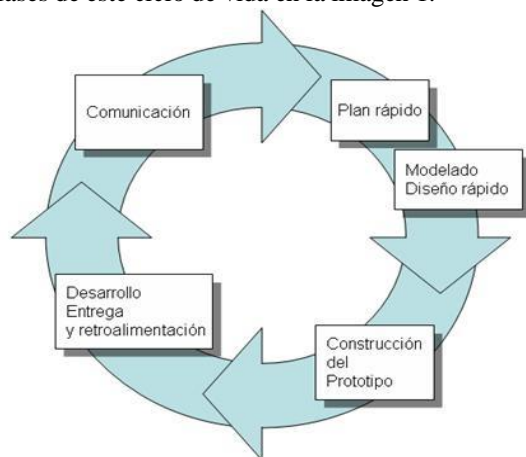


Imagen 1: Ciclo de vida prototipado evolutivo.

III. PROCESO DE INICIACIÓN, PLANIFICACIÓN Y ESTIMACIÓN DEL PROYECTO

A. Establecer la matriz de actividades para el MCVS

Nuestro diseño constara de 3 maqueta y un prototipos evolutivos, es decir, una interacción en cada una de las fases previamente detalladas, puede comprenderse esto en la imagen 2, mapa de actividades y tambien comprender los tiempos de ejecución en la imagen 3, Gantt.

B. Asignar recursos del proyecto

B.1. Recursos humanos:

El diseño de la pagina web y sus maquetas estará a cargo principalmente de Trezza Nicolás, Olmos Martin. y su desarrollo de Trezza Nicolás y Olmos Martin.

El diseño e implementacion del servidor y los controladores estarán desarrollados por Olmos Martín y Benitez Griselda; el diseño de la base de datos estará a cargo de Olmos Martin y Trezza Nicolás mientras que su desarrollo estará a cargo de Lucero José y Benitez Griselda.

Esta distribución puede cambiar acorde a los tiempos asignando más recursos, para evitar atrasos muy significativos, la asignación de tareas podrá observarse tambien el Gantt.

B.2. Recursos tecnológicos

La pagina web estará desarrollada con materialize y los controladores con python 2.7.

El servidor tiene una arquitectura CGI, la base de datos será en MySQL diseñada en workbench.

C. Definir el entorno del proyecto

El proyecto está orientado a facilitar el trabajo a la Universidad Tecnología Nacional (UTN), que tiene la necesidad de solicitar al campus de la Universidad Nacional de Lanús (UNLa), para poder crear el usuario, crear un aula o poder reutilizar y eliminar, brindar capacitaciones, actualizar virtualizaciones y por último parte de estadísticas que ciertos usuarios lo podran ver.

La interface podrá ser utilizada por los profesores de la Universidad Tecnología Nacional (UTN). No necesita estar logueado pero pedirá los datos necesarios, excepto en la parte de estadísticas pedirá ingresar un login.

D. Planificar la gestión del proyecto

El proyecto fue planificado en primera instancia fecha”, la distribución de las tareas en cada unas de las etapas esta descrita en la imagen 3.

IV. PROCESO DE SEGUIMIENTO Y CONTROL DEL PROYECTO

A. Análisis de riesgos

Por medio del análisis del entorno pudimos identificar

<u>Descripción</u>	<u>Impacto</u>	<u>Probabilidad</u>	<u>Acción a Seguir</u>
- El usuario no sabe sacar provecho máximo de la web	Muy alto, la página podría perder prestigio	Alta	Se detallará paso por paso cada una de las virtudes del programa. Se dará soporte audiovisual a la plataforma
- Error al cargar mal el archivo del Excel.	Los datos mal cargados	Media	Incorporar un manual de instrucciones para el usuario, además el administrador podrá modificar los datos de ser necesario
- No hay conexión de internet.	La página web no cumplirá las solicitudes si no posee conexión de internet.	Baja	Sin internet no se puede conectar a la página

Imagen 4: Gestión de Riesgos.

B. Realizar la planificación de contingencias

- El plan de respaldo. Para evitar la pérdida de datos o los cambios abruptos en los ideales del cliente, se

respaldan todos los archivos y documentos del proyecto en la Github situada en tres branch:

. master, maqueta, desarrollo.

<https://github.com/NicoTrezza/GestorDePedidos>

En caso de que alguno de ellos posea un problema serio por ellos tenemos varias actividades para complementarlos con preventivas antes de que se materialice una amenaza, es decir, evitar dicha materialización.

El plan de emergencia. Si algunos de los integrantes no pueden cumplimentar alguna de sus tareas, estamos preparados para esos casos a quien debe suplantar a cada persona en caso que la misma no pueda cumplir por si sola las tareas asignadas.

C. Gestionar el proyecto

Para una mejor gestión y administración, se empleó un diagrama PERT desde el momento de la creación de la primera maqueta, en el mismo se aprecian las fechas teóricas pactadas y las fechas reales dadas en el proyecto, podemos encontrarlo en la Imagen 7.

D. Implementar el sistema de informes de problemas

- 1- El primer problema critico fue el poco conocimiento del cliente real al que está destinado nuestro proyecto, pero al mismo fue solucionado mediante contacto por las entrevistas que fuimos haciendo.
- 2- Luego con las distintas versiones de programas para poder crear la página web, pero optamos en ponernos de acuerdo debido a los que se está usando ahora en la actualidad, pero no la versión nueva.

E- Archivar registros

Todos los registros de avances, o los que describen en la profundidad lo ilustrado por este documento está en el segmento de nuestro directorio online,

<https://github.com/NicoTrezza/GestorDePedidos>

V. PROCESO DE GESTIÓN DE CALIDAD DE SOFTWARE

En este segmento nos centraremos en la calidad del producto, servicio o la satisfacción del cliente. Se utiliza al aseguramiento de la calidad y el control de los procesos para obtener una calidad más consistente.

A. Planificar la garantía de calidad del software.

Los modelos de calidad del software vienen a ayudar en la puesta en práctica del concepto general de calidad que vimos en el apartado anterior, ofreciendo una definición más operacional. Unos de los modelos de calidad más antiguos y extendidos es el McCall [McCall, 1977], y de él han derivado otros modelos, como el de Boehm [Boehm 78] o el SQM [Murine, 1988]. En los modelos de calidad, la calidad se define de forma jerárquica. Es un concepto

que se deriva de un conjunto de sub-conceptos, cada uno los cuales se van a evaluar a través de un conjunto de indicadores o métricas. Para el estudio de calidad de nuestro proyecto nos basaremos en el diagrama descripto en la imagen 8.

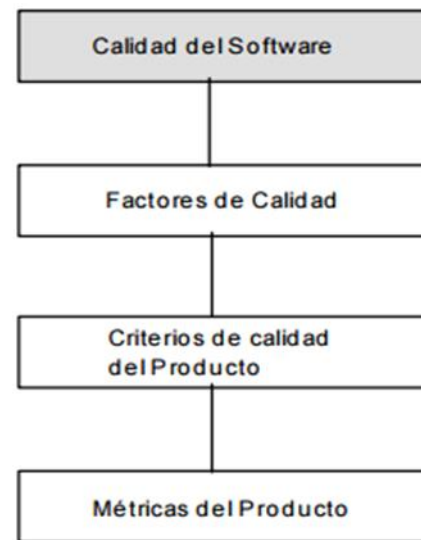


Imagen 8. Estructura de 4 niveles de la calidad de SW

B. Identificar necesidades de mejora de la calidad

Con los tres maquetado se identificaron 2 necesidades de mejoras de calidad para los modelos funcionales:

- Permitir a los usuarios puedan pasar de una página a la otra sin volver a la principal.
- Permitir que se vea bien el formato desde un celular.

VI. PROCESO DE EXPLORACIÓN DE CONCEPTOS

A. Identificar las ideas o necesidades

Necesidades a satisfacer:

- Al ingresar a la página campus virtual UNLa especialmente para la UTN habrá un botón que podrá ingresar para solicitar al campus.
- Crear solicitud de usuario.
- Crear el aula.
 - . Crear un aula
 - . Reutilizar aula.
 - . Eliminar aula.
- Solicitar tutorías y capacitaciones.
- Solicitud de virtualización material.

B. Formular las soluciones potenciales

Para combatir los riesgos descriptos en la imagen 4, proponemos las siguientes soluciones.

- En caso de ingresar habrá un botón especial que resalte que es para especialmente a los usuarios de UTN.
- Al crear la solicitud de usuario, aula u otros se le enviara una copia por correo electrónico al usuario y el

original al que le corresponda a la dirección de ese pedido.

C. Dirigir los estudios de viabilidad

Los riesgos descritos en la imagen 4 y las soluciones planteadas en el apartado anterior se considera viables para el proyecto, por los siguientes ítems:

- Viabilidad operacional: Los lenguajes de programación a implementar, el tiempo disponible y las condiciones de producción se consideran totalmente viable por los integrantes del equipo, por conocimiento previo de todas las partes.
- Viabilidad del mercado: Este proyecto será diseñado previo aun pedido por lo que posee un mercado seguro para esperar el desarrollo.
- Viabilidad conceptual: Aunque haya riesgos, pero es una página web que se está desarrollando el entendimiento del problema y la factibilidad de la solución ha sido comprendidas.

D. Refinar y finalizar la idea o necesidad.

Uno de los principales problemas sobre el entendimiento del problema a resolver, fue la comprensión del cliente como deseaba la plataforma y que algunas funciones obligatorias tendrían que estar. El problema fue resuelto, pero aún hay dudas que más adelante podremos resolver mediante las respuestas del cliente.

Link1:

<https://github.com/NicoTrezza/GestorDePedidos/blob/master/Documentaci%C3%B3n/1%20Encuesta%20Gestor%20de%20Pedidos%20.pdf>

VII. PROCESO DE ASIGNACIÓN DEL SISTEMA

Este proceso se realiza cuando el sistema requiere tanto del desarrollo de hardware como de software, o cuando no se puede asegurar que solo se necesita desarrollo de software

El informe de la necesidad se analiza para identificar las entradas, el procesamiento que se aplica a la entrada, las salidas requeridas y las funciones del sistema total, que permiten desarrollar la arquitectura del sistema e identificar las funciones del hardware, del software y de los interfaces.

Este proceso culmina con la especificación de requisitos del software e, la especificación de requisitos del hardware y la especificación del interfaz del sistema.

A. Analizar las funciones del sistema

En este analizaremos las funciones que posee la segunda iteración del maquetado. Haremos algunas menciones a la segunda maqueta, que solo posee funciones. Este recorrido mostramos las funciones operativo que tiene la página web.

La página web Gestor de pedidos, consta con una primera pantalla(Inicio), donde uno puede elegir la opción que desea el usuario. Si la persona desea hacer la solicitud de

usuario se encontrará con la imagen 6.1. se le pedirá todos los datos personales para ser un usuario.

Luego si desea crear un aula tiene que ser un usuario para poder crear, reutilizar o eliminar el aula con estos datos especifica en las siguientes imágenes: (imagen 6.2), (imagen 6.2.1) y por último la (imagen 6.2.2).

Lo mismo sucede (imagen 6.3.1) y (imagen 6.3.2) que son para capacitaciones o tutorías y microtalleres.

Maqueta 1

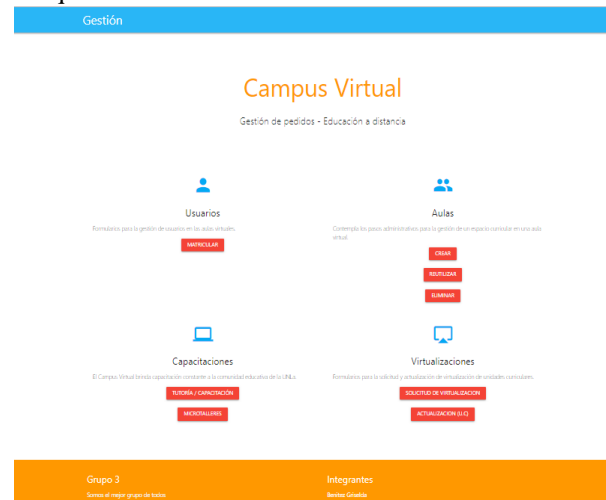


Imagen 5. Gestor de Pedidos versión 1

Maqueta 2

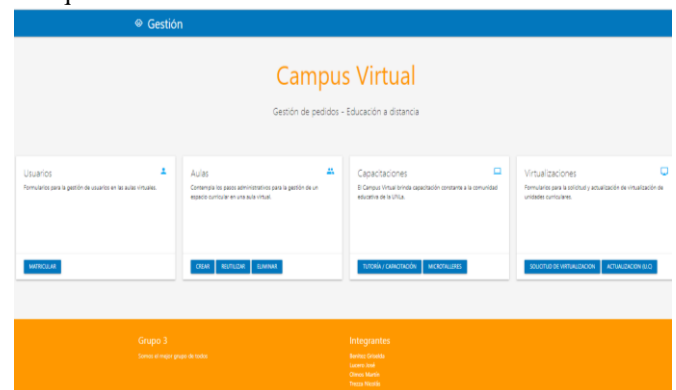


Imagen 6. Gestor de Pedidos versión 2



Imagen 6.1. Gestor de Pedidos versión 2. Matricular

Imagen 6.2. Gestor de Pedidos versión 2. Crear aula

Imagen 6.2.1 Gestor de Pedidos versión 2. Reutilizar aula

Imagen 6.2.2 Gestor de Pedidos versión 2. Eliminar aula

Imagen 6.3.1 Gestor de Pedidos versión 2. Tutoría y capacitación

Imagen 6.3.2 Gestor de Pedidos versión 2. Microtalleres

VIII. PROCESO DE DISEÑO

Se definen los pasos realizados para la creación de las 3 maquetas o mejor dicho las iteraciones. Donde muchas menciones fueron hechas en el punto VII.

A. Realizar el diseño preliminar

Para el diseño preliminar de la página web, se utilizó herramientas Material Design. En la primera maqueta se construyó una página que simplemente casi similar a la página de gestor de pedidos de la UNLa, pero luego el segundo maquetado se mejoró la plataforma (color, letra) por el pedido del cliente.

Más que nada que sea simple su utilización, solo los campos vacíos que deben ser rellenados, o incluso algunos con opciones cerradas que facilitan aún más al usuario el uso. La aplicación está disponible en:

<https://github.com/NicoTrezza/GestorDePedidos/tree/maqueta>

IX. PROCESO DE IMPLEMENTACIÓN E INTEGRACIÓN

Este proceso transforma la representación del diseño detallado de un producto de software a una realización en un lenguaje de programación.

A. Crear la documentación de operación

Todo el software, se lo puede encontrar documentado y versionado en:

<https://github.com/NicoTrezza/GestorDePedidos/tree/master>

	An.Pr.	Prototipo Maqueta (1 a 3 iteraciones)			Prototipo Evolutivo (2 a 5 iteraciones)				Instalación y Operación		
	AP	ERI	DEI	VIU	ERS	DAS	C	PP	IF	OM	R
Proceso de Selección de un MCVS											
- Identificar los posibles MCVS	X										
- Seleccionar un modelo para el proyecto.	X										
Proceso de Iniciación, Planificación y											
Estimación del Proyecto											
- Establecer la matriz de actividades para el MCVS	X										
- Asignar los recursos del proyecto.	X	X	X	X	X	X	X	X			
- Definir el entorno del proyecto.	X										
- Planificar la gestión del proyecto.	X										
Proceso de Seguimiento y Control del Proyecto											
- Analizar riesgos.	X	X	X	X	X	X	X				
- Realizar la planificación de contingencias.		X	X	X	X	X	X	X			
- Gestionar el proyecto.		X	X	X	X	X	X	X			
- Implementar el sistema de informes de problemas.		X	X	X	X	X	X	X			
- Archivar registros.		X	X	X	X	X	X	X			
Proceso de Gestión de Calidad del Software											
- Planificar la garantía de calidad del software.		X	X		X	X	X				
- Desarrollar métricas de calidad.		X	X		X	X	X				
- Gestionar la calidad del software.		X	X	X	X	X	X	X			
- Identificar necesidades de mejora de la calidad.		X	X	X	X	X	X	X			
Proceso de Exploración de Conceptos											
- Identificar las ideas o necesidades.	X	X			X						
- Formular las soluciones potenciales.	X	X			X						
- Dirigir los estudios de viabilidad.	X	X			X						
- Planificar la transición del sistema (si se aplica).	X	X			X						
- Refinar y Finalizar la idea o necesidad.	X	X			X						
Proceso de Asignación del Sistema											
- Analizar las funciones del sistema.			X			X					
- Desarrollar la arquitectura del sistema.						X					
- Descomponer los requisitos del sistema.						X					
Proceso de Análisis de Requisitos											
- Definir y Desarrollar los requisitos del software.					X						
- Definir los requisitos de interfaz.		X									
- Priorizar e Integrar los requisitos del software.					X						
Proceso de Diseño											
- Realizar el diseño preliminar.			X			X					
- Analizar el flujo de información.						X					
- Diseñar la base de datos (si se aplica).						X					
- Diseñar las interfaces.						X					
- Seleccionar o Desarrollar algoritmos (si se aplica).						X					
- Realizar el diseño detallado.						X					
Proceso de Implementación e Integración											
- Crear los datos de prueba.						X	X				
- Crear el código fuente.							X				
- Generar el código objeto.							X				
- Crear la documentación de operación.			X			X	X				
- Planificar la integración.						X	X				
- Realizar la integración.							X				
Proceso de Instalación y Aceptación											
- Planificar la instalación.									X		
- Distribuir el software.									X		
- Instalar el software.									X		
- Cargar la base de datos (si se aplica).									X		
- Aceptar el software en el entorno de operación.								X	X		
- Realizar las actualizaciones.									X		
Proceso de Operación y Soporte											
- Operar el sistema.										X	
- Proveer de asistencia técnica y consultas.										X	
- Mantener el histórico de peticiones de soporte.										X	
Proceso de Mantenimiento											
- Realizar el mantenimiento correctivo.										X	
- Reaplicar el ciclo de vida del software.										X	
Proceso de Retiro											
- Notificar al usuario.											X
- Conducir operaciones en paralelo (si se aplica).											X
- Retirar el sistema.											X

Imagen 2: Mapa de actividades

Nombre de tarea	Duración	Comienzo	Fin	
Identificar los posibles MCVS	3 días	mar 4/9/18	jue 6/9/18	
Seleccionar un modelo para el proyecto.	1 día	mié 5/9/18	mié 5/9/18	
Establecer la matriz de actividades para el MCVS	10 días	vie 7/9/18	jue 20/9/18	
Asignar los recursos del proyecto.	200 días	lun 10/9/18	vie 14/6/19	
Definir el entorno del proyecto.	8 días	lun 10/9/18	mié 19/9/18	
Planificar la gestión del proyecto.	5 días	lun 10/9/18	vie 14/9/18	
Analizar riesgos.	170 días	mié 12/9/18	mar 7/5/19	
Realizar la planificación de contingencias.	170 días	lun 1/10/18	vie 24/5/19	
Gestionar el proyecto.	170 días	lun 1/10/18	vie 24/5/19	
Implementar el sistema de informes de problemas.	170 días	lun 1/10/18	vie 24/5/19	
Archivar registros.	170 días	lun 1/10/18	vie 24/5/19	
Planificar la garantía de calidad del software.	160 días	mar 9/10/18	lun 20/5/19	
Desarrollar métricas de calidad.	160 días	mar 9/10/18	lun 20/5/19	
Gestionar la calidad del software.	160 días	mar 9/10/18	lun 20/5/19	
Identificar necesidades de mejora de la calidad.	160 días	mar 9/10/18	lun 20/5/19	
Identificar las ideas o necesidades.	80 días	mar 11/9/18	lun 31/12/18	
Formular las soluciones potenciales.	80 días	mar 11/9/18	lun 31/12/18	
Dirigir los estudios de viabilidad.	80 días	lun 17/9/18	vie 4/1/19	
Planificar la transición del sistema (si se aplica).	80 días	lun 17/9/18	vie 4/1/19	
Refinar y Finalizar la idea o necesidad.	80 días	vie 14/9/18	jue 3/1/19	
Analizar las funciones del sistema.	150 días	mar 18/9/18	lun 15/4/19	
Desarrollar la arquitectura del sistema.	120 días	vie 28/9/18	jue 14/3/19	
Descomponer los requisitos del sistema.	120 días	vie 28/9/18	jue 14/3/19	
Definir y Desarrollar los requisitos del software.	100 días	vie 28/9/18	jue 14/2/19	
Definir los requisitos de interfaz.	20 días	mar 18/9/18	lun 15/10/18	
Priorizar e Integrar los requisitos del software.	100 días	vie 28/9/18	jue 14/2/19	
Realizar el diseño preliminar.	150 días	mar 18/9/18	lun 15/4/19	
Analizar el flujo de información.	100 días	lun 19/11/18	vie 5/4/19	
Diseñar la base de datos (si se aplica).	100 días	lun 19/11/18	vie 5/4/19	
Diseñar las interfaces.	100 días	lun 19/11/18	vie 5/4/19	
Seleccionar o Desarrollar algoritmos (si se aplica).	100 días	lun 19/11/18	vie 5/4/19	
Realizar el diseño detallado.	100 días	lun 19/11/18	vie 5/4/19	
Crear los datos de prueba.	120 días	lun 19/11/18	vie 3/5/19	
Crear el código fuente.	120 días	lun 19/11/18	vie 3/5/19	
Generar el código objeto.	120 días	lun 19/11/18	vie 3/5/19	
Crear la documentación de operación.	120 días	lun 19/11/18	vie 3/5/19	
Planificar la integración.	120 días	lun 19/11/18	vie 3/5/19	
Realizar la integración.	120 días	lun 19/11/18	vie 3/5/19	
Planificar la instalación.	90 días	mar 23/4/19	lun 26/8/19	
Distribuir el software.	90 días	mar 23/4/19	lun 26/8/19	
Instalar el software.	90 días	mar 23/4/19	lun 26/8/19	
Cargar la base de datos (si se aplica).	90 días	mar 23/4/19	lun 26/8/19	
Aceptar el software en el entorno de operación.	90 días	mar 23/4/19	lun 26/8/19	
Realizar las actualizaciones.	90 días	mar 23/4/19	lun 26/8/19	
Operar el sistema.	30 días	jue 18/4/19	mié 29/5/19	
Proveer de asistencia técnica y consultas.	30 días	jue 18/4/19	mié 29/5/19	
Mantener el histórico de peticiones de soporte.	30 días	jue 18/4/19	mié 29/5/19	
Realizar el mantenimiento correctivo.	30 días	jue 18/4/19	mié 29/5/19	
Reaplicar el ciclo de vida del software.	30 días	jue 18/4/19	mié 29/5/19	
Notificar al usuario.	30 días	mié 29/5/19	mar 9/7/19	
Conducir operaciones en paralelo (si se aplica).	30 días	mié 29/5/19	mar 9/7/19	
Retirar el sistema.	30 días	mié 29/5/19	mar 9/7/19	

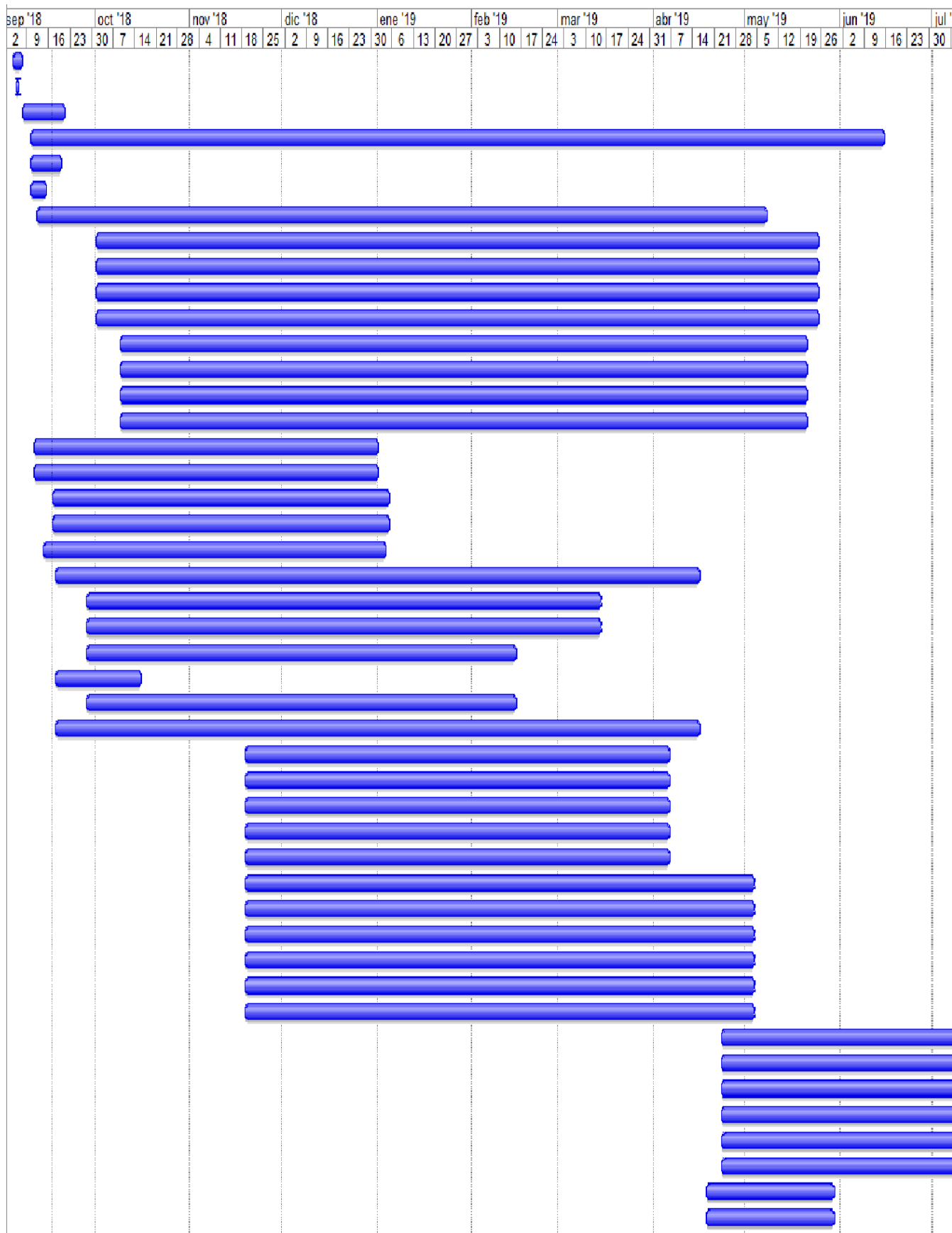


Imagen 3: Gantt