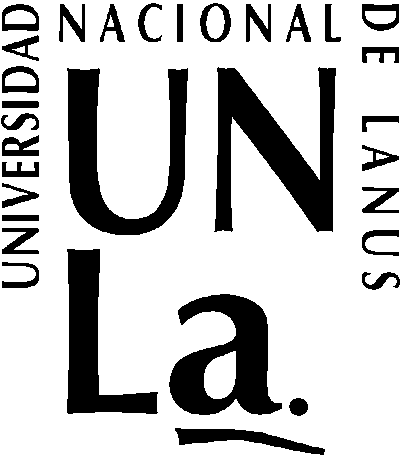
****

**Registración Espacial de Objetos en el Terreno (REOT)**

**Buczacka, Walter**

**González, Gustavo**

**Pérez, Nicolás**

**Cátedra Proyecto de Software**

**Licenciatura en Sistemas**

**2016**

*Registración Espacial de Objetos*

*en el Terreno* (REOT)

Pérez, Nicolás –Buczacka, Walter – González, Gustavo

Licenciatura en Sistemas – Departamento de Desarrollo Productivo y Tecnológico Universidad Nacional de Lanús

nperez\_dcao\_smn@outlook.com – gonzalezgustavojavier@gmail.com - buczacka.walter@gmail.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)*

# **Introducción**

La incorporación de éste proyecto surge a partir de un pedido de la Licenciatura en Gestión Ambiental Urbana. Para incorporar dos mecanismos de relevamiento de árboles en una determinada ciudad. En primera instancia el software será testeado dentro del establecimiento de la UNLa, dicho software constará de dos herramientas, la primera que consiste en la geolocalización de fotografías, y la segunda de la descripción de los árboles fotografiados. A priori no son necesarios el uso simultáneo de dichos segmentos del programa, pero al finalizar el desarrollo ambos funcionarán en simultáneo para que el cliente use los datos que considere pertinente.

La aplicación estará diseñada en 3 capas, en la primer etapa tendremos una aplicación móvil, que sacará una captura de un árbol, dicha captura estará ligada a un par ordenado(Coordenadas), y una descripción, el dispositivo móvil (si el usuario lo desea), enviará toda esta información al servidor alojado en la Universidad; dicho servidor en la segunda capa, direccionará la información a la cpu individual del cliente y guardará toda ésta información en una base de datos; ya en la 3er capa, la cpu del cliente recibirá los datos, con los que podrá generar un excel el cual será exportado a un mapa compuesto SIG, donde cada capa del compuesto reflejan datos recibidos.

Este sistema propone la solución a todo lo mencionado de forma automática.

1. **PROCESO DE SELECCIÓN DE UN MCVS**
   1. **Identificar los posibles MCVS:**

Dedicaremos este primer segmento a la comprensión y entendimiento de por que es necesario construir un ciclo de vida pertinente desde el puntapié inicial de cualquier proyecto de software.

Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al **conjunto de las fases empleadas se le denomina *“ciclo de vida”.***

Sin embargo, la forma de agrupar las actividades, los objetivos de cada fase, los tipos de productos intermedios que se generan, etc. pueden ser muy diferentes dependiendo del tipo de producto o proceso a generar y de las tecnologías empleadas.

Es fundamental conocer las principales diferencias de cada ciclo de vida para poder hacer la selección que mas se adecue a nuestro diseño.

Las principales diferencias entre distintos modelos de ciclo de vida están en:

* El **alcance** del ciclo dependiendo de hasta dónde llegue el proyecto correspondiente. Un proyecto puede comprender un simple estudio de viabilidad del desarrollo de un producto, o su desarrollo completo o, llevando la cosa al extremo, toda la historia del producto con su desarrollo, fabricación, y modificaciones posteriores hasta su retirada del mercado.
* Las **características** (contenidos) de las fases en que dividen el ciclo. Esto puede depender del propiotema al que se refiere el proyecto (no son lo mismo las tareas que deben realizarse para proyectar un avión que un puente), o de la organización (interés de reflejar en la división en fases aspectos de la división interna o externa del trabajo).
* La **estructura** de la sucesión de las fases que puede ser lineal, con prototipado, o en espiral. Veámoslo con más detalle:

Ciclo de vida en cascada

Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una **se realiza una sola vez**, cada una se realiza **tras la anterior y antes que la siguiente**. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase).

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien de antemano lo que va a ocurrir en cada fase antes de empezarla.

Ciclo de vida con prototipado

A menudo ocurre en desarrollos de productos con innovaciones importantes, o cuando se prevé la utilización de tecnologías nuevas o poco probadas, que las incertidumbres sobre los resultados realmente alcanzables, o las ignorancias sobre el comportamiento de las tecnologías, impiden iniciar un proyecto lineal con especificaciones cerradas.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa, suele recurrirse a definir especificaciones iniciales para hacer un **prototipo**, o sea, un producto parcial (no hace falta que contenga funciones que se consideren triviales o 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 del prototipo y su evaluación deben permitir la **definición de las especificaciones más completas** y seguras para el productodefinitivo.  
   
A diferencia del modelo lineal, puede decirse que el ciclo de vida con prototipado repite las fases de definición, diseño y construcción dos veces: para el prototipo y para el producto real.

|  |  |
| --- | --- |
|  |  |

Ciclo de vida en espiral

El ciclo de vida en espiral puede considerarse como una generalización del anterior para los casos en que **no basta con una sola evaluación de un prototipo** para asegurar la desaparición de incertidumbres y/o ignorancias. El propio producto a lo largo de su desarrollo puede así considerarse como una sucesión de prototipos que progresan hasta llegar a alcanzar el estado deseado. En cada ciclo (espirales) las especificaciones del producto se van resolviendo paulatinamente.

A menudo la **fuente de incertidumbres es el propio cliente**, que aunque sepa en términos generales lo que quiere, no es capaz de definirlo en todos sus aspectos sin ver como unos influyen en otros. En estos casos la evaluación de los resultados por el cliente no puede esperar a la entrega final y puede ser necesaria repetidas veces.

El esquema del ciclo de vida para estos casos puede representarse por un bucle en espiral, donde los cuadrantes son, habitualmente, fases de**especificación, diseño, realización y evaluación**  (o conceptos y términos análogos).

En cada vuelta el producto gana en “*madurez*” (aproximación al final deseado) hasta que en una vuelta la evaluación lo apruebe y el bucle pueda abandonarse.

|  |  |
| --- | --- |
|  |  |

* 1. **Seleccionar un modelo para el proyecto:**

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

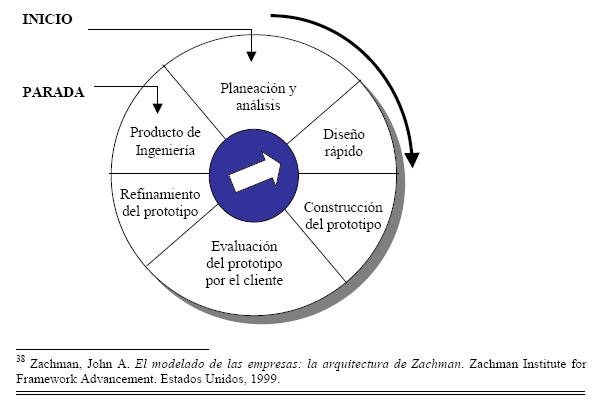


Imagen 1: Ciclo de vida prototipado evolutivo

1. **PROCESO DE INICIACIÓN, PLANIFICACIÓN Y ESTIMACIÓN DEL PROYECTO**
   1. **Establecer la matriz de actividades para el MCVS**

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

* 1. **Asignar los recursos del proyecto**

**B.1. Recursos humanos:**

El software a desarrollar consta de 3 etapas ligadas entre sí, una etapa de diseño móvil, otra etapa de transición en el servidor y una última de interacción con el usuario en una pc local. Todo lo relacionado con el diseño móvil está a cargo de Perez Nicolas, el ensamblado en el server está dirigido por Buczacka Walter , mientras que la interface visible para el usuario estará bajo la supervisión de Gonzalez Gustavo. Esta redistribución será reforzada por otros miembros en momentos críticos para evitar atrasos muy significativos, la asignacion de tareas podrá observarse también en el Gantt.

**B.2. Recursos tecnológicos:**

La programación móvil estará enfocada a dispositivos con android 4.4 en adelante, y el código será escrito íntegramente en JAVA.

El servidor de la UNLa será escrito en PYTHON 2.7, y los datos serán almacenado con Workbrench 5.1.

Mientras que el software de escritorio, que generara y unificara los archivos GVsig, será escrito en JAVA y C.

* 1. **Definir el entorno del proyecto**

El proyecto está orientado a resolver un problema planteado por la Licenciatura en Gestión Ambiental, que tiene la necesidad de censar arboles en localidades determinadas, dicha información debe estar geolocalizada y puede o no, tener una imagen adjunta para describir aun mejor este árbol. La interface podrá ser utilizada por alumnos, docentes e incluso por cualquier otra persona que desee ayudar en el censado. Por lo que no se tendrá un logueado exigente en la aplicación. Para permitir que dicho software sea lo mas masivo posible, permitiendo solo algunos permisos especiales para los miembros claves del proyecto.

* 1. **Planificar la gestión del proyecto**

El proyecto fue planificado en primera instancia para que sea terminado el 1-11-2016, la distribución de las tareas en cada una de las etapas esta descripta en la imagen 3. En el diagrama de Gantt además vemos plasmados las 4 iteraciones pensadas para el proyecto.

1. **PROCESO DE SEGUIMIENTO Y CONTROL DEL PROYECTO**
   1. **Análisis de riesgos**

En una primera instancia, por medio del análisis del entorno pudimos identificar 3 riesgos, potenciales e inevitables, descritos en la imagen 4. El primero y el segundo consta del GPS, mientras que el tercero es sobre GvSig.

1. **PROCESO DE EXPLORACIÓN DE CONCEPTOS**
   1. **Identificar las ideas o necesidades  
        
      Necesidades a satisfacer:**

* Censado rápido y móvil de arboles en municipios
* Archivar censos en servidor y poderlo visualizar desde cualquier pc
* Comprensión de datos por medio de georeferenciacion usando gvsig

|  |  |  |  |
| --- | --- | --- | --- |
| ***Decripción*** | ***Impacto*** | ***Probabildad*** | ***Accion a Seguir*** |
| -  Dependencia absoluta con GPS | El dispositivo no cumplirá las solicitudes si no posee conexión de GPS en el instante de capturar la imagen | Alta | Incorporar ubicación manual del usuario, si el cliente lo desea |
| -   Error de entre 30 y 100 metros, error controlado y causado por el Departamento de defensa de lo EEUU | Las imágenes tomadas podrán estar hasta una cuadra distante a lo dicho por el GPS, cifra muy significativa para el ideal del SW | Media | Usar convergencia de serie bidimensional en un entorno acotado para disminuir el error en (n/n+33) |
| - Fuertes de incertidumbres causada por la escasa documentación del GvSig | Desconocimiento parcial de teoría detrás del uso de segmentos del GvSig | Baja | Ver que usos necesita y usa el usuario |

**Imagen 4**: Gestión de Riesgos.

**Ideas del equipo de software:**

* Crear una aplicación para android, disponible para cualquier usuario que quiera cumplir las necesidades del cliente.
* Utilizar la posición administrada por el GPS, corrigiendo los errores utilizando convergencia de puntos
* Crear una interface web para conseguir los datos desde cualquier pc
  1. **Formular las soluciones potenciales**

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

* En caso que la conexión GPS no esté disponible dejar que el usuario seleccione la posición por medio de un mapa en forma manual.
* Si el GPS funciona, tiene un error de entre 30 y 100 mts, para minimizar este error, proponemos la solución de convergencia de puntos, para bajar este error en un (, descripto en la referencia” a”.
* Diseñar en archivo autoejecutable que interprete los datos y los cargue en GvSig sin que el usuario deba hacer nada mas que un doble click.

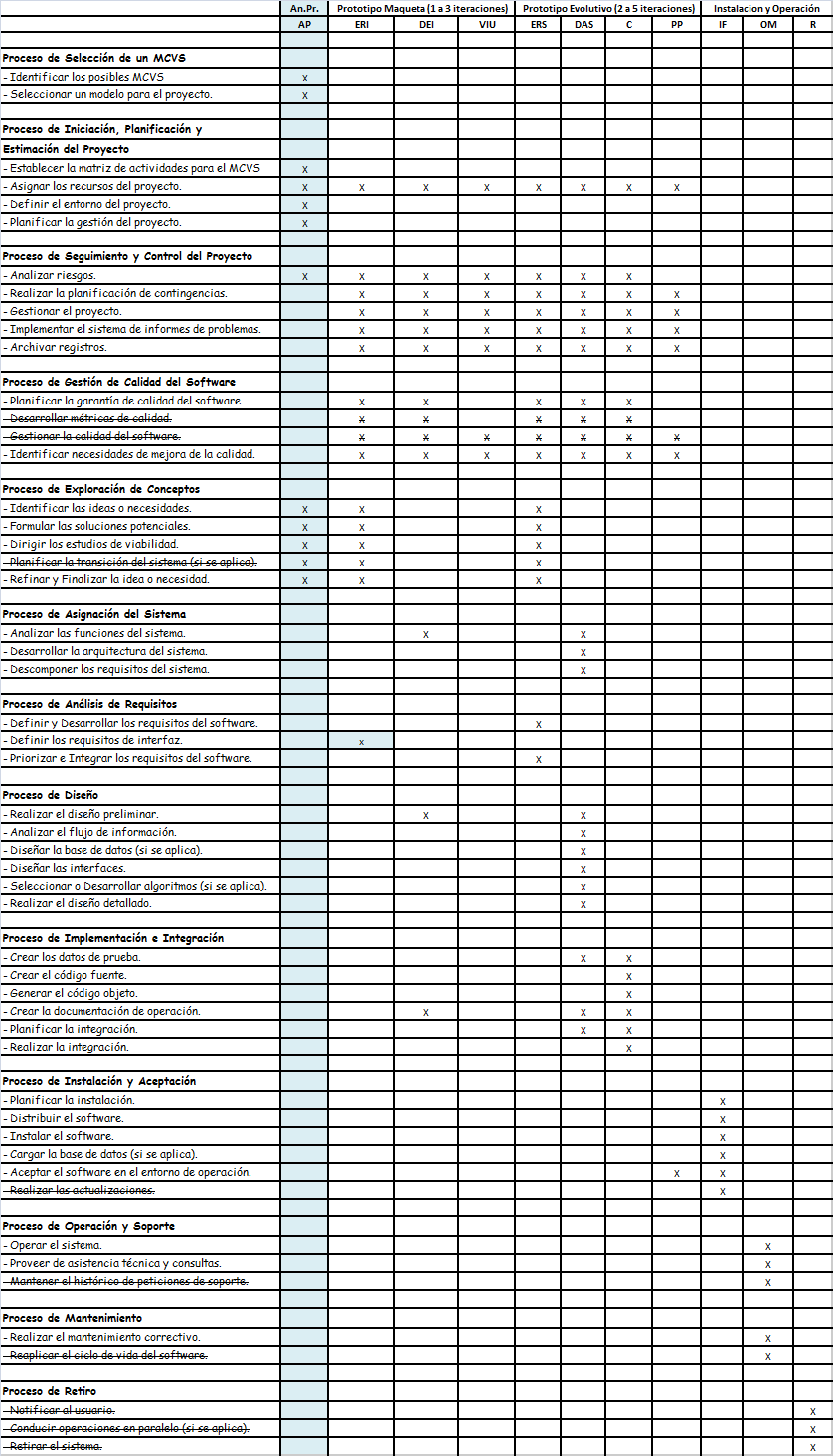


Imagen 2: Mapa de actividades

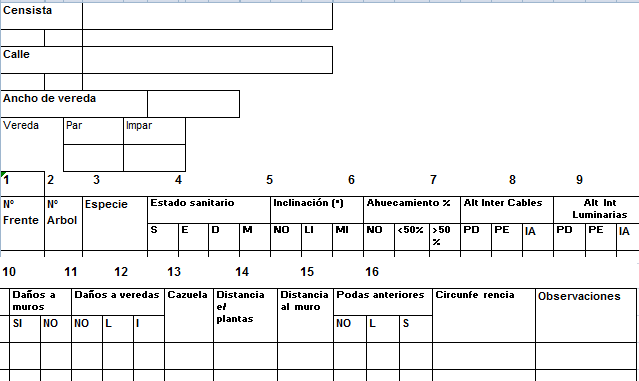
* 1. **Dirigir los estudios de viabilidad**

Los riesgos descriptos en la imagen 4 y las solucione planteada en el apartado anterior, se consideran 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 viables por los integrantes del equipo, por el conocimiento previo de todas las partes.**
* **Viabilidad del mercado: El proyecto tiene una ventaja que muchos otros proyectos desearían, el mismo no será sumergido a un mercado desconocido o inestable, el software será diseñado previo a un pedido, por lo que posee un mercado seguro esperando el desarrollo.**
* **Viabilidad conceptual: Si bien hay riesgos, y los requisitos fueron, en primera instancia muy volátiles, el comprendimiento del problema y la factibilidad de la solución han sido comprendidas por los encargados de este programa.** 
  1. **Refinar y Finalizar la idea o necesidad**

Uno de los principales problemas sobre el entendimiento del problema a resolver, fue desde un primer momento, la comprensión de lo datos a almacenar sobre el arbolado, por nuestro desconocimiento sobre el tema. Este problema fue resuelto, y refino nuestras ideas sobre la solución por medio de las respuestas del usuario, descriptas en el enlace 1, y plasmadas en la imagen 5.

Link1: <https://docs.google.com/forms/d/1duxmYBjoJQfQu0hXVa6AqDGU1VJIHG-53ZO6XSiB2dI/edit>



**Imagen 5:** Necesidad del usuario, datos que necesita el censista

# **Proceso de análisis de Requisitos**

* 1. ***Requisitos de Usuario***

**RU1**. Integrar datos excel con las herramientas gvSIG, hospedadas en la cpu personal del Usuario.

**RU2.** Capturar imágenes de árboles desde un dispositivo móvil

**RU3**. Completar una planilla con los datos pertinentes correspondientes a la imagen capturada.

**RU4**. El usuario puede cancelar el proceso en cualquier momento.

* 1. ***Requisitos del Sistema***

**RS1.** El sistema móvil, enviará las fotos georeferenciadas y etiquetadas al servidor de la UNLa

**RS2**. El sistema intermedio, alojado en el servidor, enviará los datos al usuario y guardará una copia de los datos en una base de datos Workbrench 5.5

**RS3.** El sistema que el usuario dispondrá en su pc personal recibirá los datos del servidor, los transformará en una planilla excel, y ese excel será codificado para la lectura gvSig

* 1. ***Requisitos Funcionales***

A continuación se enumera la lista de Requisitos Funcionales de la aplicación, que indican las funcionalidades que formarán parte del sistema:

**RF1.** El sistema permitirá capturar fotografías y adjuntar una descripción

**RF2.** El usuario podrá elegir si la fotografía merece o no formar parte del relevamiento de datos de arbolado.

**RF3.** El sistema georeferenciará las imágenes capturadas

**RF4.** El sistema se conectará con el servidor de la UNLa y enviará los datos al mail del usuario.

**RF5.** El sistema permitirá la interpretación de los datos por medio del motor gvSig

**RF6.** El sistema tendrá una interface web para la administración de datos y la vista de los mismos.

**RF7.** El sistema permitirá y exigirá un login para el uso.

**RF8.** El sistema determinará si las fotografías sacadas son o no de árboles.

* 1. ***Requisitos No Funcionales***

El siguiente apartado tendrá el objetivo de indicar restricciones sobre los Requisitos No Funcionales:

**RFN1.** El usuario no verá ni participará de los procesos

internos de conversión y obtención de datos.

**RFN2.** El software es íntegramente dependiente de la red GPS, en los momentos donde ésta no esté funcionando la aplicación no será utilizable.

**RFN3.** El sistema necesita conexión de internet inalámbrica.

**RFN4.** El sistema tendrá una aplicación android (superior a version 4.4) o posterior.

**RFN5.** El sistema tendrá una aplicación de escritorio (Windows)

**RFN6.** El sistema se desarrollará en JAVA (Android), Python (Servidor y Windows).

**RFN7.** El servidor contará con el Sistema Operativo Debian Jessie 10.0

**RFN8.** El sistema móvil no se encargará de la vista de información previamente subida.



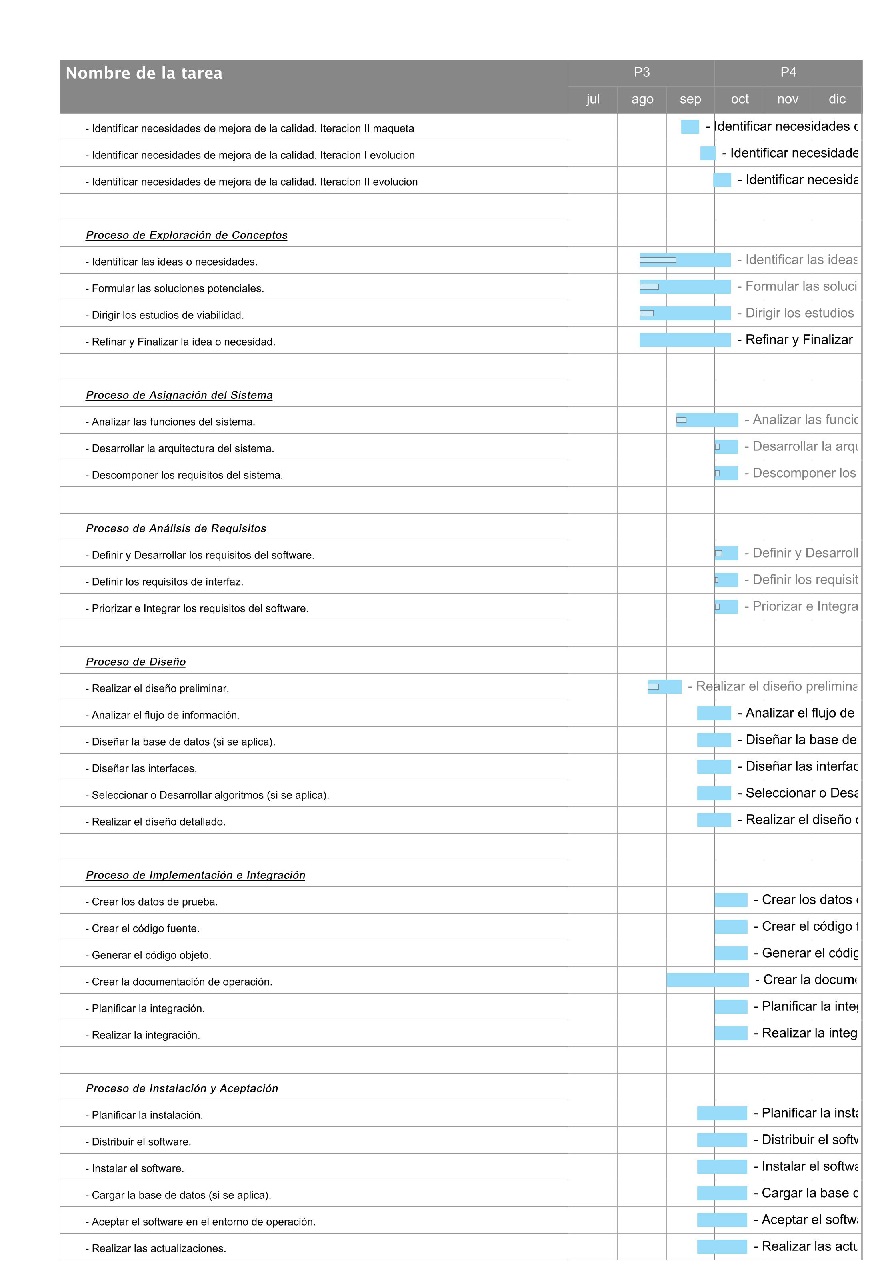


Imagen 3 Diagrama de Gantt