

# LLAMATHUST

*Planificación del Proyecto.*

# DESCRIPCIÓN DE LOS PROCESOS DEL PROYECTO

## Procesos de iniciación:

La empresa crea un equipo de empleados senior con alta experiencia para desarrollar el producto que creen que será el producto estrella de la compañía. Para formar el equipo envía una circular a los trabajadores que desean que formen parte del equipo solicitándoles que redacten un documento en el que expliquen la visión que tienen cada uno de ellos con respecto a la idea propuesta.

Posteriormente una vez formado el equipo se realizará una reunión inicial con los directivos de la empresa en la que se realizará una lluvia de ideas sobre aquellos aspectos que se desean cubrir con la aplicación y posibles implementaciones para llevarlos a cabo de modo que no se propongan cosas cuya implementación sea demasiado costosa o imposible de realizar.

Finalmente se extraerá una lista ordenada por prioridad de las propuestas realizadas de modo que se obtenga un plan inicial de tareas a desarrollar en la fase de planificación.

## Procesos de planificación:

### Alcance

Partiendo de la lista de propuestas obtenida del proceso de iniciación, se realiza un análisis con el objetivo de definir los hitos a los que se pretende llegar en cada conjunto de actividades. Usando los hitos definidos se podrán crear los conjuntos de tareas a realizar para obtenerlos.

### Definición de actividades

Conociendo los hitos a los que se deberá llegar con cada paquete de tareas a partir de la lista de tareas ordenadas por prioridad se definirán las actividades que darán lugar al hito. Las actividades se dividirán en subtareas que puedan ser asignables a trabajadores específicos para su realización.

En nuestro caso la división en tareas la hemos obtenido del trabajo realizado en la práctica anterior, dentro de cada tarea hemos creado subtareas en función de las divisiones que hemos encontrado en el progreso que habría que hacer para poder finalizar cada tarea. La división en subtareas es necesaria pues una tarea no puede finalizarse aislada del resto si no que sus subtareas deben coordinarse con las subtareas del resto de tareas. Cada subtaska se dividirá a su vez en cuatro actividades que lo que nos permitirá asignar recursos de una forma más gradual a cada una de ellas. Dichas actividades serán en todos los casos la planificación, el diseño, la programación y las pruebas unitarias y de integración.

### Definición de recursos

Nos vamos a encontrar tres tipos diferentes de recursos: los recursos humanos que hacen referencia a los trabajadores que participarán en el desarrollo del proyecto; los recursos de desarrollo que hacen referencia a todos esos recursos tanto materiales como no materiales (ordenadores, oficina, conexión a internet) que hagan falta para el correcto desarrollo del proyecto; los recursos de despliegue necesarios para poder tener el proyecto desarrollado funcionando y disponible para el público (servidores, software).

Teniendo como entrada los paquetes de actividades ya definidos anteriormente se procede a definir los recursos necesarios para poder realizar cada una de las actividades. En caso de que un recurso pertenezca a más de una actividad se distribuirá de forma adecuada de forma que no se sature ese recurso.

En el caso del proyecto los recursos que se manejarán serán los cuatro empleados que formarán parte en su desarrollo. Dichos recursos deberán ser gestionados de modo que se pueda obtener el máximo rendimiento de ellos, pero sin saturarlos en ningún momento. Para lograrlo se ha realizado una pequeña pre-planificación realizándola primero a mano y posteriormente en un Project inicial que nos ha servido de guía para crear la planificación final del proyecto completo.

### Estimación de la duración de las actividades

Conociendo las actividades y los recursos necesarios para cada una de ellas se realizará una estimación de la duración y esfuerzo para su compleción. Primero se realizará una estimación por puntos de función y posteriormente se ajustará con COCOMO II.

La estimación de las actividades se obtiene directamente de la estimación realizada en la práctica anterior.

### Planificación de la gestión de riesgos

Para la gestión de riesgos se va a usar la metodología definida por COSO. La ventaja de usar esta metodología es que intenta unir el Gobierno de las TIC con la gestión de riesgos, integrando el negocio con los procesos de tal

manera que ayude a conseguir los resultados esperados tanto en rentabilidad como en rendimiento. COSO ayudará a priorizar los objetivos, lo que permite adecuar la gestión y la toma de decisiones de forma segura para la empresa.

### **Cronograma**

Las actividades serán organizadas de forma gráfica en un cronograma en el que se indicará tanto su precedencia como duración temporal. El cronograma será una herramienta que se utilizará para indicar gráficamente el proceso realizado en el proyecto.

### **Presupuesto**

Se obtendrá un presupuesto a partir de los recursos necesarios para la realización del proyecto. Habrá que tener en cuenta la estimación realizada en COCOMO para poder determinar el presupuesto con mayor fiabilidad.

### **Desarrollo del proyecto**

Para el desarrollo del proyecto utilizaremos la metodología PMBOK que guiará cómo organizar los proyectos desde dos puntos de vista: el primer punto de vista será la organización del proyecto en función de los procesos, lo cual nos permitirá ordenar los procesos en el tiempo. Adicionalmente, tendremos otro punto de vista, que es la organización de procesos en áreas de conocimiento, lo cual nos permitirá agruparlos en diferentes materias.

## **Procesos de ejecución:**

### **Ejecución del plan del proyecto**

Para realizar este proyecto se utilizará la metodología de trabajo SCRUM, que es una metodología ágil. Las metodologías ágiles se caracterizan porque su objetivo principal consiste en satisfacer el cliente mediante entregas de prototipos funcionales en las fases tempranas del proyecto. Esto permite obtener un feedback inicial muy valioso que complementará a los requisitos iniciales del proyecto. La dificultad de estas metodologías es que por lo general son solo aplicables a equipos de trabajo de alto nivel y experiencia debido a que no son tan guiadas como las metodologías pesadas. No obstante, tiene la ventaja de poder reaccionar mejor ante cambios de especificación, requisitos o del entorno del proyecto. En concreto, SCRUM se basa en la utilización de sprints que se divide en ciclos de trabajo diarios. Cada ciclo de trabajo comienza con una reunión inicial en la que se analiza el trabajo realizado el día anterior y el trabajo que se va a realizar el día presente, así como mostrar al equipo los impedimentos o dificultades que se puedan encontrar para el desarrollo de este. Cada sprint supone definir una lista de tareas a realizar, que una vez hayan sido realizadas producirán un incremento del valor y de la funcionalidad del producto.

### **Aseguramiento de la calidad**

Para asegurar la calidad del producto nuestras decisiones se basarán en lo definido por la ISO 9001. Esta norma se centra en la identificación de procesos de la organización como actividad decisiva para su funcionamiento eficaz. Una vez identificados los procesos aplica sobre ellos el proceso de mejora continua PDCA basado en Planificar, Hacer, Comprobar y Actuar. Mediante la aplicación de la norma se pretende obtener un aumento de la satisfacción de los clientes finales lo cual es un factor determinante a la hora de determinar la calidad de un proceso. Se procurará por tanto aplicar la metodología PEDCA a todos los procesos identificados para llevar a cabo el proyecto.

### **Distribución de información**

Se identifican tres tipos de información manejar. La información de gestión que hace referencia a los datos de control, de progreso y de decisión dentro del proyecto. Esta información será manejada por un gestor de tareas que exponga un panel Kanban en el que poder expresar las tareas realizadas, por realizar y en progreso. Adicionalmente para transmitir otro tipo de información se utilizará un sistema de mensajería como Slack.

La información de negocio hace referencia al software y documentación generada a lo largo del proyecto. Dicha información será gestionada por un sistema de repositorios que permitirá mantenerla distribuida entre los trabajadores al mismo tiempo que genera un sistema de control de versiones online para proporcionar seguridad en caso de desastre.

La información de trabajo hace referencia al entorno de desarrollo que se utilizará dentro del proyecto. Dicha información se gestionará mediante contenedores que virtualizarán los programas que se utilicen de modo que puedan compartirse de forma sencilla entre los miembros del equipo, así como entre los servidores de despliegue del producto. Entre otras cosas la virtualización en contenedores permite unificar el entorno de desarrollo con el entorno de producción siendo este un único ecosistema agnóstico del hardware subyacente.

## **Procesos de control:**

### **Control de cambios**

El control de los cambios nos permitirá poder volver a versiones anteriores del proceso de desarrollo en caso de desastre al implementar una nueva funcionalidad. Dicho control se realizará mediante la acotación de cada cambio

en espacio y funcionalidad antes de realizarlo. El control de cambios y de versiones se automatizará mediante un gestor de repositorios en el caso de este proyecto en concreto dicho gestor de repositorios será git.

### Reportes de eventos

Necesitamos un sistema que controle las incidencias o eventos importantes en el desarrollo de nuestra actividad, de manera que quede claro en cada momento los aspectos a tratar más importantes.

Dicho reporte de eventos nos lo ofrece el sistema issues de GitHub, el cual no solo ofrece un sistema visual de reporte de eventos, sino que se pueden clasificar según diferentes tipos de este, ya sean errores a solucionar, temas a tratar o mejoras a considerar.

### Alcance del cambio

Un cambio podrá ser iniciado por la implementación de una nueva funcionalidad o para subsanar un error detectado. En ambos casos ante de realizar el cambio se deberá acotar su alcance en tiempo, espacio y repercusión en otras partes del proyecto.

Acotar un cambio en el tiempo permite crear un método de control de modo que se pueda identificar cuando un cambio puede llegar a desestabilizar el proyecto.

Acotar los cambios en el espacio permite determinar aquellas partes en las que físicamente se ha realizado el cambio lo que permite que estos sean trazables a lo largo del producto.

Por último, determinar la repercusión del cambio es otra forma de realizar la trazabilidad. Se define un árbol de las tareas que se verán afectadas por el cambio debido a que su entrada depende de algún modo de la salida generada por el cambio realizado. Cuanto más profundo sea dicho árbol más peligroso será el cambio realizado y más difícil será trazar el cambio a lo largo del proyecto.

### Control de la calidad

Para controlar la calidad nos basaremos en la ISO 9001, según la que todos los cambios que se realicen tendrán su actividad controlada puesto que seguirán los siguientes cuatro procesos: Planificar, hacer, verificar y actuar, de manera que en todo momento se estará comprobando que lo que se tenía que hacer se ha hecho correctamente.

Otro método para controlar la calidad es la realización de pruebas unitarias y de integración que se irán realizando sobre el software, de manera que cuando tengamos un sistema suficiente de módulos unitarios se realizarán las pruebas de integración que garanticen que todo lo que se va desarrollando funciona y se integra correctamente.

### Control del coste

Para controlar el coste utilizaremos la comparación entre las distintas curvas de control. Las curvas de control son de distintos tipos, pero todas ellas tienen en común que se compara coste con respecto a trabajo.

La primera de control hace referencia a la función definida entre el coste presupuestado y el trabajo realizado (CPTP).

La segunda curva define la función que relaciona el coste presupuestado con trabajo realizado (CPTR). Finalmente, la última curva se corresponde con la biyección entre coste real y trabajo realizado (CRTR).

Comparando las distintas curvas obtendremos información de gran relevancia sobre el progreso del proyecto con respecto a la planificación realizada del mismo. Si comparamos CPTP con CPTR obtendremos la desviación entre el trabajo programado y el trabajo realizado. Por el contrario, comparando las curvas de CPTR con CRTR obtendremos la diferencia entre el coste presupuestado con el coste real.

La primera diferencia nos indicará la desviación con respecto a al progreso planificado mientras que la segunda nos informará sobre las desviaciones en el presupuesto.

## Proceso de cierre

### Cierre administrativo

Cuando el proyecto haya finalizado se procederá a comprar cómo se ha desarrollado este con respecto a la planificación realizada. Esto permitirá a la empresa actualizar sus métricas internas de estimación de modo que en futuros proyectos sean más precisas de lo que lo hayan sido para este. El equipo podrá aprovechar el cierre del negocio para reflexionar sobre el trabajo que han realizado y como mejorarlo en futuros proyectos en los que participen.

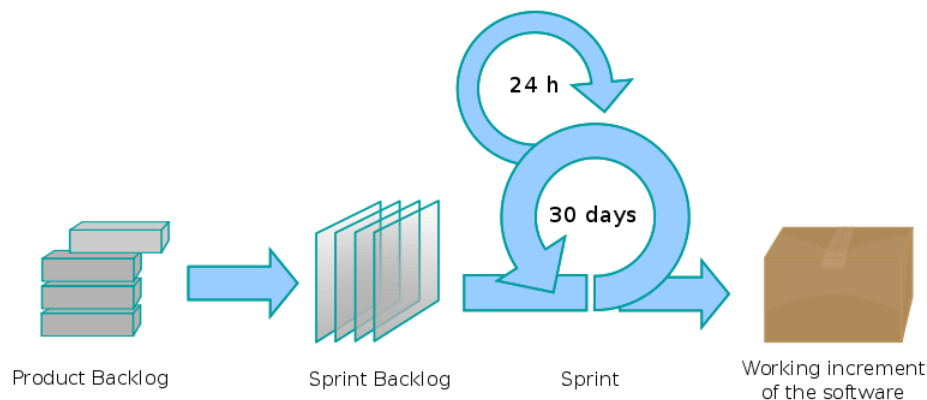
### Almacenamiento de la documentación

Tras finalizar el proyecto toda la documentación durante él generada deberá ser archivada e indexada de modo que pueda servir como referencia a futuros proyectos que la organización realice.

# ESTRUCTURAS HUMANAS Y TÉCNICAS PARA LLEVAR A CABO EL PROYECTO

## Estructuras humanas.

En esta parte hablaremos de los roles de cada uno de los trabajadores, así como de la metodología Scrum. Esta metodología pertenece a las denominadas metodologías ágiles que se caracterizan por adaptarse muy bien a los cambios en las especificaciones del producto, aunque son difíciles de aplicar si el equipo no es suficientemente experimentado.



Se definirá el rol para cada miembro del equipo, así como las partes en las que va a destacar dentro del proyecto, herramientas que se usan en el proyecto con las que él ya haya trabajado de modo que se justifique la reducción de complejidad aplicada en COCOMO. El proyecto LLAMATHRUST será realizado por 4 Ingenieros informáticos altamente cualificados:

- Juan Casado Ballesteros (Scrum Máster)
- David Menoyo Ros (Analista)
- Álvaro Vaya Arboledas (Técnico)
- David Márquez Mínguez (Programador)

El equipo deberá realizar reuniones diarias dirigidas por el Scrum Máster, de manera que durante el periodo de desarrollo se llevará un control total y directo en las diferentes tareas. Además, el hecho de que el equipo tenga experiencia en el sector de las IoT nos permitirá lograr una gran agilidad en la dirección e implementación del proyecto.

Las principales actividades que realizara cada miembro del equipo según su rol son las siguientes:

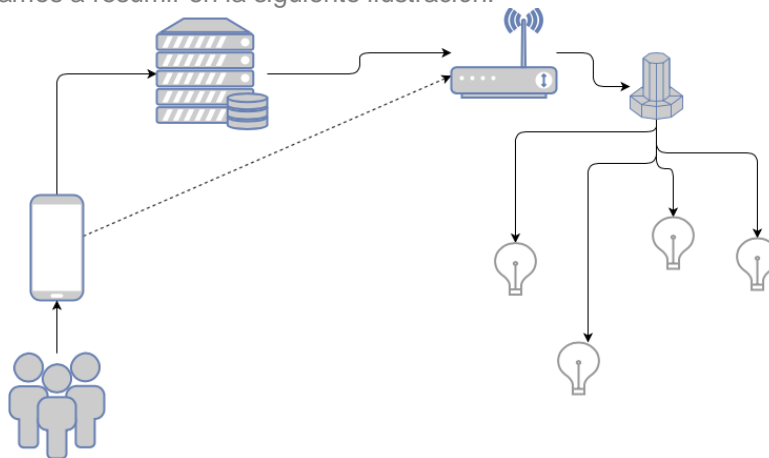
- Scrum Máster: encargado de la gestión del equipo y del desarrollo del proyecto, así como su posible participación en otras tareas durante el desarrollo del mismo. En nuestro caso el Scrum Máster es una persona muy experimentada en programación de bajo nivel por lo que sus tareas por defecto serán aquellas relacionadas con los enchufes y la puerta de enlace.
- Analista: el analista es el encargado de realizar todo lo relativo a la captura de requisitos y el diseño del sistema a implantar. Nuestro analista está especializado en el diseño del producto, así como en la planificación y la programación a más alto nivel por lo que sus tareas por defecto serán las relacionadas con la aplicación y la base de datos.
- Técnico: el técnico se encarga de todo lo relativo al diseño e implantación de la arquitectura que soportara al sistema y aplicación para su correcto funcionamiento. Está experimentado en el bajo nivel y por lo general ayudará al Scrum Master en las tareas de más bajo nivel.

- Programador: el programador se encarga de las principales tareas de implantación de la aplicación con el desarrollo de todo el código necesario. Está especializado en programación y pruebas y ayudará al analista a realizar las tareas de más alto nivel.

Para el desarrollo del proyecto además se requerirán diversas herramientas como GitHub para controlar las versiones del programa, contenedores para la encapsulación de programas, o herramientas como Jira para controlar el progreso del equipo conseguido mediante el uso de la metodología ágil SCRUM.

## Estructuras técnicas.

A continuación, se va a realizar una exposición de las estructuras técnicas necesarias para el desarrollo de nuestro proyecto. Para ello las vamos a resumir en la siguiente ilustración:



En dicho diagrama se puede apreciar:

- La utilización por parte de los clientes de una aplicación en el smartphone (o en la web) que les permite interactuar con el sistema IoT implantado en su hogar.
- La comunicación Cliente-Sistema IoT presenta nuestros servidores como punto intermedio de conexión para ofrecer mejores técnicas de seguridad.
- El router del hogar es la primera puerta de entrada de las órdenes hacia los dispositivos IoT. Pero antes de llegar a los mismos, la orden se filtra en un núcleo de usuario (microcontrolador) el cual es el encargado de interpretar y mandar las órdenes hacia los dispositivos correctos.

Otros aspectos concretos acerca de la infraestructura son:

- La base de datos operativa será Azure (representada con nuestro servidor), y contará con una configuración optimizada en función de las necesidades actuales de la empresa, es decir, de manera que se optimizará el coste. Esta base de datos estará diseñada para almacenar los datos del consumo de los usuarios que es el activo principal que se manejará.
- Existirá datawarehouse también en Azure que estará enfocada a realizar inferencia de información sobre los datos recogidos de los clientes. En esta base de datos estarán los datos de los otros sensores incluidos en los dispositivos, así como datos obtenidos de otras fuentes como el precio del kw/h en cada región. Un cliente solo guardará datos en esta base de datos cuando sea cliente premium.
- En la primera conexión que el usuario haga de nuestro sistema IoT, tendrá que conectar el núcleo de usuario a su router (por cable o inalámbricamente por conexión de red básica), de manera que nuestro servidor pueda establecer y validar las primeras configuraciones y direcciones IP.
- Cómo se actualizará la base de datos
- Los ciclos de pushing de los dispositivos será de una media por hora, dichas medidas subidas a la base de datos se irán reemplazando según el siguiente esquema:
  - 1 medida por hora en 7 días
  - 1 medida por día en 3 meses
  - 1 medida por mes en 10 días
  - En total serán 370 medidas por dispositivo en 10 años.

Hay que aclarar que estas medidas están optimizadas y filtradas, ya que el dispositivo realmente realiza una medición por segundo de los datos de sus sensores para las operaciones y servicios en tiempo real, pero solo almacenaremos permanentemente aquellos datos promedio.

- Los sensores que disponemos son:
  - Temperatura
  - Humedad
  - Cantidad de luz

Los sensores serán de tipo analógico y cada uno estará conectado a una entrada del microprocesador. estará conectado a una de las entradas de microcontrolador que será una ATtinny1634. Las medidas obtenidas tendrán una precisión de 12 bits (0 a 4096).

# ESTIMACIÓN DE FASES

Estimación de la duración de cada una de las fases en las que se va a dividir el proyecto atendiendo a lo estimados en COCOMOII aplicando la cantidad de personas que se necesiten para realizar cada tarea.

Para automatizar la tarea de calcular los tiempos en función de los trabajadores hemos realizado un pequeño programa en Java que toma como entrada lo calculado en COCOMO y la cantidad real de personas que habrá realizado cada tarea y nos proporciona los tiempos que la tarea durará.

```
run:
-----2-DISPOSITIVOS ENCHUFE-----
-3-Dispositivo Enchufe-Tomar datos de los sensores
    Dias: 7.0964994
    Dias: 18.818998
    Dias: 38.40165
    Dias: 11.2005
-8-Dispositivo Enchufe-Conectar los enchufes con la puerta de enlace
    Dias: 8.279249
    Dias: 21.955498
    Dias: 71.68308
    Dias: 13.067249
-13-Dispositivo Enchufe-Gestión de tareas que tomarán los datos periódicamente
    Dias: 8.279249
    Dias: 21.955498
    Dias: 71.68308
    Dias: 13.067249
-----DISPOSITIVOS DE PUERTA DE ENLACE-----
-19-Dispositivos de puerta de enlace-Conectar la puerta de enlace con los enchufes
    Dias: 1.2702
    Dias: 1.71
    Dias: 6.9508495
    Dias: 2.0358
-24-Dispositivos de puerta de enlace-Conectar la puerta de enlace con la base de datos
    Dias: 0.95265
    Dias: 2.565
    Dias: 10.426275
    Dias: 3.0537
-29-Dispositivos de puerta de enlace-Procesar las medidas recibidas de los sensores
    Dias: 1.4819
    Dias: 3.99
    Dias: 12.163987
    Dias: 3.5626497
-34-Dispositivos de puerta de enlace-Enlazar la puerta enlace con la aplicación de usuario
```

## Dispositivos Enchufe

- Tomar datos de los sensores: 30%
  - Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 7 días.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 18 días.



- Programación:
  - Personas: Scrum Master, Técnico, Analista, Programador.
  - Tiempo: 38 días.
- Integración y prueba:
  - Personas: Scrum Master, Técnico, Analista, Programador.
  - Tiempo: 11 días.
- Conectar los enchufes con la puerta de enlace: 35%
  - Planificación:
    - Personas: Scrum Master, Programador.
    - Tiempo: 8 días.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 21 días.
  - Programación:
    - Personas: Scrum Master, Técnico, Analista [25%], Programador [25%].
    - Tiempo: 71 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 13 días.
- Gestión de tareas que tomarán los datos periódicamente: 35%
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 8 días.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 22 días.
  - Programación:
    - Personas: Scrum Master, Técnico, Analista [25%], Programador [25%].
    - Tiempo: 71 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 13 días.

## Dispositivos Puerta de Enlace

- Conectar la puerta de enlace con los enchufes 20%:
  - Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 1 día.
  - Programación:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 7 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 2 días.
- Conectar la puerta de enlace con la base de datos 15%:
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 2 días.

- Programación:
  - Personas: Analista, Programador.
  - Tiempo: 10 días.
- Integración y prueba:
  - Personas: Analista, Programador.
  - Tiempo: 3 días.
- Procesar las medidas recibidas de los sensores 35%:
  - Planificación:
    - Personas: Scrum Master, Técnico, Analista.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico, Analista.
    - Tiempo: 5 días.
  - Programación:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 13 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico, Analista, Programador.
    - Tiempo: 10 días.
- Enlazar la puerta de enlace con la aplicación del usuario 30%:
  - Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 5 días.
  - Programación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 20 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 10 días.

## Aplicación de Clientes

- Creación de interfaz 40%:
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 2 días.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 7 días.
  - Programación:
    - Personas: Analista, Programador.
    - Tiempo: 30 días.
  - Integración y prueba:
    - Personas: Analista, Programador.
    - Tiempo: 8 días.
- Enlazar la aplicación con la base de datos 25%:
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 4 días.

- Programación:
  - Personas: Analista, Programador.
  - Tiempo: 18 días.
- Integración y prueba:
  - Personas: Analista, Programador.
  - Tiempo: 5 días.
- Mostrar los datos recibidos en la interfaz 35%:
  - Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 2 días.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 6 días.
  - Programación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 26 días.
  - Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 14 días.

## Base de Datos

- Crear la estructura de los datos 50%:
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 2 días.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 6 días.
  - Programación:
    - Personas: Analista, Programador.
    - Tiempo: 26 días.
  - Integración y prueba:
    - Personas: Analista, Programador.
    - Tiempo: 12 días.
- Recibir peticiones externas 20%:
  - Planificación:
    - Personas: Analista, Programador.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Analista, Programador.
    - Tiempo: 3 días.
  - Programación:
    - Personas: Analista, Programador.
    - Tiempo: 15 días.
  - Integración y prueba:
    - Personas: Analista, Programador.
    - Tiempo: 4 días.
- Gestión de cuentas 30%:
  - Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 1 día.
  - Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 3 días.

- Programación:
  - Personas: Scrum Master, Técnico.
  - Tiempo: 15 días
- Integración y prueba:
  - Personas: Técnico.
  - Tiempo: 4 días

# PROGRAMACIÓN DE LAS ACTIVIDADES DEL PROYECTO

Hemos dividido cada una de las tareas que componen el proyecto en un conjunto de subtareas. Cada subtaska tiene cuatro fases Planificación, Diseño, programación e Integración y pruebas. Dichas fases tendrán que ser secuenciales entre sí de modo que en ningún caso podrá realizarse por ejemplo la programación antes del análisis o la planificación.

Distintas fases de distintas subtareas sí podrán ser concurrentes entre sí.

Con respecto a las subtareas existirá el siguiente orden de precedencia:

1. Tomar datos de los sensores. (A)
2. Crear la interfaz de la aplicación. (B)
3. Conectar los enchufes con la puerta de enlace. (C)
4. Conectar la puerta de enlace con los enchufes. (C)
5. Procesar las medidas recibidas de los sensores. (C)
6. Crear la estructura de los datos. (D)
7. Recibir las peticiones externas. (D)
8. Enlazar la aplicación con la base de datos. (E)
9. Conectar la puerta de enlace con la base de datos. (E)
10. Gestión de las tareas que tomarán los datos periódicamente. (E)
11. Mostrar los datos en la interfaz. (F)
12. Enlazar la puerta de enlace con la aplicación de usuario. (F)
13. Gestión de cuentas. (F)

Las primeras tareas se corresponden con la toma de datos de los sensores y la creación de la interfaz. Se han elegido estas dos tareas pues se corresponden con polos opuestos del proyecto y por tanto podrán generar entregable de forma rápida al inicio de este. Además, podrán hacerse de forma concurrente.

Las subtareas de conectar los enchufes con la puerta de enlace, conectar la puerta de enlace con los enchufes, procesar las medidas recibidas de los sensores se realizarán a continuación pues forman la estructura de generación de los datos con los que se trabajará en la aplicación.

Crear la estructura de los datos, recibir las peticiones externas, enlazar la aplicación con la base de datos son las tareas que se realizarán a continuación. Estas tareas permiten enlazar la aplicación con los enchufes, así como almacenar los datos recogidos. Es necesario que sean realizadas en este punto pues de otro modo no podríamos transmitir los datos entre los enchufes y la aplicación.

A continuación, se realizarán las tareas que permitan pasar los datos de los enchufes hasta la base de datos, conectar la puerta de enlace con la base de datos, Gestión de las tareas que tomarán los datos periódicamente. Esta última tarea se corresponde con la optimización de esta comunicación en paquetes de datos en vez de enviar los datos dispersos.

Mostrar los datos en la interfaz, enlazar la puerta de enlace con la aplicación de usuario, gestión de cuentas se corresponden con las tareas finales en las que los datos serán mostrados en la aplicación.

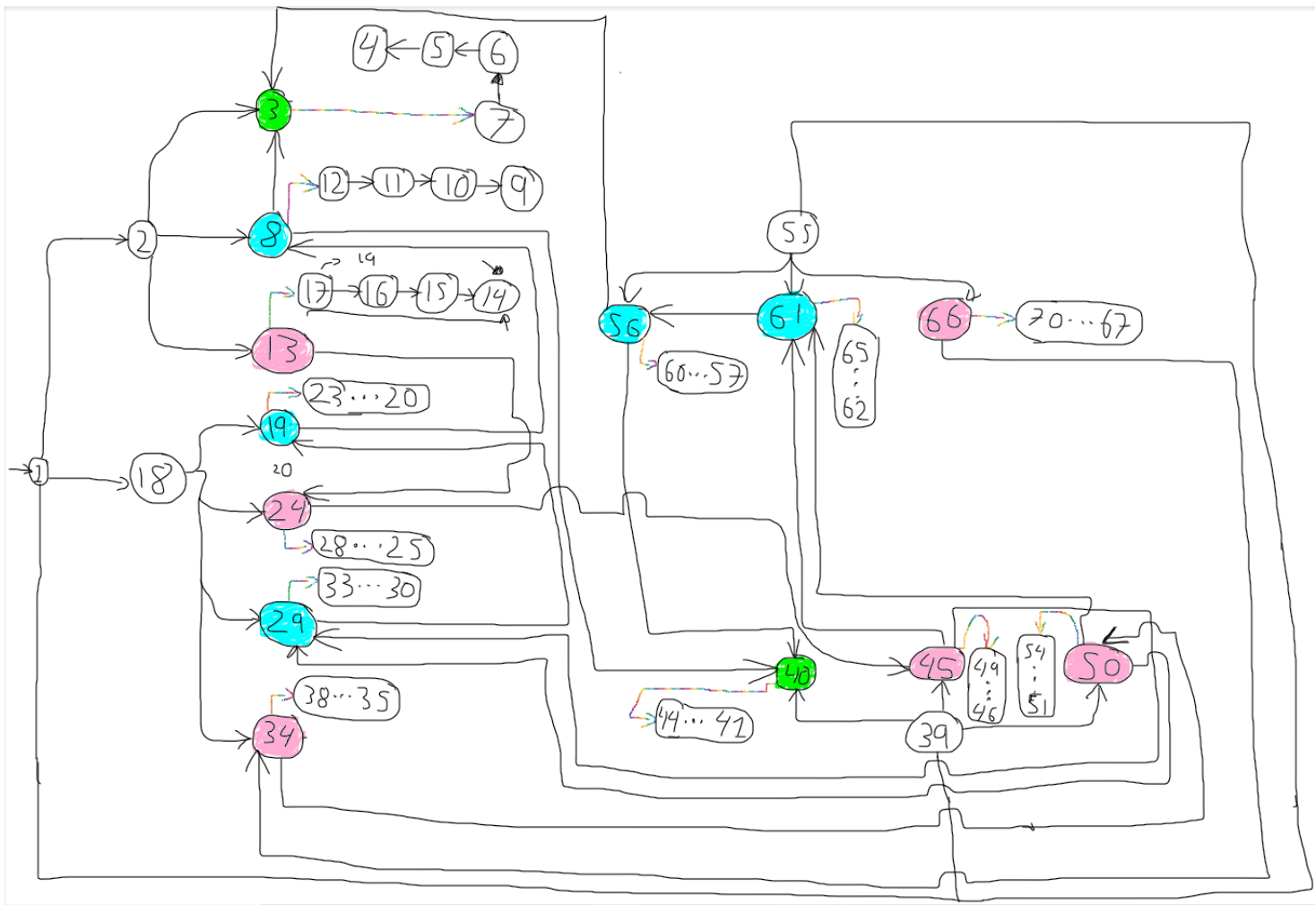
## Grupos de subtareas.

Cada subtaska se ha incluido en un grupo (A, B, C, D, E, F). Tareas de distintos grupos consecutivos podrán ser concurrentes en parejas de dos. Es decir, las tareas de grupo A se pueden hacer junto con las de grupo B, Las tareas de grupo C se pueden hacer a la vez que las del grupo D y finalmente las tareas del grupo E podrán realizarse junto a las del grupo F.

Las tareas que pertenecen al mismo grupo deben de ser secuenciales entre sí.

## Organización previa.

Antes de comenzar con Project realizamos un estudio previo sobre cómo organizar las tareas de modo que todas se realizaran en un orden adecuado. Una tarea que toma como entrada la salida de otra deberá siempre realizarse después de esa otra. De este modo creamos la dependencia entre las tareas.



## ENTREGABLES DEL PROYECTO

En nuestro proyecto distinguiremos:

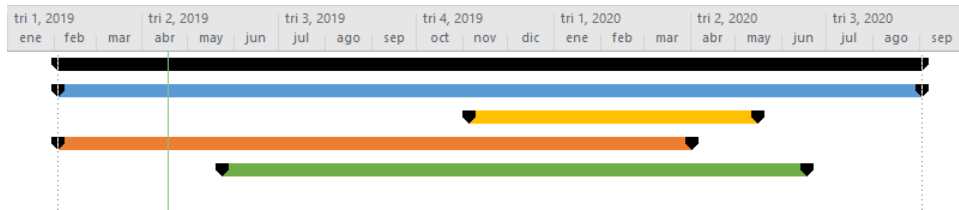
- Tareas:
  - Se corresponde con un objetivo mayor que supone una pieza clave y bien diferencia del proyecto.
  - Ejemplo:
    - Dispositivos de Enchufe
    - Dispositivos de Puerta de Enlace
    - Aplicación de Clientes
    - Base de Datos
  - Cada vez que una tarea se complete, se producirá un **Hito**.
- Subtareas:
  - Se corresponden con objetivos que permiten la realización de una tarea.
  - Por ejemplo:
    - La tarea Base de Datos tiene 3 subtareas:
      - Crear la estructura de los datos
      - Recibir peticiones externas
      - Gestión de cuentas
  - Cada vez que una subtarea se complete, se producirá un **Entregable**
- Actividades:
  - Se corresponden con los objetivos que permiten completar una subtarea.
    - Por ejemplo:
      - La subtarea de “Recibir peticiones externas” (y en realidad todas las subtareas de nuestro proyecto tienen las mismas actividades) precisa de las siguientes 4 actividades:
        - Planificación
        - Diseño del Producto
        - Programación
        - Integración y Prueba

Todas las subtareas presentan 4 actividades que se realizarán de manera secuencial, es decir, planificación, diseño, programación y por último integración y prueba se realizarán secuencialmente y completarán la subtarea correspondiente.

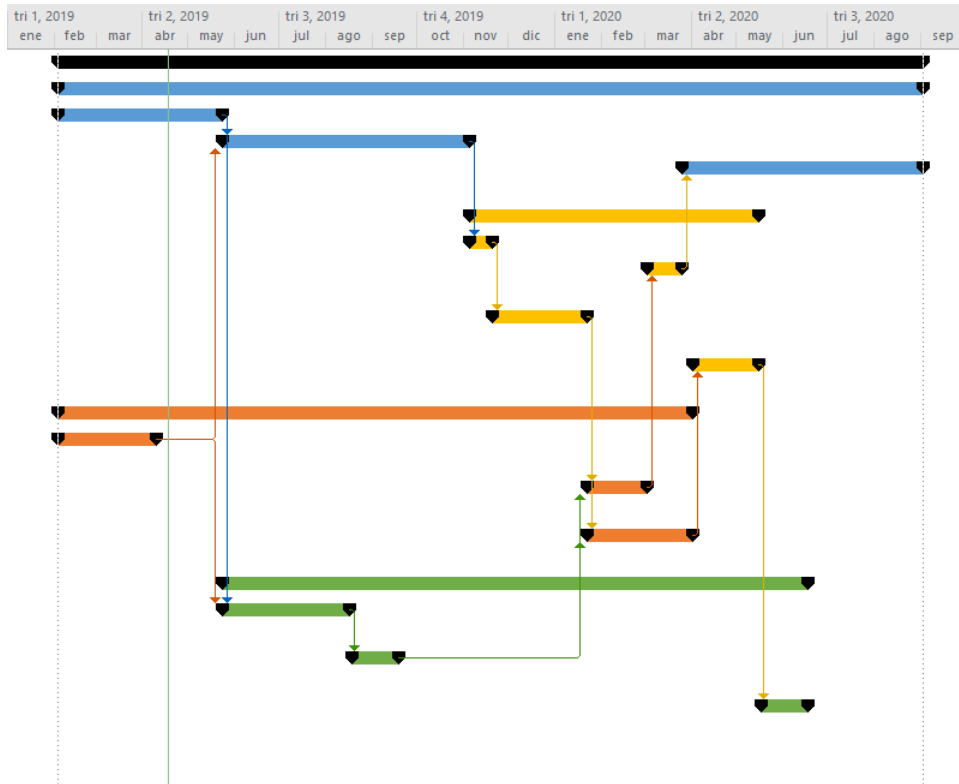
Nuestro proyecto empieza con las tareas de Tomar datos de los sensores y Creación de la interfaz, los cuales generarán los dos primeros entregables del proyecto, correspondientes a la tarea de Dispositivos de Enchufe y de Base de Datos respetivamente.

El informe de la primera subtarea se envía a la siguiente subtarea de la propia actividad, Conectar los enchufes con la puerta de enlace; en cambio, el informe de la segunda subtarea se envía a dos subtareas pertenecientes a tareas diferentes, concretamente se envía a Dispositivos Enchufe-Conectar los enchufes con la puerta de enlace y a Aplicación de clientes-Crear la estructura de los datos.

Podemos ver que los entregables del proyecto no empezarán a ocurrir hasta pasado el primer año de desarrollo. El primer entregable será la aplicación y posteriormente vendrán la puerta de enlace, la base de datos y por último los dispositivos de enchufe. No obstante, esto no representa del todo a cómo se han desarrollado cada uno por lo que será necesario estudiar el proyecto a nivel de subtareas y no de tareas.

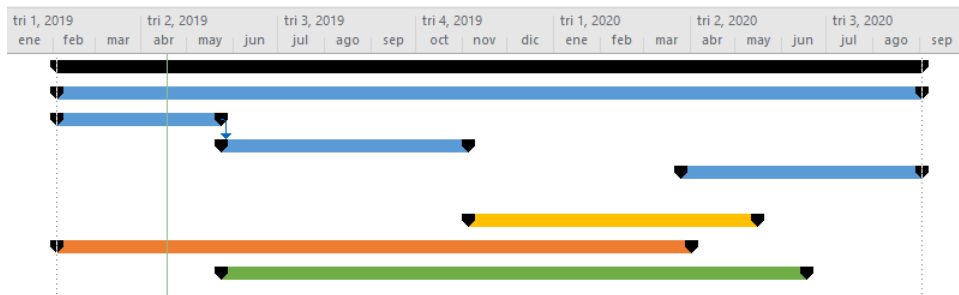


A nivel de tareas podemos ver como las primeras en comenzar son las de los enchufes y la aplicación, aunque a nivel de subtareas podemos ver que en realidad el grueso de la aplicación se desarrolla a mediados del proyecto. Algo similar sucede con la base de datos en la que se comienza a trabajar al inicio del proyecto, pero hasta no es terminada hasta casi el final.



## Dispositivos enchufe.

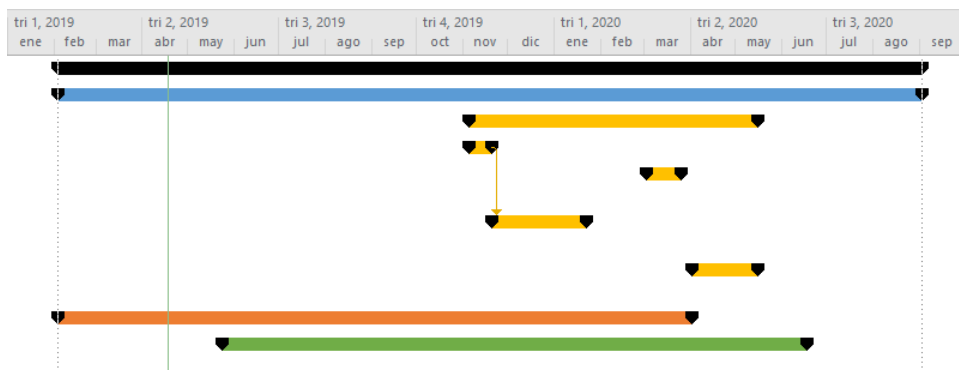
Su desarrollo es continuo a lo largo de todo el proyecto salvo el periodo en el que se integran con el resto del producto, es decir, los enchufes deben de integrarse con la puerta de enlace y la aplicación. Se ve como en el periodo en el que el trabajo en los enchufes está paralizado las otras partes están en pleno desarrollo.





## Puerta de enlace.

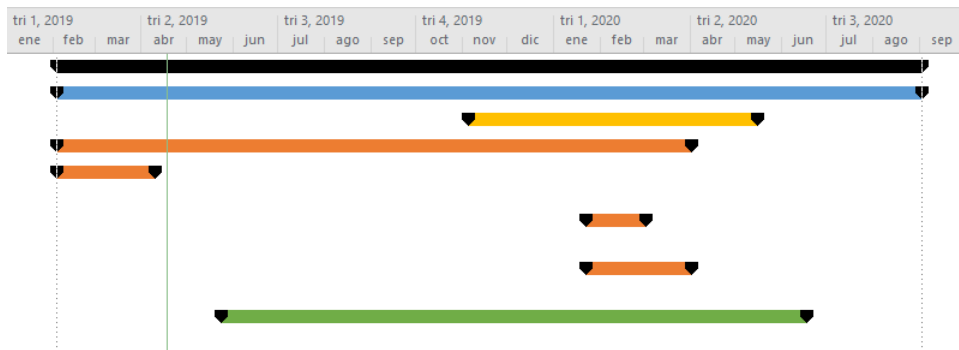
La puerta de enlace se desarrolla de forma acompasada con la aplicación. Ambos desarrollos son prácticamente paralelos pues deben cooperar para poder transmitir correctamente las medias de los sensores hasta el usuario final.



## Aplicación.

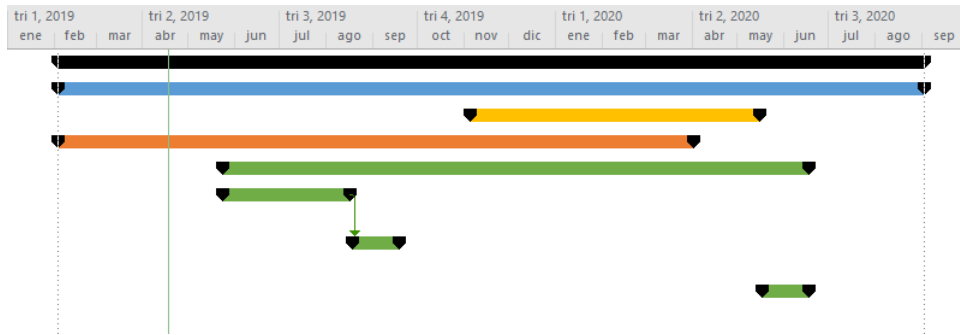
Se ha considerado que realizar el hito de la interfaz al inicio del proyecto es bueno, aunque no aporte funcionalidad significativa sin estar conectado a otras partes del mismo. Se toma esta decisión debido a que teniendo la interfaz es más fácil comenzar a realizar tareas de máquetin en las que esta se podrá mostrar. Adicionalmente se cree que es motivador para el equipo trabajar viendo resultados directos en el producto final dando vida progresivamente a la interfaz según se completan otros hitos que no cuando está todavía no existe.

El desarrollo final de la aplicación sucede al final del proyecto cuando el resto de la infraestructura ha sido completada.



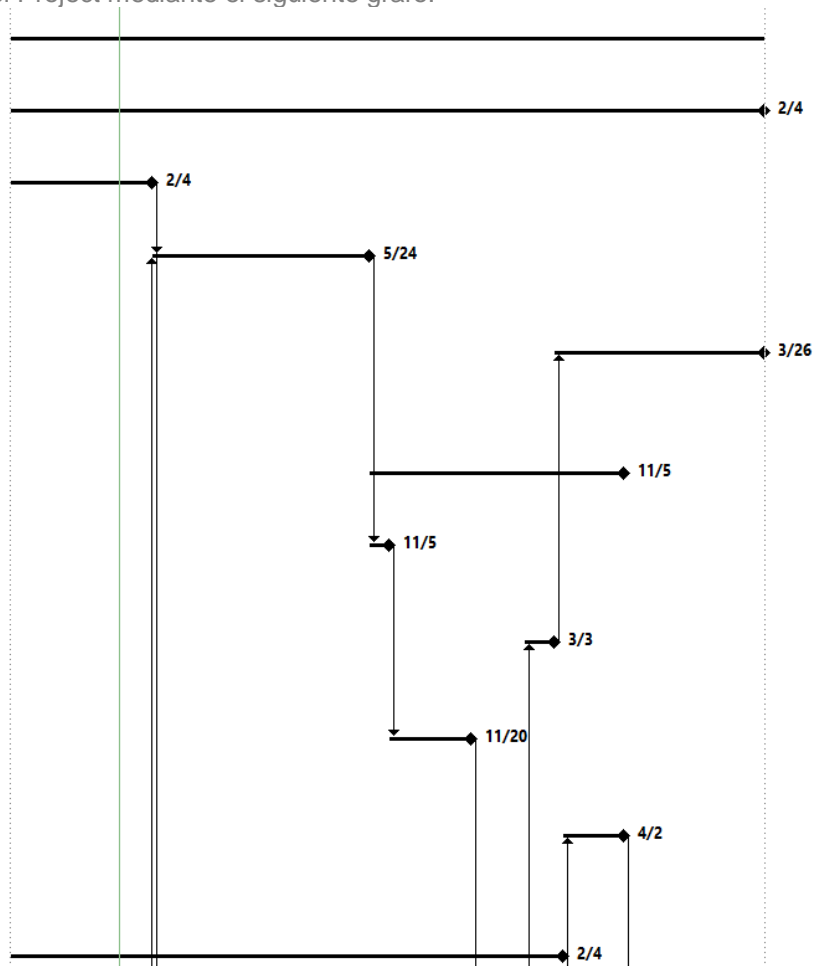
## Base de datos.

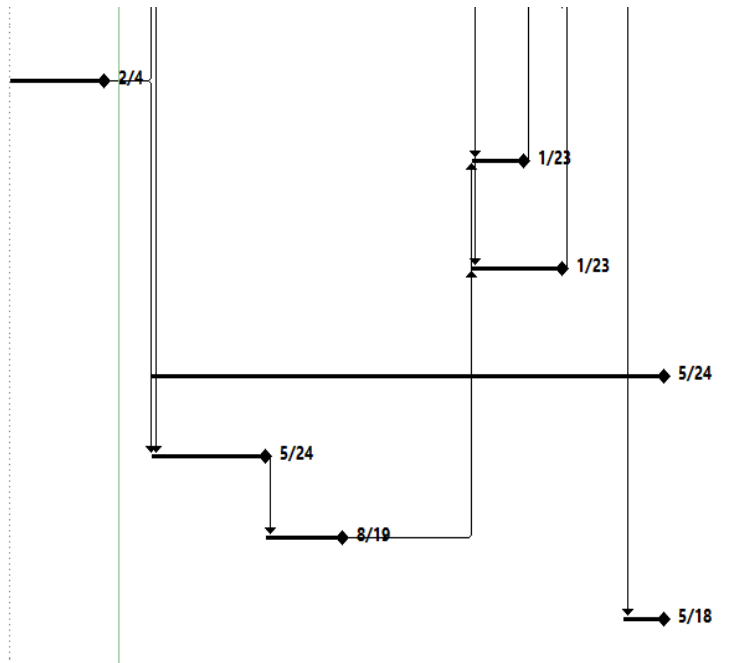
El trabajado en la base de datos se divide en dos fases diferenciadas. Una al inicio del proyecto dónde se producen los hitos de crear la estructura de los datos y conectar la base de datos con el resto del producto. No obstante, la parte final de esta, la parte correspondiente con la gestión de cuentas es retrasada hasta esta que el proyecto avance, momento en que esta fase se hará más necesaria según nos aproximemos al lanzamiento del producto.



### Hitos.

Como hemos mencionado cada Tarea y cada Subtarea serán un hito que dará lugar a un nuevo entregable. Esto puede visualizarse en el Project mediante el siguiente grafo.



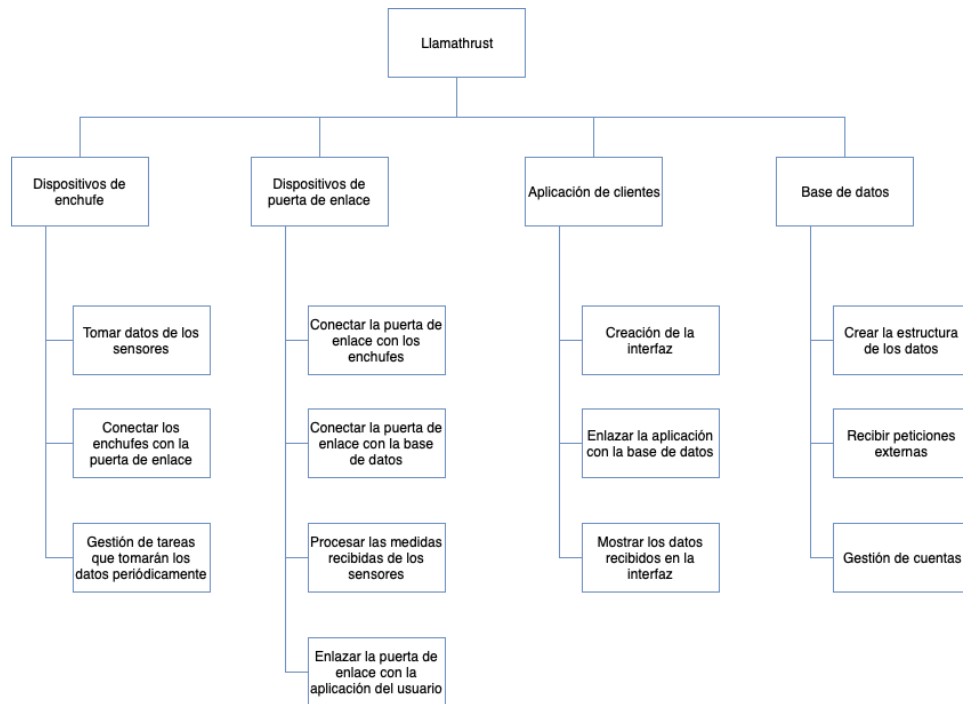


# WORK BREAKDOWN STRUCTURE

Se presenta la estructura de tareas que componen el proyecto.

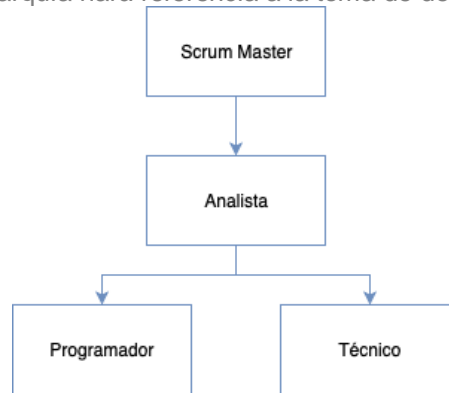
Cada una de las tareas se descompone a su vez en una serie de subtareas que una vez completadas darán lugar a una funcionalidad nueva del producto. A nivel organizativo cada una de las subtareas se descompondrá en cuatro actividades correspondientes a planificación, diseño, programación y pruebas.

Cada actividad será llevada a cabo por uno o varios miembros del equipo en función de su disponibilidad y cualificación.

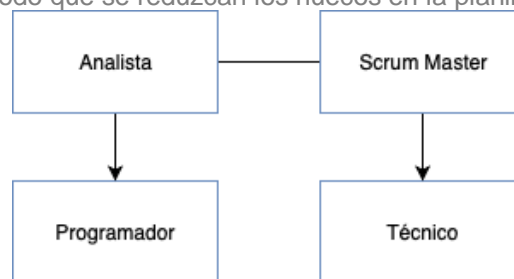


## RESOURCE BREAKDOWN STRUCTURE

Realizaremos dos diagramas de Breakdown Structure de los trabajadores en la organización. En el primero nos centraremos en el aspecto directivo de gestión del proyecto. En este diagrama mostraremos la jerarquía existente entre los miembros de este. Dicha jerarquía hará referencia a la toma de decisiones.



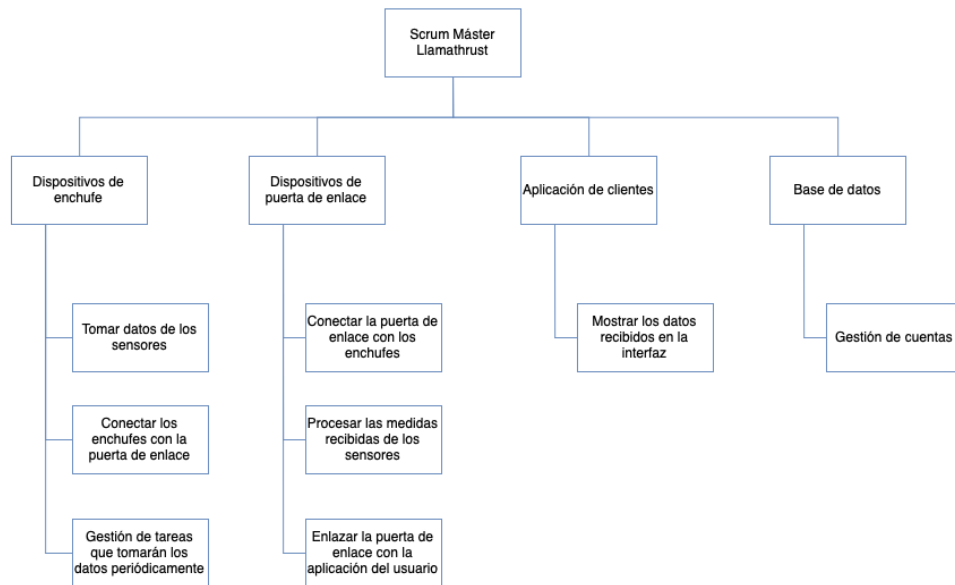
El siguiente diagrama por el contrario hace referencia a los grupos de trabajo por defecto dentro del proyecto. Por lo general el Scrum Master y el Técnico realizarán aquellas tareas que sean de bajo nivel (enchufes y puerta de enlace) mientras que el Analista y el programador realizarán las de más alto nivel (aplicación móvil y base de datos). Dicha estructura existe debido a que son los campos en los que cada uno de los miembros tiene más experiencia, no obstante, todos los miembros del equipo se apoyarán entre ellos ya sea aportando ideas en las reuniones como proponiendo la metodología Scrum como ayudando directamente en la realización de las tareas como método para acortar las tareas más largas de modo que se reduzcan los huecos en la planificación del trabajo.



# ORGANIZATION BREAKDOWN STRUCTURE

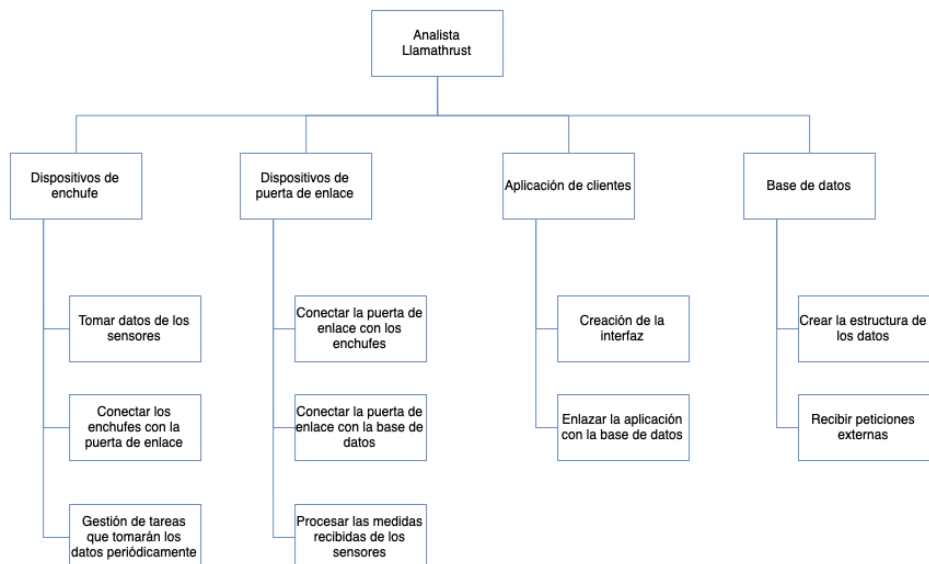
## Scrum Master

Tal y como hemos indicado previamente el Scrum Master se dedicará a realizar las tareas de más bajo nivel correspondientes con los dispositivos enchufe y la puerta de enlace. Adicionalmente en la aplicación y la base de datos realizará aquellas subtareas relacionadas con conectar las partes del alto nivel con las de bajo nivel.



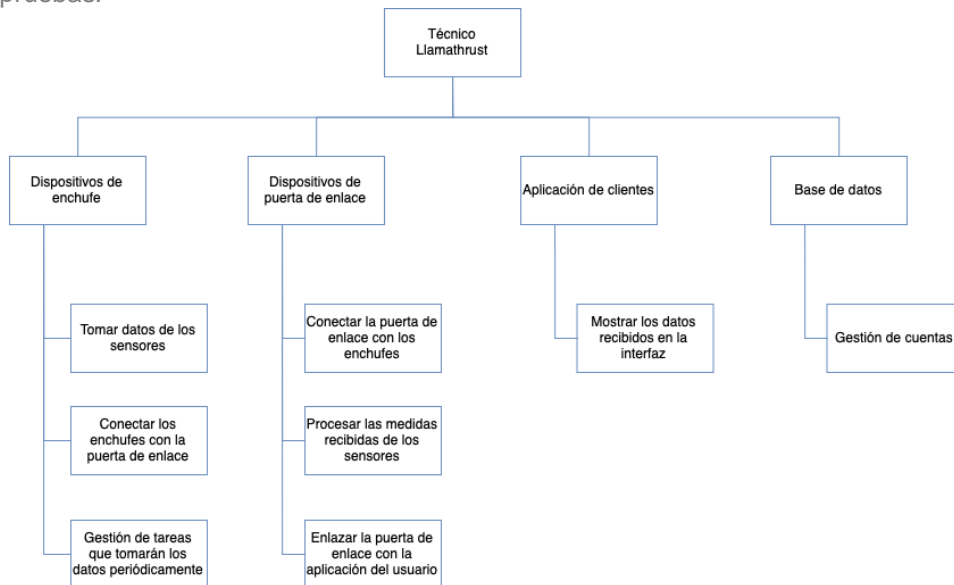
## Analista

Realizará principalmente las tareas de más alto nivel, aunque adicionalmente ayudará al Técnico y al Scrum Master en las teas de bajo nivel. Esto as así pues el enchufe y la puerta de enlace son las tareas de mayor duración y debido a que el equipo de trabajo está formado tan solo por cuatro miembros necesitan de apoyo extra para no retrasar el desarrollo total del producto.



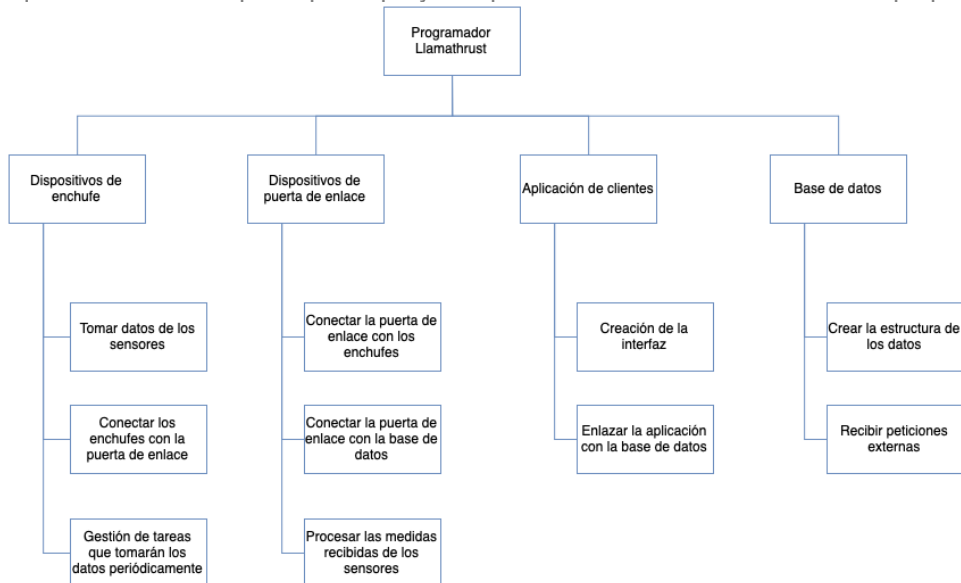
## Técnico

Ayudará al Scrum Master en sus tareas, especialmente en las de más bajo nivel y en las relacionadas con programación y pruebas.



## Programador

Trabjará junto al Analista en las tareas de alto nivel, aunque al igual que él dará su apoyo en las tareas de mayor duración siempre que sea necesario para que el proyecto pueda realizarse en el menor tiempo posible.



## TAREAS CRÍTICAS

Se puede ver como el camino crítico queda reducido a unas pocas tareas al final del proyecto. Estas tareas son aquellas que pertenecen al final del desarrollo donde todos los miembros del equipo cooperan para finalizar el proyecto.

En el resto de las tareas existirán pequeños márgenes de retraso admisible debido a los periodos existentes entre las reasignaciones de trabajadores de unas a otras tareas.

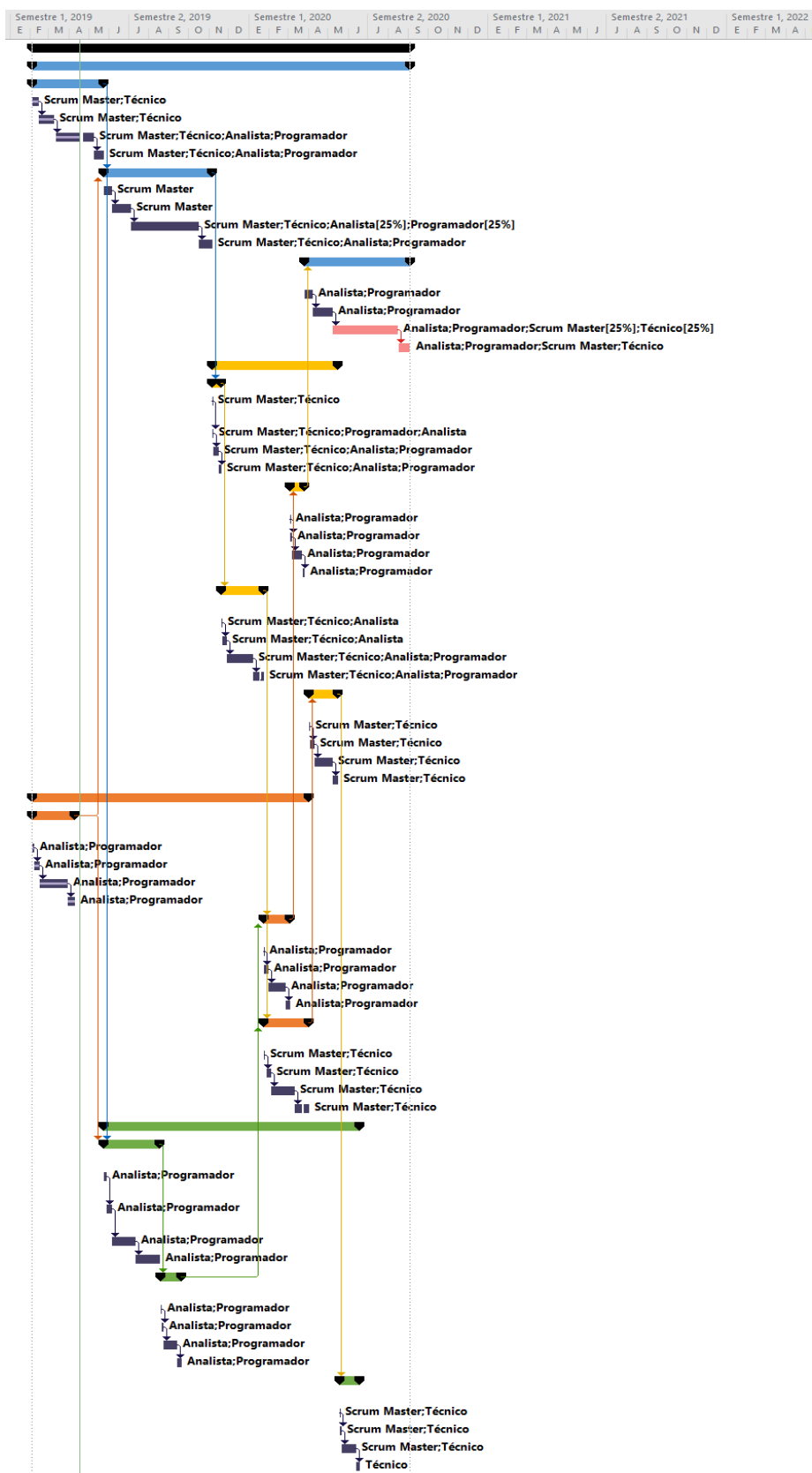
Podemos ver como el proyecto comienza realizando la infraestructura mínima de tomar los lados de los sensores y de crear la interfaz de la aplicación. Esto permite obtener resultados iniciales de gran valor para validar las especificaciones del producto.

Posteriormente el desarrollo de la infraestructura de los enchufes continuará llevado a cabo por el Scrum Master y el Técnico mientras de forma paralela el Analista y el Programador crean la infraestructura de la base de datos.

Una vez esto haya sido completado se creará la puerta de enlace parte en la que todo el equipo cooperará. Para concluir en Scrum Master y el Técnico unirán la base de datos con la puerta de enlace mientras el Analista y el técnico realizan la parte de gestión de las tomas de medias en los enchufes. Por último, se vuelve a trabajar en la aplicación conectándola con el resto de las componentes del producto mientras se finalizan los enchufes que junto a la gestión de cuentas serán las últimas tareas en realizarse.

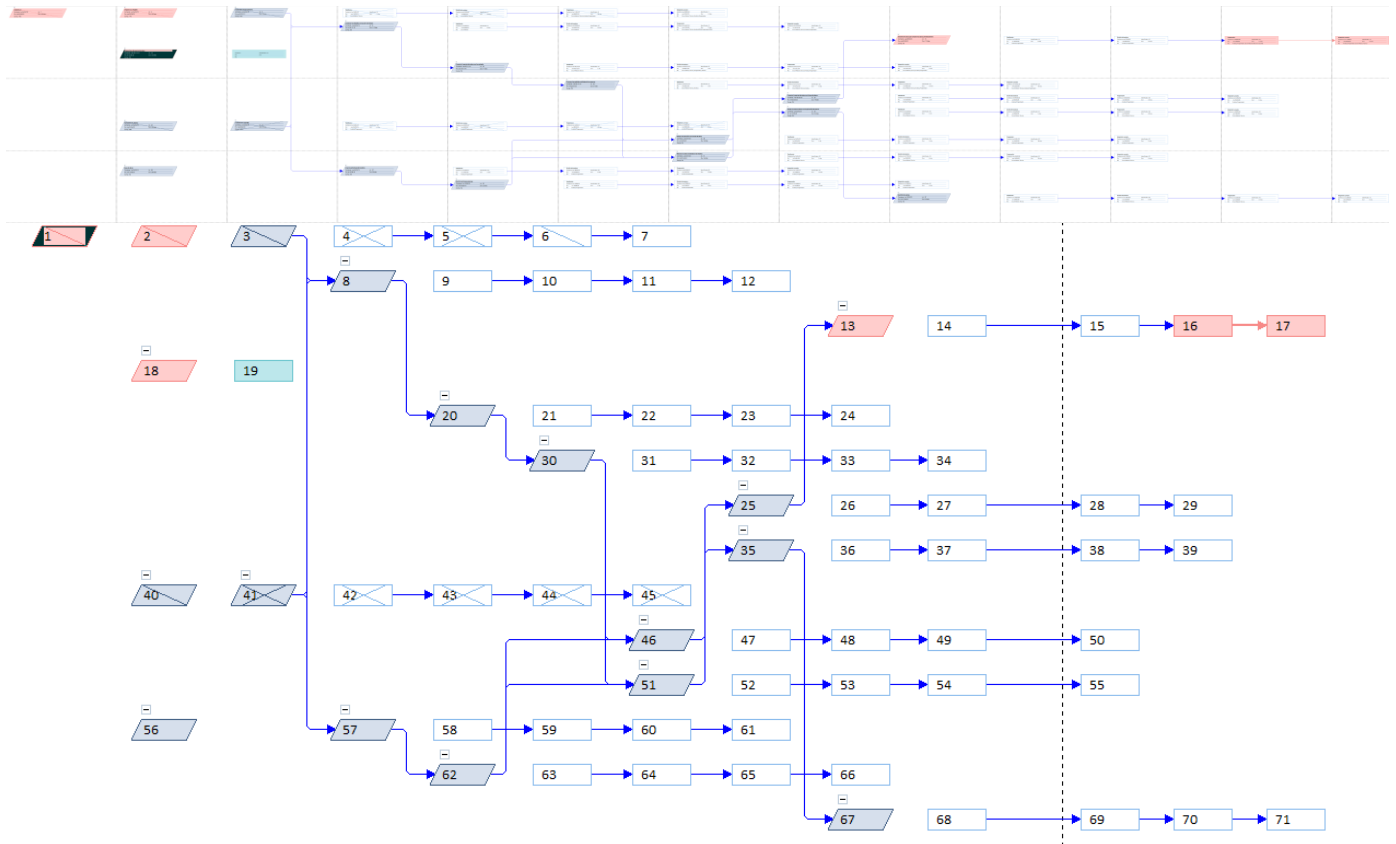
Como se puede apreciar el proyecto dura 399 días de trabajo que acaban siendo 19 meses de trabajo real.





## DIAGRAMA PERT

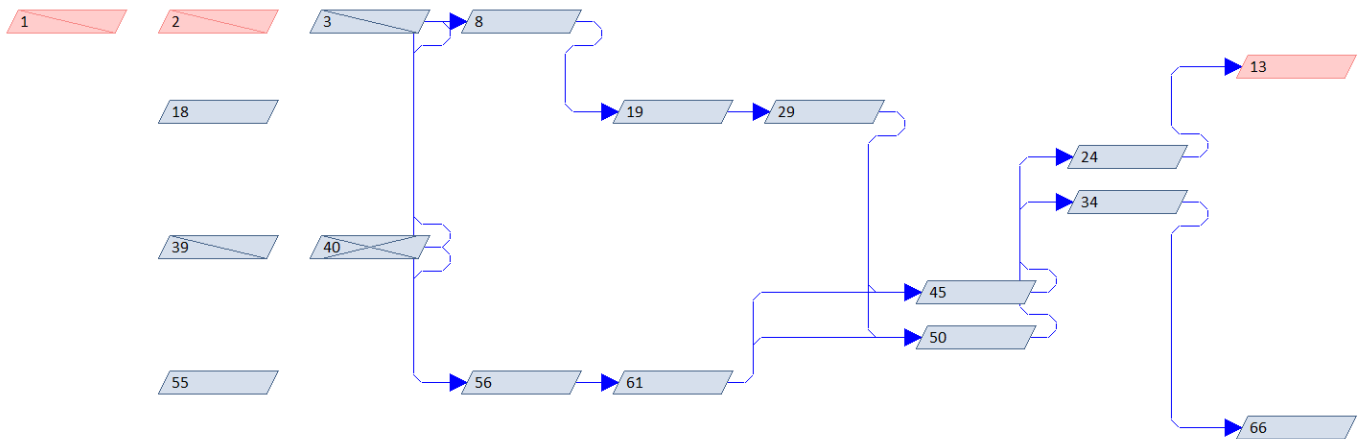
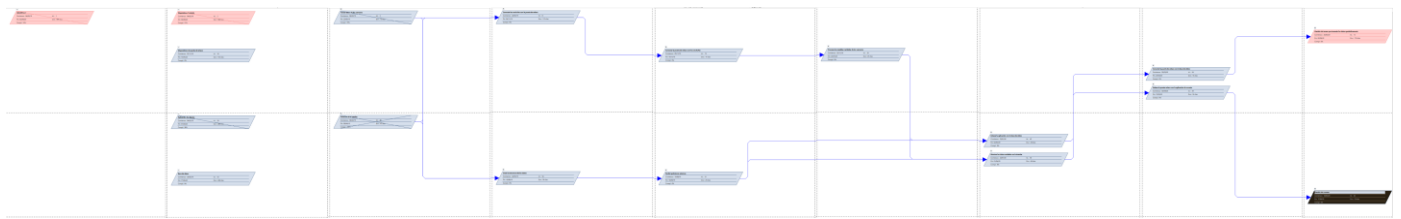
Se puede apreciar en el diagrama cómo la organización del proyecto se ha realizado a nivel de subtareas tras lo cual se completa un nuevo hito. Dentro de cada subtarea existen cuatro actividades entre las que habrá relaciones de dependencia. Es decir, se deberán realizar siempre en el mismo orden, planificación, diseño, programación y pruebas.



Las subtareas se ordenan de modo que se produzca un avance progresivo y coordinado de la funcionalidad. Primero se realizan las tareas con funcionalidad base. Poco a poco el proyecto irá creciendo encima del trabajo realizado previamente.

El diagrama muestra de forma clara la dependencia existente entre las tareas indicando cuáles son realizables de forma concurrente y cuáles deberán realizarse secuencialmente.

Esto es de gran ayuda para tomar decisiones como dónde ubicar a trabajadores que hayan finalizado una tarea. Habrá ocasiones que dos tareas concurrentes deban haber finalizado antes de poder comenzar una nueva. Esto sucede en los puntos tarea 3 y 40 deben finalizar antes que empezar 8 y 56, tarea 29 y 61 deben finalizar antes de empezar 45 y tareas 45 y 50 deben finalizar antes de comenzar 24 y 34. Si la diferencia de tiempo entre las tareas concurrentes predecesoras es muy elevada se toma la decisión de que aquellos trabajadores que hayan finalizado la tarea predecesora más breve ayuden a los trabajadores de la tarea predecesora más larga ya que no pueden comenzar ninguna de las tareas sucesoras. Realizando esto se reduce en gran medida la duración del proyecto y se aprovechan mejor los recursos. Se debe mencionar que si la duración de las tareas no es lo suficientemente distinta esto no merece la pena. En un caso realista un trabajador necesita un periodo de adaptación para poder comenzar una tarea nueva.



1. Llamathust.
2. Dispositivos enchufe.
18. Dispositivos puerta de enlace.
39. Aplicación de clientes.
55. Base de datos.
3. Tomar datos de los sensores.
40. Creación de la interfaz.
8. Conectar los enchufes con la puerta de enlace.
56. Crear la estructura de los datos.
19. Conectar la puerta de enlace con los enchufes.
61. Recibir peticiones externas.
29. Procesar las medidas recibidas de los sensores.
45. Enlazar la aplicación con la base de datos.
50. Mostrar los datos recibido en la interfaz.
24. Conectar la puerta de enlace con la base de datos.
34. Enlazar la puerta de enlace con la aplicación de usuario.
13. Gestión de tareas que tomarán los datos periódicamente.
66. Gestión de cuentas.

# SEGUIMIENTO Y CONTROL DEL PROYECTO

Es muy complicado que la planificación inicial de un proyecto se cumpla. Por lo general surgirán desviaciones que deberemos detectar lo antes posibles para responder ante ellas si fuera necesario.

Para poder hacerlo necesitamos partir de la programación del proyecto y comprar la con los resultados reales a cada momento. Dichos resultados podrán obtenerse a partir de evaluar el uso de los recursos en el tiempo.

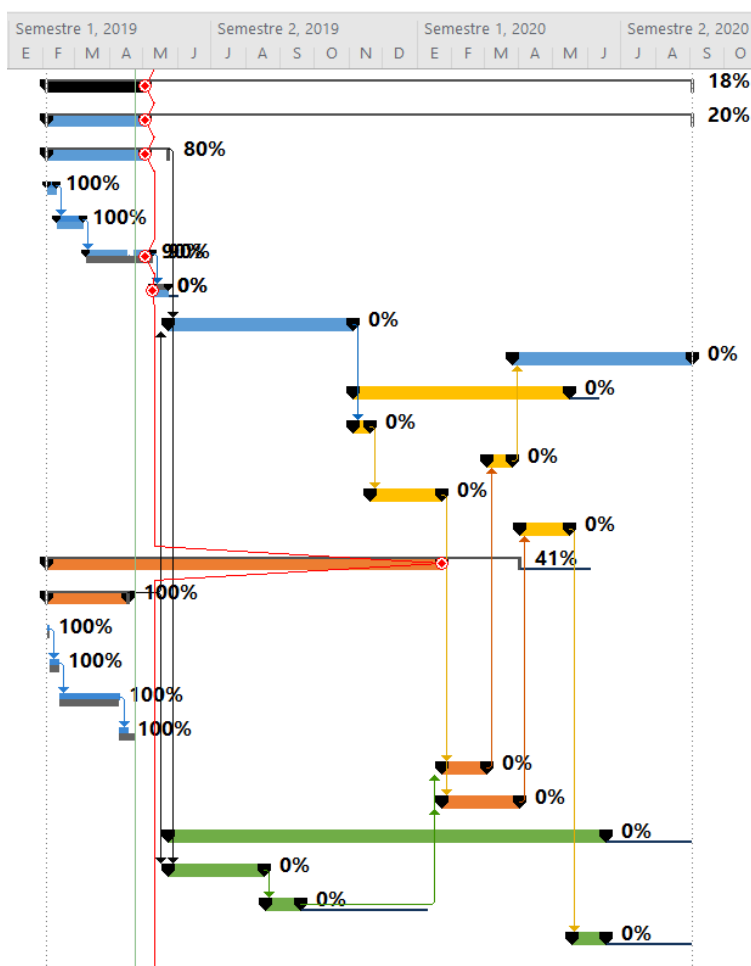
En caso de desviación de las mediciones realizadas sobre el proyecto con respecto a lo esperado en la planificación el proyecto deberá replanificarse o incluso renegociarse con los contratantes. Es muy importante poder proporcionar información del estado del proyecto al contratante y ser transparente respecto de la correcta o incorrecta evolución de este. Las desviaciones podrán manifestarse tanto en forma de retrasos como en forma de sobrecostos.

En seguimiento podrá ser tanto dinámico mediante reuniones y replanificaciones, estático mediante medidas del progreso y la productividad como preventivo mediante el análisis a futuros y la estimación.

Es importante conocer la situación real y actual del proyecto en todo momento, especialmente en los proyectos informáticos donde las desviaciones se pueden producir en el día a día y arrastrarse en el tiempo.

Es recomendable establecer fechas clave para el seguimiento y control del proyecto que en ocasiones coincidirán con las fechas previstas para los hitos y los entregables. En estas fechas se generarán informes para la dirección en las que se plasmará la situación real del proyecto y las decisiones de control tomadas para hacer que este se ajuste lo mejor posible a las previsiones.

Se exponen ahora las curvas de control más habituales realizadas para la fecha del 11 de mayo del 2019 habiendo el proyecto comenzado el 4 de febrero del 2019. En dicha fecha se habrá producido un progreso del 100% hasta el mes de marzo y un 80% en el mes de abril.



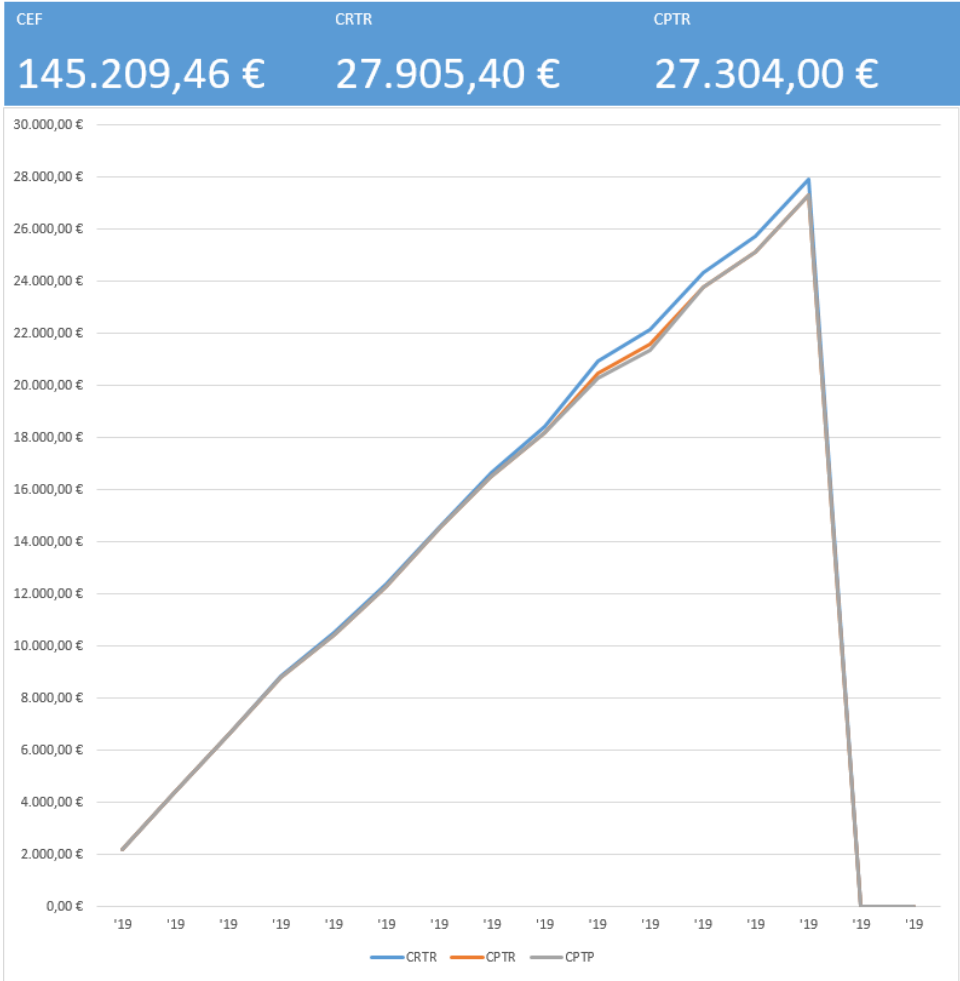
Tras haber actualizado el estado del proyecto con los nuevos porcentajes de completitud en las tareas hemos podido comprobar que se ha producido un retraso respecto de la programación inicial. Para corregir dicho retraso hemos tomado la decisión de corregirlo poniendo 12h extra a los trabajadores cuyas tareas se han retrasado (Analista y programador). Adicionalmente ya que esas tareas no estaban en el camino crítico se ha retrasado ligeramente su fecha de finalización para ser realistas con el progreso realizado.

# Curvas de Control: CPTP, CPTR y CRTR

Se ve que el coste del proyecto ha aumentado pues el CRTR (coste real en tiempo real) está por encima del CPTR (coste presupuestado en tiempo real).

Adicionalmente se ve que el proyecto también se retrasó durante un breve momento pues se aumentó la fecha del fin de las tareas. Se aprecia cuando el CPTP (gris) está por encima del CPTR (naranja), aunque en un pequeño periodo de tiempo la situación era de adelanto a la tarea puesto que CPTR (naranja) estaba por encima de CPTP (gris), pero rápidamente la línea gris supera a la naranja, reflejando el leve atraso en el proyecto.

Finalmente, las curvas vuelven a caer pues el proyecto no ha llegado todavía a ese tiempo (no ha sucedido por lo que no hay datos).

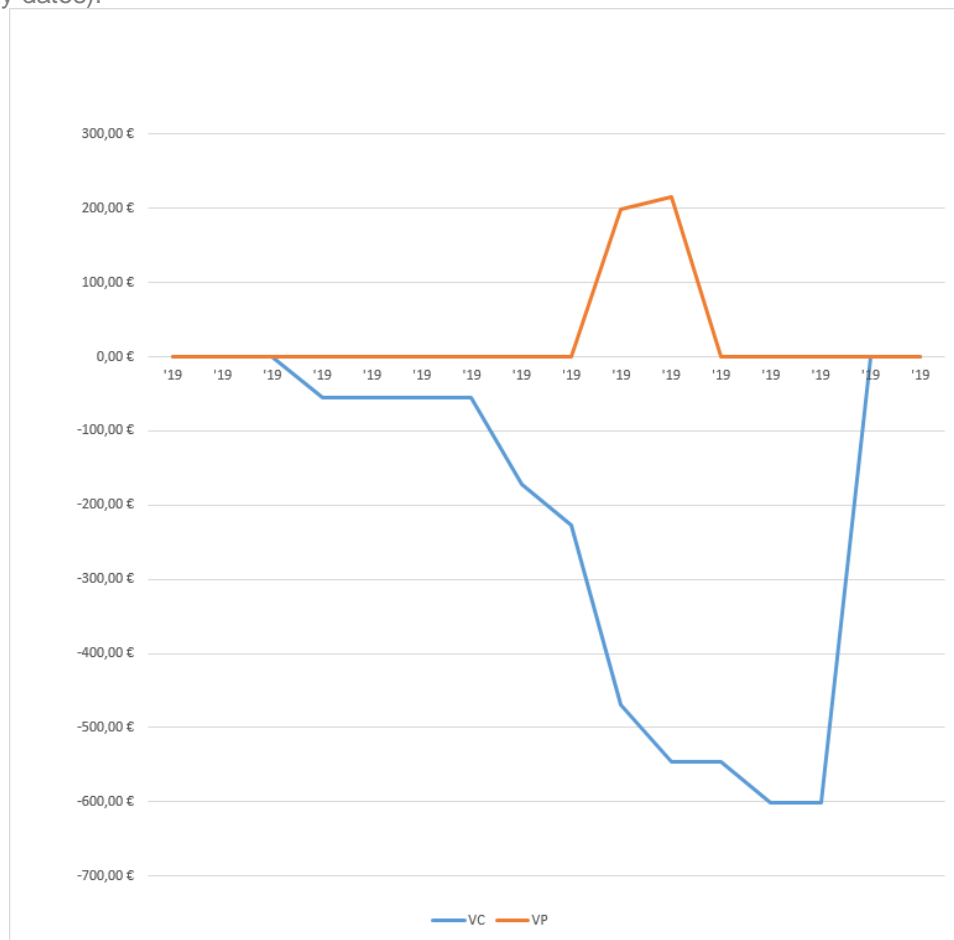


## Variación del Coste (VC) y Variación del Programa (VP)

Se puede ver en las gráficas que el coste del proyecto ha aumentado debido a la incorporación de las horas extra cuyo coste es mayor que la hora de trabajo normal.

Adicionalmente se ve que proyecto se ha retrasado ya que la duración de las tareas se ha ampliado.

Finalmente, ambas curvas vuelven a converger debido a que no formaban parte del camino crítico (no ha sucedido por lo que no hay datos).



# CONCLUSIONES

## Estimación de tiempos

Para determinar el tiempo de cada una de las tareas hemos partido de los datos de COCOMO que nos han servido de guía. No obstante, hemos podido comprobar que la duración total del proyecto en nuestro caso es superior a la estimada.

Esto se debe a dos factores. Nuestro equipo tiene solo cuatro trabajadores de modo que, aunque COCOMO para ciertas tareas nos recomiende tener más gente para nosotros eso no es posible. Es por tanto que el calendario debe durar más para esas tareas hasta que puedan ser realizadas por el personal disponible en cada momento.

Adicionalmente debido a la coordinación entre las tareas y la posibilidad de paralelizar algunas de ellas la duración del proyecto puede variar. En nuestro caso, aunque hay tareas paralelas ya que estamos limitados en gente la mayor parte del trabajo será secuencial. Además, las limitaciones de precedencia que son muy fuertes entre las tareas no nos permiten realizar ciertas paralizaciones.

En nuestro caso como se ha podido ver esto acaba produciendo que finalmente se produzca una elongación en la duración total del proyecto.

## Planificación y control

La planificación es una parte fundamental en todos los proyectos. Cuando un proyecto se divide en procesos resulta mucho más claro no solo de visualizar como conjunto para encontrar fallos de concepto y cuestiones en las que no se había pensado inicialmente. De forma adicional esta división resulta fundamental para poder organizar las tareas. Cuando las tareas se organizan como procesos es mucho más fácil pensar en su precedencia para de ese modo crear un esquema temporal en el que visualizar como se desarrollará el proyecto.

Del mismo modo resulta más fácil decidir los participantes que trabajarán para realizar cada proceso por lo que se acaba obteniendo un mecanismo para realizar la trazabilidad.

Haber realizado una buena planificación en una herramienta adecuada como es Project nos servirá de guía posteriormente para realizar el seguimiento y control del proyecto de modo que podremos comparar la planificación con el estado del proyecto según este se vaya realizando.

Podemos decir por tanto que la planificación nos aporta una visión general del proyecto en su estado inicial a partir de la cual podremos decidir cómo idealmente el proyecto debería de ser realizado.

Realizar un adecuado seguimiento y control en un proyecto tiene dos ventajas, permite mediar las desviaciones que han sucedido en un proyecto tanto en tiempo como dinero lo cual produce que se puedan tomar decisiones mientras que el proyecto se está realizando para corregir las desviaciones de forma precisa pues se conoce la magnitud de la desviación. Adicionalmente podremos informar sobre el estado del proyecto al cliente de modo que seremos más transparente ante él.

Es por tanto que un proyecto que haya sido planificado previamente y sobre el que se esté realizando un control de su ejecución tendrán más probabilidades de éxito solo por el hecho de que tendremos más información sobre el proyecto que si no lo hubiéramos realizado.