# LLAMATHUST

Planificación del Proyecto.

## DESCRIPCIÓN DE LOS PROCESOS DEL PROYECTO

##### PROCESOS

### 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 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.

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.

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.

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 del mismo. 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 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 contendores 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 del mismo, 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 acortar 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.

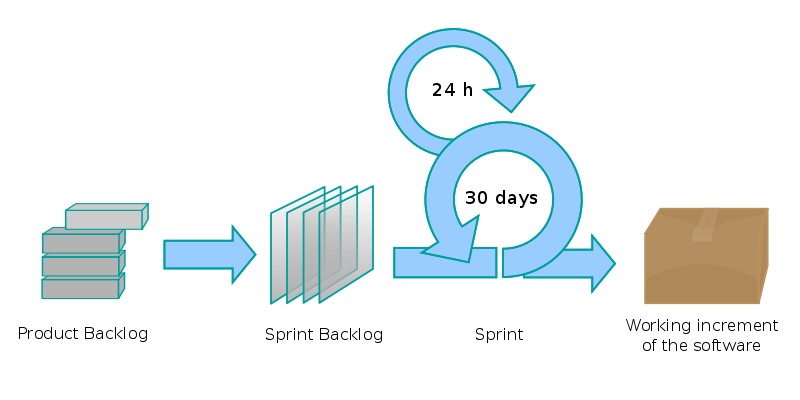
Almacenaje de la documentación

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 Scum.



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.

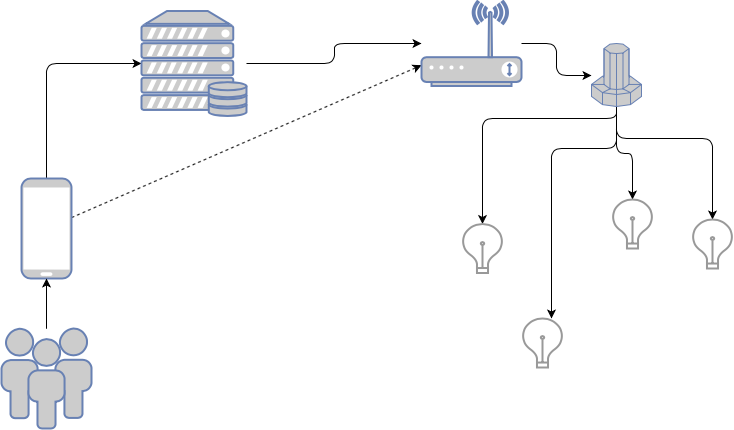
Las principales actividades que realizara cada miembro del equipo según su rol son las siguientes:

* 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.
* Analista: el analista es el encargado de realizar todo lo relativo a la captura de requisitos y el diseño del sistema a implantar.
* 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.
* Programador: el programador se encarga de las principales tareas de implantación de la aplicación con el desarrollo de todo el código necesario.

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 par te 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 mismo, 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.

##### 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.
    - Tiempo: 76 días.
  + Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 22 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.
    - Tiempo: 89 días.
  + Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 26 días.
* Gestión de tareas que tomarán los datos periódicamente: 35%
  + Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 8 días.
  + Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 21 días.
  + Programación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 89 días.
  + Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 26 días.

##### Dispositivos Puerta de Enlace

* Conectar la puerta de enlace con los enchufes 20%:
  + Planificación:
    - Personas:
    - Tiempo:
  + Diseño del producto:
    - Personas:
    - Tiempo: 6 días
  + Programación:
    - Personas: Técnico
    - Tiempo: 28 días
  + Integración y prueba:
    - Personas: Técnico
    - Tiempo: 8 días
* Conectar la puerta de enlace con la base de datos 15%:
  + Planificación:
    - Personas: Programador.
    - Tiempo: 1 día
  + Diseño del producto:
    - Personas: Programador.
    - Tiempo: 5 días
  + Programación:
    - Personas: Programador.
    - Tiempo: 21 días
  + Integración y prueba:
    - Personas: Programador.
    - Tiempo: 6 días
* Procesar las medidas recibidas de los sensores 35%:
  + Planificación:
    - Personas: Programador.
    - Tiempo: 4 días
  + Diseño del producto:
    - Personas: Programador.
    - Tiempo: 12 días
  + Programación:
    - Personas: Programador.
    - Tiempo: 48 días
  + Integración y prueba:
    - Personas: Programador.
    - Tiempo: 14 días
* Enlazar la puerta de enlace con la aplicación del usuario 30%:
  + Planificación:
    - Personas: Analista.
    - Tiempo: 2 días
  + Diseño del producto:
    - Personas: Analista.
    - Tiempo: 5 días
  + Programación:
    - Personas: Analista.
    - Tiempo: 21 días
  + Integración y prueba:
    - Personas: Analista.
    - Tiempo: 12 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: 29 días.
  + Integración y prueba:
    - Personas: Analista, Programador.
    - Tiempo: 8 días.
* Enlazar la aplicación con la base de datos 25%:
  + Planificación:
    - Personas:
    - Tiempo:
  + Diseño del producto:
    - Personas:
    - Tiempo:
  + Programación:
    - Personas:
    - Tiempo:
  + Integración y prueba:
    - Personas:
    - Tiempo:
* Mostrar los datos recibidos en la interfaz 35%:
  + Planificación:
    - Personas:
    - Tiempo:
  + Diseño del producto:
    - Personas:
    - Tiempo:
  + Programación:
    - Personas:
    - Tiempo:
  + Integración y prueba:
    - Personas:
    - Tiempo:

##### Base de Datos

* Crear la estructura de los datos 50%:
  + Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 2 días
  + Diseño del producto:
    - Personas: Técnico.
    - Tiempo: 6 días.
  + Programación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 26 días.
  + Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 7 días.
* Recibir peticiones externas 20%:
  + Planificación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 1 día
  + Diseño del producto:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 3 días
  + Programación:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 11 días
  + Integración y prueba:
    - Personas: Scrum Master, Técnico.
    - Tiempo: 3 días
* Gestión de cuentas 30%:
  + Planificación:
    - Personas:
    - Tiempo:
  + Diseño del producto:
    - Personas:
    - Tiempo:
  + Programación:
    - Personas:
    - Tiempo:
  + Integración y prueba:
    - Personas:
    - Tiempo:

## PROGRAMACIÓN DE LAS ACTIVIDADES DEL PROYECTO

Hemos dividido cada una de las tareas que componen el proyecto en un conjunto de subtareas. Cada subtarea 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)- 3
2. Crear la interfaz de la aplicación. (B)- 40
3. Conectar los enchufes con la puerta de enlace. (C)-8
4. Conectar la puerta de enlace con los enchufes. (C)-19
5. Procesar las medidas recibidas de los sensores. (C)- 29
6. Crear la estructura de los datos. (D)- 56
7. Recibir las peticiones externas. (D)- 61
8. Enlazar la aplicación con la base de datos. (E)-45
9. Conectar la puerta de enlace con la base de datos. (E)-24
10. Gestión de las tareas que tomarán los datos periódicamente. (E)-13
11. Mostrar los datos en la interfaz. (F)- 50
12. Enlazar la puerta de enlace con la aplicación de usuario. (F)- 34
13. Gestión de cuentas. (F)- 66

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 tareas 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 subtarea 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 si.

### ENTREGABLES DEL PROYECTO

Se producirá un entregable cada vez que una subtarea haya sido finalizada de modo que una parte de la funcionalidad del producto esté realizada por completo. Por ejemplo, haber conectado la aplicación con la base de datos será un nuevo entregable.

Se producirá un hito cada vez que una tarea haya sido finalizada, es decir, todas sus subtareas se hayan completado, un hito será haber completado por ejemplo la base de datos.

### ORGANIZATION BREAKDOWN STRUCTURE

Diagrama

### TAREAS CRÍTICAS

Después de haber hecho Project

### DIAGRAMA PERT / CPM

## SEGUIMIENTO Y CONTROL DEL PROYECTO

### Curvas de Control: CPTP, CPTR y CRTR

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

## CONCLUSIONES

## BIBLIOGRAFÍA