**DOCUMENTACIÓN TÉCNICA**

**TALLER DE PROYECTOS I**

**SISTEMAS ELECTRÓNICOS**

Logotipo, Icono, nombre de la empresa

Descripción generada automáticamente

**Datalogger for IoT**

*Autores*: *Tutor*:

Andrés Martín Jesús Hernández Mangas

Pablo Villacorta

Rubén Serrano

Óscar Martín

XX de xxxxx de 2023

**Índice**

**Índice de figuras**

# Capítulo 1

# 1. Resumen Cierre de Proyecto

## 1.1 Datos básicos del proyecto

**Nombre del proyecto**: Datalogger for IoT

**Tipo de proyecto**: Diseño electrónico

**Sector**: Telecomunicaciones

**Subsector**: Electrónica

**Ubicación**: Escuela Técnica Superior de Ingenieros de Telecomunicación

**Localidad**: Valladolid

**Dimensiones del proyecto**: Cuatrimestral

**Nombre de los responsables del proyecto**: Andrés Martín

**Nombre de los responsables del proyecto**: Pablo Villacorta

**Nombre de los responsables del proyecto**: Rubén Serrano

**Nombre de los responsables del proyecto**: Óscar Martín

**Email de los responsables del proyecto**: andres.martin@estudiantes.uva.es

**Email de los responsables del proyecto**: @estudiantes.uva.es

**Email de los responsables del proyecto**: @estudiantes.uva.es

**Email de los responsables del proyecto**: oscar.martin.casares@estudiantes.uva.es

## 1.2 Resultado económico

## 1.3 Informe de situación final

# Capítulo 2

# Informe Técnico Detallado

## 2.1 Objetivos del proyecto

Antes de comenzar con el diseño de un proyecto, es conveniente definir cuál es el objetivo que se persigue. En primera instancia, se busca desarrollar un sistema de comunicación inalámbrica que permita capturar diferentes magnitudes físicas del medio ambiente a través de diferentes sensores. Asimismo, se quiere que los datos capturados mediante los sensores sean transmitidos a una infraestructura de ordenadores (*cloud*) para poder procesarlos posteriormente.

La propuesta inicial para obtener el objetivo mencionado fue el desarrollo de una satélite de comunicaciones o CubeSat, como los utilizados por agencias espaciales NASA o ESA. Dichos satélites poseen unas dimensiones de alrededor de 10x10x10 cm tal y como podemos observar en la figura XXXX, se estima que desde que se lanzó el primero en el año 2003 hasta día de hoy, se encuentran en el espacio más de 3.200 CubeSats. (Referencia: Wikipedia en inglés: <https://en.wikipedia.org/wiki/CubeSat>).

[[[SE PODRÍA PONER MUCHA MÁS INFORMACIÓN SI SE QUISIERA]]]



Figura 1. CubeSat desarrollado en Noruega.

Sin embargo, debido a su alta complejidad y coste, es completamente inviable para el taller de proyectos.

En segunda instancia, surge otra idea de diseño consistente en el desarrollo de una radio sonda capaz de obtener diferentes magnitudes. Éstas son comúnmente utilizadas en meteorología, se usan en globos meteorológicos que usan gases de elevación con el objetivo de medir parámetros atmosféricos y transmitirlos a un receptor fijo. Entre los parámetros más importantes nos podemos encontrar la presión, la altitud, la temperatura o la humedad relativa. (Referencia: Wikipedia en español <https://es.wikipedia.org/wiki/Radiosonda>)

[[[SE PODRÍA PONER MUCHA MÁS INFORMACIÓN SI SE QUISIERA]]]

Al contrario del satélite miniatura, la opción de la radiosonda es más plausible, por ello buscaremos aproximarnos a esta idea. Sin embargo, dado que una radiosonda se puede poner en marcha sin necesidad de lanzarla a las alturas, nosotros simplemente la desplazaremos con la ayuda de un vehículo y obtendremos datos de temperatura, humedad, presión, monóxido de carbono, partículas que estén geolocalizadas y transmitiremos dicha información mediante radio.

## 2.2 Especificaciones del proyecto

En conocimiento del sistema que se quiere diseñar, es conveniente presentar las ideas y componentes generales necesarios para la herramienta, podemos observar estos en la figura 2. Entre los componentes principales nos encontramos con una antena, una CPU, una FPGA o matriz de puertas lógicas programable en campo, una memoria *flash*, los sensores y su correspondiente alimentación.

Antena 
Inalámbricas 
Depuración 
Programación 
USB 
Señal Sensor 1 
Acondicionamiento 
Señal Sensor 2 
Acondicionamiento 
Señal Sensor 3 
Acondicionamiento 
Actuachr 1 
Memoria 
Acondicionamiento 
Actuaú»r 2 
Acondicionamiento 
Actuador 3 
Sensor 1 
Sensor 2 
Sensor 3 
Actuador 1 
Actuador 2 
Actuador 3 

Figura 2. Esquema general del Datalogger.

Dicho sistema se dispondrá en diferentes placas de circuito impreso o PCBs, montadas unas sobre otras de la manera mostrada en la figura 3. Con este montaje conseguiremos que se puedan reusar o sustituir partes no funcionales de manera muy sencilla, además de poder así rediseñar nuevas partes en el futuro.

En la primera placa nos encontraremos con la alimentación y la programación incluida en la unidad central de procesamiento o CPU. Contendrá los conectores de la batería y USB, el microntrolador o un puerto USB – serie. En segundo lugar, tendremos otra placa que incluirá la FPGA, conectores de acceso a parte del patillaje de la FPGA o la memoria *flash.* La tercera placa estará destinada a la incorporación de los diferentes de sensores, así como sus circuitos de acondicionamiento necesarios. Por último, la PCB restante contendrá las comunicaciones, es decir, el módulo de comunicaciones inalámbricas WiFi y el transceptor LoRa.

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Figura 3. Disposición de las placas de circuito impreso.

## 2.3 Planificación

Para poder gestionar el proyecto, existen diferentes herramientas de planificación y control. Una de dichas herramientas es el Diagrama de Gantt, la cual hemos utilizado para planificar nuestro trabajo y que nos proporciona una vista general de las tareas programadas además de cuáles han de completarse y en qué fecha.

En primer lugar, se debe definir la estructura de descomposición del trabajo en la cual deben listarse las tareas y proporcionarlas una estimación del tiempo necesario para completar cada una de ellas, uno de los cometidos más complicados. En los anexos Ay B [[[FALTA INCLUIR ANEXOS (GANTT)]]] se pude observar que la planificación inicial dista de la final. Al comienzo de un proyecto es muy complicado anticipar los problemas que van a surgir y más cuando se carece de experiencia en ello.

[[[EXPLICAR LAS DIFERENCIAS QUE HABRÁ ENTRE AMBOS DIAGRAMAS]]]

## 2.4 Herramienta de control de versiones

En el mundo profesional cada vez es más común el trabajo colaborativo y por ende las herramientas destinadas a ello. Es importante hoy en día estar familiarizado con sistemas de control versiones, la cuales permiten almacenar todas las versiones que ha habido durante la realización del proyecto, permitiéndonos además acceder a versiones antiguas en cualquier momento, actuando así a modo de ‘copias de seguridad’.

Para la realización de este proyecto se ha utilizado la mundialmente herramienta conocida como GitHub, disponible nuestro desarrollo en <https://github.com/pvillacorta/TP1SE>. Dicho repositorio ha sido organizado en distintas subcarpetas creadas sobre la marcha según las necesidades, con el objetivo de poder encontrar todo de manera ordenada e intuitiva, tal y como podemos observar en la figura 4. [[[ACTUALIZAR FIGURA]]]

En éste además hemos ido añadiendo en el fichero README.md todos los comentarios o instrucciones de utilidad necesarias durante todo el proyecto, para poder así recordar en cualquier momento cómo hicimos determinada acción o cosas de suma importancia.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 4. Esquema del repositorio de GitHub

Para manejarnos con la herramienta se han hecho uso de comandos básicos, entre los cuales nos podemos encontrar con:

* git clone ’url’: con este comando se ha podido clonar el repositorio en los respectivos ordenadores para poder trabajar de forma más cómoda
* git add \*: comando utilizado para añadir cambios de archivos en el directorio clonado.
* git commit -m ’nombre del commit’: Este comando toma todos los cambios de archivos descritos por el ’git add’, crea un nuevo objeto de confirmación y se establece una rama para que apunte a esta nueva confirmación.
* git push: comando que envía todos los objetos modificados localmente el respositorio remoto. Con el uso de este comando se ha podido ir subiendo las nuevas actualizaciones.
* git pull: comando utilizado para actualizar el repositorio local.
* git status: este comando muestra el estado de los archivos que hay en el repositorio remoto en comparación con los del directorio de trabajo. Esto ha servido para verificar que en todo momento el repositorio local se encontraba en buen estado sin errores.

[[[SE PODRÍAN INCORPORAR ESTADÍSTICAS DE GITHUB SI QUISIERAMOS]]]

## 2.5 Diseño electrónico y esquemático