

# Plan Driven

## Planificación y Diseño de Sistemas Computacionales

Escuela Ingeniería Informática de Valladolid  
Universidad de Valladolid

19 de enero de 2020



Universidad de Valladolid

Valdunciel Sánchez, Pablo

# Índice

<b>1. Introducción</b>	<b>4</b>
1.1. Metodología de planificación . . . . .	4
<b>2. Especificación de requisitos</b>	<b>5</b>
2.1. Requisitos funcionales . . . . .	5
2.1.1. Requisitos de información . . . . .	5
2.1.2. Requisitos de interacción . . . . .	5
2.2. Requisitos no funcionales . . . . .	6
2.3. Restricciones . . . . .	6
<b>3. Descripción del entorno y tecnología a utilizar</b>	<b>7</b>
<b>4. Arquitectura del sistema</b>	<b>7</b>
<b>5. Productos y tareas del proyecto</b>	<b>8</b>
5.1. Estructura de descomposición del producto (PBS) . . . . .	8
5.2. Estructura de descomposición de tareas (WBS) . . . . .	8
<b>6. Esfuerzo del proyecto</b>	<b>11</b>
6.1. Medidas de tiempo . . . . .	11
6.2. Cálculo de los Puntos de Función . . . . .	11
6.2.1. Identificación de componentes y cálculo del grado de complejidad . . . . .	11
6.2.2. Cálculo de los Puntos de Función en Bruto (UFC) . . . . .	12
6.2.3. Cálculo de los Puntos de Función Finales (FPC) . . . . .	12
6.2.4. Cálculo del tiempo en días de esfuerzo . . . . .	14
6.2.5. Comparación con el esfuerzo real del proyecto . . . . .	14
<b>7. Plan de tareas</b>	<b>15</b>
7.1. Estimación del esfuerzo de las actividades . . . . .	15
7.2. Diagrama de actividades . . . . .	15
7.3. Diagrama de Gannt. Cálculo del camino crítico del proyecto . . . . .	15
<b>8. Anexo I - <i>JSON schema</i> para los ficheros del sistema</b>	<b>20</b>

## Índice de figuras

1.	Estructura de descomposición del producto (PBS) . . . . .	9
2.	Estructura de descomposición de tareas (WBS) . . . . .	10
3.	Diagrama de actividades . . . . .	17
4.	Diagrama de Gantt . . . . .	18
5.	Asignación de tareas . . . . .	19

## Índice de cuadros

1.	Productos generados por cada actividad planificada . . . . .	8
2.	Tabla de complejidad EI, EO, EQ . . . . .	11
3.	Tabla de complejidad ILF y ELF . . . . .	11
4.	EI - Identificación y cálculo de la complejidad . . . . .	12
5.	EO - Identificación y cálculo de la complejidad . . . . .	12
6.	EQ - Identificación y cálculo de la complejidad . . . . .	12
7.	ILF - Identificación y cálculo de la complejidad . . . . .	12
8.	Cálculo de los Puntos de Función en Bruto (UFC) . . . . .	13
9.	Evaluación de las Características Generales del Sistema (GSC) . . . . .	13
10.	Cálculo de los Puntos de Función Finales (FPC) . . . . .	13
11.	Equivalencia de los FPC del proyecto . . . . .	13
12.	Tiempo estimado y gastado en los diferentes hitos del proyecto realizado . . . . .	14
13.	Tiempo estimado y gastado en las etapas planificadas - Modelo en cascada . . . . .	14
14.	Estimación del esfuerzo de cada actividad . . . . .	15

# 1. Introducción

En el presente documento se desarrolla la planificación *plan-driven* del sistema de simulación de la Ley Electoral española actual requerido en la asignatura *Planificación y Diseño de Sistemas Computacionales (PDSC)*.

## 1.1. Metodología de planificación

La planificación del proyecto se basará en el modelo conocido como *Waterfall model*, o **Modelo en Cascada** en el que se distinguirán las etapas habituales citadas a continuación.

1. **Recogida de requisitos:** determinar las características y cualidades requeridas por los usuarios o clientes.
2. **Análisis de requisitos:** convertir los requisitos en equivalentes que los desarrolladores puedan entender.
3. **Diseño:** especificación de la arquitectura del sistema.
4. **Codificación:** desarrollo del software y test unitarios.
5. **Pruebas** (*Testing*): test de integración, debugging.
6. **Operaciones:** despliegue, instalación, soporte y mantenimiento.

La utilización de el Modelo en Cascada impondrá una serie de hitos que serán los siguientes:

1. Requisitos
2. Especificación de sistema.
3. Diseño del sistema.
4. Código
5. Código integrado y probado
6. Sistema completo y operativo

La elección de esta metodología de planificación tiene, en primer lugar, el objetivo de evitar rehacer actividades, limitando la iteración a lo estrictamente necesario para solucionar errores. En segundo lugar, aplicando este modelo de planificación se pretende conseguir una predicción lo más precisa posible de los tiempos de completitud del proyecto, estableciendo así el máximo control sobre el desarrollo del mismo [1].

Dado que se partía de un documento donde se recogían la funcionalidad del sistema, la etapa de *Recogida de requisitos* no se incluirá en la planificación. Por otro lado, dado la etapa de *Operaciones* tampoco se incluirá en la planificación al no ser necesario el despliegue del sistema ni las actividades que se derivan del mismo.

## 2. Especificación de requisitos

### 2.1. Requisitos funcionales

El sistema debe permitir...

- **[RF-01]** identificar de forma única a cada dispositivo de usuario utilizado para acceder al sistema.
- **[RF-02]** introducir los datos de la configuración (*RI-01*), la elección (*RI-02*), las circunscripciones (*RI-03*) y las candidaturas (*RI-04*) mediante una interfaz gráfica (GUI).
- **[RF-03]** introducir los datos de la configuración (*RI-01*), la elección (*RI-02*), las circunscripciones (*RI-03*) y las candidaturas (*RI-04*) mediante la subida de un archivo.
- **[RF-04]** visualizar la configuración por defecto de cada dispositivo de usuario. Inicialmente la configuración por defecto de un dispositivo de usuario se corresponde con un *umbral mínimo de representación* del **3 %** (0,03) (ver *RI-01*).
- **[RF-05]** visualizar los datos de la configuración (*RI-01*), la elección (*RI-02*), las circunscripciones (*RI-03*) y las candidaturas (*RI-04*) una vez han sido introducidos.
- **[RF-06]** modificar la configuración por defecto.
- **[RF-07]** modificar los datos de la configuración (*RI-01*), la elección (*RI-02*), las circunscripciones (*RI-03*) y las candidaturas (*RI-04*) una vez han sido introducidos.
- **[RF-08]** exportar en un fichero los datos de la configuración (*RI-01*), la elección (*RI-02*), las circunscripciones (*RI-03*) y las candidaturas (*RI-04*) una vez han sido introducidos.
- **[RF-09]** visualizar el historial de elecciones calculadas por cada dispositivo de usuario.
- **[RF-10]** una visualización del reparto de escaños en forma de tabla y del resultado final.

#### 2.1.1. Requisitos de información

El sistema deberá almacenar la información sobre:

- **[RI-01] Configuración:** umbral mínimo de representación.
- **[RI-02] Elección:** tipo de elección (municipal, autonómica, congreso, senado), fecha.
- **[RI-03] Circunscripción:** nombre, número de representantes (escaños), número de censados, número de votos en blanco, número de votos nulos.
- **[RI-04] Candidatura:** nombre largo, nombre abreviado, votos obtenidos.
- **[RI-05] Dispositivo de usuario:** fecha del primer acceso, fecha del último acceso.

#### 2.1.2. Requisitos de interacción

Los casos de uso del sistema son:

1. CU1 - Obtener el reparto de escaños
2. CU2 - Introducir los datos de unas elecciones
  - a) CU2.A - Introducir los datos de unas elecciones mediante GUI
  - b) CU2.B - Introducir los datos de unas elecciones mediante la subida de un archivo
3. CU3 - Modificar los parámetros de la configuración por defecto
4. CU4 - Modificar los datos de unas elecciones (una vez introducidos)
5. CU5 - Exportar los datos de unas elecciones
6. CU6 - Ver la distribución de escaños de unas elecciones del historial

## 2.2. Requisitos no funcionales

El sistema deberá...

- [NF-01] soportar varios usuarios simultáneos realizando operaciones.
- [NF-02] tener una cobertura de test de por lo menos el 75 % del código.
- [NF-03] responder en menos de 10 segundos a cualquier petición.
- [NF-04] ser capaz de manejar más de 25 peticiones correctamente por minuto.

## 2.3. Restricciones

Se deberán verificar las siguientes restricciones:

- [RT-01] La asignación de escaños en cada circunscripción se realiza de acuerdo con el Sistema D'Hont, explicado en el artículo 163 de la [Ley Orgánica 5/1985, de 19 de junio, del Régimen Electoral General](#).
- [RT-02] El *umbral mínimo de representación* de la **Configuración** (*RI-01*) es un número real comprendido entre 0.0 y 1.0.
- [RT-03] El *nombre* de una **Circunscripción** (*RI-03*) contiene al menos una letra, mayúscula (A-Z) o minúscula (a-z), como primer carácter y está formado únicamente por letras, mayúsculas (A-Z) o minúsculas (a-z), con una longitud máxima de 120 caracteres.
- [RT-04] El *número de censados* de una **Circunscripción** (*RI-03*) es un número entero positivo.
- [RT-05] El *número de representantes (escaños)* de una **Circunscripción** (*RI-03*) es un número entero positivo y menor que el *número de censados*.
- [RT-06] El *número de votos en blanco* de una **Circunscripción** (*RI-03*) es un número entero mayor o igual que 0.
- [RT-07] El *número de votos nulos* de una **Circunscripción** (*RI-03*) es un número entero mayor o igual que 0.
- [RT-08] La suma del *número de votos en blanco* y el *número de votos nulos* es menor o igual que el *número de censados* de una **Circunscripción** (*RI-03*).
- [RT-09] El *nombre largo* de una **Candidatura** (*RI-04*) contiene al menos una letra, mayúscula (A-Z) o minúscula (a-z), como primer carácter y está formado únicamente por letras, mayúsculas (A-Z) o minúsculas (a-z), dígitos (0-9), guión (-) o guión bajo (\_) con una longitud máxima de 70 caracteres.
- [RT-10] El *nombre abreviado* de una **Candidatura** (*RI-04*) contiene al menos una letra, mayúscula (A-Z) o minúscula (a-z), como primer carácter y está formado únicamente por letras, mayúsculas (A-Z) o minúsculas (a-z), dígitos (0-9), guión (-) o guión bajo (\_) con una longitud máxima de 6 caracteres.
- [RT-11] El *número de votos obtenidos* de una **Candidatura** (*RI-04*) es un número entero mayor o igual que 0.
- [RT-12] La suma del *número de votos obtenidos* de todas las **Candidaturas** (*RI-04*) de una misma Circunscripción es menor o igual que (*número de censados* - (*número votos nulos* + *número de votos en blanco*)) de dicha **Circunscripción** (*RI-03*).
- [RT-13] El fichero de entrada deberá atenerse a la siguiente especificación (JSON Schema). Ver *Anexo I*.

### 3. Descripción del entorno y tecnología a utilizar

La aplicación descrita en este documento se implementará como una aplicación web. Los motivos son, entre otros:

- familiaridad con el diseño de interfaces web.
- gran documentación online disponible .
- mejor accesibilidad para el usuario, que no necesitará instalar nada.

La aplicación será implementada en el lenguaje de programación **Python**, en su última versión estable al momento de escribir, **3.8**. Para el desarrollo de la aplicación se utilizará el framework **Django**, en su última versión estable, 2.2, y como base de datos relacional, **SQLite**, la cual está incorporada con **Python**. Para la validación del fichero de entrada se usará **JSON Schema**.

Para desarrollar el *frontend* no se usará ninguna librería específica. Se utilizarán los lenguajes **HTML**, **CSS** y **JavaScript** estándar según lo requieran las diferentes pantallas.

El proyecto se ejecutará en contenedores **Docker**. Para facilitar la tarea, se usará **Docker Compose**, un conjunto de scripts de Python que permiten trabajar con varios contenedores a la vez de forma cómoda y de modo que se comuniquen entre sí. Habrá dos contenedores principales, cada uno de los cuales contendrá:

1. La aplicación *Django*.
2. El servidor web *nginx* que hará de proxy inverso y servirá el contenido estático de la web.

Para realizar los tests unitarios se usará **PyTest**. Para comprobar la calidad del código se usará **Flake8**. Para la cobertura se usará la herramienta **coverage.py** . Para las pruebas de rendimiento se usará **Apache JMeter**.

### 4. Arquitectura del sistema

El sistema será desarrollado utilizando el *framework* de desarrollo de aplicaciones web Django. La utilización de este *framework* determina en gran parte la arquitectura del sistema. Django sigue un patrón arquitectónico *Front Controller*. Este patrón arquitectónico implica la existencia de un único controlador que maneja todas las peticiones al sitio web, y que se divide en dos partes:

1. **Handler**: recibe las peticiones del servidor web, extrae la información de la URL y la petición, y determina la acción a realizar. Es el propio *framework* el que realiza el trabajo de *Handler* y proporciona un comando abstracto, la clase *View*, para poder implementar comandos concretos.
2. **Comandos concretos**: realizan las acciones, actúan sobre los modelos y deciden las vistas. En Django, estos comandos concretos se denominan *views*.

El patrón arquitectónico *Front Controller* utilizado por Django da lugar a una arquitectura en la que se diferencian, al menos, los siguientes módulos:

- **templates**: ficheros HTML y DTL(Django Template Language), donde se presentan los datos al usuario.
- **views**: son la comandos concretos, que junto con el propio *framework* realizan las tareas propias de un controlador.
- **models**: modelos de datos en los que se encapsulan los datos y la funcionalidad

A mayores, la arquitectura de nuestro sistema contará con los siguientes módulos:

- **styles**: ficheros CSS que modifican la apariencia de los *templates*.
- **scripts**: ficheros JS que proporcionan un comportamiento dinámico a los *templates*.
- **services**: encapsulan la funcionalidad de la aplicación, haciendo que el módulo *models* se limite a albergar los datos.

## 5. Productos y tareas del proyecto

### 5.1. Estructura de descomposición del producto (PBS)

En la *Figura 1* podemos observar la *Estructura de descomposición del producto (PBS)* del sistema a desarrollar. Los productos *tangibles* o *entregables*, los cuales se sitúan en los nodos hoja, aparecen coloreados en amarillo.

### 5.2. Estructura de descomposición de tareas (WBS)

Si bien es cierto que la *PBS* suele ir acompañada de un *Diagrama de Flujo del Producto (PDF)*, incluiremos en este caso la *Estructura de descomposición de tareas (WBS)*, ya que el orden de generación de los productos a través de tareas podrá verse con todo detalle en el **Plan de tareas** (ver *Sección 7*).

El enfoque utilizado para construir la *WBS* ha sido el enfoque basado en actividades, dividiendo la estructura de tareas en las principales fases del ciclo de vida del proyecto. Dado que se trata de un proyecto pequeño y las tareas serán, en general, tareas abordables por una sola persona, se ha construido una *estructura de descomposición de tareas (WBS)* con únicamente 2 niveles de profundidad. Aumentar la profundidad de esta estructura no mejoraría el control del proyecto, sino que lo empeoraría ya existirían un gran número de tareas muy pequeñas, lo que dificultaría posteriormente tanto la creación como la interpretación del plan de tareas.

La *Estructura de Descomposición de Tareas* de se puede ver en la *Figura 2*. En esta figura sí se incluyen las etapas de *Recogida de requisitos* y *Operaciones* para tener una visión completa de todas las actividades que sería necesario desarrollar si el desarrollo de este sistema tuviera como última instancia la distribución del software construido.

En el *Cuadro 1* se puede ver la correspondencia entre los productos que aparecen en el *PBS* y las actividades del *WBS* que los generan.

<b>Etapas del Modelo en Cascada</b>	<b>Tarea</b>	<b>Producto(s) generado</b>
Análisis de requisitos	Especificación de los requisitos del sistema	Especificación del sistema
	Modelos de análisis	
Diseño	Diseño BD	Data Model
	Diseño arquitectónico	Modules styles.
	Diseño de la interacción	Diagramas de secuencia.
	Diseño GUI	Bocetos GUI
Codificación	Preparación del entorno	1. Configuración Docker. 2. Configuración del Servidor Web
	Codificar / Probar modelos	1. Código Py modelos. 2. Test unitarios modelos.
	Codificar templates	HTML + JS Scripts.
	Codificar estilos	Estilos CSS
	Codificar / probar controladores	1. Código Py views. 2. Test unitarios views.
	Codificar /probar servicios	1. Código Py servicios. 2. Test unitarios servicios
Puebas	Pruebas de integración del sistema	Test de integración
	Pruebas de calidad, cobertura y rendimiento	1. Informes de cobertura, calidad y rendimiento de software. 2. Software probado e integrado

Cuadro 1: Productos generados por cada actividad planificada



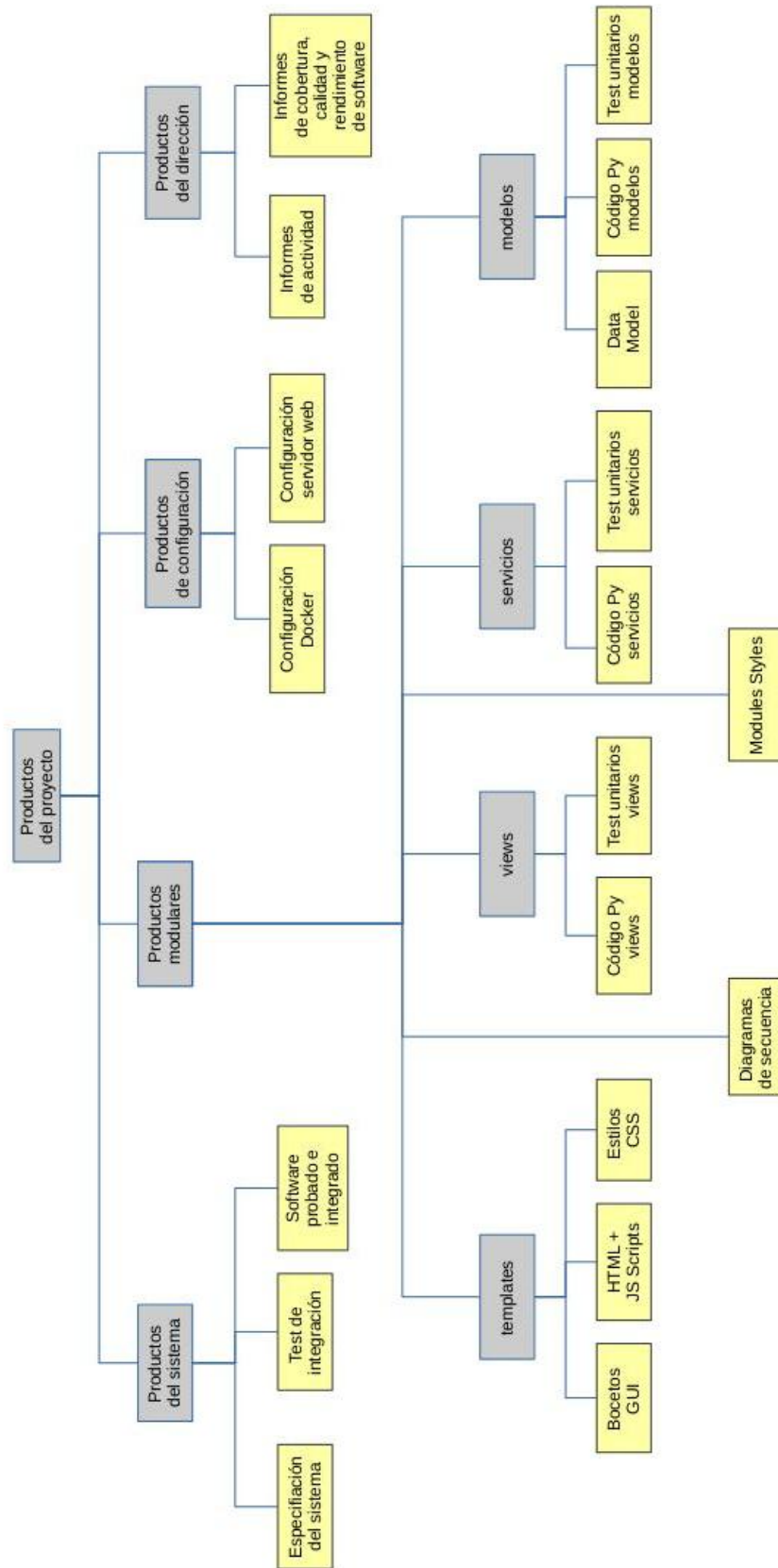


Figura 1: Estructura de descomposición del producto (PBS)

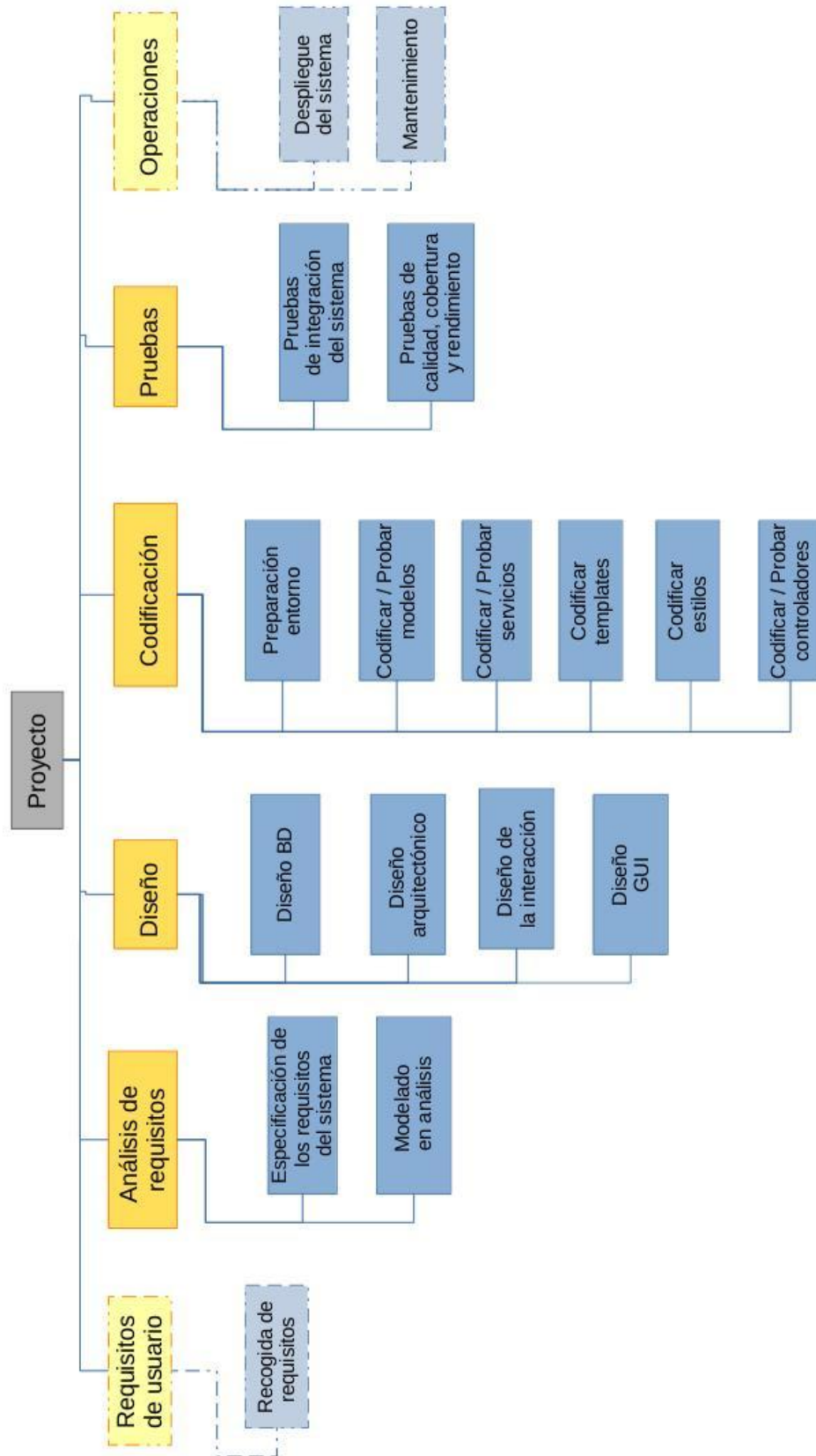


Figura 2: Estructura de descomposición de tareas (WBS)

## 6. Esfuerzo del proyecto

### 6.1. Medidas de tiempo

Para medir el esfuerzo del proyecto se han utilizado las siguientes unidades:

- Meses (mo, *months*)
- Semanas (w, *weeks*)
- Días (d, *days*)
- Horas (h, *hours*)
- Minutos (min, *minutes*)

Las equivalencias que se han establecido entre estas unidades son las que se utilizan por defecto en las aplicaciones de planificación o seguimiento de un proyecto, en nuestro caso la plataforma *GitLab*[2]. Las equivalencias entre las unidades son:

- 1mo = 4w: 1 mes de esfuerzo es equivalente a 4 semanas de esfuerzo.
- 1w = 5d: 1 semana ( es equivalente a 5 días.
- 1d = 8h: 1 día es equivalente a 8 horas .

### 6.2. Cálculo de los Puntos de Función

Para realizar el cálculo de los Puntos de Función del sistema se ha utilizado el método de Albrecht [3]. A continuación se desarrollan las diferentes etapas realizadas para calcular los Puntos de Función.

#### 6.2.1. Identificación de componentes y cálculo del grado de complejidad

En esta sección se identifican los diferentes componentes principales presentes en el sistema ya desarrollado. La complejidad de los componentes EI, EO y EQ se calculará en base lo estipulado en el *Cuadro 2*, y la complejidad de los componentes ILF y ELF se calculará en base a lo estipulado en el *Cuadro 3*.

FTR	DET		
	1-5	6-15	>15
0-1	Low	Low	Average
0-3	Low	Average	High
>3	Average	High	High

*FTR*: número de archivos actualizados o referenciados

*DET*: número de datos elementales reconocibles por el usuario

RET	DET		
	1-5	6-15	>15
0-1	Low	Low	Average
2-5	Low	Average	High
>5	Average	High	High

*RET*: número de elementos de dato reconocibles por un usuario

*DET*: número de campos reconocibles por el usuario

**EI - External inputs**

n	Descripción	FTR	DET	Complejidad
1	Introducir los datos de unas elecciones mediante la subida de un fichero <i>json</i>	1	>15	Avarage
2	Introducir los datos de unas elecciones mediante la interfaz gráfica	0	>15	Avarage
3	Introducir una nueva configuración por defecto	1	1	Low

Cuadro 4: EI - Identificación y cálculo de la complejidad

**EO - External outputs**

n	Descripción	Nº operaciones	FTR	DET	Complejidad
1	Mostrar el historial de elecciones asociado al dispositivo	x 1	1	3	Low

Cuadro 5: EO - Identificación y cálculo de la complejidad

**EQ - External queries**

n	Descripción	Nº operaciones salida	FTR	DET	Complejidad
1	Mostrar los resultados de una elección	x 1	3	>15	High
2	Exportar los datos de unas elecciones en un fichero <i>json</i>	x 1	0	>15	Average
3	Volver a la pantalla de inicio	x 1	0	>15	Average
4	Ir a la pantalla con la configuración	x 1	0	1	Low

Cuadro 6: EQ - Identificación y cálculo de la complejidad

**ILF - Internal logical files**

Desde el punto de vista del usuario final, se pueden identificar los siguientes grupos de datos identificables a nivel lógico y que según éste están dentro del sistema:

n	Descripción	RET	DET	Complejidad
1	Configuración por defecto	1	1	Low
2	Historial de elecciones	>5	3	High

Cuadro 7: ILF - Identificación y cálculo de la complejidad

**ELF - External logical files**

No se ha identificado ningún grupo de datos que se agrupe lógicamente desde el punto de vista del usuario final y que residan externamente fuera del sistema.

**6.2.2. Cálculo de los Puntos de Función en Bruto (UFC)**

En segundo lugar, se calculan los Puntos de Función en Bruto tal y como se especifica en el *Cuadro 8*.

**6.2.3. Cálculo de los Puntos de Función Finales (FPC)**

Una vez evaluadas las Características Generales del Sistema (GSC) (ver *Cuadro 9*) se aplica la fórmula

$$FPC = UFC * (0,65 + 0,01 \sum GSC_i)$$

y se obtienen los Puntos de Función Finales (FPC) (ver *Cuadro 10*).

Complejidad				Total	
Tipo	Low	Average	High	Fórmula	UFC
EI	1	2	0	$I*3 + a*4 + h*6$	11
EO	1	0	0	$I*4 + a*5 + h*6$	4
EQ	1	2	1	$I*3 + a*4 + h*5$	16
ILF	1	0	1	$I*7 + a*10 + h*15$	22
ELF	0	0	0	$I*5 + a*7 + h*10$	0

Cuadro 8: Cálculo de los Puntos de Función en Bruto (UFC)

Factor	Puntuación	Justificación
GSC 01	2	Salida remota y entrada remota de datos.
GSC 02	1	Ejecución de código en el cliente web.
GSC 03	0	No hay requerimientos especiales de rendimiento.
GSC 04	0	Sin restricciones de operaciones declaradas.
GSC 05	2	El 50 % de las transacciones se pueden ver afectadas por picos de tráfico.
GSC 06	5	Más del 30 % de las funciones son entradas interactivas de datos.
GSC 07	2	Se incluyen las siguientes 4 tareas de diseño para consideraciones especiales del usuario: scrolling, ratón, pantallas reducidas, copia impresa de las transacciones en línea.
GSC 08	3	Actualización en línea de ficheros importantes internos.
GSC 09	0	
GSC 10	0	No hay que reutilizar código.
GSC 11	0	El usuario no ha declarado consideraciones especiales para instalación y conversión.
GSC 12	1	Proceso específico de arranque: mediante la ejecución de un contenedor Docker.
GSC 13	0	No hay requerimientos de usuario para más de un lugar.
GSC 14	2	Se prevé una flexible capacidad de peticiones para modificaciones medias.
<b>TOTAL</b>	18	

Cuadro 9: Evaluación de las Características Generales del Sistema (GSC)

Tipo	Complejidad			UFC	FPC
	Low	Average	High		
EI	1	2	0	11	9,13
EO	1	0	0	4	3,32
EQ	1	2	1	16	13,28
ILF	1	0	1	22	18,26
ELF	0	0	0	0	0
<b>Total</b>				53	43,99

Cuadro 10: Cálculo de los Puntos de Función Finales (FPC)

Puntos de Función	Días de esfuerzo	Horas de esfuerzo	Esfuerzo <w d h min>
41,99	41,99	335,92	8w 1d 7h 55 min

Cuadro 11: Equivalencia de los FPC del proyecto

#### 6.2.4. Cálculo del tiempo en días de esfuerzo

Para calcular el tiempo en días de esfuerzo, utilizamos la estimación 1 Punto de Función = 1 día de esfuerzo. A partir de los 43,99 FPC se pueden obtener las equivalencias presentes en el *Cuadro 11*. Como podemos ver, los 43,99 Puntos de Función Finales son equivalentes a 8 semanas, 1 día, 7 horas y 55 minutos de esfuerzo.

#### 6.2.5. Comparación con el esfuerzo real del proyecto

La estimación de esfuerzo obtenida se encuentra muy alejada de lo reflejado en el seguimiento del proyecto *GitLab* ([gitlab.inf.uva.es/pdsc-19-20-02-pabvald/pydsc-19-29-02](https://gitlab.inf.uva.es/pdsc-19-20-02-pabvald/pydsc-19-29-02)), en el cuál este es el esfuerzo que se ha reflejado en los diferentes hitos (ver *Cuadro 12*). Es necesario aclarar que en el **Hito 5** se han incluido aquellas tareas incompletas o pendientes y que realizarán de aquí a la defensa de la práctica. Como se puede

ver, la diferencia entre el tiempo estimado y el tiempo gastado es sobre todo relevante en las tareas agrupadas dentro del hito *Modelos de análisis*, siendo el tiempo gastado casi el doble del tiempo estimado. Si sumamos los tiempos gastados de los hitos alcanzados y el tiempo estimado del hito *Defensas de prácticas* todavía sin alcanzar, vemos como la totalidad del esfuerzo suma **2w 3d 3h 30min** o **107,5 horas**, lo cual es únicamente la tercera parte del esfuerzo estimado a partir de los Puntos de Función (ver *Sección 6.2.4*). La diferencia gran diferencia entre la estimación y el esfuerzo reflejado en *GitLab* puede deberse a las siguientes razones:

1. En ocasiones, no se ha reflejado el tiempo empleado en una tarea cuando estaba siendo repetida para solventar algún error.
2. En algunas de las tareas, las menos, no se registró el tiempo empleado, por lo que se ha considerado que se empleó el estimado, el cual posiblemente era menor que el real.

Hito	Nombre	Tiempo estimado	Tiempo gastado
1	Especificación y entorno	1d 6h	1d 7h
2	Modelos de análisis	1d 6h	2d 1h 45min
3	Modelos de diseño	1w 1d 15min	1w 5h 45min
4	Planificación del proyecto	1d 7h	2d
5	Defensas de prácticas	1d 6h 30min	Sin realizar

Cuadro 12: Tiempo estimado y gastado en los diferentes hitos del proyecto realizado

Etapas	Tiempo en horas	Tiempo %
Análisis de requisitos	32,25	34,86 %
Diseño	22,75	24,59 %
Codificación	33,25	35,95 %
Pruebas	4	4,32 %
<b>Total</b>	<b>92,5</b>	

Cuadro 13: Tiempo estimado y gastado en las etapas planificadas - Modelo en cascada

Es posible establecer una correspondencia con las tareas reales realizadas y las diferentes tareas de la planificación, pudiendo así calcular el esfuerzo real en cada etapa en caso de haber aplicado el Modelo en Cascada. Para ello, en primer lugar, es necesario tener en cuenta que dentro del hito *Modelos de diseño* se encuentran tanto las actividades de la etapa de *Diseño* como las actividades de la etapa de *Codificación*. En segundo lugar, en el hito *Defensa de prácticas*, casi todas las tareas pertenecen a la etapa de *Codificación*, menos la elaboración de los test de integración del sistema que constituye en sí misma la etapa de *Pruebas*. La única tarea realizada para el hito *Planificación del proyecto* es la elaboración de este documento, que no tiene ninguna equivalencia con las actividades de la planificación (ver *Figura 2*).

Sumando los tiempos de las actividades correspondientes a cada etapa en el caso de haber aplicado el Modelo en Cascada se puede saber la proporción de esfuerzo empleado en cada etapa (ver *Cuadro 13*).

## 7. Plan de tareas

### 7.1. Estimación del esfuerzo de las actividades

A pesar de la gran diferencia entre el tiempo registrado y el tiempo estimado a partir de los puntos de función (ver *Sección 6.2.5*), podemos utilizar los tantos por ciento del *Cuadro 13* para estimar el esfuerzo de las actividades planificadas. La estimación del esfuerzo de cada etapa y de cada actividad la podemos ver en el *Cuadro 14*, donde se incluyen las horas totales y las horas por persona suponiendo un grupo de trabajo de **3 personas**. Se obtiene un total de **111,97** horas por persona.

Etapas	Tarea	Esfuerzo por actividad (h)	Esfuerzo por actividad (h/persona)	Esfuerzo por etapa (h)	Esfuerzo por etapa (h/persona)	Esfuerzo por etapa (%)
Análisis de requisitos	Especificación de los requisitos del sistema	40	13,33	100,78	33,59	30,00 %
	Modelos de análisis	60,776	20,26			
Diseño	Diseño BD	4	1,33	67,18	22,39	20,00 %
	Diseño arquitectónico	26	8,67			
	Diseño de la interacción	20	6,67			
	Diseño GUI	18,184	6,06			
Codificación	Preparación del entorno	6	2,00	134,37	44,79	40,00 %
	Codificar / Probar modelos	15	5,00			
	Codificar templates	25	8,33			
	Codificar estilos	25	8,33			
	Codificar / probar controladores	25	8,33			
	Codificar /probar servicios	38,368	12,79			
Puebas	Pruebas de integración del sistema	25	8,33	33,59	11,20	10,00 %
	Pruebas de calidad, cobertura y rendimiento	8,59	2,86			
<b>Total</b>		<b>335,92</b>	<b>111,97</b>	<b>335,92</b>	<b>111,97</b>	

Cuadro 14: Estimación del esfuerzo de cada actividad

### 7.2. Diagrama de actividades

En la *Figura 3* se puede ver el diagrama de actividades con las horas/persona (h/p) de cada actividad.

### 7.3. Diagrama de Gannt. Cálculo del camino crítico del proyecto

En la *Figura 4* se puede ver el *Diagrama de Gannt* del proyecto. En el diagrama se muestran los hitos naturales que generan las etapas del Modelo en Cascada planificadas, así como el camino crítico de actividades. El camino crítico del proyecto está constituido por las siguientes tareas:

1. Especificación de requisitos del sistema
2. Modelos de análisis
3. Diseño de GUI / Diseño arquitectónico
4. Diseño de la interacción
5. Preparación del entorno
6. Codificar/Probar modelos
7. Codificar/Probar servicios Codificar/Probar controladores
8. Pruebas de integración

## 9. Pruebas de calidad, cobertura y rendimiento

La duración de las actividades está expresada en **días/persona** ya que el programa *Gantt Project* no permite indicarlo en horas; esto hace que las actividades cuya duración no sea, más o menos, un múltiplo de 8 aparezcan con una duración en días redondeada; esto provoca que actividades como la *preparación del entorno*, que sólo supone 2 horas/persona, aplase un día entero la fecha de finalización del proyecto. Teniendo esto en cuenta y atendiendo al diagrama de Gantt generado, si el proyecto comenzara el día **7 de octubre de 2019** y estaría finalizado el día **25 de octubre de 2019**.

En la *Figura 5* podemos ver una posible asignación de tareas disponiendo de **1 analista, 2 desarrolladores web y 1 un encargado de pruebas**.



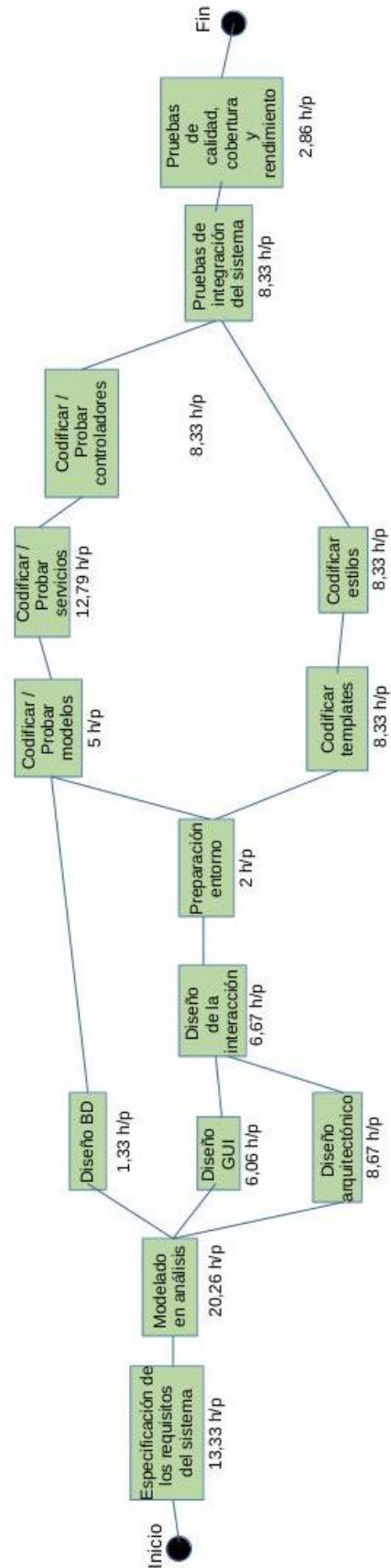


Figura 3: Diagrama de actividades

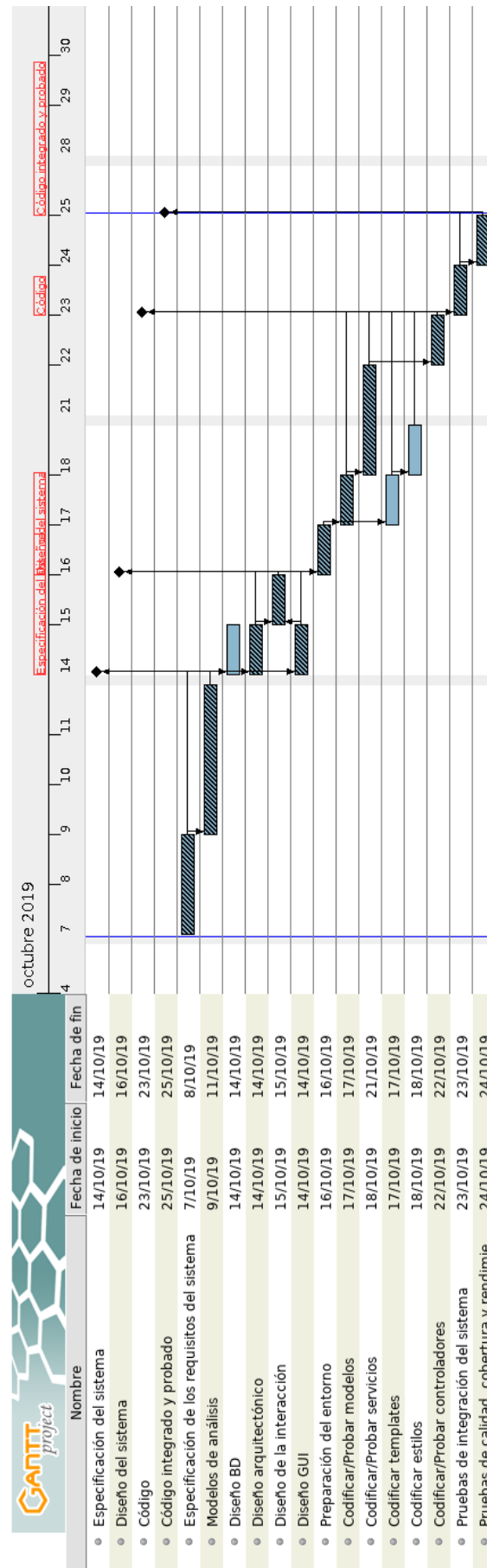


Figura 4: Diagrama de Gantt

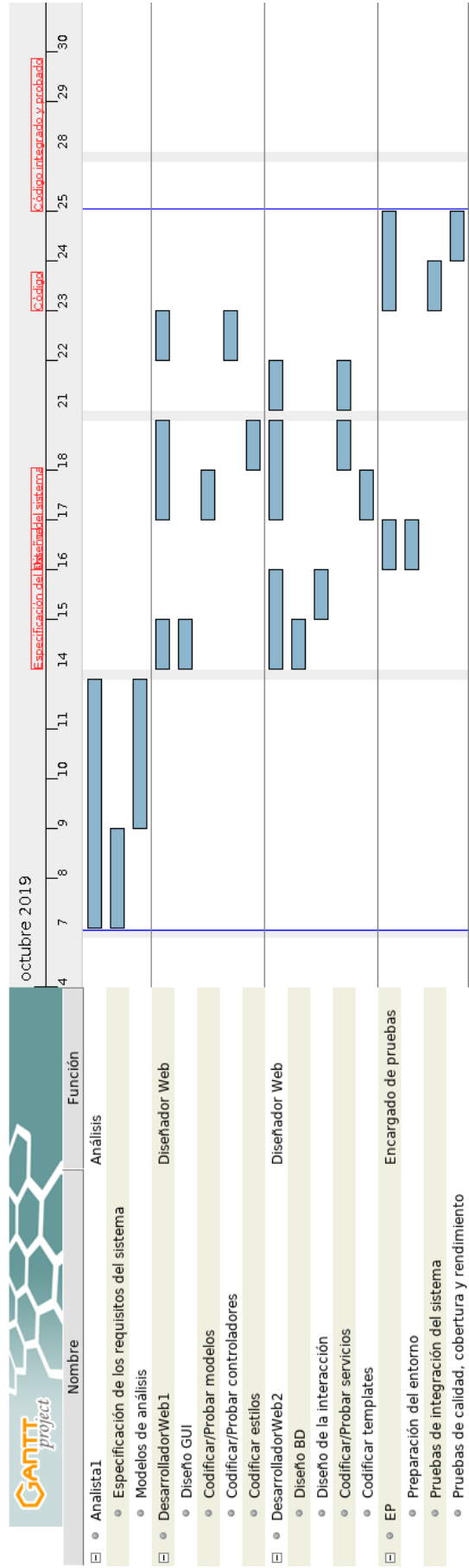


Figura 5: Asignación de tareas

## 8. Anexo I - *JSON schema* para los ficheros del sistema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Import/Export election file",
  "description": "File used to import/export all election data",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "date",
    "type",
    "configuration",
    "districts"
  ],
  "properties": {
    "date": {
      "type": "string",
      "format": "date"
    },
    "type": {
      "type": "string",
      "enum": [
        "regional",
        "congress",
        "senate",
        "local"
      ]
    },
    "configuration": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "threshold"
      ],
      "properties": {
        "threshold": {
          "type": "number",
          "minimum": 0,
          "maximum": 100
        }
      }
    },
    "districts": {
      "type": "array",
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "name",
          "voters",
          "representatives",
          "blank",
          "null",
          "candidatures"
        ],
        "properties": {
```



## Referencias

- [1] B. Hughes and M. Cotterell, “Software project management, 5th ed.,” 2009.
- [2] GitLab, “Time tracking.” [https://docs.gitlab.com/ee/user/project/time\\_tracking.html](https://docs.gitlab.com/ee/user/project/time_tracking.html). [Online, accedido: 18-12-2019].
- [3] César Llamas, “Método albrecht para el análisis de los puntos función,” 2019.