



aprenderaprogramar.com

Planificación del proyecto de un programa. Complejidad del problema vs recursos. (CU00228A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

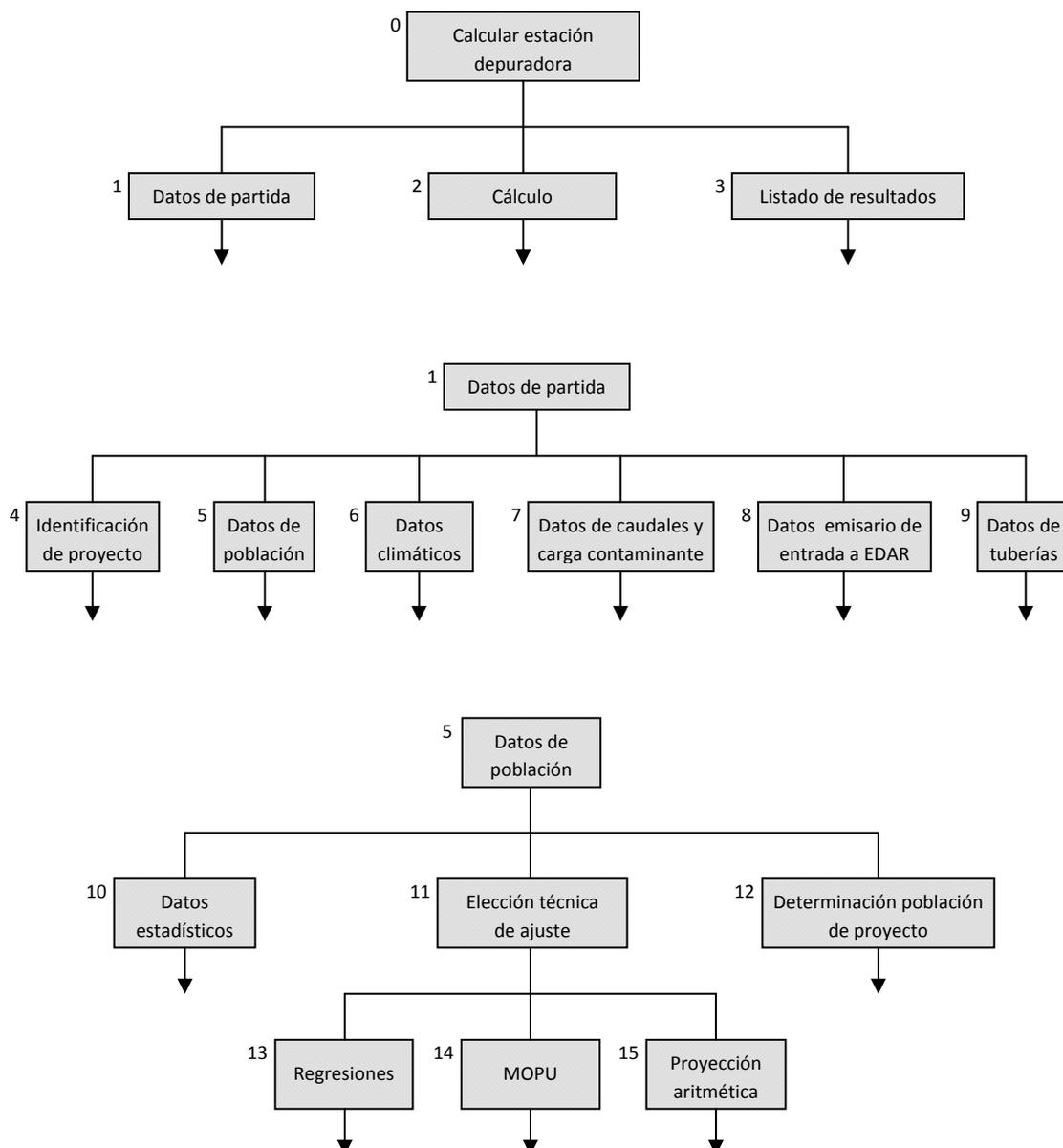
Resumen: Entrega nº 27 del Curso Bases de la programación Nivel II

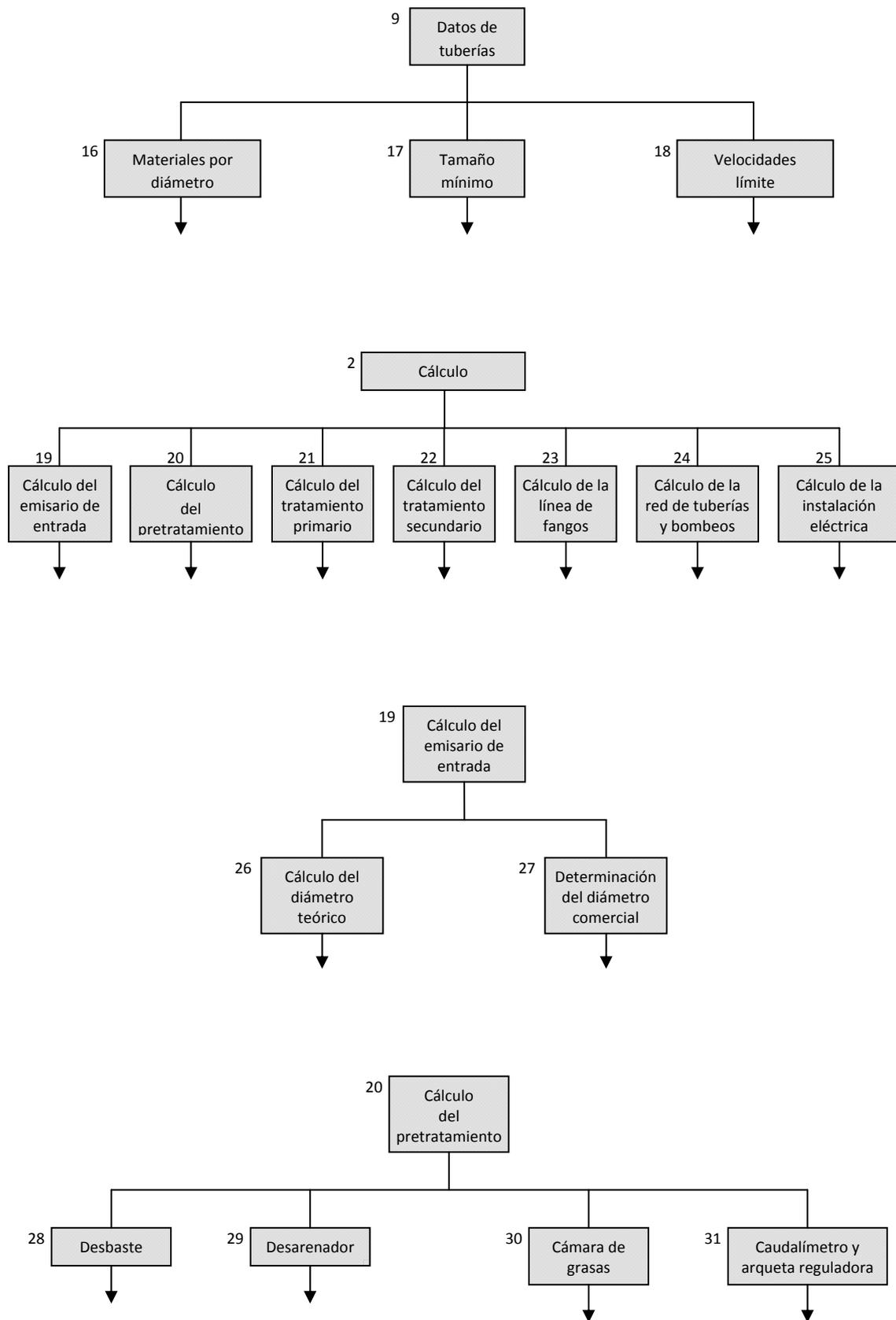
24

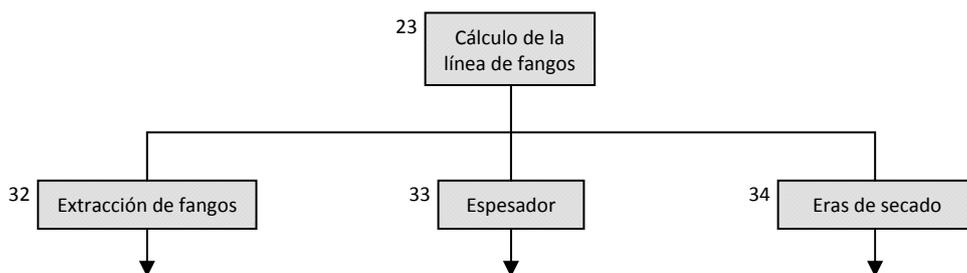
PLANIFICACIÓN DEL PROYECTO DE UN PROGRAMA

Hemos visto algunos casos de esquemas descendentes para pequeños algoritmos. Muchos programadores se saltan el realizar un esquema descendente para programas pequeños porque son capaces de elaborarlos mentalmente, de modo que empiezan a trabajar el pseudocódigo directamente. La utilidad más notable de los esquemas o índices descendentes la encontramos cuando los programas son largos y complejos. En ese caso en nuestra cabeza no cabe todo y el problema “se nos escapa”. El esquema o índice descendente se convierte así en nuestra guía de trabajo.

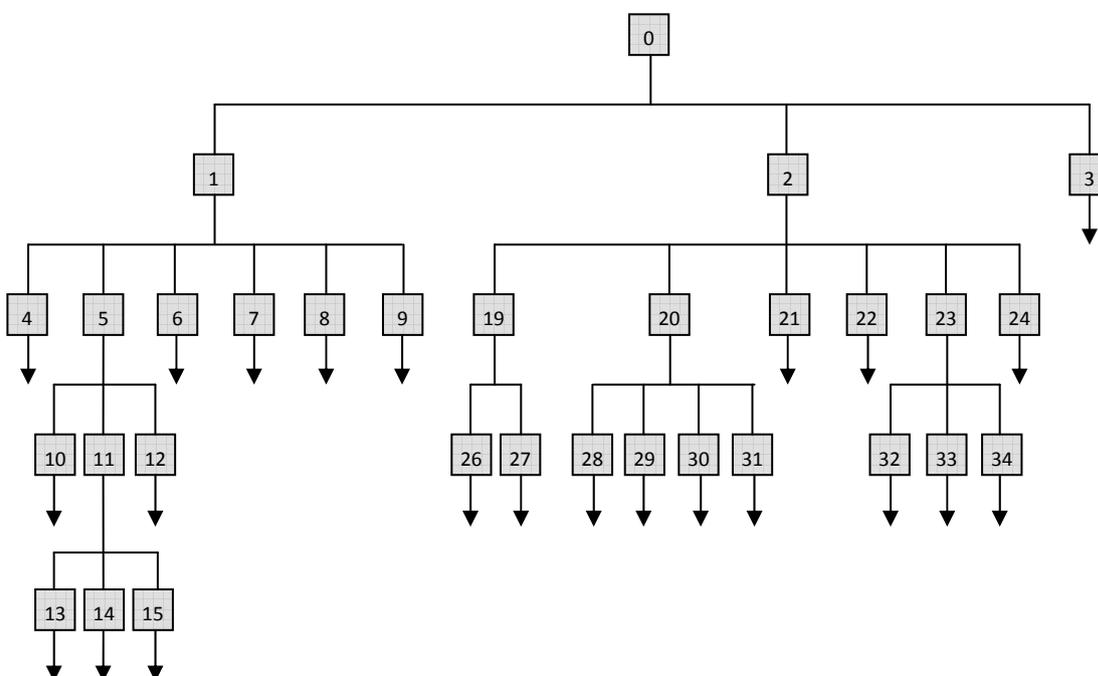
Supongamos que una empresa que realiza proyectos de estaciones depuradoras de aguas residuales (*EDARs*) para poblaciones urbanas ha llegado a establecer unos estándares de diseño para *EDARs* de hasta 10000 habitantes equivalentes. Dado que el proceso es similar para muchos proyectos desean “automatizarlo” a través de un programa de ordenador, cosa que encargan a dos ingenieros. ¿Cómo afrontar el problema? El esquema o índice descendente dará la respuesta. El mismo puede ser del tipo que se muestra:







Se han desarrollado algunos esquemas sin haber conseguido siquiera llegar a un nodo final. Veamos “por dónde vamos”:



La numeración es preliminar y corregible una vez se tuviera todo el árbol. A pesar de tener ya un número de niveles y nodos relativamente elevados, todavía no aparece ningún nodo cerrado que no se considere necesario descomponer. No vamos a proseguir con el desarrollo porque sería ocupar páginas y páginas. Queremos quedarnos con la idea de lo que puede ser intentar programar algo realmente complejo, de tener un árbol “frondoso”. ¿Cómo enfrentarnos a un problema de este tipo? En primer lugar sumaremos *Conocimiento profundo del problema + Conocimientos de programación*. Nuestro conocimiento personal nos debe guiar: si no nos sentimos seguros cuando realizamos programas relativamente sencillos, no convendrá ser temerario. Una vez analizada una situación y comprobada su dificultad, conviene reflexionar. Meternos de cabeza porque pensemos que la máxima “Podemos desarrollar programas que exceden de la capacidad humana” es una varita mágica que lo va a resolver todo será un error. Como programadores habremos de ir ampliando progresivamente nuestro “techo” de complejidad.

Supongamos que *conocimientos del problema + conocimientos del programador = "suficientes"*. Tendremos que enfrentar ahora *complejidad del problema con recursos disponibles*. El objetivo es hacer una valoración preliminar de la pregunta ¿Qué tiempo y esfuerzo me va a requerir desarrollar el programa? Una respuesta aproximada puede consistir en partir del esquema descendente y hacer una estimación de cuántos módulos de extensión media compondrían el programa, así como del tiempo de diseño y desarrollo del módulo medio. El tiempo necesario resultaría:

$$t(\text{días}) = n^{\circ} \text{ de módulos} \times \frac{\text{tiempo necesario para diseño + desarrollo (días)}}{\text{módulo medio y persona}}$$

Normalmente sabemos el tiempo en que debe estar listo el programa (o en el que aspiramos a que esté listo). Si el tiempo necesario supera ampliamente el deseado supone que nos estamos enfrentando a un objetivo demasiado ambicioso para los recursos de que disponemos. Supongamos una estimación de 30 módulos con duración estimada de dos días por módulo y persona dedicada. Una persona tardaría 60 días; dos personas 30 días; tres personas 20 días, y cuatro 15 días. Los recursos y limitaciones de tiempo definirán si el proyecto es viable o no.

Volvamos ahora al ejemplo de la pequeña empresa de estaciones depuradoras de aguas residuales. Supongamos que los ingenieros desarrollan un detallado esquema descendente y parten de estos datos.

- Conocimiento profundo del problema: "Sí, la empresa y nosotros mismos llevamos varios años en el sector".
- Conocimientos de programación: "Hemos desarrollado programas para cálculos complejos, pero nunca uno tan extenso como éste".
- Recursos: "dos personas (obviamente con 2 ordenadores)".
- Número de módulos estimados: 72
- Duración estimada para el módulo medio: 2,5 días

Obtendríamos un tiempo total $t(\text{días}) = 72 \times 2,5 = 180$

Para los recursos disponibles una *duración estimada = 180 días / 2 personas = 90 días* de trabajo.

Considerando una media de 20 días laborales por mes natural, el tiempo natural de desarrollo sería:

$$90 \text{ días laborales} \times \frac{30 \text{ naturales}}{20 \text{ laborales}} = 135 \text{ días}$$

$$135 \text{ días} \approx \frac{135}{30} \text{ meses} \approx 4,5 \text{ meses}$$

En resumen, cuatro meses y medio de trabajo por delante, aproximadamente, sin experiencia previa en desarrollos de este tipo. Se decide no afrontarlo por estar descompensados los recursos y complejidad del problema. Ante una situación de este tipo caben cuatro alternativas:

- 1) Reducir el objetivo subordinando el que se llegue a alcanzar el objetivo general al desarrollo previo de fases. Tiene la ventaja de que supone un test real a la complejidad del problema, que una vez se enfrenta puede magnificarse o reducirse, dándonos información para los procesos posteriores. El inconveniente puede ser una más difícil integración a la hora de unificar todas las fases.
- 2) Aumentar recursos: más programadores u otros programadores con mayores conocimientos o rendimiento.
- 3) Encargar el desarrollo a una consultoría que trabaje con ingeniería del software.
- 4) Renunciar o replantear el problema en otros términos.

Supongamos que nuestra empresa ficha a un programador más, que aporta experiencia en proyectos de esa envergadura. Se vuelven a estudiar los recursos frente a la complejidad del problema y se decide que es viable. ¿Cuál sería el siguiente paso? Entre decidir que se acomete y empezar a trabajar debe haber un estudio o análisis previo en el que la persona que va a actuar como coordinador toma decisiones importantes de cara a un desarrollo a medio o largo plazo. Las decisiones serán:

- Forma de ataque: el trabajo sobre el árbol irá por partes y seguirá un orden. Dicha estrategia se decidirá durante el estudio previo. Si por ejemplo contamos con 3 programadores, tendremos que decidir si se sitúan los tres en una misma rama o en distintas ramas. A su vez, se decidirá si se comienza con desarrollos parciales de las ramas o con desarrollos completos, etc.
- Características de la programación: se verán influidas por el lenguaje de programación que utilicemos y por las directrices que se establezcan al comenzar a trabajar en el proyecto. Incluye diversos aspectos como el utilizar una mayor o menor extensión de los módulos, uso en mayor o menor grado del control directo de flujos, grados de anidamiento y recursividad permitidos, sistema de comentarios, patrones para nombrar módulos y variables, etc.

En resumen, si el proyecto va a durar varias semanas, meses o incluso años, conviene “dedicarle minutos” a pensar qué sistema de organización va a ser el más adecuado, sin perjuicio de que siempre, durante el desarrollo del mismo, se continuarán estableciendo las pautas y reorientaciones que se estime oportuno. Pequeñas cuestiones, de las que ya hemos hablado, como organización de las variables, formas de nombrar los módulos, etc. pueden convertirse en serios problemas si en vez de planificarse desde el principio se tratan de corregir cuando existen desarrollos avanzados.

Próxima entrega: CU00229A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60