

Introducción a la programación competitiva

Marco Antonio Gómez Martín



Irie

Institut de Recerca
i Innovació Educativa

UIB - GOIB

Activitat realitzada en el marc del projecte d'innovació concedit per l'IRIE (UIB) nº PID222428, Títol "Desenvolupament d'un marc de treball per a l'entrenament en programació competitiva"

Gracias por la invitación

Activitat realitzada en el marc del projecte d'innovació concedit per l'IRIE (UIB) nº PID222428, Títol “Desenvolupament d'un marc de treball per a l'entrenament en programació competitiva”



Irie

Institut de Recerca
i Innovació Educativa

UIB - GOIB

Computing Curricula (CC2020)

- Computación
- Ingeniería de computadores
- Ingeniería del software
- Sistemas de Información
- Tecnologías de la Información
- Ciberseguridad
- Ciencia de datos

Programación

2x02 - De la creación de videojuegos a ingeniero en Facebook

Tweet



Borja Moreno

2x02 - De la creación de videojuegos a ingeniero en Facebook

La Tecnología Entrevistas

Suscribirse

Descargar App

1x

10

▶

10

↓

00:00

01:19:23

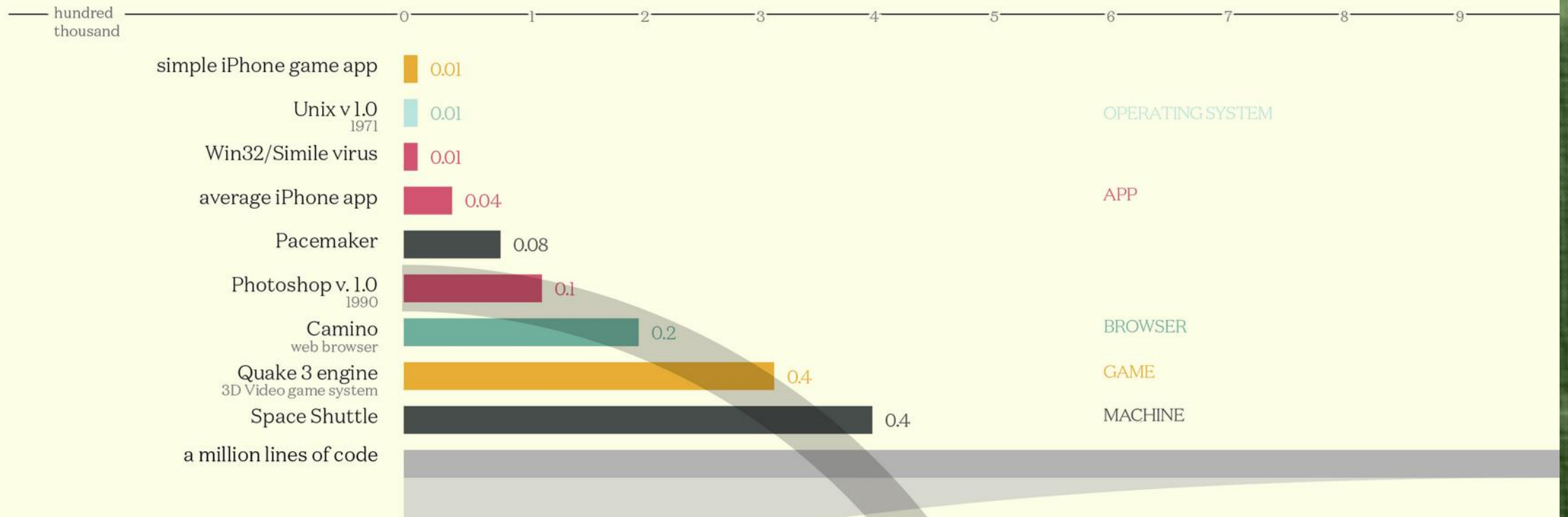
ivoox



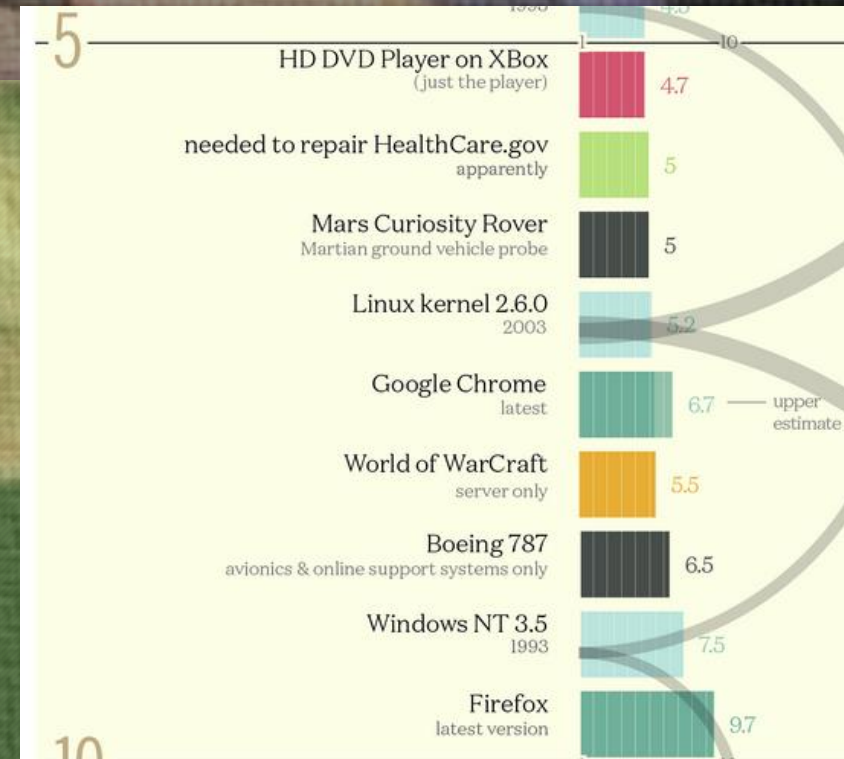
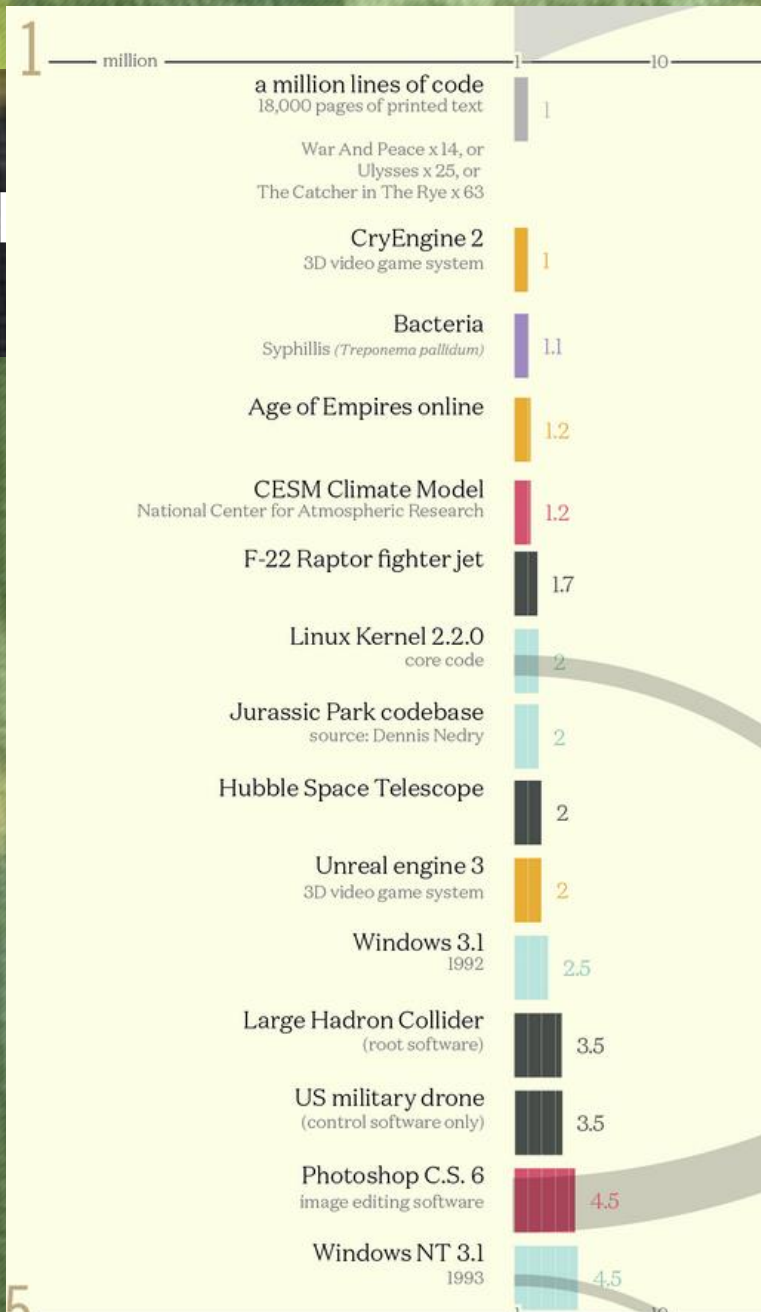
Programación

Codebases

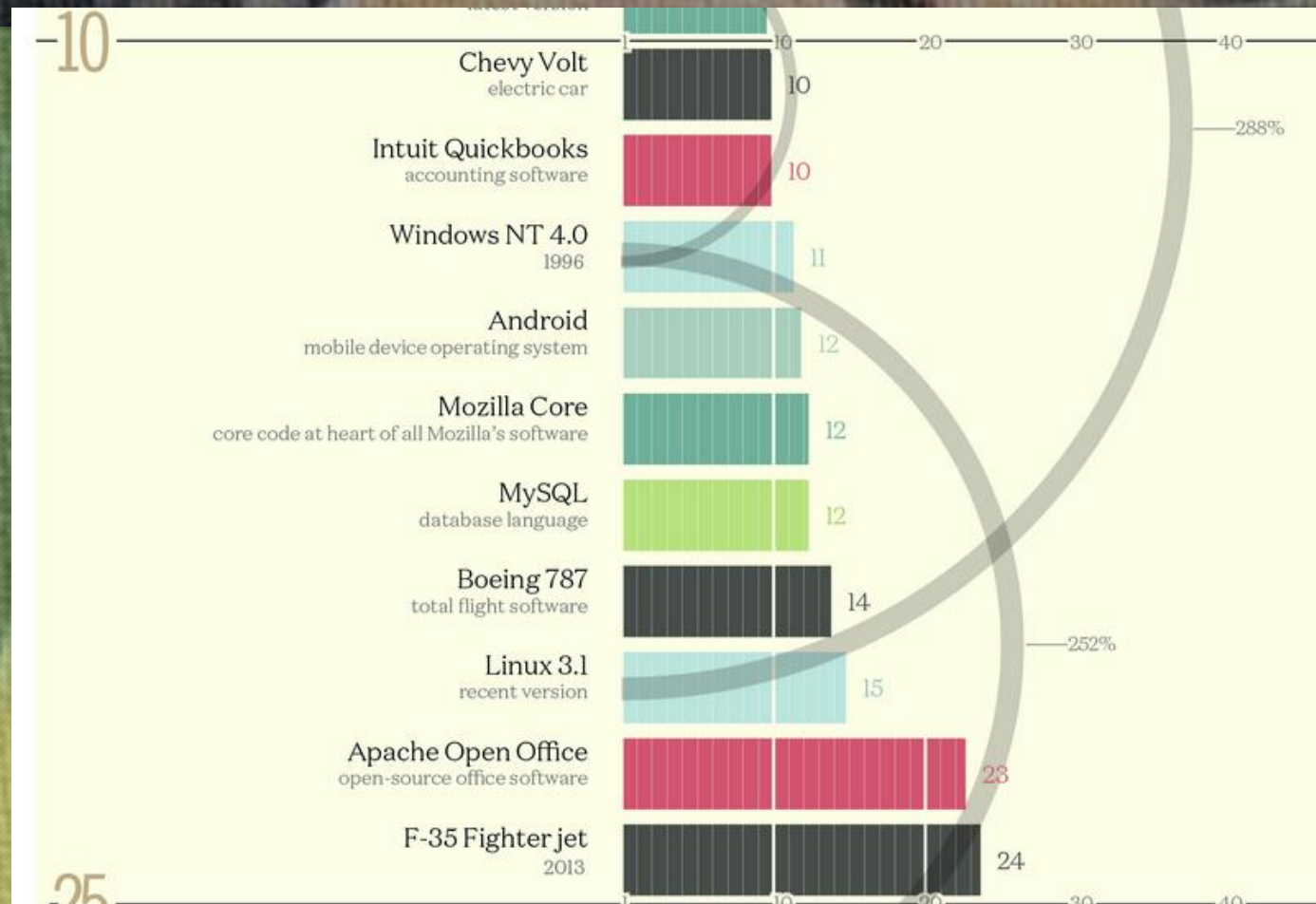
Millions of lines of code



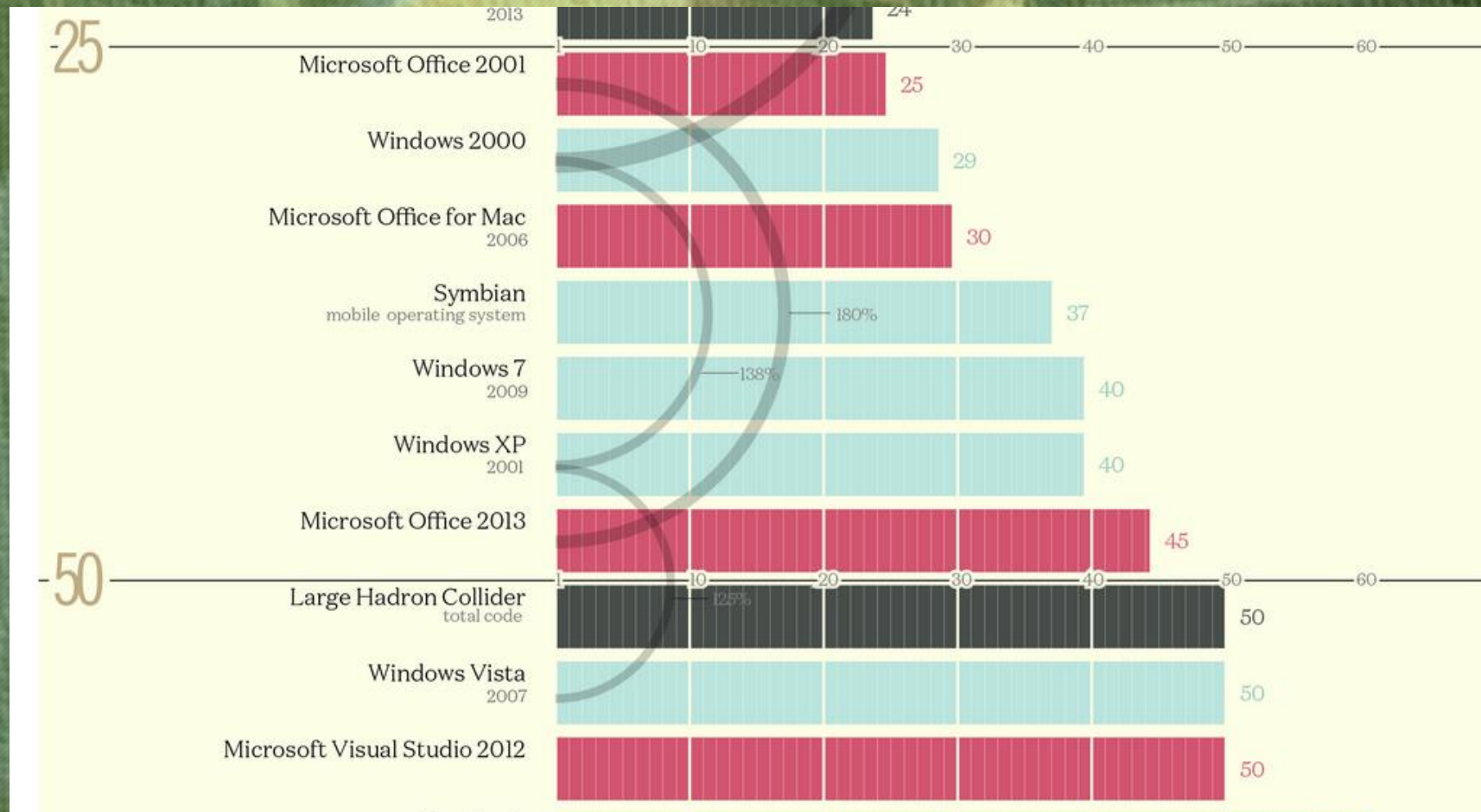
Progr



Programación



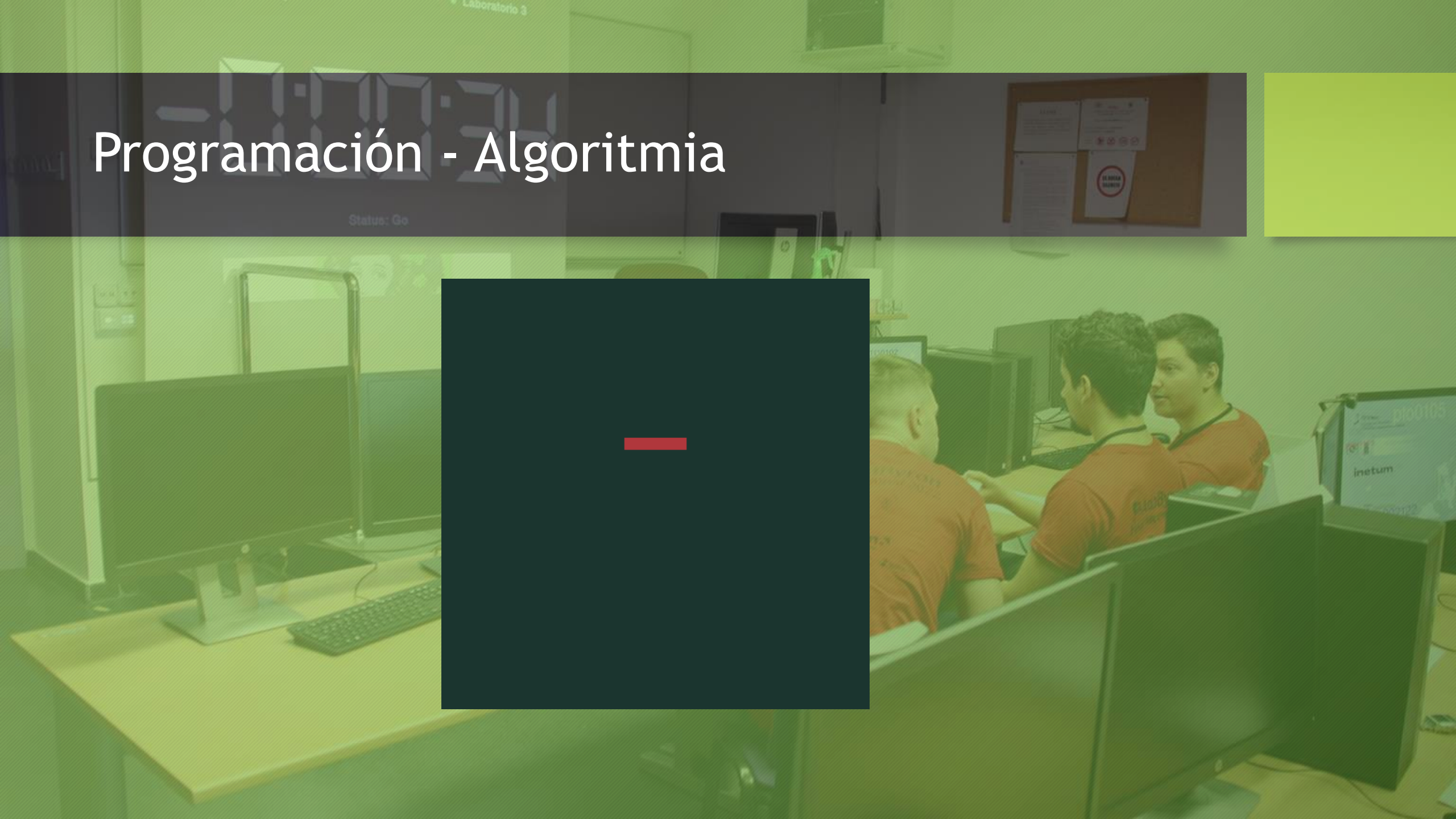
Programación




```
@Configuration
public class SecurityConfig {
    @Bean @Order(1)
    public SecurityFilterChain authorizationServerSecurityFilterChain(HttpSecurity
http)
        throws Exception {
        OAuth2AuthorizationServerConfiguration.applyDefaultSecurity(http);
        http.getConfigurer(OAuth2AuthorizationServerConfigurer.class)
            .oidc(Customizer.withDefaults()); // Enable OpenID Connect 1.0
        http
            // Redirect to the login page when not authenticated from the
            // authorization endpoint
            .exceptionHandling((exceptions) -> exceptions
                .authenticationEntryPoint(
                    new LoginUrlAuthenticationEntryPoint("/login"))
            )
            // Accept access tokens for User Info and/or Client Registration
            .oauth2ResourceServer(OAuth2ResourceServerConfigurer::jwt);

        return http.build();
    }
    ...
}
```


Programación - Algoritmia



Entrevistas de trabajo

- ¿Qué hace lo siguiente?

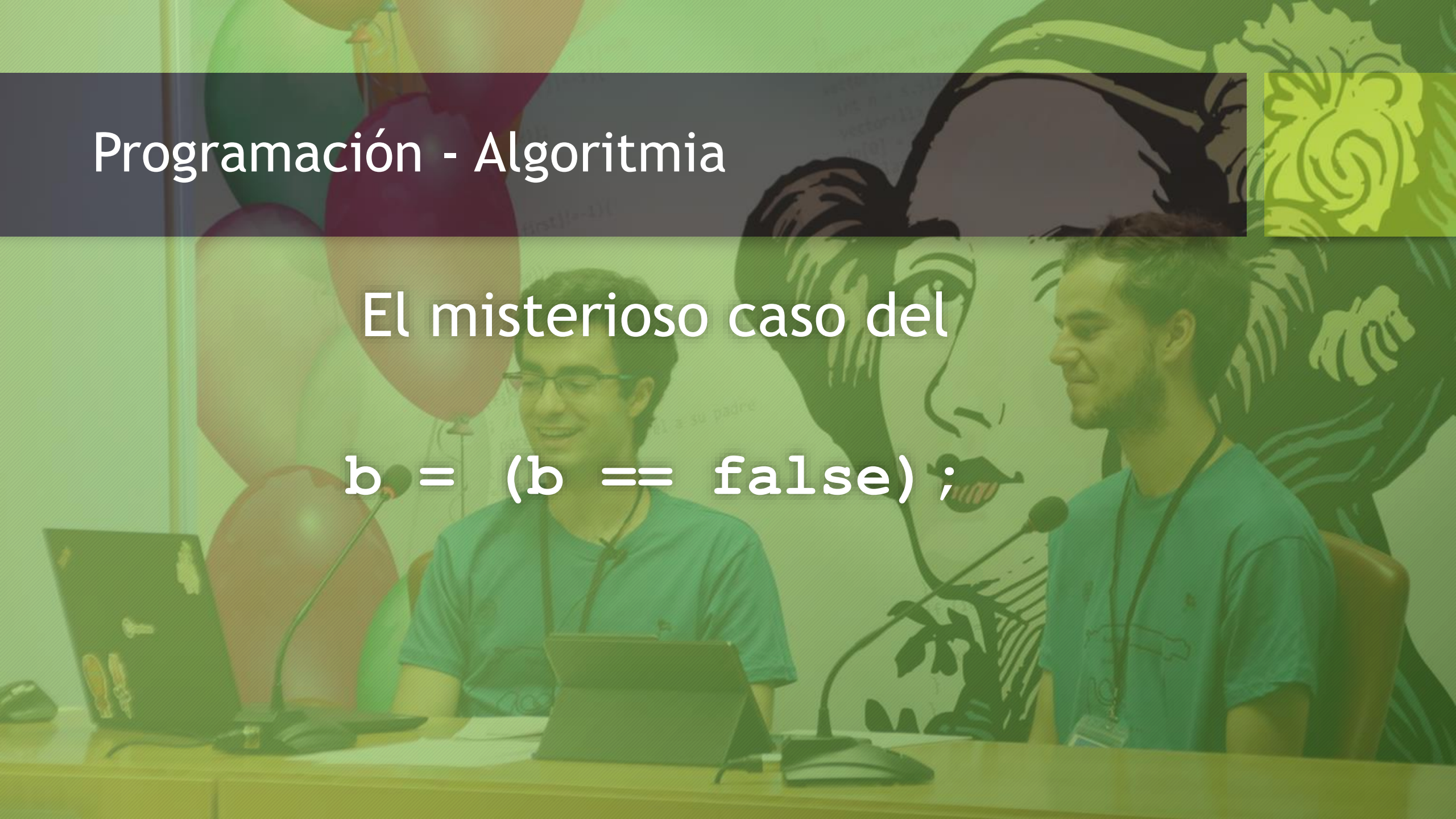
$! (a \ \& \ (a-1))$

- Dada una lista (larga) de palabras, ¿cómo calcularías cuántas son anagrama de otras?

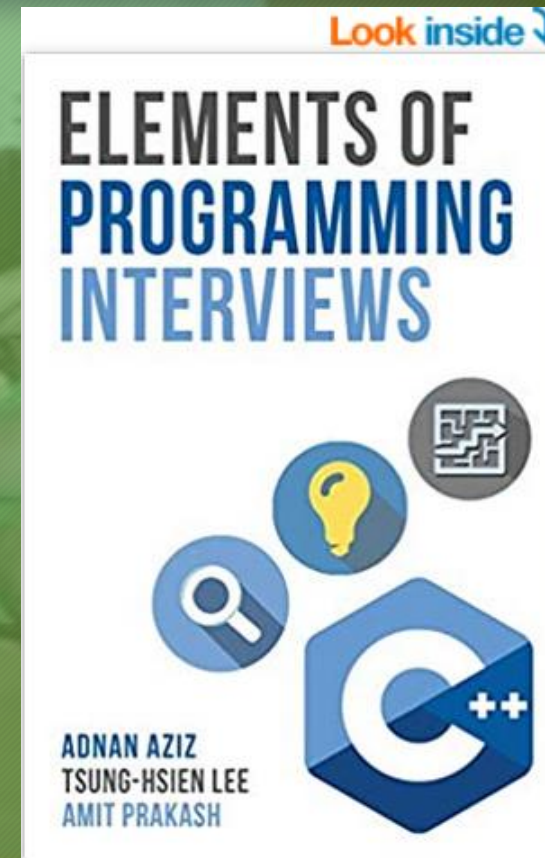
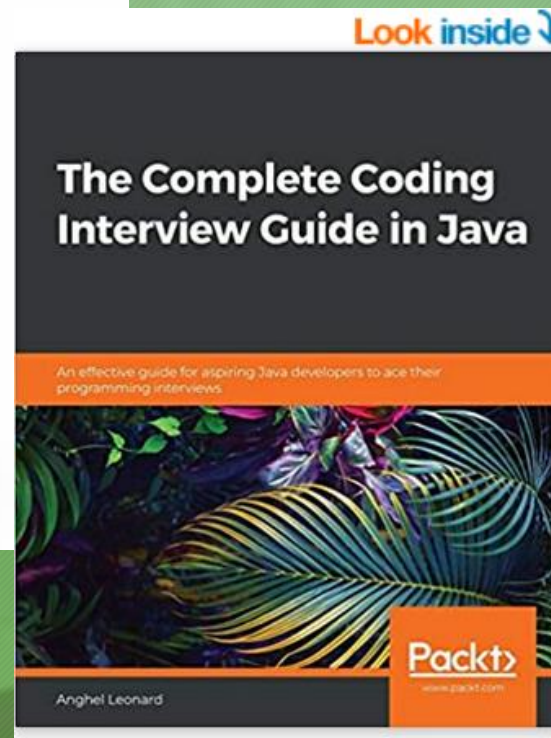
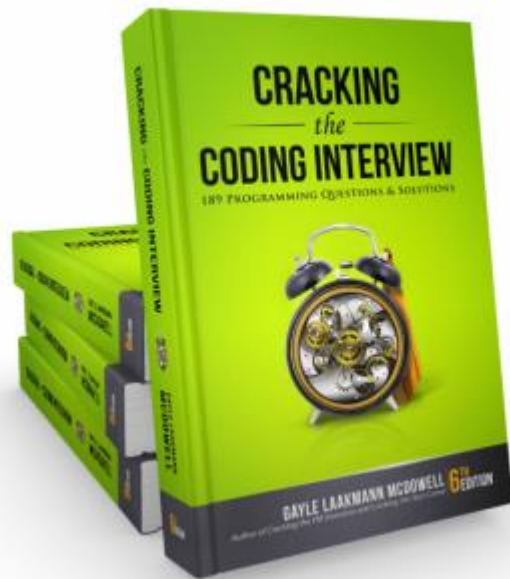
Programación - Algoritmia

El misterioso caso del

```
b = (b == false);
```



Entrevistas de trabajo



Entrevistas de trabajo

HackerRank[Products](#)[Solutions](#)[Resources](#)[Pricing](#)[For Candidates](#)[Request Demo](#)[Free Trial](#)

— **Leave the binary tree behind.**

Build coding questions using our library of dev-friendly content that challenges them to solve the problems they'd actually tackle on the job.

Start free trial →

Available questions: **4020**

Pagination
React

Autocorrect Prototype
Problem solving

Responsive form
CSS

Inventory lists
Javascript

Top Articleless
REST API

1

2

3

4

5

...

24

>

Filter questions by

Roles

Front-end developer

Difficulty

Easy

Medium

Hard

Skills

Type

Clear filter

Apply

Objetivos

- Para vosotros
 - Preparación (sin saberlo) para entrevistas de trabajo
- Para nosotros
 - Que vayáis “por el mundo” ganando concursos en nombre de la UCM

Programación

2x02 - De la creación de videojuegos a ingeniero en Facebook

Tweet



Borja Moreno

2x02 - De la creación de videojuegos a ingeniero en Facebook

La Tecnología Entrevistas

Suscribirse Descargar App

1x 10 Play 10 Download

00:00 01:19:23



Programación competitiva

- Problemas con fuerte componente algorítmica
 - Con solución conocida (pero secreta)
 - Cortos de programar (no más de unos pocos centenares de líneas, como mucho)
 - Corrección automática mediante jueces automáticos

Programación competitiva

- Normalmente:
 - Datos leídos por la entrada estándar
 - Respuesta dada por la salida estándar
- Lenguajes habituales: C, C++, Java, Python, ...

P Polisílaba es polisílaba

Tiempo máximo: 1,000 s



Memoria máxima: 4096 KiB

La palabra "*polisílaba*" es polisílaba ya que tiene más de tres sílabas. Este curioso fenómeno de una palabra describiéndose a sí misma también se da en otros vocablos que describen propiedades de las palabras. Por ejemplo, tenemos que "*esdrújula*" es esdrújula y que "*llana*" es llana.

Si vamos a palabras con más de un significado, el mismo esquema de frase aparentemente tautológica podemos tenerlo por ejemplo al afirmar que la naranja es naranja.

Sea como sea, si alguien no sabe el significado de *polisílaba* o de *esdrújula* las frases anteriores serán, efectivamente, simples tautologías vacías de contenido.



Entrada

La entrada comienza con una línea con un número indicando cuántos casos hay que procesar.

Cada caso tiene una frase con el esquema "*x es y*" en la que tanto *x* como *y* son palabras sin espacios con hasta 20 caracteres del alfabeto inglés.

Salida

Por cada caso de prueba se escribirá `TAUTOLOGIA` si la frase es una tautología (sin distinguir entre mayúsculas y minúsculas) y `NO TAUTOLOGIA` en caso contrario.

Programación competitiva

Entrada de ejemplo

3
Polisilaba es polisilaba
NARANJA es naranja
Mandarina es naranja

Salida de ejemplo

TAUTOLOGIA
TAUTOLOGIA
NO TAUTOLOGIA

Jueces automáticos



Veredictos habituales

- Accepted (AC): aceptado
- Presentation error (PE): Error de presentación
- Wrong answer (WA): respuesta incorrecta
- Compilation error (CE): Error de compilación
- Run-time error (RTE): Error durante la ejecución
- Time limit exceeded (TLE): Tiempo límite superado
- Memory limit exceeded (MLE): Límite de memoria superado

Mini demo

Tensión descompensada

Tiempo máximo: 3,000 s



Memoria máxima: 4096 KiB

La hipertensión arterial es un mal de "nuestro tiempo" que afecta a gran parte de la población, especialmente los mayores. Y eso para mí es malo, muy malo. Como es una enfermedad muy extendida, cuando les cuento a los médicos lo mal que me siento desde que soy hipertenso no me hacen demasiado caso. Me dicen que me tranquilice, que siga el tratamiento y la dieta y que no me preocupe más.

Pero yo estoy muy alarmado con el asunto, que tan mayor no soy. Para poder controlarme la tensión regularmente, tengo un tensiómetro en cada habitación. Así, me pille donde me pille la siguiente crisis, puedo medir mi presión arterial en cuestión de segundos y ver si tengo que salir corriendo al hospital o no.

La gente que tengo alrededor dice que estoy exagerando. Incluso los médicos de urgencias ya me conocen y me dejan horas en la sala de espera antes de llamarme. Y cuando me atienden noto cierto desdén en sus palabras supuestamente tranquilizadoras.

Lo peor de todo es que mi enfermedad va a peor. He leído en "*El rincón del hipocondríaco*" que es muy malo tener la tensión descompensada, algo que ocurre cuando la "*mínima*" y la "*máxima*" están demasiado cerca. Tras leerlo me he tomado la tensión y el susto ha sido mayúsculo, tanto que he salido disparado al hospital. Y la verdad es que me he enfadado bastante al ver cómo se han reído de mí en urgencias cuando les he dicho que la tenía tan descompensada que la *mínima* estaba por encima de la *máxima*. Me han dicho algo de que igual he leído los números al revés.



Mini demo

Último dígito del factorial

Tiempo máximo: 3,000 s



Memoria máxima: 4096 KiB

Tu primo Luis, de 12 años, está aprendiendo a usar la calculadora. Su profesor le ha dicho que calcule el factorial de varios números. Pero, para evitar que le tengan que copiar números muy largos en el cuaderno, les ha pedido únicamente el último dígito, el de más a la derecha.

Recordando que el factorial es la multiplicación de todos los números entre el número y el uno (por ejemplo, el factorial de 8, escrito $8!$, es $8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$), demuestra a tu primo Luis que tú eres capaz de hacerlo mucho más rápido que él.

Entrada

El programa recibirá en la primera línea de la entrada el número de casos de prueba. A continuación, cada caso de prueba estará compuesto de una única línea que contendrá un número (positivo).

Concursos

- Muchos tipos y reglas distintas
 - Por equipos, individuales
 - Ámbito local/privado, comarcal, nacional, internacional
 - Presenciales, multisede, a distancia
 - Solución vía código o vía respuesta
 - En jueces online abiertos o jueces privados para el concurso
 - ...

Concursos

- Dos modelos fundamentales
 - Modelo “ICPC” (concursos universitarios)
 - Modelo “IOI” (concursos para secundaria)

Modelo ICPC

- Veredictos “binarios”: AC / no AC
- Ganador: el que más problemas resuelva
 - En caso de empate, se desempata por tiempo
- Los envíos incorrectos a problemas resueltos penalizan en tiempo

Modelo ICPC



Pos.	PARTICIPANTE	PUNTOS	A 	B 	C 	D 	E 	F 	G 	H 	I 	J 	K 	L 
1	 C intenta otra vez 	9 842	1/31	5	1/9	1/22	2/144	1/37	1	1/70	1	1/152	1/254	1/103
2	 a 	8 869	1/11		3/24	1/41	1/243	1/102		1/29	3	1/165	3	2/194
3	 _("/)_ 	8 1119	1/22		2/36	2/39	4	1/60		1/50		1/262	3/208	5/282
4	 UMU+1 	7 741	1/68	1	2/28	1/38	2/239	1/77	1	1/93	1	1/158	2	1
5	 C y los mases 	7 1079	2/43		6/92	3/95	2/292	2/103		1/118	4	1/136		
6	 Tortilla Enjoys 	6 268	1/23		1/19	1/31	7	1/44		1/36		1/115		
7	 C++ 	6 639	1/29		5/99	3/41		1/88		1/107	4	1	4	1/155
8	 MeuGang 	6 780	2/33		2/91	1/56	5	2/76		2/127	4	4/257		4

Modelo IOI

- Problemas con “subproblemas”
 - Con límites distintos
- Cada subproblema da un número de puntos
- Ganador el que más puntos consiga

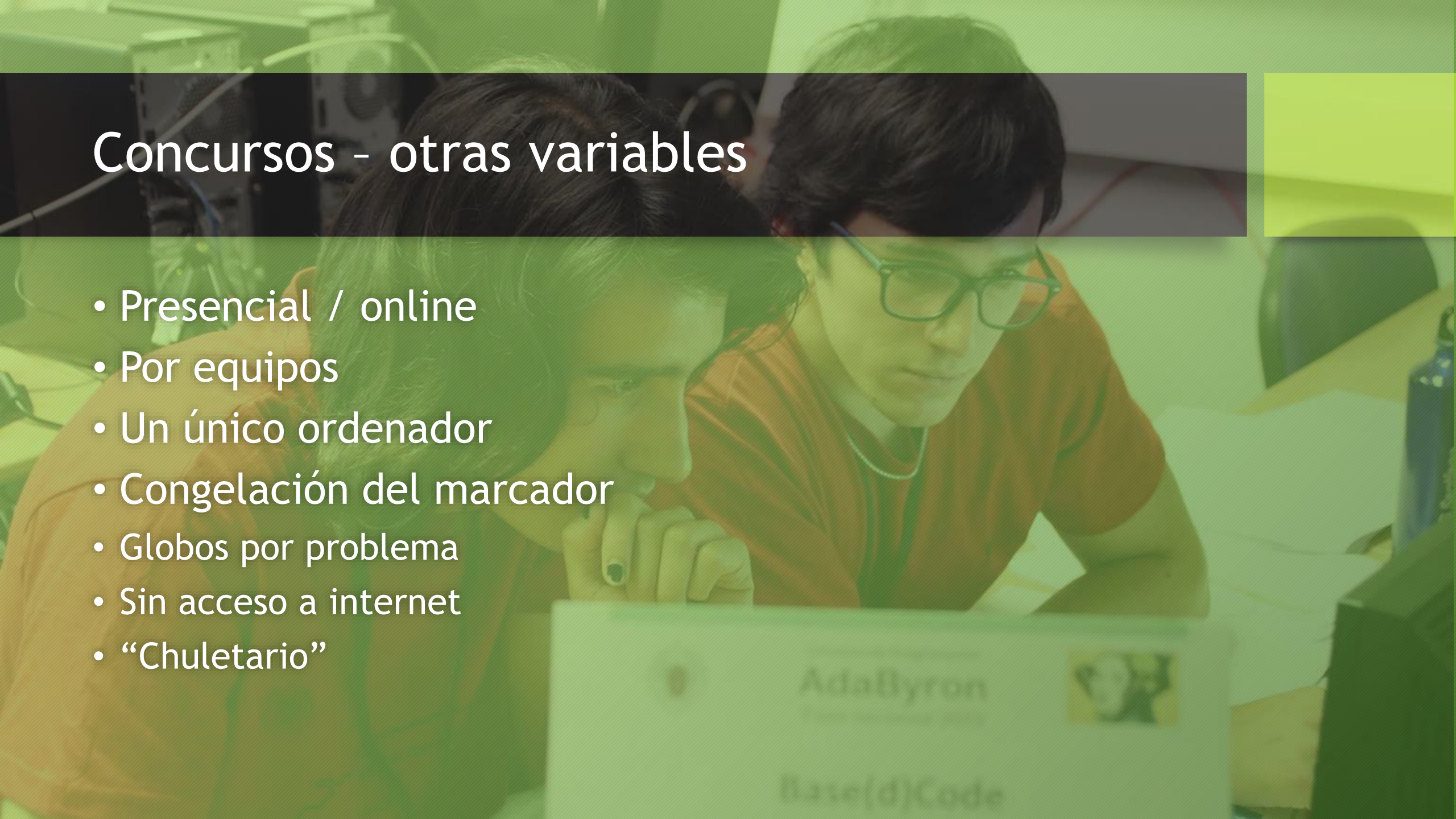
Modelo IOI



✓	Rank	Nombre	Apellidos	Team	A-Mi...	B-Ma...	C-Ae...	D-A...	E-Ju...	F-Su...	G-Lo...	H-Re...	Regiona...	Global
	1	Huize	Mao		100	100	75	25	100	23	100	0	523	523
	2	Daniel	López Piris		100	100	0	25	100	0	100	4	429	429
	3	Mateo	Martín Nieto		80	100	0	0	100	0	100	0	380	380
	4	Jaime	Pérez Gil		30	100	0	0	100	0	56	0	286	286
	5	Santiago	Qin		5	5	0	25	100	0	100	0	235	235
	6	Miguel	Martín Rodríguez		80	5	0	0	15	0	100	0	200	200
	7	Nicolas	Clamer Fuentes		30	40	0	3	0	0	100	0	173	173
	8	Miguel	Garnica Del Amo		30	0	0	25	0	100	0	0	155	155
	9	Samuel	Caraballo Chichiraldi		80	0	0	0	0	58	0	0	138	138
	10	Carlos	Quereda Molina		30	5	0	0	100	0	0	0	135	135
	11	Noah	Dris Sánchez		80	40	0	0	0	0	0	4	124	124
	12	Jaime	Torres Romero		80	5	0	3	15	13	0	0	116	116

Concursos - otras variables

- Presencial / online
- Por equipos
- Un único ordenador
- Congelación del marcador
- Globos por problema
- Sin acceso a internet
- “Chuletario”

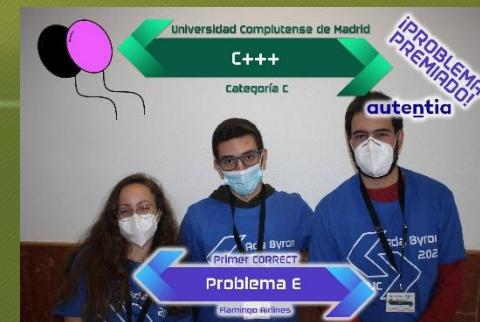
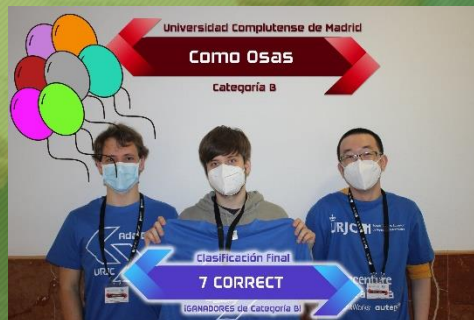


Concurso de



UNIVERSIDAD
COMPLUTENSE
MADRID

¿De verdad sirve para algo?



¿De verdad sirve para algo?

[...] Al final me han cogido en Google :D
Quería darte (daros) las gracias, porque sin duda las asignaturas de EDA, MARP y PCOM han sido fundamentales.

Las entrevistas parecían literalmente ejercicios de PCOM, y no tendría esa práctica sin vuestras asignaturas.

Yo por ejemplo sigo llevándome bien con [...] estudiantes de titulaciones de otras universidades omitidas [...] y ellos a día de hoy no tienen nada de práctica en ese tipo de problemas.

Concurso de Programación Ada Byron

¡Me gusta! ¿Cómo empiezo?

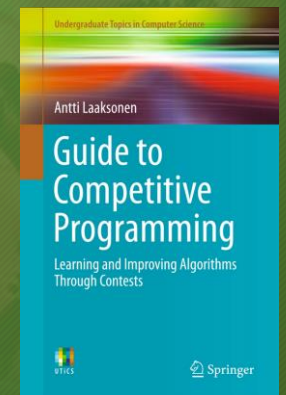


Jueces en línea

- ¡Acepta el reto! <https://aceptaelreto.com>
- Online judge: <https://onlinejudge.org/>
- Codeforces: <https://codeforces.com/>
- Topcoder: <https://www.topcoder.com/community/arena>
- Atcoder: <https://atcoder.jp/>
- Codechef: <https://www.codechef.com/>
- OmegaUp: <https://omegaup.com/>
- ...

Recursos

- Competitive Programming 4 (Halim)
- USACO guide <https://usaco.guide/>
- Aprende de la OIE <https://aprende.olimpiada-informatica.org/>
- Guide to Competitive Programming (Springer)
- ...



Recursos

- Libros de algoritmia “clásicos”
 - Habituales en la bibliografía de cualquier asignatura de algoritmia
- ¡Las asignaturas de algoritmia y programación!

Concursos presenciales

- AdaByron: <https://ada-byron.es/>
~Marzo/Abril 2023
- SWERC: <https://swerc.eu/>
 - Cerrado ya para el 2022/2023
- Google Code Jam
- ...

Concursos online

- Las 12 uvas: <https://las12uvas.es/>
31 de diciembre de 2022

[Las 12 uvas](#) [¿Qué es?](#) [Historia](#) [Normas](#) [¿Cómo participo?](#) [¿Puedo entrenar?](#) [Premios](#) [Concurso](#) [Más información](#)



Concurso Las 12 uvas

El concurso *Las 12 uvas* es un concurso de programación on-line organizado desde la [Facultad de Informática](#) de la [Universidad Complutense de Madrid](#). Tiene lugar el 31 de diciembre por la mañana sobre el juez en línea de la UCM [¡Acepta el reto!](#) y el objetivo es resolver en menos de seis horas hasta 12 problemas de programación (uno por cada campanada). Si quieres conocer la historia del concurso y por qué lo hacemos ese día, date una vuelta por la sección de [historia](#).

El concurso es individual y on-line. Los problemas a resolver se publican el día 31 a las 9:00 a.m., dándose el pistoletazo de salida. Entre campanada y campanada hay 30 minutos, por lo que la última es a las 14:30, acabándose el concurso a las 15:00. En ese momento el que tenga mayor puntuación gana el concurso. Muchos de los problemas del concurso son problemas *fáciles* que no requieren grandes destrezas de programación ni algoritmia.

Asómate al concurso en cualquier momento de la mañana del 31 de diciembre

Aunque el concurso comienza a las 9 y termina a las 15, no necesitas competir las 6 horas. Los problemas pueden resolverse en menos tiempo y puedes unirte en cualquier momento. Eso sí, si quieres llevarte un premio, tendrás que madrugar.

Una campanada cada media hora, una uva cada media hora

Hay seis horas para resolver los doce problemas. ¿Podrás hacerlo, o se te atragantará alguna uva?

Si crees que programas más rápido que otros compañeros de tu clase o trabajo, convénceles para que acepten el reto y averigüad de una vez por todas quién es capaz de resolver más problemas en menos tiempo. Y si simplemente quieres pasar un buen rato y romper la monotonía de las Navidades durante unas horas, o descansar de la preparación de exámenes, pásate por aquí ese día para intentar resolver los problemas.

Además, gracias a la colaboración con distintos patrocinadores, habrá [premios para los ganadores](#). ¿Te animas a intentarlo?

Premios patrocinados por


Twittear a [@lasdoceuvas](#)

Concursos online

- Rondas de codeforces
- Advent of code: <https://adventofcode.com/>

¿Se puede usar en clase?

- ¡Si!



Programación competitiva sin concursos

Camino al cole

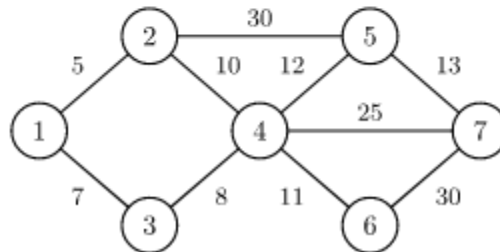
Tiempo máximo: 2,000-3,000 s



Memoria máxima: 20480 KiB

Lucas va todos los días en bici al cole. Como le cuesta una barbaridad levantarse por las mañanas, siempre lleva prisa y tiene que ir por el camino más corto. Pero le aburre ir siempre por el mismo camino, por lo que va alternando, eso sí, recorriendo siempre la misma distancia mínima, para no llegar tarde. Como es un poco temerario, no respeta el sentido de circulación de las calles por lo que es capaz de recorrer cualquiera de ellas en ambos sentidos. Ahora se pregunta de cuántas formas distintas puede ir de su casa al colegio sin recorrer ni un centímetro de más.

Por ejemplo, si el siguiente esquema representara el pueblo de Lucas, con 7 intersecciones y 10 calles, donde junto a cada calle aparece su longitud, Lucas viviera en la intersección número 1 y su colegio se encontrara en la intersección 7, entonces Lucas podría ir al colegio de 4 formas distintas recorriendo una distancia de 40 ($1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 7$, $1 \rightarrow 2 \rightarrow 4 \rightarrow 7$, $1 \rightarrow 3 \rightarrow 4 \rightarrow 7$ y $1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7$).



Programación competitiva sin concursos

Entrada de ejemplo

```
7
10
1 2 5
1 3 7
2 4 10
3 4 8
2 5 30
4 5 12
4 7 25
4 6 11
5 7 13
6 7 30
```

Salida de ejemplo

```
4
```


Prog

```
16  int main() {
17      while (cin >> n >> m) {
18          G = W = vector<vector<int>>(n + 1);
19
20          while (m--) {
21              int u, v, d; cin >> u >> v >> d;
22
23              G[u].push_back(v), W[u].push_back(d);
24              G[v].push_back(u), W[v].push_back(d);
25          }
26
27          dis.resize(n + 1, INF), cont.resize(n + 1, 0);
28
29          dijsktra();
30
31          cout << cont[n] << '\n';
32
33          G.clear(), W.clear(), dis.clear(), cont.clear();
34      }
35
36      return 0;
37  }
```


Jueces en línea como almacenes de problemas

¡Acepta el reto!

Problemas

Estadísticas

Documentación

Profesor FDI

Buscar

Estás en: Inicio

¿Qué es?

¡Acepta el reto! es un almacén y juez en línea de problemas de programación en español que acepta soluciones en C, C++ y Java.

No es un mero listado de problemas, sino mucho más. ¡Es un corrector automático!

Si quieres poner a prueba tu habilidad programando y compararla con la de otros, ¡éste es tu sitio!

¿Por dónde empiezo?

Lo primero que querrás hacer será leer algunos de los múltiples problemas disponibles. Si no sabes por cuál empezar, puedes recorrer las diferentes categorías o mirar el *problema de la semana* que te proponemos abajo.

Si te llama la atención algún problema, crees que eres capaz de resolverlo y quieres intentarlo, [regístrate](#). ¡Es fácil, rápido y no te enviaremos spam! Con tu cuenta, podrás enviar tus soluciones y compararlas con las de otros usuarios.

¿Aceptas el reto?

Problema de la semana

Ordenando a los pacientes en urgencias

En el Hospital ACR (*Aquí Curamos Rápido*) se han puesto a mejorar las Urgencias para que los enfermos que llegan con dolencias más graves sean atendidos antes que los demás. Para eso, han comprado una UCM (*Unidad de Catalogación Médica*) que es capaz de valorar instantáneamente el estado de un paciente con un número entero entre 0 y 1.000.000, donde 0 indica que su dolencia es menor (quizá incluso inexistente y sea un mero hipocondríaco) y 1.000.000 indica que el enfermo está casi caminando hacia la luz.

Por desgracia, la afluencia de enfermos es tal que incluso así es muy complicado saber rápidamente quién debería ser el próximo en ser atendido. No hacen más que entrar pacientes nuevos a la vez que los más graves son atendidos, y no es fácil mantenerlos ordenados. Cuando un médico queda libre, debe ser atendido el enfermo a la espera con una valoración más grave. Si hay dos pacientes evaluados con la misma gravedad, deberá ser atendido el que más tiempo lleve esperando.

Para ayudar en la tarea de decidir quién es el próximo, desde ACR se ha hecho un llamamiento para buscar ayuda entre los mejores programadores. ¿Eres tú uno de ellos?



[Seguir leyendo](#)

Abierta la inscripción del concurso Las 12 UVas del 31 de diciembre de 2022

(c) Acepta el reto. 2013-2022

tema, compuesto por un número natural. Los identificadores se asignan de manera n en la plataforma.

puesto de 100 problemas. Los problemas de un mismo volumen no tienen por qué en el sistema en un periodo cercano, de manera que sus identificadores los hacen

veriguar el número exacto de problemas disponibles, y enfrentarse a ellos sin un ede dar pistas sobre su resolución. Si se desea practicar algún tipo concreto de

ón	Nº de problemas	AC/Envíos
ias: [100-199]	100	93922/270941
ias: [200-299]	100	40509/118256
ias: [300-399]	100	41058/103421
ias: [400-499]	100	27119/75722
ias: [500-599]	100	18459/51126
ias: [600-699]	40	6496/16693

¿Qué significan estos números?

Algoritmos iterativos

Periodo de tiempo: Viernes, 16 Sep 2022 13:00 - Viernes, 30 Sep 2022 13:00
[Envíos totales: 351]

Algunos algoritmos iterativos.

Lo importante no es únicamente conseguir el AC, sino implementarlos con un código compacto y claro, pensando el algoritmo antes de programarlo.

#	Problem Title	AC	NOS
219	La lotería de la peña Atlético	1945	4265
314	Temperaturas extremas	823	1431
222	Serie de potencias	506	2487
242	Erasmús	986	3786
152	Va de modas...	855	3420
171	Abadías pirenaicas	2045	6597
346	El hombre con seis dedos	427	1215

Live Submissions (hide)

Show : 5 | 10 | 25 | 50 | 100

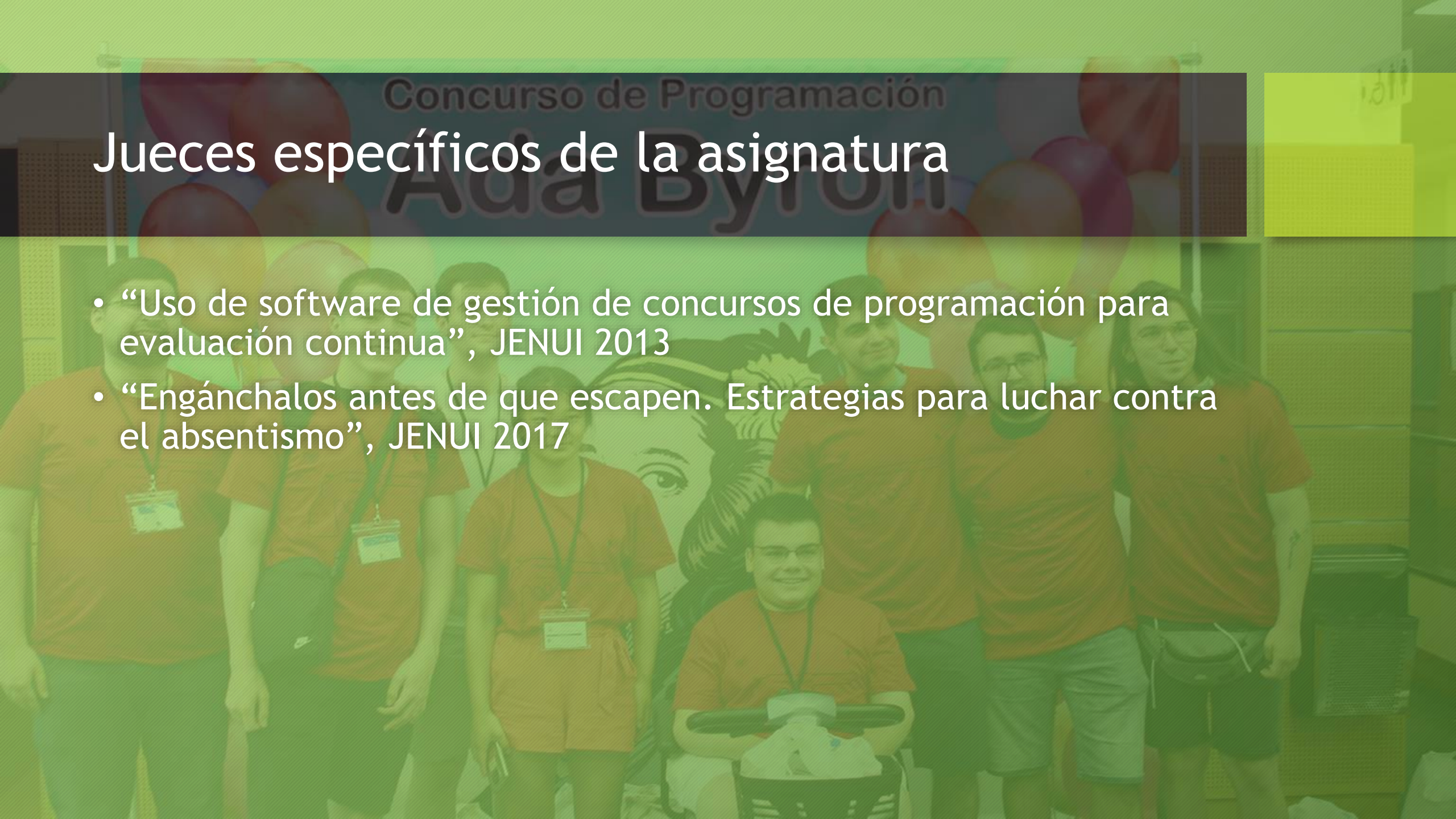
#		Problem Title	User (username)	Verdict	Lang	Time	Best	Rank	Submit Time
649276	346	El hombre con seis ...	Juan Diego Martínez Banda (...)	Accepted	C++	0.424	0.000	125	2022-09-30 12:12
649272	219	La lotería de la peñ...	Juan Diego Martínez Banda (...)	Accepted	C++	0.324	0.000	820	2022-09-30 12:08
649270	219	La lotería de la peñ...	Juan Diego Martínez Banda (...)	Time limit	C++	-	0.000	-	2022-09-30 12:07
649263	314	Temperaturas extre...	Juan Diego Martínez Banda (...)	Accepted	C++	0.152	0.000	342	2022-09-30 11:55
649248	222	Serie de potencias	Pablo Ferrando Galán (pfg-o...	Accepted	C++	1.860	0.000	307	2022-09-30 11:43

Clasificación

#	Usuario (nick)	Resueltos	219	314	222	242	152	171	346
1	Laura De Cara Molina ...	7 / 9d04	1d	1d(1)	1d(3)	1d	1d(2)	1d	1d
2	Clara Fernández Fort...	7 / 15d14	23:03	23:24	1d(4)	1d	5d(8)	1d	5d(1)
3	Javier Gómez Arribas ...	7 / 19d05	2d	2d	2d(1)	2d	2d	2d	2d
4	Daixiang Chen (DAIXI...	7 / 26d11	3d	3d	3d	4d	4d(2)	4d	4d
5	Jiahui You (jiahuiyou)	7 / 27d00	7d	06:44(1)	7d	04:59	5d	04:58	7d
6	Franz Rodrigo Quispe ...	7 / 30d23	3d	3d	6d(2)	3d	5d	3d	6d
7	Rodrigo Castañón Mar...	7 / 34d06	2d	8d	6d(1)	2d	4d(1)	3d	5d
8	Adrián Salinas Ruiz (a...	7 / 41d19	4d(1)	4d	7d(12)	7d(3)	5d	01:29(3)	11d
9	José Ignacio Barrios ...	7 / 44d11	6d(1)	6d	6d	6d	6d	6d	6d
10	Ángel Vila Sánchez (a...	7 / 47d22	4d	9d(1)	7d(7)	5d(3)	9d	4d	8d
11	Andrés Sánchez Casa...	7 / 64d16	11d	11d	3d(3)	5d(2)	8d(5)	11d	11d
12	Francisco Prieto Galle...	7 / 72d07	10d	10d	10d	10d	10d	10d	10d
13	Alejandro Rodríguez ...	6 / 28d15	1d(5)	1d	4d	7d	-	7d(1)	7d
14	Li Jie Chen Chen (LiO...	6 / 38d07	5d	5d	7d(7)	7d	7d(1)	7d	-
15	Santiago Castro Mate...	6 / 56d12	9d	9d(1)	13d(7)	2d	8d(1)	13d	-
16	Carlos Alba Tejada (C...	6 / 78d16	12d	12d	12d	12d	12d	12d	-

Jueces específicos de la asignatura

- “Uso de software de gestión de concursos de programación para evaluación continua”, JENUI 2013
- “Engánchalos antes de que escapen. Estrategias para luchar contra el absentismo”, JENUI 2017





Introducción a la programación competitiva

Marco Antonio Gómez Martín



Irie

Institut de Recerca
i Innovació Educativa

UIB - GOIB

Activitat realitzada en el marc del projecte d'innovació concedit per l'IRIE (UIB) nº PID222428, Títol “Desenvolupament d'un marc de treball per a l'entrenament en programació competitiva”