

---

# GYMKANA: LOS CÓDIGOS DEL HAMBRE

---

La razón de realizar esta actividad es determinar en qué nivel la gamificación favorece el proceso de aprendizaje. Nuestra intención es mejorar nuestro proceso de enseñanza por lo que agradecemos vuestra colaboración y pedimos disculpas de antemano por los errores que puedan surgir. Esta actividad no podría llevarse a cabo sin la colaboración desinteresada de todo el personal del Centro, por lo que os pedimos que seáis respetuosos, agradecidos e indulgentes con todos los participantes, jugadores o no.

---

## MECANISMO DEL JUEGO

---

El juego consta de tres pruebas. En cada prueba hay que resolver un cálculo (cuyo resultado puede ser un dato numérico o una cadena de caracteres) y contestar correctamente a tres preguntas de tipo test sobre la asignatura.

La primera prueba os la entregará la profesora de la asignatura en la sala Hedy Lamarr cuando dé comienzo el juego. Para obtener el enunciado de la siguiente prueba se necesitan los dos resultados obtenidos en la actual:

- El número o el texto obtenido en el cálculo realizado por el programa será la clave que habrá que decirle a la persona que os entregará el sobre de la siguiente prueba. Si la clave es incorrecta podréis volver a intentarlo.
- Para saber a quién tenéis que dirigiros para obtener el siguiente enunciado necesitaréis las respuestas correctas a las preguntas de tipo test y buscar el código que forman en la tabla del anexo.

Por ejemplo, si como resultado del cálculo habéis obtenido el número 1000 y respondéis A a la primera pregunta, B a la segunda y C a la tercera, tenéis que ir al despacho 1D009 (como podéis ver en el anexo, el código ABC se corresponde con ese lugar) entregarle a la persona que esté allí un papel con el mensaje 1000 y ella os dará el sobre con la siguiente prueba si la respuesta es correcta.

---

## MATERIAL NECESARIO

---

Sólo necesitáis:

- Un ordenador portátil, como mínimo, por equipo.
- El ordenador debe tener instalado algún entorno de desarrollo para Java (Eclipse, NetBeans,...)
- Lápiz y papel.

---

## REGLAS DEL JUEGO (GRUPO T3)

---

Las reglas que vamos a seguir son muy básicas y pueden resumirse en:

- El juego se realizará el miércoles 15 de noviembre de 2017.
- Comenzará a las 12:00h en la sala Hedy Lamarr y terminará transcurridas 2 horas desde el inicio.
- Podrán participar todos los alumnos del **grupo 3 de Fundamentos de Programación** que se hayan inscrito previamente.
- Los equipos estarán formados por 3 o 4 personas.
- Todos los participantes responderán a una **ENCUESTA ANÓNIMA** tras la realización de la actividad.
- Cada equipo deberá elegir un capitán que funcionará como portavoz del grupo.
- Cada equipo tendrá asignada una mesa de trabajo.
- Los sobres que contienen los enunciados de las pruebas deben permanecer siempre encima de la mesa de cada equipo.
- Cuando localicéis el lugar en el que tenéis que pedir un sobre:
  - Sólo entrará el capitán del equipo.
  - El número obtenido como clave lo entregará escrito en papel que se incluye dentro de cada sobre, **NUNCA SE DIRÁ EN VOZ ALTA.**
  - Si el lugar está cerrado o vacío es que el código es incorrecto, habéis fallado en alguna de las respuestas del test.
- La mayor parte de la actividad se realizará en la sala Hedy Lamarr.
- **NADIE PUEDE ABANDONAR EL JUEGO ANTES DE QUE FINALICE.**
- No debéis gritar ni correr por el edificio. Respetemos a las personas que están trabajando.

---

## FIN DEL JUEGO

---

El juego terminará en cuanto 10 equipos finalicen exitosamente todas las pruebas o se haya agotado el tiempo límite (2 horas).

El primer equipo que termine las pruebas será el ganador, pero recibirán premio los 10 primeros equipos que acaben la gymkana.

El premio es 1 punto para cada uno de los componentes del primer equipo que finalice las pruebas, 0.9 puntos para el segundo, 0.8 para el tercero, 0.7 para el cuarto, 0.6 para el quinto, 0.5 para el sexto, 0.4 para el séptimo, 0.3 para el octavo, 0.2 para el noveno y 0.1 puntos para los componentes del décimo equipo. Esas notas se sumarán a su nota final de la convocatoria ordinaria siempre que hayan superado el mínimo exigido en el examen.

**PASADAS DOS HORAS DESDE EL INICIO DEL JUEGO SE DARÁ POR CONCLUIDA LA ACTIVIDAD** sea cual sea el número de equipos que ha terminado las pruebas.

Muchas gracias por vuestra participación.

**¡BUENA SUERTE!**

---

## ANEXO

---

En la siguiente tabla se muestra la correspondencia entre códigos obtenidos en las preguntas de tipo test y lugares en los que se puede encontrar la siguiente prueba.

<b>Código</b>	<b>Lugar del edificio</b>
AAA	Dirección del Centro. Secretaría de Dirección (Marisa y Celia)
AAB	Despacho 1D005 Secretaría Administrativa del Dpto. de Informática (Rebeca)
AAC	Despacho 1D006 Belarmino Pulido Junquera
ABA	Despacho 1D007 Mercedes Martínez González
ABB	Despacho 1D008 Jesús M <sup>a</sup> Vegas Hernández
ABC	Despacho 1D009 Carmen Hernández Díez
ACA	Despacho 1D010 César Llamas Bello
ACB	Despacho 1D012 Alejandra Martínez Monés
ACC	Despacho 1D013 Arturo González Escribano
BAA	Despacho 1D014 Manuel Barrio Solórzano
BAB	Despacho 1D015 M <sup>a</sup> Luisa González Díaz
BAC	Despacho 1D016 Quiliano Isaac Moro Sancho
BBA	Despacho 1D018 Carlos Enrique Vivaracho Pascual
BBB	Despacho 1D019 Miguel Ángel Laguna Serrano
BBC	Despacho 1D020 César Vaca Rodríguez
BCA	Despacho 1D021 M <sup>a</sup> Aránzazu Simón Hurtado
BCB	Despacho 1D022 M <sup>a</sup> Esperanza Manso Martínez
BCC	Despacho 1D023 César González Ferreras
CAA	Despacho 1D031 Grupo Universitario Informática (GUI)
CAB	Laboratorio 1L017 Javier Ramos López
CAC	Laboratorio 2L018 Mario Corrales
CBA	Despacho 2D050 Margarita Gonzalo Tasis
CBB	Despacho 2D054 Aníbal Bregón Bregón
CBC	Despacho 2D060 Yania Crespo González-Carvajal
CCA	Despacho 2D062 Javier Bastida Ibáñez
CCB	Despacho 2D064 Benjamín Sahelices Fernández
CCC	Despacho 2D065 Joaquín Adiego Rodríguez

# LOS CÓDIGOS DEL HAMBRE

La cosecha ha comenzado y en breve se celebrarán los primeros *Códigos del Hambre* que conmemorarán nuestra gran victoria en la guerra contra los rebeldes. Si quieres ser un tributo voluntario y eres **alumno del grupo T3** de Fundamentos de Programación solo tienes que seleccionar tu distrito y apuntarte en la siguiente lista.

Capitolio: Gobernantes

Distrito 1: Joyeros

Distrito 2: Canteros

Distrito 3: Electrónicos

Distrito 4: Pescadores

Distrito 5: Técnicos

Distrito 6: Transportistas

Distrito 7: Madereros

Distrito 8: Sastres

Distrito 9: Cerealeros

Distrito 10: Ganaderos

Distrito 11: Agricultores

Distrito 12: Mineros

Distrito 13: Rebeldes

# PRIMERA PRUEBA:

## LA ENTREVISTA

La entrevista televisada con Caesar Flickerman es el último paso antes del comienzo de los juegos y sabes que no es un mero trámite. La opinión de Caesar es muy importante y puede hacer que el público esté a tu favor o en tu contra. Si está de tu parte puede convertirse en un gran aliado y la mejor ayuda fuera de la arena, pero no es fácil caerle bien. Tu escolta de distrito, Effie Trinket, te dice que hay una frase a la que Caesar no puede resistirse, y que si se la dices durante la entrevista se convertirá en tu más ferviente admirador. El miedo que Effie tiene a las represalias del presidente Snow no le permite decirte cuál es esa frase, pero sí que te da unas pautas para que la obtengas tú mismo.

Te dice que si tienes una frase que está completamente en minúsculas del alfabeto inglés, puedes relacionarla con un número entero si a cada letra le asocias su número de orden empezando por el 1 ('a' → 1, 'b' → 2, ..., 'z' → 26) y al espacio en blanco un 0, y sumas el valor de cada carácter multiplicado por 27 elevado a su distancia al final de la frase. Por ejemplo la clave "la a" tendría asociado el número 236926 ya que:

	Valor del carácter	Potencia de 27 según su posición	Valor que aporta
'l'	12	$27^3 = 19683$	$12 \times 27^3 = 236196$
'a'	1	$27^2 = 729$	$1 \times 27^2 = 729$
' '	0	$27^1 = 27$	$0 \times 27^1 = 0$
'a'	1	$27^0 = 1$	$1 \times 27^0 = 1$
			SUMA = 236926

Gracias al ingenio que has desarrollado en las clases de *Códigos, Caza y Criptografía* que recibiste durante tu entrenamiento, te das cuenta de que esa relación es como trabajar en base 27 (en lugar de base 2 ó 10 ó 60), usando letras en lugar de dígitos, y sabes cómo implementar el procedimiento inverso, es decir, a partir de un número entero obtener la cadena de caracteres que tiene asociada. Y ese conocimiento será fundamental, ya que la única pista que te da Effie es el número 13891392744057. Así que si quieres el apoyo de Caesar en estos juegos deberás...

... **HALLAR EL TEXTO ASOCIADO AL NÚMERO 13891392744057**

Cuando sepas la frase exacta llévala como contraseña para recoger el enunciado de la siguiente prueba. Para saber dónde tienes que buscarlo tendrás que responder correctamente a las siguientes preguntas:

---

Si `x` es una variable de tipo `int` que tiene asignado un valor, la sentencia

```
if (2<x<1.5) System.out.println ("Si");
```

Es incorrecta en java porque...

- A** Ningún valor de `x` puede cumplir la condición
- B** No se puede comparar un `boolean` con un `double`
- C** No se puede comparar un `int` con un `double`

---

Para calcular el valor aproximado de una cierta función en el punto 10, se realiza un determinado proceso en el que `x` va de 0 hasta 10 en pequeños pasos. En teoría, cuanto menor sea el paso, más preciso será el valor obtenido. Se hacen experimentos con un código como el siguiente, para obtener el valor deseado después de 100 iteraciones y, sorprendentemente, produce un bucle infinito.

```
double paso = 0.1;
double x = 0, xfin = 10;
while (x != xfin) {
    // cálculos que se usan para obtener un valor y
    x = x + paso;
}
```

¿Cuál puede ser la explicación?

- A** Seguramente algún cálculo modifica el valor de `x` y no se han dado cuenta
- B** Es imposible que haya un bucle infinito con ese código.
- C** Es un problema de inexactitud en la representación en la máquina de los números reales.

---

¿Cuál de las siguientes sentencias no es equivalente a las otras dos? (Equivalente en el sentido de producir el mismo efecto o la misma modificación en el entorno)

- A** `{int x = -1; while (x < 0) x = x + 1;}`
- B** `{int x = 0;}`
- C** `{int x = -1; while (x < 0) ; x = x + 1;}`

# SEGUNDA PRUEBA:

## LA CORNUCOPIA

¡Enhorabuena! Ya tienes el apoyo de Caesar y el público te adora, pero ahora comienza la lucha de verdad. Te encuentras sobre la plataforma que te lleva directamente a la arena de juego desde donde tendrás acceso a la cornucopia. Es importante que llegues a ella antes que tus competidores, porque allí hay herramientas básicas para sobrevivir, pero no puedes bajarte de la plataforma antes de tiempo o te desintegrarán.

Tu mentor, Haymitch Abernathy, que ya ha pasado por esa situación, sabe que existe una fórmula basada en la **secuencia de Conway** para calcular el tiempo exacto que tienes que esperar antes de bajar de la plataforma sin correr peligro.

Gracias a la atención que has prestado en tus clases de *Fundamentos de Programación y Supervivencia* sabes que en la secuencia de Conway cada número se obtiene a partir del anterior ( $n$ ) aplicando las siguientes reglas:

- Si  $n$  es par, el siguiente número es  $n/2$
- Si  $n$  es impar, el siguiente es  $3n+1$
- Pero si  $n=1$ , no hay siguiente, la secuencia termina.

Por ejemplo el número 13 produce la cadena: 13 – 40 – 20 – 10 – 5 – 16 – 8 – 4 – 2 – 1, que consta de 10 elementos, por lo que se dice que tiene longitud 10.

Haymitch te indica que para saber los milisegundos que debes esperar tienes que...

**... CALCULAR EL NÚMERO ENTRE 1 Y 1000000 QUE GENERA LA CADENA DE CONWAY MÁS LARGA**

Cuando sepas el número exacto de milisegundos llévalo como contraseña para recoger el enunciado de la siguiente prueba. Para saber donde tienes que buscarlo tendrás que responder correctamente a las siguientes preguntas:

---

Una pequeña parte de la definición BNF para sentencias Java, sin expandir las expresiones `e` ni los identificadores `id`, podría ser la siguiente:

```
<sent> ::= { <sts> } | ; | while (e) <sent> | id = e;  
<sts> ::= <sts> <sent> |
```

¿Cuál de las siguientes sentencias no está descrita en esta definición?

- A** `while (e) {}`
- B** `while (e);`
- C** `while (e)`

---

Dada la declaración `String s;` y suponiendo que a la variable `s` ya se le ha asignado algún valor, no es posible realizar la asignación `s.length() = 7;` a pesar de que los Strings tienen longitud variable. ¿Cuál es la razón?

- A** `s.length()` no es una variable sino una expresión y por lo tanto no puede ser el destino de una asignación.
- B** `s.length()` es realmente un apuntador a una zona de memoria y Java se protege de las asignaciones directas a memoria prohibiendo asignaciones a punteros.
- C** El tipo `String` implantado en Java por defecto tiene un número fijo de elementos, por lo que se prohíbe la asignación a la variable `s.length()`. Se dice que es inmutable.

---

Suponiendo `b` y `c` variables definidas y dada la siguiente iteración:

```
while (b != 0 && c < 5) {  
    b--;  
    c++;  
    //otras sentencias que no modifican ni b ni c  
}
```

Cuándo el bucle termine, ¿qué condición se puede garantizar?

- A** `b = 0` y `c = 5`
- B** `b = 0` o `c = 5`
- C** `b = 0` o `c >= 5`



# TERCERA PRUEBA:

## LA ARENA

Llevas semanas oculto y hace tiempo que no escuchas cañonazos. Si tus cálculos no fallan, solo quedáis vivos otro tributo y tú. Aunque estás exhausto, podrías permanecer así indefinidamente ya que la caza es abundante en este lugar y tú eres hábil con el arco. Pero la organización de los juegos no puede tolerar esta situación y plantea una solución al conflicto: el tributo que antes resuelva un acertijo será el ganador de los juegos (el otro será sacrificado).

La adivinanza se basa en calcular la diferencia entre el cuadrado de la suma y la suma de los cuadrados de los números menores o iguales que un número dado. Por ejemplo, si el número dado fuese 10, el resultado sería 2640 (la diferencia entre 3025 y 385), ya que:

$$(1+2+3+4+5+6+7+8+9+10)^2 = 3025$$
$$1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+10^2 = 385$$

La propuesta final que Séneca Crane, el organizador de los juegos os hace es...

**... CALCULAR LA DIFERENCIA DETALLADA ANTERIORMENTE UTILIZANDO CÓMO NÚMERO DADO EL VALOR 100000**

Si no quieres ser sacrificado tendrás que ser rápido y eficiente al realizar el cálculo. Aunque parece sencillo, nada en Panem puede serlo; de hecho, te das cuenta de que para ese valor (100000) el problema no es trivial. Cuando estás a punto de darte por vencido, uno de los patrocinadores que has obtenido gracias a tu fama te envía una pista...

**... PUEDE AYUDARTE HACER LO MISMO CON LOS VALORES 10, 100, 1000 Y 10000 Y BUSCAR UN PATRÓN EN LA RESPUESTA**

¿Ya tienes esa diferencia? Pues ahora solo tienes que coger el arco que se encuentra en la sala Hedy Lamar y entregárselo a la jefa de vigilantes...

...Mucha suerte