FCEyN - UBA - Segundo Cuatrimestre 2013

Álgebra I
Práctica 2 - Números naturales e inducción

1. i) Reescribir cada una de las siguientes sumas usando el símbolo de sumatoria

(a) $1+2+3+4+\cdots+100$,
(b) $1+2+4+8+16+\cdots+1024$,
(c) $1+(-4)+9+(-16)+25+\cdots+(-144)$,
(d) $1+9+25+49+\cdots+441$,
(e) $1+3+5+\cdots+(2n+1)$,
(f) $n+2n+3n+\cdots+n^2$.

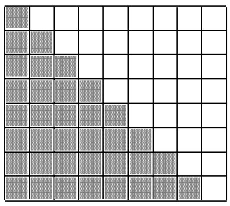
ii) Reescribir cada una de los siguientes productos usando el símbolo de productoria y/o de factorial

(a) $5\cdot 6\cdots 99\cdot 100$, (b) $1\cdot 2\cdot 4\cdot 8\cdot 16\cdots 1024$, (c) $n\cdot 2n\cdot 3n\cdots n^2$.

2. Escribir los dos primeros y los dos últimos términos de las expresiones siguientes

i) $\displaystyle\sum_{i=6}^{n} 2(i-5)$, ii) $\displaystyle\sum_{i=n}^{2n} \frac{1}{i(i+1)}$, iii) $\displaystyle\sum_{i=1}^{n} \frac{n+i}{2i}$, iv) $\displaystyle\sum_{i=1}^{n^2} \frac{n}{i}$, v) $\displaystyle\prod_{i=1}^{n} \frac{n+i}{2i-3}$.

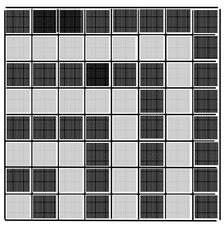
3. i) Probar que, $\forall\, n\in\mathbb{N}$, $\displaystyle\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ contando de dos maneras la cantidad de cuadraditos sombreados del diagrama



ii) Deducir que, $\forall\, n\in\mathbb{N}$, $2+4+6+\cdots+2n = n(n+1)$.

4. Probar que, $\forall\, n\in\mathbb{N}$, $\displaystyle\sum_{i=1}^{n}(2i-1) = n^2$:

i) contando de dos maneras la cantidad total de cuadraditos del diagrama



ii) usando el ejercicio 3,

iii) usando el principio de inducción.

5. Calcular

i) $\displaystyle\sum_{i=1}^{n}(4i+1)$,

ii) $\displaystyle\sum_{i=6}^{n} 2(i-5)$.

6. *(Suma de cuadrados y de cubos)* Probar que para todo $n\in\mathbb{N}$ se tiene

i) $\displaystyle\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$,

ii) $\displaystyle\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$.

7. Probar que para todo $n \in \mathbb{N}$ se tiene

i) $\displaystyle\sum_{i=1}^{n} (-1)^{i+1}\, i^2 = \frac{(-1)^{n+1} n(n+1)}{2}$,

ii) $\displaystyle\sum_{i=0}^{n} \frac{-1}{4i^2-1} = \frac{n+1}{2n+1}$,

iii) $\displaystyle\sum_{i=1}^{n} (2i+1)\, 3^{i-1} = n\, 3^n$,

iv) $\displaystyle\sum_{i=1}^{n} \frac{i\, 2^i}{(i+1)(i+2)} = \frac{2^{n+1}}{n+2} - 1$,

v) $\displaystyle\prod_{i=1}^{n} \frac{n+i}{2i-3} = 2^n(1-2n)$.

8. Sea $(a_n)_{n\in\mathbb{N}}$ una sucesión de números reales. Probar que $\displaystyle\sum_{i=1}^{n}(a_{i+1}-a_i) = a_{n+1} - a_1$.

9. Calcular

i) $\displaystyle\sum_{i=1}^{n} \frac{1}{i(i+1)}$ (Sugerencia: $\displaystyle\frac{1}{i(i+1)} = \frac{1}{i} - \frac{1}{i+1}$),

ii) $\displaystyle\sum_{i=1}^{n} \frac{1}{(2i-1)(2i+1)}$ (Sugerencia: calcular $\displaystyle\frac{1}{2i-1} - \frac{1}{2i+1}$).

10. Calcular $\displaystyle\sum_{i=1}^{n} \frac{(-1)^i i}{(2i-1)(2i+1)}$, $\forall\, n \in \mathbb{N}$.

11. Probar que para todo $n \geq 3$ vale que

i) la cantidad de diagonales de un polígono de n lados es $\displaystyle\frac{n(n-3)}{2}$,

ii) la suma de los ángulos interiores de un polígono de n lados es $\pi\,(n-2)$.

12. Probar que las siguientes desigualdades son verdaderas para todo $n \in \mathbb{N}$

i) $n < 2^n$,

ii) $3^n + 5^n \geq 2^{n+2}$,

iii) $3^n \geq n^3$,

iv) $\displaystyle\sum_{i=1}^{n} \frac{n+i}{i+1} \leq 1 + n(n-1)$,

v) $\displaystyle\sum_{i=n}^{2n} \frac{i}{2^i} \leq n$,

vi) $\displaystyle\sum_{i=1}^{2^n} \frac{1}{2i-1} > \frac{n+3}{4}$.

13. Probar que, $\forall\, n \in \mathbb{N}$, vale

i) $\displaystyle n! \geq \frac{3^{n-1}}{2}$,

ii) $\displaystyle\sum_{i=1}^{n} \frac{1}{i!} \leq 2 - \frac{1}{2^{n-1}}$.

14. Probar que $\displaystyle\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0$ (no hace falta usar inducción).

15. i) ¿Cuántos subconjuntos de 4 elementos tiene el conjunto $\{1, 2, 3, 4, 5, 6, 7\}$?

ii) ¿Y si se pide que 1 pertenezca al subconjunto?

iii) ¿Y si se pide que 1 no pertenezca al subconjunto?

iv) ¿Y si se pide que 1 ó 2 pertenezcan al subconjunto, pero no simultaneamente los dos?

16. En este ejercicio no hace falta usar inducción: se puede pensar en el significado del número combinatorio en cuanto a cantidad de subconjuntos de cierta cantidad de elementos en un conjunto dado.

i) Probar que $\displaystyle\sum_{k=0}^{2n}\binom{2n}{k} = 4^n$ y deducir que $\displaystyle\binom{2n}{n} < 4^n$.

ii) Calcular $\displaystyle\sum_{k=0}^{n}\binom{2n+1}{k}$.

iii) Probar que $\displaystyle\sum_{k=1}^{n} k\binom{n}{k} = n\,2^{n-1}$ $\left(\text{sug: } k\binom{n}{k} = n\binom{n-1}{k-1}\right)$.

iv) Probar que $\displaystyle\sum_{k=0}^{n}\binom{n}{k}^2 = \binom{2n}{n}$ $\left(\text{sug: } \binom{n}{k} = \binom{n}{n-k}\right)$.

17. Derivar a izquierda y derecha la igualdad $(x+1)^n = \sum_{k=0}^{n}\binom{n}{k}x^k$ y evaluar lo obtenido en $x = 1$. ¿Qué se obtiene? (Comparar con el Ej. 16(iii).)

18. Sea $a, b \in \mathbb{R}$. Probar que para todo $n \in \mathbb{N}$, $a^n - b^n = (a-b)\displaystyle\sum_{i=1}^{n} a^{i-1}b^{n-i}$. Deducir la fórmula de la serie geométrica: para todo $a \neq 1$, $\displaystyle\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$.

19. Sea $a \in \mathbb{R}$, $a \geq -1$. Probar que, $\forall\, n \in \mathbb{N}$, $(1+a)^n \geq 1 + na$.
¿En qué paso de la demostración se usa que $a \geq -1$?

20. Probar que

i) $n! \geq 3^{n-1}$, $\forall\, n \geq 5$,

ii) $3^n - 2^n > n^3$, $\forall\, n \geq 4$,

iii) $\displaystyle\sum_{i=1}^{n}\frac{3^i}{i!} < 6n - 5$, $\forall\, n \geq 3$,

iv) $\displaystyle\binom{2n}{n} > n\,2^n$, $\forall\, n \geq 4$.

21. i) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión de números reales definida recursivamente por

$$a_1 = 5, \qquad a_{n+1} = 3a_n - 2^n \qquad (n \in \mathbb{N})$$

Probar que $a_n = 2^n + 3^n$.

ii) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión de números reales definida recursivamente por

$$a_1 = 2, \qquad a_{n+1} = 2\,n\,a_n + 2^{n+1}n! \qquad (n \in \mathbb{N})$$

Probar que $a_n = 2^n\,n!$.

iii) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión de números reales definida recursivamente por

$$a_1 = 0, \qquad a_{n+1} = a_n + n(3n+1) \qquad (n \in \mathbb{N})$$

Probar que $a_n = n^2(n-1)$.

iv) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión de números reales definida recursivamente por

$$a_1 = 2, \qquad a_{n+1} = 4a_n - 2\frac{(2n)!}{(n+1)!\,n!} \qquad (n \in \mathbb{N})$$

Probar que $a_n = \displaystyle\binom{2n}{n}$.

22. Para cada una de las siguientes sucesiones definidas por recursión, hallar una fórmula para el término general y demostrar su validez

i) $a_1 = 1$, $a_{n+1} = (1 + \sqrt{a_n})^2$ $(n \in \mathbb{N})$

ii) $a_1 = 3$, $a_{n+1} = 2a_n + 3^n$ $(n \in \mathbb{N})$

iii) $a_1 = 1$, $a_{n+1} = n\, a_n$ $(n \in \mathbb{N})$

iv) $a_1 = 2$, $a_{n+1} = 2 - \dfrac{1}{a_n}$ $(n \in \mathbb{N})$

23. Hallar una fórmula para el término general de la sucesión $(a_n)_{n\in\mathbb{N}}$ definida por

i) $a_1 = 1, \;\; a_{n+1} = a_n + n^3$ $(n \in \mathbb{N})$.

ii) $a_1 = 1, \;\; a_{n+1} = a_n + (-1)^{n+1} n^2$ $(n \in \mathbb{N})$.

iii) $a_1 = 3, \;\; a_{n+1} = a_n + (2n+1)3^{n-1}$ $(n \in \mathbb{N})$.

(Sugerencia: usar los Ejercicios 8, 6 y 7.)

24. Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión definida por

$$a_1 = 1, \qquad a_{n+1} = a_n + n \cdot n! \qquad (n \in \mathbb{N})$$

Probar que $a_n = n!$ y, aplicando el Ej. 8, calcular $\displaystyle\sum_{i=1}^{n} i \cdot i!$

25. Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión definida por

$$a_1 = 1, \qquad a_{n+1} = a_n + 3n^2 + 3n + 1 \qquad (n \in \mathbb{N})$$

Probar que $a_n = n^3$ y, aplicando el Ej. 8, calcular de otra manera $\displaystyle\sum_{i=1}^{n} i^2$ (c.f. Ej. 6).

26. Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión definida por

$$a_1 = 1, \qquad a_{n+1} = \frac{2n+1}{n+1}\, a_n \qquad (n \in \mathbb{N})$$

i) Probar que $a_n \le \dfrac{1}{2n}\dbinom{2n}{n}$ para todo $n \in \mathbb{N}$

ii) Probar que $a_n > \dfrac{1}{3^{n-1}}\dbinom{2n}{n}$ para todo $n \ge 3$

27. Para cada una de las siguientes sucesiones definidas por recursión, conjeturar una fórmula para el término general y probar su validez

i) $a_1 = 1, \; a_2 = 2, \qquad a_{n+2} = n\, a_{n+1} + 2(n+1)a_n \qquad (n \in \mathbb{N})$

ii) $a_1 = 1, \; a_2 = 4, \qquad a_{n+2} = 4\sqrt{a_{n+1}} + a_n \qquad (n \in \mathbb{N})$

iii) $a_1 = 1, \qquad a_{n+1} = 1 + \displaystyle\sum_{i=1}^{n} i\, a_i \qquad (n \in \mathbb{N})$

iv) $a_1 = \frac{1}{2}, \qquad a_{n+1} = \frac{1}{2}\left(1 - \displaystyle\sum_{i=1}^{n} a_i\right) \qquad (n \in \mathbb{N})$

28. i) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión definida por

$$a_1 = 1, \quad a_2 = 3, \qquad a_{n+2} = a_{n+1} + 5a_n \qquad (n \in \mathbb{N})$$

Probar que $a_n < 1 + 3^{n-1}$ para todo $n \in \mathbb{N}$.

ii) Sea $(a_n)_{n\in\mathbb{N}}$ la sucesión definida por

$$a_1 = 1, \quad a_2 = \frac{3}{2}, \qquad a_{n+2} = a_{n+1} + \frac{2n+1}{n+2}\, a_n \qquad (n \in \mathbb{N})$$

Probar que $a_n > n + \frac{1}{3}$ para todo $n \in \mathbb{N}$, $n \ge 4$.

29. Cálculo de a^{2^n} para un número a dado y $n \in \mathbb{N}$ (más adelante se verá el cálculo rápido de a^n)

Se asume un modelo ideal donde dados dos números b y c, se puede calcular $b \cdot c$ exactamente realizando una sola operación.

i) Algoritmo recursivo *secuencial*: Calcular las distintas potencias de a mediante

$$a^{k+1} = a \cdot a^k, \ \forall\, k \geq 1$$

Esribir el algoritmo en pseudocódigo y en algún lenguaje funcional.
¿Cuántas operaciones se realizaron para calcular a^n? ¿y a^{2^n}?

ii) Algoritmo recursivo *dividir y conquistar*: Calcular a^{2^n} mediante

$$a^{2^1} = a \cdot a \quad , \quad a^{2^{k+1}} = a^{2^k} \cdot a^{2^k}, \ \forall\, k \geq 1$$

Esribir el algoritmo en pseudocódigo y en algún lenguaje funcional.
¿Cuántas operaciones se realizaron para calcular a^{2^n}?

30. En este ejercicio se admite idealísticamente que sumar, multiplicar, dividir, y sacar raíz cuadrada requieren cada uno una sola operación. Comparar la cantidad de cuentas que se tienen que hacer para calcular el 2^n-ésimo número de Fibonacci F_{2^n} usando por un lado la definición recursiva y por otro lado la fórmula general obtenida. (Más adelante se hará para F_n en general.)

31. Algoritmos de ordenación de datos:

Dada una lista ordenada $(a(1), \dots, a(n))$ de n números reales, se la quiere ordenar de menor a mayor. Por ejemplo dada la lista $(4, 3, 5, 7, 2, 9, 1, 8)$ se quiere obtener la lista $(1, 2, 3, 4, 5, 7, 8, 9)$ haciendo comparaciones entre elementos ¿$a(i) < a(j)$? y en función de la respuesta intercambiando si necesario los elementos comparados.

i) Algoritmo ingenuo ("burbujeo"): Compara el 1er elemento de la lista con el 2do intercambiándolos si necesario, luego pasa al 2do y hace lo mismo con el siguiente, luego al 3ro etc. hasta recorrer todos los elementos de la lista (una pasada). En una pasada por todos los elementos de la lista, el elemento más grande quedará entonces a la derecha ¿Por qué? Luego repite el procedimiento desde el comienzo hasta no haber producido ningún cambio en una pasada. En ese punto la lista estará ordenada ¿Por qué?
Por ejemplo una pasada del ejemplo de arriba da:

$$\underline{4}\ \underline{3}\ 5\ 7\ 2\ 9\ 1\ 8 \ \rightarrow\ 3\ \underline{4}\ \underline{5}\ 7\ 2\ 9\ 1\ 8 \ \rightarrow\ 3\ 4\ \underline{5}\ \underline{7}\ 2\ 9\ 1\ 8 \ \rightarrow$$

$$3\ 4\ 5\ \underline{7}\ \underline{2}\ 9\ 1\ 8 \ \rightarrow\ 3\ 4\ 5\ 2\ \underline{7}\ \underline{9}\ 1\ 8 \ \rightarrow\ 3\ 4\ 5\ 2\ 7\ \underline{9}\ \underline{1}\ 8 \ \rightarrow$$

$$3\ 4\ 5\ 2\ 7\ 1\ \underline{9}\ \underline{8} \ \rightarrow\ 3\ 4\ 5\ 2\ 7\ 1\ 8\ 9$$

¿Cuántas comparaciones se usan en la primer pasada de esta lista? ¿Y en la segunda? ¿Cuántas comparaciones son necesarias para ordenar esta lista?
¿Cuántas comparaciones se usan en la primer pasada de una lista de n elementos? ¿Y en la segunda? ¿Cuántas comparaciones son necesarias para ordenar una lista de n elementos?

ii) Algoritmo "merge" (fusiona): Funciona en 3 etapas usando el importantísimo mecanismo de "dividir y conquistar". Por simplicidad lo vamos a aplicar aquí para listas de longitud 2^n:

(a) "Divide": divide la lista en dos sublistas de tamaño 2^{n-1}.
(b) "Conquista": ordena cada sublista por separado (utilizando el mismo algoritmo para listas de tamaño 2^{n-1}).
(c) "Fusiona": fusiona en forma adecuada las dos sublistas en una sola.

Ejemplo: Sea la lista $(4, 3, 5, 7, 2, 9, 1, 8)$ de longitud 2^3:

(a) "Divide": divide la lista en las dos sublistas $(4, 3, 5, 7)$ y $(2, 9, 1, 8)$ de longitud 2^2.
(b) "Conquista": Aplica el algoritmo para estas dos sublistas (en forma recursiva, o sea partiendo cada una de ellas en dos sublistas de longitud 2) y termina obteniendo: $(4, 3, 5, 7) \rightarrow (3, 4, 5, 7)$ y $(2, 9, 1, 8) \rightarrow (1, 2, 8, 9)$.

(c) "Fusiona": fusiona inteligentemente las dos sublistas, como lo hacemos naturalmente sin pensar: Se compara el primer elemento de cada sublista, se pone el más chico primero, luego se compara los dos primeros elementos que quedaron (sin ese) de cada lista, y se repite el procedimiento hasta agotar. Para fusionar $(3, 4, 5, 7)$ y $(1, 2, 8, 9)$ se obtiene

$$
\begin{array}{lcl cl}
\boxed{3}\ 4\ 5\ 7 & \text{y} & \boxed{1}\ 2\ 8\ 9 & \to & \boxed{1} \\
\boxed{3}\ 4\ 5\ 7 & \text{y} & \cancel{1}\ \boxed{2}\ 8\ 9 & \to & 1\ \boxed{2} \\
\boxed{3}\ 4\ 5\ 7 & \text{y} & \cancel{1}\ \cancel{2}\ \boxed{8}\ 9 & \to & 1\ 2\ \boxed{3} \\
\cancel{3}\ \boxed{4}\ 5\ 7 & \text{y} & \cancel{1}\ \cancel{2}\ \boxed{8}\ 9 & \to & 1\ 2\ 3\ \boxed{4} \\
\cancel{3}\ \cancel{4}\ \boxed{5}\ 7 & \text{y} & \cancel{1}\ \cancel{2}\ \boxed{8}\ 9 & \to & 1\ 2\ 3\ 4\ \boxed{5} \\
\cancel{3}\ \cancel{4}\ \cancel{5}\ \boxed{7} & \text{y} & \cancel{1}\ \cancel{2}\ \boxed{8}\ 9 & \to & 1\ 2\ 3\ 4\ 5\ \boxed{7} \\
\cancel{3}\ \cancel{4}\ \cancel{5}\ \cancel{7} & \text{y} & \cancel{1}\ \cancel{2}\ \boxed{8}\ 9 & \to & 1\ 2\ 3\ 4\ 5\ 7\ \boxed{8} \\
\cancel{3}\ \cancel{4}\ \cancel{5}\ \cancel{7} & \text{y} & \cancel{1}\ \cancel{2}\ \cancel{8}\ \boxed{9} & \to & 1\ 2\ 3\ 4\ 5\ 7\ 8\ \boxed{9}
\end{array}
$$

Esribir el algoritmo en pseudocódigo y en algún lenguaje funcional.
¿Cuántas comparaciones son necesarias para ordenar la lista original del ejemplo?
¿Cuántas comparaciones son necesarias para ordenar una lista de 2^n elementos?

iii) Algoritmo "quick sort": Es el algoritmo de ordenamiento más rápido conocido "en la práctica". También utiliza el mecanismo de dividir y conquistar, eligiendo un pivote p en la lista (que puede ser el primer elemento, o elegido al azar, o de alguna otra manera): luego de una primer pasada el algoritmo colocará el pivote elegido en su lugar correcto en la lista, colocando todos los menores que él a su izquierda y todos los mayores a su derecha. Luego ordena las dos sublistas (de la izquierda y de la derecha) por separado. La forma establecida de colocar el pivote en su lugar correcto (que resulta más conveniente que cualquier forma ingenua) es aplicando lo que se llama "partitioning in place", trabajando con dos punteros, i por izquierda (que se mueve de izquierda a derecha) y d por derecha (que se mueve de derecha a izquierda) y compara los elementos correspondientes a esos punteros. Por convención el pivote p se intercambia con el elemento $a(n)$ de más a la derecha de la lista, para quedar él a la derecha. Se tiene entonces $p = a(n)$.

- Fijar los punteros i y d a la izquierda y a la derecha de la lista (sin el pivote).
- Incrementar i hasta que coincida con un elemento $a(i)$ mayor que p
- Disminuir d hasta que coincida con un elemento $a(d)$ menor que p.
- Intercambiar $a(i)$ y $a(d)$.
- Repetir hasta que i y d se crucen (es decir $i \geq d$).
- Intercambiar el pivote $p = a(n)$ con $a(i)$.

En el ejemplo anterior, si el puntero p elegido es el 4, primero se intercambia con el 8 de la derecha:

$$\boxed{4}\ \ 3\ \ 5\ \ 7\ \ 2\ \ 9\ \ 1\ \ 8 \to 8\ \ 3\ \ 5\ \ 7\ \ 2\ \ 9\ \ 1\ \ \boxed{4}$$

Y luego la primer pasada del algoritmo funciona de la manera siguente:

$$\overset{i}{\underline{8}}\ 3\ 5\ 7\ 2\ 9\ \overset{d}{\underline{1}}\ \boxed{4} \to \overset{i}{\underline{1}}\ 3\ 5\ 7\ 2\ 9\ \overset{d}{\underline{8}}\ \boxed{4} \to 1\ \overset{i}{\underline{3}}\ 5\ 7\ 2\ 9\ \overset{d}{\underline{8}}\ \boxed{4} \to$$

$$1\ 3\ \overset{i}{\underline{5}}\ 7\ 2\ 9\ \overset{d}{\underline{8}}\ \boxed{4} \to 1\ 3\ \overset{i}{\underline{5}}\ 7\ 2\ \overset{d}{\underline{9}}\ 8\ \boxed{4} \to 1\ 3\ \overset{i}{\underline{5}}\ 7\ \overset{d}{\underline{2}}\ 9\ 8\ \boxed{4} \to$$

$$1\ 3\ \overset{i}{\underline{2}}\ 7\ \overset{d}{\underline{5}}\ 9\ 8\ \boxed{4} \to 1\ 3\ 2\ \overset{i}{\underline{7}}\ \overset{d}{\underline{5}}\ 9\ 8\ \boxed{4} \to 1\ 3\ 2\ \overset{i\,d}{\underline{7}}\ 5\ 9\ 8\ \boxed{4} \to$$

$$1\ 3\ \overset{d}{\underline{2}}\ \overset{i}{\underline{7}}\ 5\ 9\ 8\ \boxed{4} \to 1\ 3\ \overset{d}{2}\ \overset{i}{\boxed{4}}\ 5\ 9\ 8\ 7$$

Entender cómo funciona este algoritmo.
Este algoritmo tiene la particularidad que "en promedio" funciona muy rápido (como el algoritmo merge, incluso más rápido), aunque para algunas listas iniciales requiere una cantidad de intercambios similar a la del algoritmo burbujeo.