Calculator Polinomial

Documentatie



**Profesor Coordonator:**  **Student:**

Dr.Pop Cristina Bianca Vaduva Calin Liviu

**Grupa**: 30228 **An:** 2

Cuprins

1. Obiectiv
2. Studiul Problemei
3. Proiectarea

3.1 Diagrama UML

3.2 Clase

3.3 Metode

3.4 GUI

1. Implementare
   1. Monom
   2. Polinom
   3. Operatii
   4. View
   5. Controlle
2. Rezultate
3. Concluzii
4. Obiectivul temei

Scopul principala al temei este implementarea si proiectarea unui Calculator polinomial care va fi implementat cu ajutorul unei interfete grafice in care vor fi introduse 2 polinoame si asupra carora se vor executa operatiile de adunare, scadere, inmultire, impartire cat si derivarea si integrarea unui al polinom introdus separat .

Obiectivele secundare pentru inplementare sunt:

* Creearea de clase si metode pentru lucru cu polinoame
* Descrierea operatiilor necesare lucrului cu polinoame
* Afisarea rezultatului in urma operatiilor
* Testarea aplicatiei

Pentru indeplinirea obiectivului principal este nevoie de o *analiza* a problemei atat pentru scenariile de functionare cat si pentru cazurile de utilizare a aplicatiei, informatii care voi fi amanuntite in *capitolul 2;* Tot pentru a indeplini scopul temei este necersara atat *proiectarea* (*capitolul 3*) cat si *impementarea* proiectului (*capitolul 4*) iar in final testarea acestuia care va fi prezentata in *capitolul 5.*

1. Analiza Problemei

Pentru a putea implementa si proiecta Calculatorul de polinoame trebuie in

primul rand sa stim ce este un polinom. Un polinom este o expresie matematica construita din monomane. Monoamele la randu lor sunt alcatuite dintr-o constanta numita coeficient inmultite cu o variabile(X de obicei) la o putere/exponent numit grad, astfel o forma obisnuita a unui monom sau polinom este de forma:

Monom: X^2

Polinom: X^2+3\*X^1+5

In urma intelegerii conceptului de polinom este necesara proiectarea unei

Interfete usor de utilizat si de inteles pentru utilizator , pentru aceasta am creeat o

diagrama Use-Case. Aceasta este prezentata mai jos.

Cerintele functionale ale problemei sunt:

* Calculatorut trebuie sa permita introducerea a doua polinoame la alegere
* Calculatorul trebuie sa permita utilizatorului sa selecteze operatia dorita
* Calculatorul trebuie sa poata aduna doua polinoame
* Calculatorul trebuie sa poata scadea doua polinoame
* Calculatorul trebuie sa poata inmultii doua polinoame
* Calculatorul trebuie sa poata deriva un polinom
* Calculatorul trebuie sa poata integra un polinom

Cerintele non-functionale sunt:

* Interfata grafica trebuie sa fie atat usor de inteles cat si de utilizat
* Calculatorul ar trebuii sa lucreze rapid si eficient, iar rezultatele sa fie cat mai vizibile pentru utilizator.

Diagrama Use-Case:

Calculator Polinomial

Introducere Polinom

Adunare

Scadere

Inmultire

Integrare

Derivare

Impartire

User

Vizualizare Rezultat

Descriere diagrama Use-Case:

Actorul principal este User-ul

Scenariul corect de utilizare:

1. Utilizatorul introduce doua polinoame in interfata grafica pentru operatiile de adunare, scadere, inmultire, impartiire si respectiv un polinom pentru operatiile de derivare si integrare.
2. Utilizatorul selecteaza butonul specific unei operatii pe care doreste sa o efectueze.
3. Programul verifica daca cele doua polinoame sunt introduse corect dupa forma “a1X^b1+a2X^b2+....” .
4. Calculatorul de polinoame face calculul operatiei selectate
5. Ulterior rezultatul calculului este afisat in casuta text corespunzatoare.

Scenariu alternativ de functionare:

6. Utilizatorul introduce unu sau polinoame incorect .

7. Programul ofera o eroare

8. Se revine la pasul numarul 1 .

Use-case pentru operatiile de integrare sau derivare:

Actorul principal este User-ul

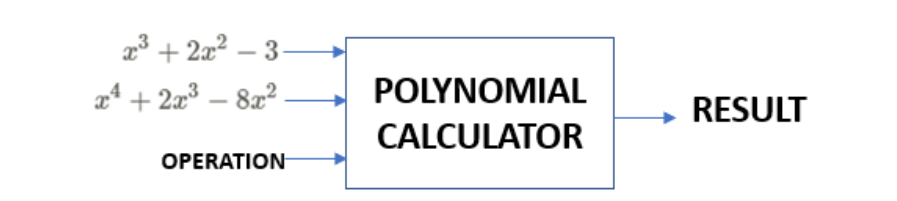
Scenariul corect de utilizare:

1. Utilizatorul introduce un polinoam in casuta text denumita Polinom
2. Utilizatorul selecteaza dintre butoanele de derivare sau integrare a polinomului introdus la pasul 1
3. Programul verifica daca polinomul este itrodus corespunzator
4. Se face operatia selectata mai sus
5. In final rezultatu elste afisat in casuta text corespunzatoare

Scenariul alternativ de functionare

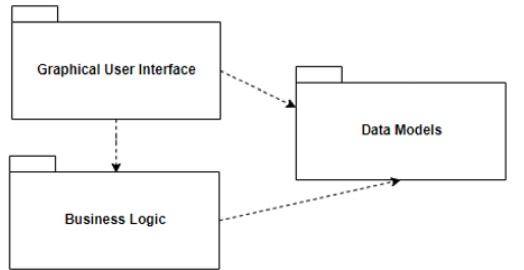
1. Utilizatorul introduce incorect polinomul.
2. Programul da o eroare
3. Se revine la pasul 1.
4. Proiectare

Schema bloc a sistemului are 3 intrari pe care se polinoamele pe care dorim

le introducem si operatia de efectuat si o iesire care este rezultatul operatiei dintre cele 2 polinoame.

Aplicatia este impartita in 3 pachete : GUI, Logica si Model. Pachetul GUI contine clasele Controller si View si metodele lor, in care este descris codul pentru interfata grafica . In pachetul Logica se afla o singra clasa , numita Operatii; in ea se afla toate metodele care fac operatiile pe polinoame posibile , iar in pachetul Model se afla clasele Monom si Polinom in care deifnesc polnoamele, monoamele si lucrul cu ele. Diagrama UML pentru pachete este prezentata mai jos.

* Diagrama UML pentru pachete:



use

use

use

package

Logica

package

Model

package

GUI

1. Implementare

In capitolul acesta sunt prezentate pachetele si implementarea claselor si a metodelor corespondente fiecari pachet.

1. **Pachetul Model**:

1.Clasa Monom

Clasa monom contine metodele pentru extragerea monoamelor din polinoame. Aceasta are 3 atribute: -grad, de tipul in, acesta retine exponentul monomului

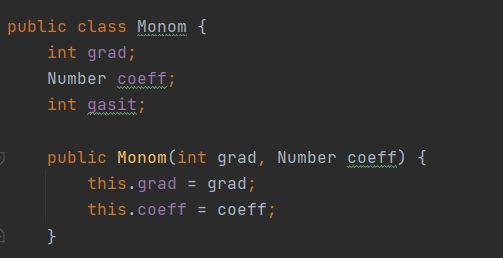
-coeff, de tipul Number, acesta retine coeficientul fiecarui momon definit. Coeficientul este de tipul Number chiar daca nu este obligatoriu lucrul cu coeficienti reali la metoda de

integrare cand acesta se imparte la gradul monomului+1, acea impartire mai mult ca si probabil va da cu rest, astfel daca monomul este definit ca si number vom putea sa facem si operatii cu virgula.

-gasit, acest atribut este utilizat pentru a face adunarea respectiv scaderea polinoamelor.

Metodele acestei clase sunt alcatuire din getter-ul si setter-ul atributelor pentu a se putea obtine si modifica atat gradul cat si coeficientrul, iar atributul gasit are un rol foarte important la operatia de adunare si scadere.

Constructorul clasei contine ca parametrii doar pentru grad si coeficient, deoarece pentru a initializa un obiect de tipul monom avem nevoie doar de grad si de coeficientul acestuia , atributul gasit il folosim doar cand facem operatii cu polinomul.



Tot clasa monom mai are si metoda toString prezentata mai jos care face posibila printarea rezultatului la toate operatiile din aplicatie . Codul este atasat mai jos .

*public String toString() {*

*if (coeff.doubleValue() == 0) {*

*return "";*

*} else {*

*if (grad == 0) {*

*if (coeff.doubleValue() > 0)*

*return "+" + coeff;*

*else*

*return coeff+"";*

*} else {*

*if (coeff.doubleValue() > 0)*

*return "+" + coeff + "X^" + grad;*

*else*

*return coeff + "X^" + grad;*

*}*

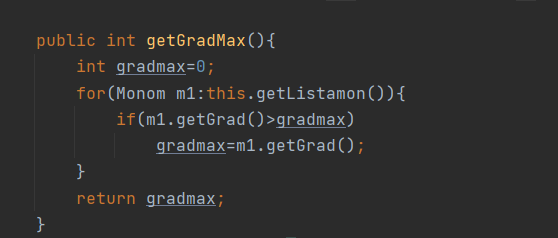
*}*

*}*

2.Clasa Polinom

Clasa polinom contine o lista de monoame care formeaza polinomul propriu-zis. Ca metode, acesta are metoda getList care returneaza lista de monoame, metoda addMonom care introduce un obiect de tip monom in lista de monoame iar pe langa metoda toString care ajuta la afisarea polinomuli am mai implementat metoda getGradMax(), aceasta returneaza gradul cel mai mare dintre monoamele polinomului.

Metoda folosete un for de tipul for each pentru a face parcurgerea prin lista de monoame, si retine in variabila gradmax gradul monomului cu cel mai mare grad; aceasta metoda este utilizata in special la inmultirea si impartirea polinoamelor.



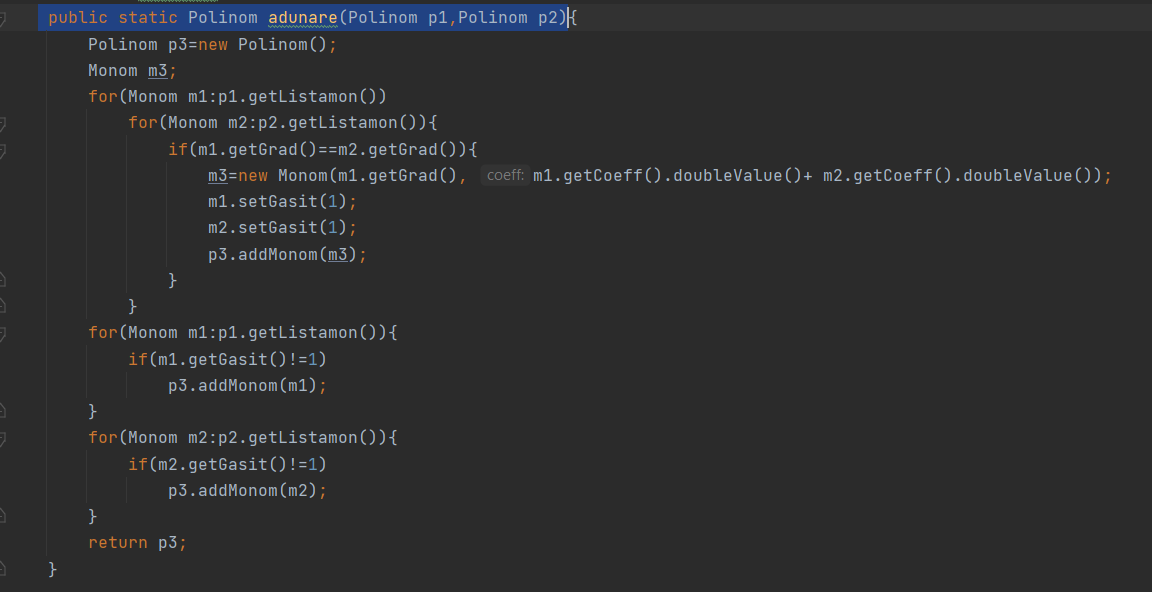
1. **Pachetul Logica**
2. Clasa Operatii

Clasa operatii este singura clasa din pachetul Logica, iar dupa cum ii spune

si numele este clasa in care se fac operatiile pe polinoame.

Metodele componente sunt statice si aceste sunt:

* *public static Polinom adunare(Polinom p1, Polinom p2)* , aceasta metoda se ocupa de adunarea a doua polinoame. In interiorul metodei, parcurgem fiecare polinom prin intermediul unui for de tipul for-each, pentru a gasii monoamele din fiecare polinom cu acelasi grad. Dupa ce gasim monoamele, ne folosim de atributele acestora, modificam valoarea atributului “gasit” a monoamelor gasite iar apoi adunam gradele monoamelor. In codul de mai jos se poate observa mai clar toata implementarea acestei metode. Aceasta metoda returneaza polinomu rezultat din adunarea celor doua polinoame initiale.

**

X y a s a w e r t y u y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y

* *public static Polinom scadere(Polinom p1, Polinom p2)*

*-*metoda de scaderea a doua polinoame este asemanatoare cu adunarea acestora; Se dau ca si parametrii cele doua polinoame. Prin doua for-uri de tip for-each asociem fiecare monom din primul cu fiecare monom din al 2-lea polinom si verificam daca au gradul la fel. In cazul in care se gasesc, notam atributele monoamelor prin gasit=’1’ si introducem in noul polinom monomul rezultat din scaderea monomului polinomului 2 din primul. Ulterior sa gauta in fiecare polinom introdus pentru a gasii monoamele care nu au avut acelasi grad in ambele polinoame. Acestea se introduc in noul polinom pe care il v-om returna la finalul metodei.

In urma explicarii in detaliu si asemanarii cu metoda de adunare, nu v-oi mai atasa codul aferent operatiei de scadere.

* *public static Polinom inmultire(Polinom p1,Polinom p2)*

*-*metoda de inmultire a doua polinoame difera de restu metodelor de pana acum; Aceasta prin intermediul a doua for-uri de tipul for-each inmulteste coeficientii fiecarui monom din primul polinom cu fiecare coeficient din al 2-lea si pune rezultatul intr-ul al 3-lea.

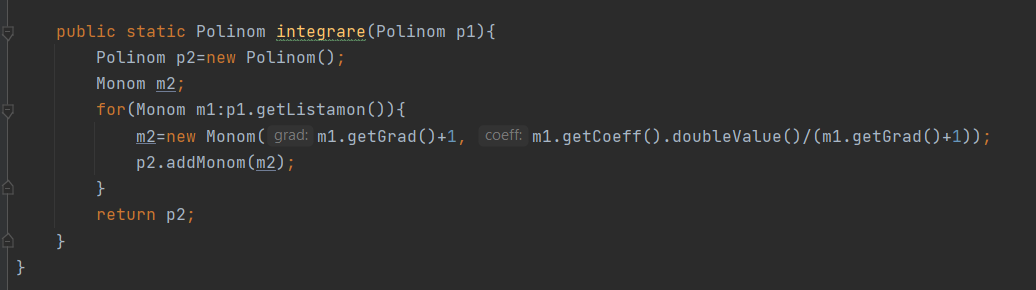
Inmultirea a doua monoame se refera la inmultirea coeficientiilor intre ei dar si in acelasi timp adunarea gradelor acestorea.

Dupa ce am pus rezultatul inmultiri in al 3-lea polinom , acesta trebuie optimizat prin adunarea sau scaderea monoamelor cu acelasi grad in cazul in care exista) astfel aici am folosit metoda *gatGradMax ()* din clasa Polinom, toate acestea pentru a gasii polinoamele cu acelasi grad. Dupa ce am folosit un for de la cel mai mare grad al monomului din polinom respectiv am cautat monoamele cu grad la fel si am facut calculele de adunare sau scadere intre ele ,iar numai apoi am pus reultatul intr-um polinom final pe care l-am si returnat. Toate explicatiile se pot oserva usor in codul atasat mai jos .

t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y t y

* *public static Polinom integrare(Polinom p1)*

*-*Metoda de integrare a unui polinom este relativ simpla; aceasta consta

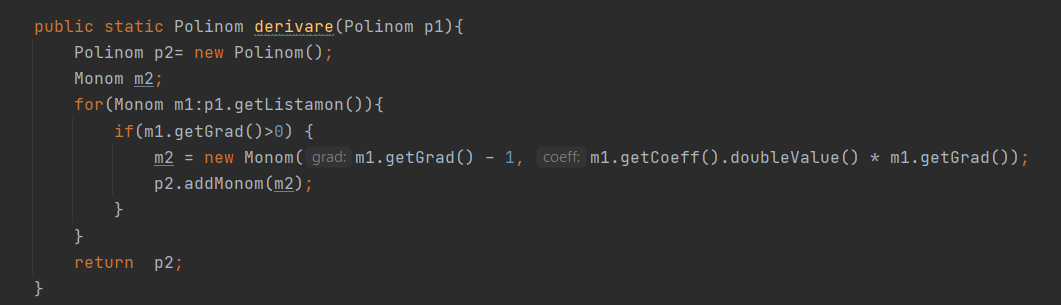
Intr-ul for care parcurge toate elementele polinomului si pentru fiecare element se va face operatia matematica de integrare adica se va creea un nou monom in care va avea coeficientul monomului gasit inpartit la gradul acestuia adunat cu 1 iar gradul noului monom va fi vechiul grad + 1 . Codul pentru integrare este atasat mai jos.

* *public stati Polinom derivare(Polinom p1)*

-metoda de derivare a polinomului este asemanatoare cu ceea de

integrare. Metoda are doar un parametru acesta este polinomul la care se va face integrarea.

Prin intermediul unui for vom cauta fiecare in monom din polinom daca are gradul mai mare ca si 0. In caz afirmativ, vom face operatia matematica de derivare, adica noul monom rezultat va fi de tipul gradul vechiului monom inmultit cu coeficientului acestuia , iar ca si grad va avea vechiul grad – 1 .



1. **Pachetul GUI**
2. Clasa View

In cadrul acestei clase am implementat interfata cu utilizatorul

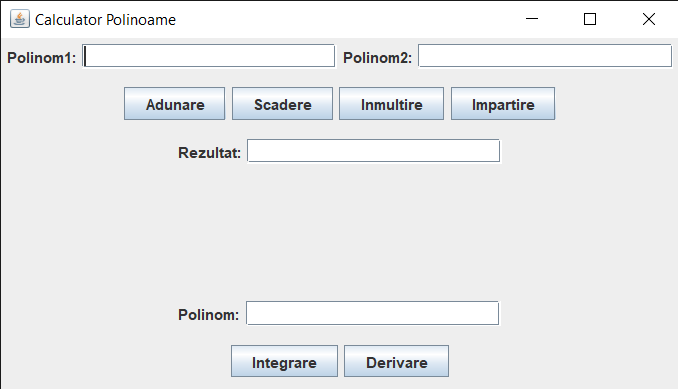
Am facut interfata in doua parti distincte pentru a fi cat mai explicit si cat mai usor de folosit.

1. In prima parte am pus operatiile de baza pentru cele 2 polinoame: scaderea, adunarea, inmultirea si impartirea.

In aceasta zona utilizatorul poate sa introduca in 2 zone de text doua polinoame si sa apese pe unul dintre butoanele denumit dupa operatia pe care o face, iar intr-o alta zona de text se va afisa rezultatul operatiei dintre ele.

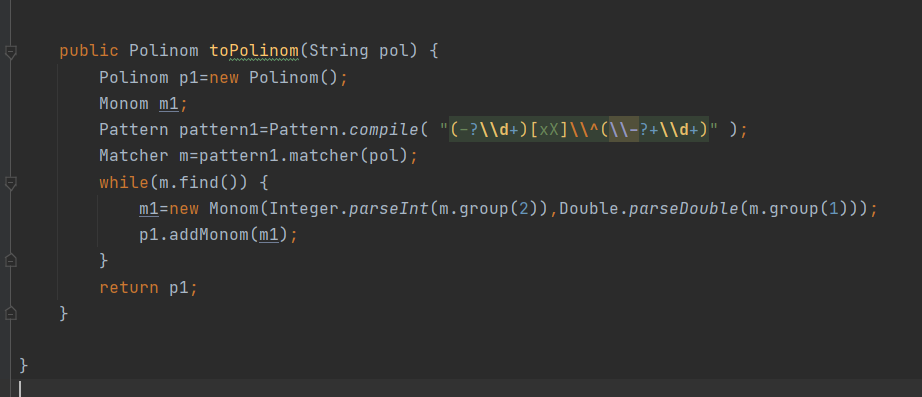
1. In a 2-a parte am pus operatiile de integrare si derivare

Tot tot asemanator primei parti am asignat butoane specifice operatiilor dar de data aceasta o zona de text pentru polinom si o zona de text pentru rezultatul operatiei.



1. Clasa Controller

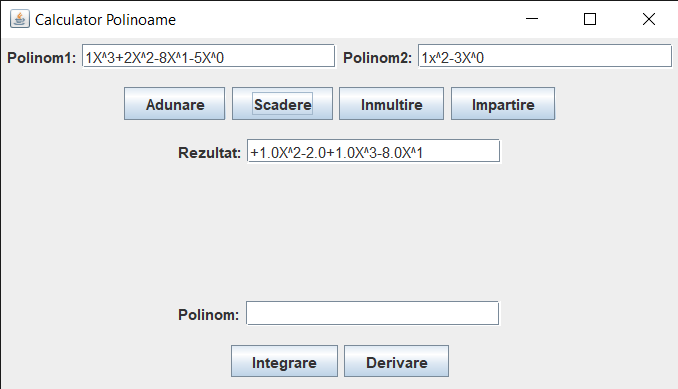
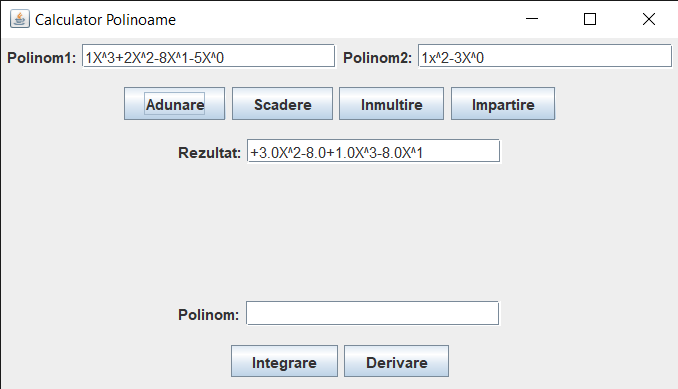
Clasa aceasta contine metoda: *public Polinom toPolinom(String pol) ,* aceasta converteste un string primit ca si parametru la un obiect de tipul Polinom. Pentru extragerea gradelor si a coeficientilor s-a folosit un pattern.



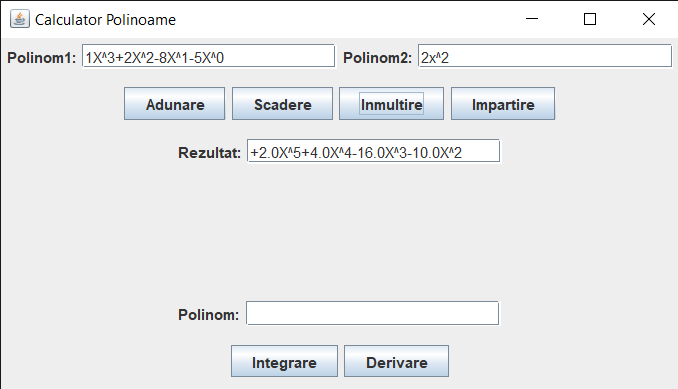
A d s a d a s d a d a s d a s d a s d a d s a s d a s d

1. Rezultate

In urma implementarii claselor, pentru testarea si verificarea programului s-au introdus doua cate doua polinoame pentru verificarea corectitudini acestora, iar pentru fiecare in parte s-a atasat cate o imagine .

Adunare Scadere

Inmultire



1. Concluzie

In concluzie pot sa spun ca in urma realizarii acestui proiect mi-am

imbofatit cunostiintele legate de proiectare pe obiecte in limbajul de programare java. Am invatat sa structurez aplicatia pe pachete, sa folosesc patter matching

pentru interfata grafica, altfel spus am mai facut un pas in cunoasterea limbajului java.

O inbunatatire care ar putea fi adusa proiectului ar fi: in primul rand implementarea operatiei de impartire creearea unui test cu Junit si bineinteles dezvoltarea dimensiuni polinoamelor, precum creearea de polinoame complexe cu coeficienti de tip imaginar.