ULTIMATE LaTeX REPOSITORY

EXAMPLE

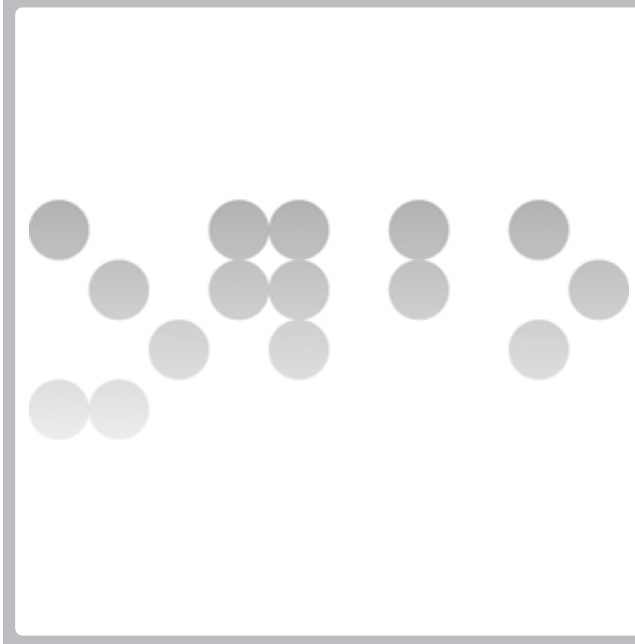
PAC1

P³CTEX
Exemple pràctic d'ús

Repte 1: Introducció a P³CTEX

01312

Universitat Oberta de Catalunya



Vladimir Scoriae

Jo **Vladimir Scoriae**, declaro que ostento l'autoria total i plena de totes les tasques que es duen a terme al present document. Sóc l'única persona que ha fet cada exercici. No he compartit els enunciats amb ningú i l'únic ajut que he rebut ha estat a través de l'aula de la UOC i el seu professorat.

21 de març de 2026

Índex

Introducció

En aquest document es mostra un exemple pràctic de l'ús de P3CTEX per fer PACs i PRs seguint l'estil oficial de la UOC.

Tot i això, el repositori P3CTEX i les llibreries que el componen tenen moltes mes opcions possibles. Per això recomanem revisar la documentació de cada mòdul situat a la carpeta "/doc". Pots contribuir a ampliar aquest projecte a `https://github.com/DidacLL/P3CTeX`

F.A.Q.

Per què P3CTEX ?

P3CTEX es una eina desenvolupada durant els estudis d'enginyeria informàtica a la UOC. Son eines, comandes i entorns complets per facilitar la resolució de les proves de l'avaluació continua o pràctiques mantenint el format oficial de la UOC i els estàndards requerits en documents acadèmics, cientifics o tècnics adequats.

Però.. fer les PACs en LATEX?

Quan et planteges per primer cop fer les PACs en LATEXpot semblar un complicació innecessària, però en la meva experiència personal puc dir que aprendre a utilitzar LATEXamb comoditat és la millor decisió que he pres durant els meus estudis.
Per una banda ofereix un format estandarditzat i adequat per a la avaluació. Això es valora a l'hora de puntuar i sens dubte els professors ho reben d'una altra manera si es presenten les entregues en un format polit i adequat.
Però no nomes per al format ni el resultat, sinó perquè a la llarga t'estalvia molt temps i permet crear un ritme de treball mes òptim, centrar-te en el contingut sense preocupar-te per al format.

A mes a mes, P3CTEX , ha estat pensat des del principi per apropar les funcions mes complexes i modernes de LATEXd'una manera senzilla per als usuaris mes nous. Així doncs es proporcionen eines per a automatitzar tasques i calculs, molt útil en pràctiques en les que s'han d'introduir moltes captures de pantalla, gràfiques, diagrames etc..

En que es diferencia de LATEX?

P3CTEX està construït per damunt de LATEX2e (la versió actual) i combina funcions de *expl3* (conegut com LATEX3) però sempre accessibles des de documents LATEXestàndard. Com a repositori pensat per al usuari final, aquest paquet carrega automàticament altres paquets [1]per tant recomanem consultar-ne la documentació oficial per veure totes les funcions que aporten

[1]Si bé els mòduls individuals si son independents i es poden carregar o utilitzar sense necessitat de carregar tota la classe

1 Document Setup

1.1 Com utilitzar la classe de document P3CTeX

Per començar a utilitzar la classe de document P3CTeX en els teus documents LaTeX, primer cal definir-la a l'inici del fitxer principal. Aquest pas permet carregar els estils i les funcionalitats proporcionades pel paquet.

```
\documentclass{P3CTeX}
```

Aquesta comanda assegura que l'estructura, el format i les opcions específiques de la UOC es carreguin automàticament. Pots afegir opcions addicionals si ho vols, per exemple:

```
\documentclass[ca-exercise]{P3CTeX}
```

A continuació, pots configurar la informació bàsica del treball mitjançant l'ordre `\pxAuthorSetup` per establir el teu nom, NIU i altres detalls rellevants:

```
\pxAuthorSetup{
  author       = {Nom Cognom},
  niu          = {1234567},
  subject      = {Nom de l'assignatura},
  ca-fullname  = {Nom de la pràctica o PAC}
}
```

També pots fer servir altres comandes com `\pxSRCsetup` i `\pxTABsetup` per personalitzar el comportament de càrrega d'imatges o l'estil de les taules, respectivament.

Finalment, com en tot document LaTeX, el contingut va entre `\begin{document}` i `\end{document}`. L'exemple següent mostra una estructura bàsica:

```
\documentclass{P3CTeX}
\begin{document}
 \section{Introducció}
  Això és un document d'exemple amb \PECTeX.
\end{document}
```

Pots ampliar el document utilitzant les funcionalitats addicionals de la classe, com ara comandaments per inserir codi, taules millorades, gestió de captures de pantalla i molt més.

2 LaTeXBasic

LaTeXés un sistema de composició de textos que permet crear documents estructurats i formatejats de manera eficient.

LaTeXbàsic es basa en l'ús d'entorns per a estructurar el contingut. Els entorns més comuns són:

- Organitzacio del document

  Seccions `\section{Títol}` `\subsection{Títol}` `\subsubsection{Títol}`

  Paràgrafs `\paragraph{Títol}` `\subparagraph{Títol}`

- Format del text

  `\textbf{ }` **per a escriure en negreta.**

  `\textit{ }` *per a escriure en itàlica.*

  `\underline{ }` per a escriure en subratllat.

  `\emph{ }` *per a escriure en emphàtic.*

  `\texttt{ }` `per a escriure en mono.`

- Mida del text

  de gran a petit:

  `\normalsize` `\small` `\footnotesize` `\scriptsize` `\tiny`

  de petit a gran:

  `\normalsize` `\large` `\Large` `\LARGE` `\huge` `\Huge`

- Referencies

  `\cite{referencia}`: per a citar una referència.

  `\label{etiqueta}`: per a etiquetar un element del document.

  `\ref{etiqueta}`: per a referir-se a un element etiquetat.

  `\pageref{etiqueta}`: per a referir-se a la pàgina d'un element etiquetat.

També es pot formatejar l'espai vertical o horitzontal:
`\hspace{\fill}` o `\hspace*{2ex}`

`\vspace{1in}` o `\vspace*{0.3em}`

2.1 Com utilitzar les comandes de LaTeX

Algunes comandes de LaTeXnomes es poden executar en determinats entorns com es el cas del contingut del document mateix.
`\begin{comanda}` per a iniciar un entorn.
Aqui dins podem executar les comandes d'aquest entorn `\end{comanda}` per a finalitzar un entorn.

alguns exemples d'aquests entorns son:

- verbatim : permet escriure codi o caracters especials sense ser comptat com a codi LaTeX
- document : engloba tot el que s'imprimeix en el document final, a la part anterior se'n diu Preamble i s'utilitza per declarar comandes i configuració.
- itemize/ enumerate : crea llistes o enumeracions respectivament amb les comandes `\item`,`\subitem` i `\subsubitem`.

```
\begin{itemize}
 \item Llista desordenada
  \subitem primer subelement
   \subsubitem primer subsubelement
  \subitem segon subelement
   \subsubitem primer subsubelement
   \subsubitem segon subsubelement
\end{itemize}
```

- Llista desordenada
  - primer subelement
    - primer subsubelement
  - segon subelement
    - primer subsubelement
    - segon subsubelement

```
\begin{enumerate}
 \item Llista desordenada
 \subitem primer subelement
 \subsubitem primer subsubelement
 \subitem segon subelement
 \subsubitem primer subsubelement
 \subsubitem segon subsubelement
\end{enumerate}
```

1. Llista ordenada
   - primer subelement
2. segon element
3. tercer element
   - primer subelement
     - primer subsubelement

2.2 crear comandes propies

També es pot crear comandes propies per a utilitzar-les en el document. Per exemple:

```
\newcommand{\mycommand}[1]{Hello #1}
```

Aquesta comanda crea una comanda `\mycommand` que es pot utilitzar en el document a partir d'aquest moment:

```
\mycommand{World}
```

Això imprimeix: Hello World
També es pot crear comandes amb arguments opcionals:

```
\newcommand{\mycommand}[2][default]{#1 #2}
```

Aquesta comanda crearia una comanda `\mycommand` que es pot utilitzar en el document com ara:

```
\mycommand{Això és un argument}
\mycommand[Aquest es opcional]{Això és un argument}
```

També disposem de la manera primitiva de declarar comandes, si bé té algunes avantatges i es més àgil, no està protegida contra sobreescriptura i pot causar comportaments no desitjats. Però és molt útil per entendre com funciona LaTeX. LaTeXanalitza el document per tokens, tokens que va expandint (intercanviant per el seu valor) quan troba alguna macro.
Un token pot ser tant:

- A — cada caracter es un token
- \PECTeX — cada comanda o macro es un sol token
- `{Patatas pata todos}` — el contingut de dues { } es tracta com un sol token

Aixi podem entendre la comanda `\def` que es la manera de definir macros a un nivell més baix, concretament demana el nom de la macro, els parametres i el codi que expandir en lloc de la macro.
`\def\foomacro{PATATA}` → `\foomacro` simplement escriu "PATATA"
`\def\foomacro#1{Patata #1}` → `\foomacro{frita}` espera 1 parametre (o un token) es per això que posem les {} tot i que les podriem ometre i nomes agafaria el primer token.
Sabent això podem definir altres formes de separar els parametres entenent que LaTeXva escanejant la comanda esperant el caracter definit per entendre la separació entre parametres. Si no hi ha cap separador agafar un unic token.

- `\def\patata#1#2#3`
  es crida com `\patata{param 1}{param 2}{param 3}`
- `\def\patata#1.#2.#3:`
  es crida com `\patata{param 1}.{param 2}.{param 3}:`
  o `\patata param 1.param 2.param 3:`

3 Taules

Les taules a LaTeXes solen fer amb l'entorn *tabular* però té una sintaxi molt farragosa per un ús habitual, per això P3CTEX n'ofereix una serie de metodes alternatius.

L'entorn **pxTAB** de P3CTEX ofereix uan seria d'estils de taules automàtiques predisenyat, a continuació es mostra alguns dels mes habituals i com utilitzarlos.

En aquest cas ofereix la opció de configurar el disseny de les taules individualment. Aixi doncs quan apliquem la configuració s'aplicarà la nova configuracio en les taules següents sense modificar les anteriors:

```
\pxTABsetup{style=header}

\pxTable{Name, Age, City}
  {Alice, 20, Barcelona}
  {Bob, 22, Madrid}
```

Table Styles

Name	Age	City
Alice	20	Barcelona
Bob	22	Madrid

Item	Quantity
Apples	3
Oranges	5
Bananas	2

Code	Description
PX1	First code
PX2	Second code

m11	m12	m13
m21	m22	m23

Centered header-style table using the new key.

Language	Users
LaTeX	Many
P3CTeX	Felizment cada cop més

Float table with caption, label, and explicit placement.

Taula 1: Sample pxTAB float table

Module	Feature
pxSRC	Safe image inclusion
pxTAB	Predefined table styles

Table 1 shows a small pxTAB float that can be cross-referenced in the usual way.

Composed styles

Composed styles: rubric-style grid and striped dataset. Row backgrounds fill the entire cell area for all striped-like styles, including with explicit width/tabwidth settings.

Criterion	Weight	Notes
Design	40%	Structure and clarity
Implementation	40%	Correctness and style
Testing	20%	Coverage and edge cases

Student	PEC 1	PEC 2	Final
Alice	8.0	9.0	8.5
Bruno	7.5	8.0	7.8
Carla	9.0	9.5	9.3
Daniel	6.5	7.0	6.8

Paragraph-cell

Paragraph-cell mode: multi-line wrapping in equal-width columns.

Component	Weight	Description
Design	30%	Justify architectural decisions and document the chosen patterns. Include a class diagram and a brief rationale for each design trade-off.
Implementation	40%	Deliver clean and well-structured source code with meaningful variable names. The build must succeed without errors or critical warnings.
Testing	20%	Provide unit tests covering the main public API and at least two edge cases per module.
Report	10%	Summarise the work in a concise document using the P3CTeX template. Include references and an appendix with test logs.

Preset system examples

Custom preset: define once, reuse everywhere.

Module	Credits	Grade
Algebra	6	A
Calculus	6	B+

Built-in preset (registered at load time).

Component	Status
API Design	Complete
Implementation	In progress

Override a preset key inline.

Test	Result
Unit	Pass
Integration	Pass

Alternative tables

pxTBLtop: legacy header-body shorthand and includes longtables

Name	Course	Grade
Alice	CS101	A
Bob	CS102	B+

Columna 1	**Columna 2**	**Columna 3**
11	12	13
21	22	23
31	32	33

1	2	3	4	5	6	7	8
Aquesta es una mostra d'una taula que s'exten a varies pagines							
X	a mes no respecta el numero de columnes en totes les files				SENSE MOSTRAR CAP ERROR		
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y						TUDO BEM
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					

1	2	3	4	5	6	7	8
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
X	Y	Z					
P	A	T	A	T	A	S	
P	A	T	A	T	O	D	OS

4 Imatges

Tot i estar disponible la comanda habitual `\includegraphics[keyvals]{imagefile}` P3CTEX ofereix una opció mes segura, en la que si la imatge no existeix simplement imprimeix un placeholder amb la mateixa mida i a la mateixa posicio, permetent incloure una imatge que encara no tenim i podent seguir redactant i formatejant.

Aquesta comanda es :

```
\pxIMG[width]{imagename}[placeholderColor]{Caption text}[label]
```

On els parametres entre "[]" son opcionals.
La amplada es pot indicar en cualsevol unitat valida (cm,in,pt,em,ex) o bé amb fraccions d'una mida vàlida (`0.5\linewidth`)

Aixi la versio minima de la comanda seria: `\pxIMG{imagename}{Caption text}`

També tenim les opcions per inserir dues imatges una al costat de l'altre amb un sol peu de imatge
`\pxIMGpair{imgname1}{imgname2}[errcolor]{caption}[label]`

i per últim una opció per introduir llistats de imatges nombrades sequüencialment (sense 0s a l'esquerra) en una mateixa carpeta
`\pxIMGList{nom}{Caption de la primera imatge, caption de la segona, tercera, ...n}`
això carregaria 4 imatges anomenades nom1,nom2,nom3,nom4 ja que hi ha 4 elements a la llista de captions

Exemples

Figura 1: Aquesta imatge medeix 12em (o 12 caracters de la mida de la font actual)

Figura 2: Aquesta imatge medeix 0.3*linewidth i a mes la imatge no existeix, hem modificat el color del placeholder a primaryColor que es el per defecte de P3CTEX

Figura 3: Pair caption

Figura 4: Imatge automatitzada amb la IMGList 1

Figura 5: Imatge automatitzada amb la IMGList 2

Figura 6: Imatge automatitzada amb la IMGList 3

Figura 7: Imatge automatitzada amb la IMGList 4

5 pxPRP: Objectes i OOP

Aquest mòdul de P3CTEX ens permet crear i enmagatzemar objectes lògics. Es pot utilitzar simplement per enmagatzemar informació amb una estructura lògica, o per implementar funcions mes avançades (com per exemple les funcions de UML s'han fet utilitzant aquesta llibreria independent)

5.1 Objects, keys, and access

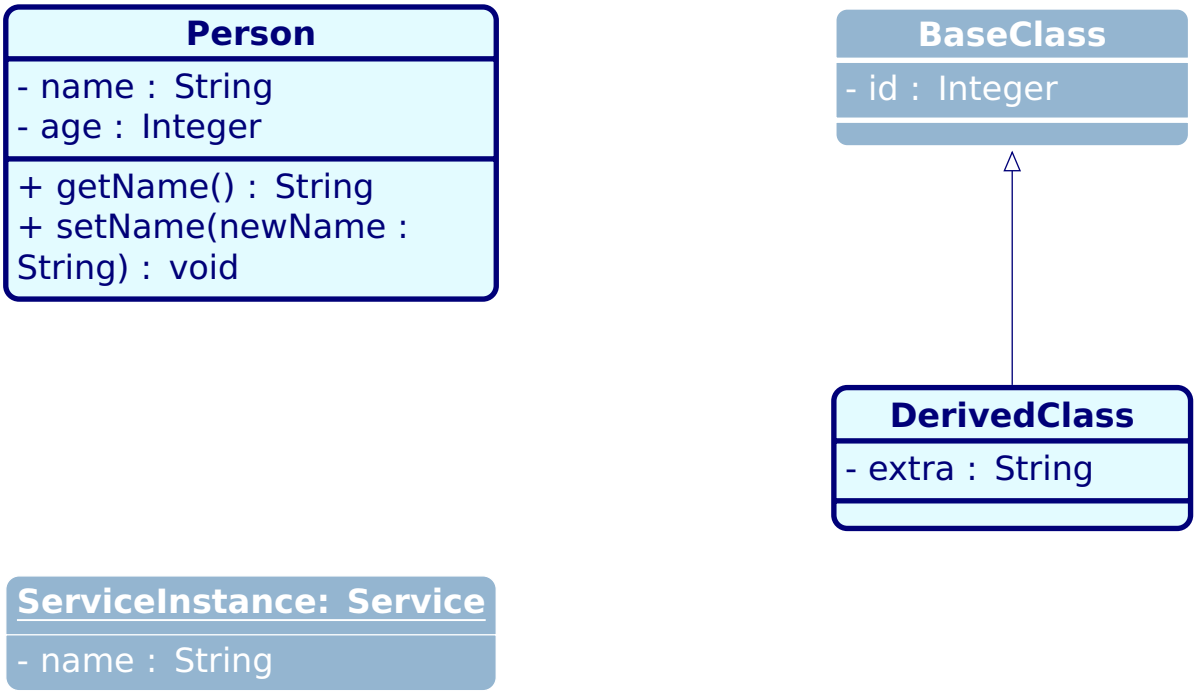
Document il·lustratiu de com utilitzar P3CTEX
Redactat per: V. Scvria
(O pots dir-me nomes Scvria)

6 pxUML: class diagrams

Per als UML P3CTEX crea una ampliació de les comandes de *pgfuml-cd* a mes de carregar directament *pgfuml-sd* per tant les comandes habituals d'aquests dos reconeguts paquets estan disponibles igualment i per defecte al activar el mòdul pxUML de P3CTEX

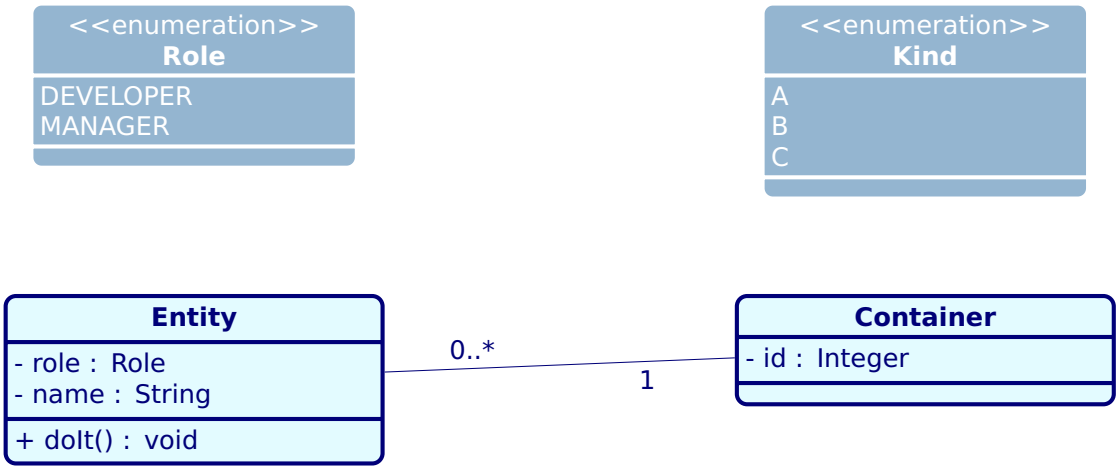
A mes de les opcions habituals, P3CTEX ofereix una abstracció de la sintaxi similar al OOP en la que es separa la declaració de les classes i les propietats, de dibuixar el diagrama, permetent aixi reutilitzar el codi de les mateixes classes al llarg del document en diferents diagrames i anar afegint mes atributs dinamicament.

6.1 Classes and diagram



Inspection: Person attributes: [- name : String,- age : Integer].
Service operations: [+ run() : void].
Parents of DerivedClass: [BaseClass].
Non-existent class: [].

6.2 Enums and associations



7 Code

A P3CTEX es proporciona un metode rapid per a poder incloure codi formatejat per als llenguatges i estils mes habituals.

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

FFSDA

```
    @Overwrite
public class Fibonacci<?> {
    public static int fib(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int tmp = a + b;
            a = b; b = tmp;
        }
        return b;
        [false == 23] patata;
    }
}
```

codebox:[lst-alg-ex] **Algorisme: factorial**

```
Funcio factorial(n: Integer): Integer
  Si n <= 0 Llavors
    Retornar 1
  Sinó
    resultat <-- 1
    Per i = 1 A n Fer
      resultat <-- resultat * i
    Fi Per
    Retornar "resultat" @tag
  Fi Si
Fi Funcio
```

codebox:[lst-alg-ex] Calcul del factorial de manera iterativa.

La llista 7 mostra l'algorisme iteratiu del factorial.

lst-yaml-ex

```
project:
  name: P3CTeX
  version: "0.3"
  languages: [ca, en]
  debug: false
```

```
#!/bin/bash
cd tex/
pdflatex -halt-on-error main.tex
echo "Build complet."
```

7.1 Exercici: codi inline

En aquest exercici es mostren fragments de codi inline. Fragment en Java: `int n = 0;` i `return n + 1;`. Un altre fragment en Bash: `echo "fi"`.