# Workflow

## Rularea si compilarea programului:

Pentru a putea fi rulat, programul trebuie sa fie compilat pe fiecare sectiuna a sa, asa cum este prezentat mai departe:

Intrarea in folderul /Functions si rularea comenzii ‘make’.

Intrarea in folderul /serverAPI si rularea comenzii ‘make’.

Intrarea in folderul /Users si rularea comenzii ‘make’.

Workflow-ul programului este urmatorul:

Se ruleaza procesul administrator, folosind comanda ./admin.

Urmatorul pas este rularea proceselor ‘worker’, din folderul Functions. Acestea pot fi lansate in executie fie manual (folosind ./[nume executabil]), fie in mod automat printr-un script pus la dispozitie (prin rularea ./script.sh).

Urmatorul pas este folosirea aplicatiei prin rularea fie a proceselor exemplu, din folderul /User (prin rulari succesive si manuale de tipul ‘./[nume executabil]’), fie prin crearea unei alte surse ‘.c’.

Pentru crearea altei surse si rularea corespunzatoare, se recomanda pasii 2,3,4.

## Inițializarea parametrilor de conexiune:

Definiți obiectul params și setați adresa IP și numărul portului utilizând params.ipadd și params.portNumber.

## Apeluri sincrone:

Apelați funcțiile sincrone pentru a obține rezultate imediate:

Apelați funcția Hash("deliaandreea", params) pentru a calcula hash-ul unui text și salvați rezultatul într-o variabilă, de exemplu h.

Apelați funcția Base64("deliaandreea", params) pentru a realiza o conversie Base64 și salvați rezultatul într-o variabilă, de exemplu b.

Apelați funcția StergereDuplicate("ana ana are mere multe multe si bune bune", params) pentru a elimina duplicatelor dintr-un text și salvați rezultatul într-o variabilă, de exemplu sd.

Apelați funcția DeterminaFrecventa("ana ana are mere multe multe si bune bune", "ana", params) pentru a determina frecvența unui cuvânt într-un text și salvați rezultatul într-o variabilă, de exemplu fa.

## Apeluri asincrone:

Apelați funcția Encrypt("delia", params) pentru a cripta un text și salvați ID-ul cererii într-o variabilă, de exemplu a.

Apelați funcția getRet(a, params) pentru a obține rezultatul asociat cu ID-ul cererii și salvați rezultatul într-o variabilă, de exemplu aa.

Afișați rezultatul utilizând printf.