

# Photoformer *the tiny image editor*



By Guillermo Ramos  
and Alejandro López

The page features abstract line art in the corners. In the top-left and bottom-left corners, there are overlapping, curved lines that resemble stylized leaves or petals. In the top-right and bottom-right corners, there are concentric, curved lines that resemble stylized ripples or waves. The central text is enclosed in a dotted rectangular border.

# **Content**

**01 Diagrams**

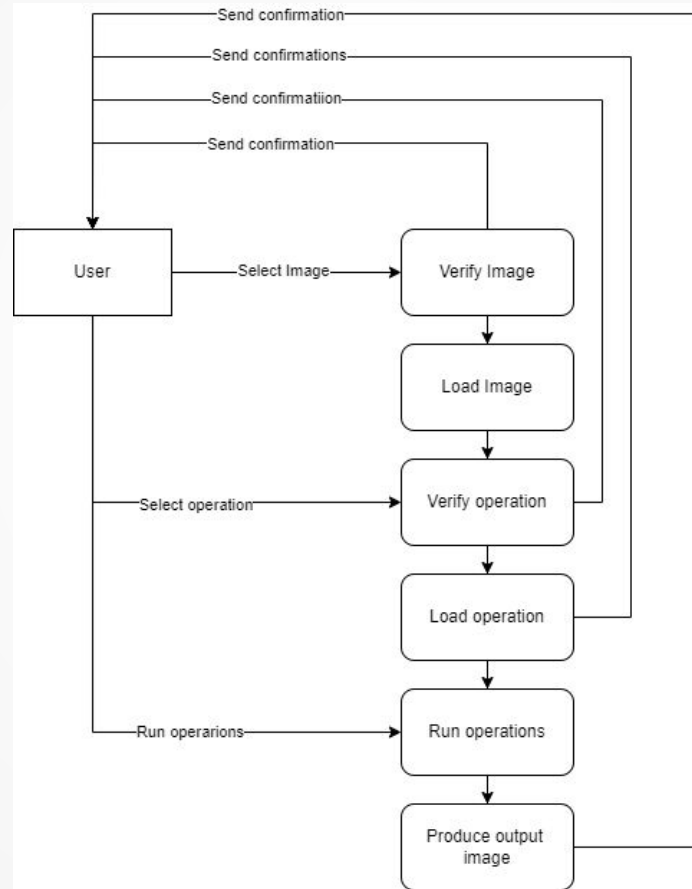
**02 Reasoning**

**03 Adaptations**

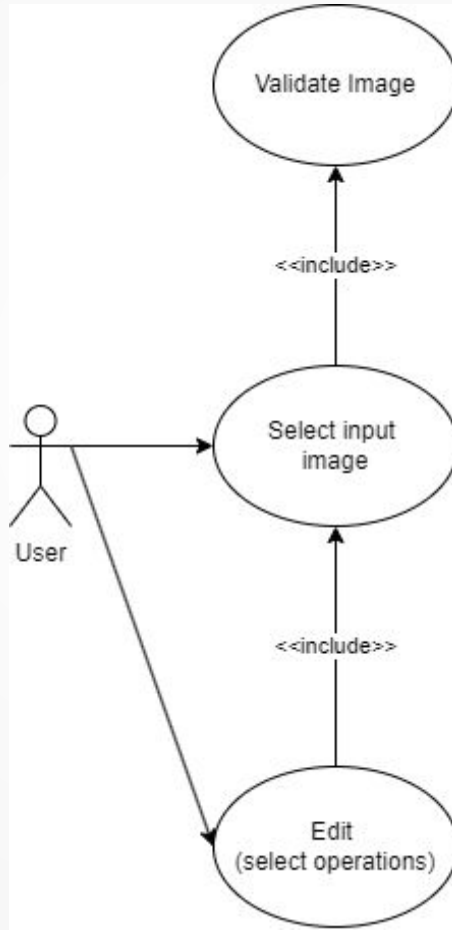
**04 Explanations**

**05 Conclusions**

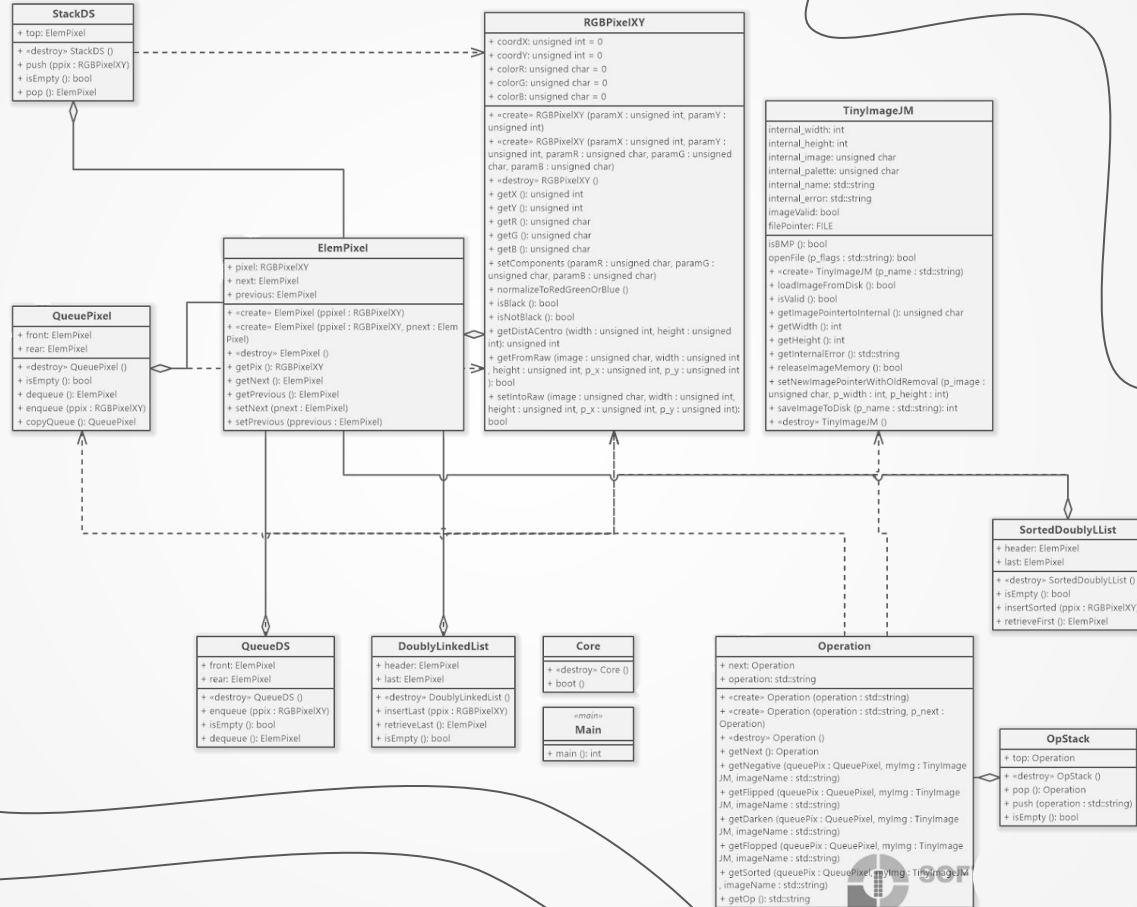
# Data flow diagram



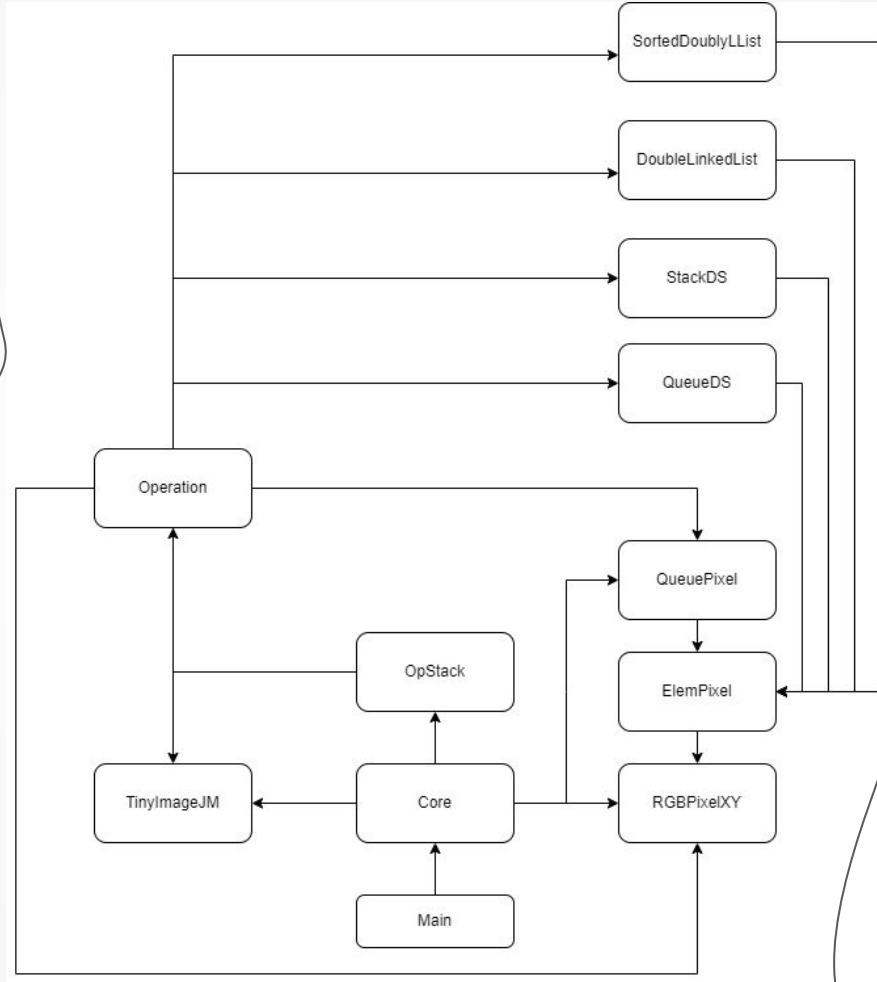
# Use case diagram



# Class diagram



# Source file diagram



# ADTs REASONING

getNegative - Queue

getDarken - Queue

getFlipped - Stack

getFlopped - Doubly Linked List

getSorted - Sorted Doubly Linked List

# ADTs ADAPTATIONS: QUEUEPIXEL

SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue

dequeue: queue -> item

front: queue -> item

makenull: queue -> queue

empty: queue -> boolean

ENDSPEC

SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue

dequeue: queue -> item

empty: queue -> boolean

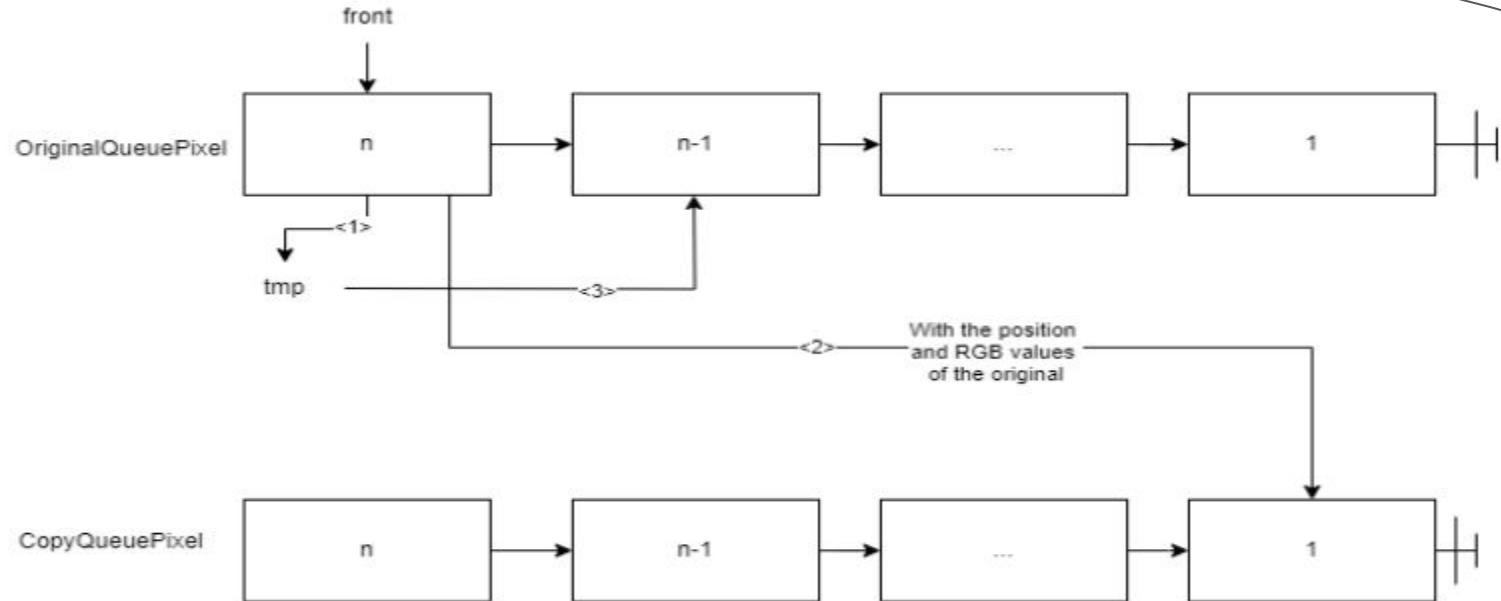
copyQueue: queue -> queue

ENDSPEC



# ADTs ADAPTATIONS: QUEUEPIXEL

COPYQUEUE



# ADTs ADAPTATIONS: QUEUEDS

SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue

dequeue: queue -> item

front: queue -> item

makenull: queue -> queue

empty: queue -> boolean

ENDSPEC

SPEC QUEUE[ITEM]

GENRES queue, item

OPERATIONS:

enqueue: queue item -> queue


dequeue: queue -> item

empty: queue -> boolean

ENDSPEC

# ADTs ADAPTATIONS: OPSTACK

```
SPEC STACK[ITEM]
  GENRES stack, item
  OPERATIONS:
    push: stack item -> stack
    pop: stack -> item
    top: stack -> item
    makenull: stack -> stack
    empty: stack -> bool
ENDSPEC
```



```
SPEC STACK[ITEM]
  GENRES stack, item
  OPERATIONS:
    push: stack item -> stack
    pop: stack -> item
    empty: stack -> bool
ENDSPEC
```

# ADTs ADAPTATIONS: STACKDS

SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

top: stack -> item

makenull: stack -> stack

empty: stack -> bool

ENDSPEC

SPEC STACK[ITEM]

GENRES stack, item

OPERATIONS:

push: stack item -> stack

pop: stack -> item

empty: stack -> bool

ENDSPEC

# ADTs ADAPTATIONS: DOUBLYLINKEDLIST

SPEC LIST[ITEM]

GENRES list, item, position

OPERATIONS:

insert: item position list -> list

delete: position list -> list

locate: item list -> position

retrieve: position list -> item

next: position list -> item

previous: position list -> item

makenull: list -> list

empty: list -> bool

ENDSPEC

SPEC LIST[ITEM]

GENRES list, item

OPERATIONS:

insertLast: item list -> list

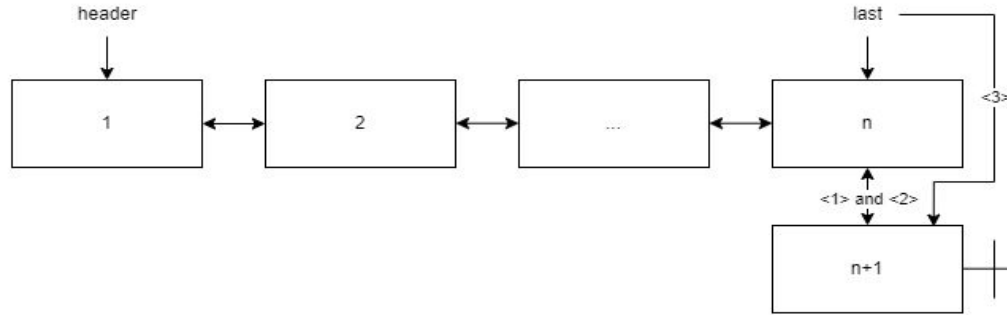
retrieveLast: list -> item

empty: list -> bool

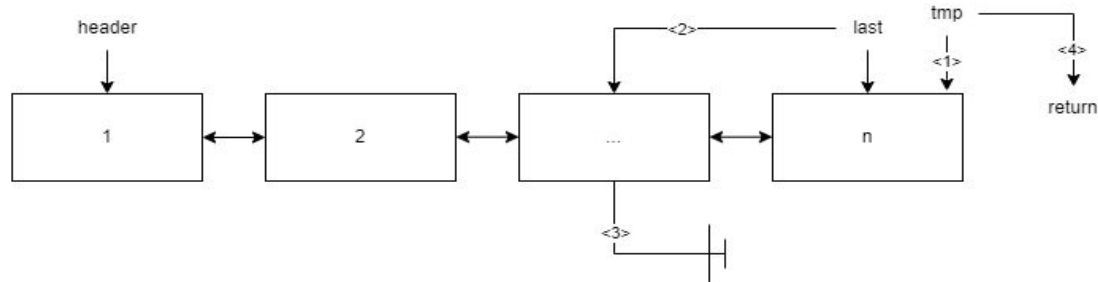
ENDSPEC

# ADTs ADAPTATIONS: DOUBLYLINKEDLIST

INSERTLAST



RETRIEVELAST



# ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST

SPEC LIST[ITEM]

GENRES list, item, position

OPERATIONS:

insert: item position list -> list

delete: position list -> list

locate: item list -> position

retrieve: position list -> item

next: position list -> item

previous: position list -> item

makenull: list -> list

empty: list -> bool

ENDSPEC

SPEC LIST[ITEM]

GENRES list, item

OPERATIONS:

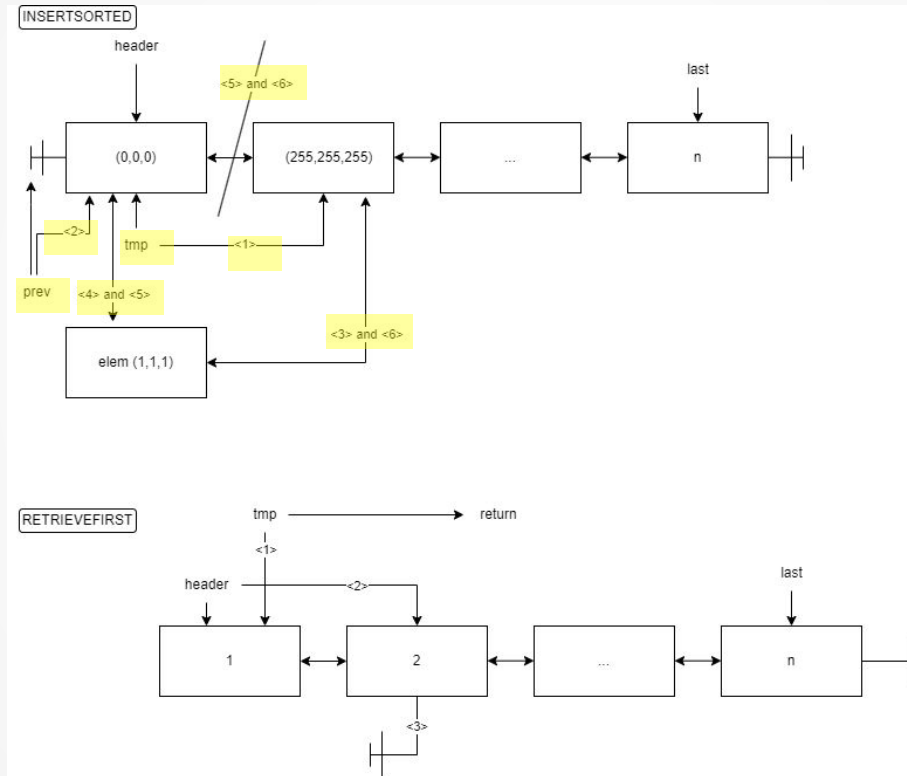
insertSorted: item list -> list

retrieveFirst: list -> item

empty: list -> bool

ENDSPEC

# ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST





# CLASSES EXPLANATION

Core:

- void boot()

ElemPixel

Operation:

- void getNegative()
- void getFlipped()
- void getDarken()
- void getFlopped()
- void getSorted()

OpStack

QueueDS

QueuePixel:

- QueuePixel\* copyQueue()

StackDS

DoublyLinkedList

SortedDoublyLList:

- void insertSorted()

# RUNNING TIME EXPLANATION

## LOADING PHASE

Pixel queue creation ->  $O(1)$

Validation of the image format ->  $O(1)$

Pixel enqueueing ->  $O(N)$

## USER INTERACTION PHASE

Insert operations in OpStack ->  $O(1)$

Where  $N$  is the total number of pixels in the image

# RUNNING TIME EXPLANATION

## DATA PROCESSING PHASE

getNegative()  $\rightarrow O(N + N + N) \rightarrow O(3N) \rightarrow O(N)$

getFlipped()  $\rightarrow O(N + N + N + N) \rightarrow O(4N) \rightarrow O(N)$

getFlopped()  $\rightarrow O(N + N + N + N) \rightarrow O(4N) \rightarrow O(N)$

getDarken()  $\rightarrow O(N + N + N + N) \rightarrow O(4N) \rightarrow O(N)$

getSorted()  $\rightarrow O(N + N*N + N) \rightarrow O(2N + N^2) \rightarrow O(N^2)$

The background features two thin, dark gray curved lines. One line starts from the left edge and curves upwards towards the top right. The other line starts from the bottom left and curves upwards towards the right edge, positioned below the first line.

# **CONCLUSIONS**