

# MANUAL TÉCNICO

## MODELO

**Client:** Clase para manejar la información de los clientes.

- Atributos
  - id: Integer
  - name: String
  - colorImage: Integer
  - colorlessImage: Integer
  - steps: Integer
- Getters and Setters: Hay uno para cada uno de los atributos.

**Window:** Clase para manejar la información de las ventanillas.

- Atributos
  - id: Integer
  - client: Client
  - images: Stack<Image>
- Getters and Setters: Hay uno para cada uno de los atributos.

**Image:** Clase para manejar la información de las imágenes.

- Atributos
  - id: Integer
  - type: String
- Getters and Setters: Hay uno para cada uno de los atributos.

## ESTRUCTURA DE DATOS

**Node:** Clase para manejar la información de los nodos que utiliza template.

- Atributos
  - id: Integer
  - element: T
  - after: Node<T>
  - before: Node<T>
- Getters and Setters: Hay uno para cada uno de los atributos.

**LinkedList:** Clase de estructura de lista simple que utiliza template.

- Atributos
  - index: Integer. Sirve para generar id para cada nodo.
  - count: Integer. Sirve para contar la cantidad de elementos en la lista
  - head: Node<T>. Este contiene la cabeza de la lista.
  - last: Node<T>. Tiene el ultimo elemento agregado.
- Métodos y Funciones
  - get(): Retorna la cabecera de la lista.
    - Retorna: Node<T>
  - getCount(): Retorna la cantidad de elementos en lista.
    - Retorna: Integer

- add(T element): Agrega un nuevo elemento al final de la lista.
  - Parámetros:
    - element: T
- dequeue(): Elimina el primer elemento de la lista y lo retorna.
  - Retorna: T

**Stack:** Clase de estructura de pila que utiliza template.

- Atributos
  - index: Integer. Sirve para generar id para cada nodo.
  - count: Integer. Sirve para contar la cantidad de elementos en la lista
  - head: Node<T>. Este contiene la cabeza de la lista.
- Métodos y Funciones
  - get(): Retorna la cabecera de la lista.
    - Retorna: Node<T>
  - getCount(): Retorna la cantidad de elementos en lista.
    - Retorna: Integer
  - add(T element): Agrega un nuevo elemento al inicio de la lista.
    - Parámetros:
      - element: T
  - pop(): Elimina el primer elemento de la lista y lo retorna.
    - Retorna: T

**CircularDoubleLinkedList:** Clase de estructura de lista circular doblemente enlazada que utiliza template.

- Atributos
  - index: Integer. Sirve para generar id para cada nodo.
  - count: Integer. Sirve para contar la cantidad de elementos en la lista
  - head: Node<T>. Este contiene la cabeza de la lista.
- Métodos y Funciones
  - get(): Retorna la cabecera de la lista.
    - Retorna: Node<T>
  - getCount(): Retorna la cantidad de elementos en lista.
    - Retorna: Integer
  - add(T element): Agrega un nuevo elemento al final de la lista.
    - Parámetros:
      - element: T
  - pop(): Elimina el primer elemento de la lista y lo retorna.
    - Retorna: T