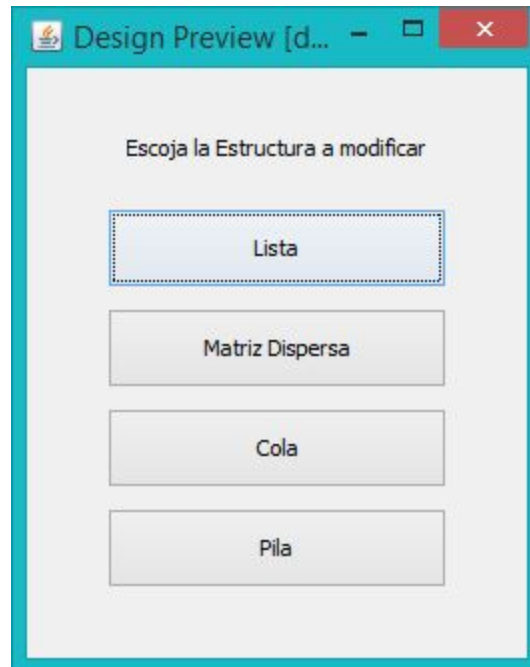
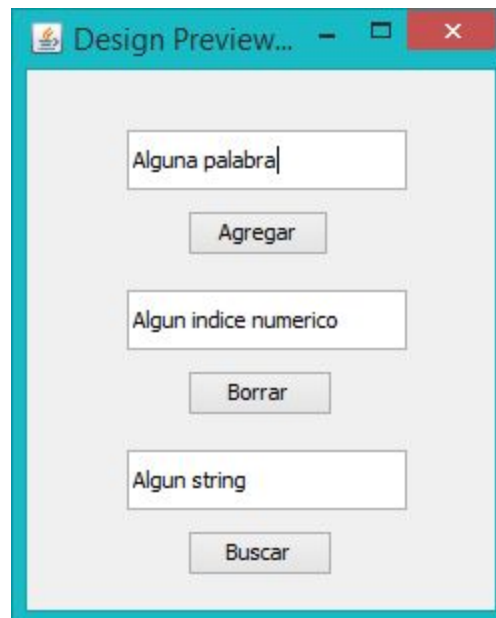


El menú principal tendrá estas opciones:

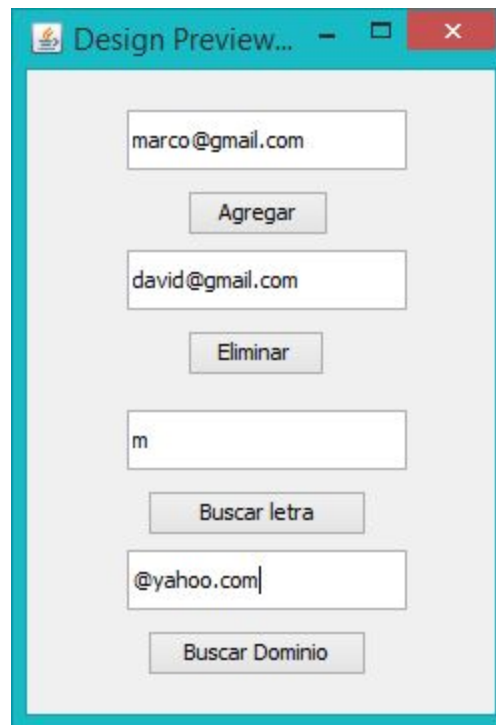


- Al escoger lista deberán aparecer estas opciones:



- El indice numerico va a empezar desde 0. No se necesita verificar si el índice es más grande que el tamaño actual de la lista.
- Al escoger la función buscar, en Java se deberá imprimir en consola el mensaje "NO SE ENCONTRÓ EL DATO" si no existe el string en la lista y el mensaje "EL DATO SE ENCUENTRA EN EL ÍNDICE <#indice>" si el string existe en la lista.

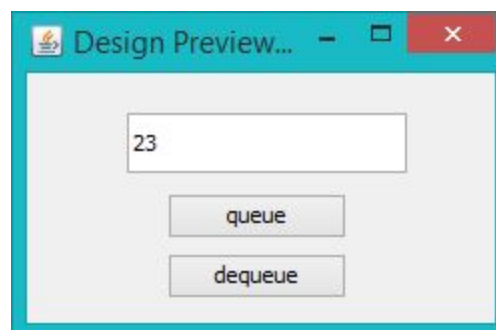
- Al escoger matriz se mostrara esto:



The screenshot shows a 'Design Preview' window with a light gray background. It contains the following elements from top to bottom: a text input field with 'marco@gmail.com', an 'Agregar' button, a text input field with 'david@gmail.com', an 'Eliminar' button, a text input field with 'm', a 'Buscar letra' button, a text input field with '@yahoo.com|', and a 'Buscar Dominio' button.

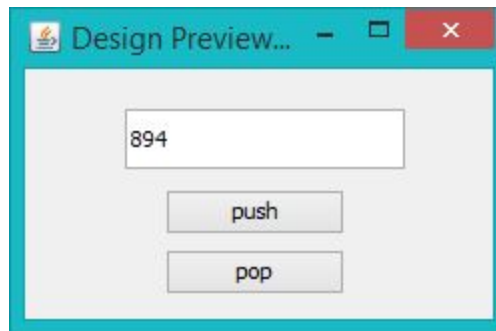
- En agregar se deberá de almacenar el dato en la matriz dispersa.
- En eliminar se deberá de eliminar el correo que ingresamos de la matriz dispersa.
- En buscar letra se mostrarán todos los correos que comiencen con esa letra. Se imprimirá una lista con los nombres en la consola de Java.
- En buscar dominio se mostrarán todos los correos que pertenezcan a ese dominio. Se imprimirá una lista con los nombres en la consola de Java.

- Al escoger cola se mostrara esto:



The screenshot shows a 'Design Preview' window with a light gray background. It contains the following elements from top to bottom: a text input field with '23', a 'queue' button, and a 'dequeue' button.

- Al escoger “queue” se deberá de ingresar el valor (siempre numérico) del campo de texto en la cola.
- Al escoger “dequeue”, el dato que arroje la cola será impreso en la consola de Java.
- Al escoger pila se mostrara esto:



- Al escoger “push” se deberá de ingresar el dato (siempre numérico) en la pila.
- Al escoger “pop” el dato que arroje la pila se deberá de imprimir en la consola de Java.

## Forma de trabajo

Los datos se ingresaran desde Java y se trasladaran a Python con el web service implementado con la herramienta de Flask. Siempre que se ejecute alguna acción, se deberá de refrescar el reporte correspondiente.

## Reportes

Los reportes deben estar hechos en Graphviz y se deben de mostrar por medio de algún visor de imágenes.

Las gráficas que se deben de mostrar son:

- La cola
- La lista simple
- La pila
- La matriz dispersa

Ejemplos de los reportes serán dados en clase.

## Restricciones

Los reportes son esenciales para verificar si se trabajó correctamente la estructura, así que **sin reportes, no se calificará el proyecto.**

## Observaciones

- Las estructuras deben de ser realizadas por el estudiante. No se permite el uso de ninguna librería dentro de las estructuras.
- Lenguaje a utilizar: Python y Java, cualquier IDE.
- Se debe de crear un repositorio en GitHub con el nombre de Practica2s12017\_#carnet.
- Los documentos deben estar sincronizados don GitHub así mismo teniendo un historial de que trabajo en dicha plataforma
- Forma de entrega: vía GitHub
- Fecha de Entrega: Domingo 5 de Marzo del 2017 a las 11:59 P.M. (Última sincronización de GitHub).
- **COPIAS SERÁN PENALIZADAS con una nota de 0.**
- **COPIAS SERÁN PENALIZADAS conforme al reglamento.**