

Trabajo Práctico n° 3: “Censo 2022”

Alumnos:

- Nissero, Matías
- Avalos, Nazareno
- Gonzalez Cabral, Francisco

Lógica de la interfaz de usuario:

- **class “VentanaPrincipal”:**
- “Inicio” inicio: muestra la primera ventana de la aplicación.
- “CensoSinArchivo” censoSinArchivo: muestra la segunda ventana de la aplicación donde el usuario ingresa cierta información.
- “Asignaciones” censistasAsignados: la tercera ventana de la aplicación, donde se muestra mediante texto qué censista le toca qué manzanas/cuadras (sí las hubiera disponibles).
- el llamado a las operaciones para el funcionamiento de la lógica de negocio: -
“RadioCensal” radioCensal: la clase directriz de la parte de negocio.

Las operaciones más importantes son:

crearCensoSinArchivo()	Oculto el panel Inicio y se crea la nueva ventana a mostrar: “CensoSinArchivo”.
crearAsignaciones()	Oculto el panel “CensoSinArchivo” y se crea la nueva ventana a mostrar: “Asignaciones”.
volverInicio()	Se muestra el panel Inicio y se eliminan los demás paneles sí existen. Además se reinicia la lógica de negocio: ‘radioCensal’ = null;
iniciarRadioCensal(int vértices)	Dado una cantidad de vértices previamente propuestos por el usuario, se ejecuta la lógica de negocio: new RadioCensal(vértices);
agregarCensista(String nombre, int DNI)	Dado un ‘nombre’ y un ‘DNI’ se lo agrega a radio censal.

conectarManzanas(int origen, int destino)	Dado un vértice 'origen' (manzana 'origen') y otro vértice 'destino' (manzana 'destino') se añade dicha arista (representación de dos manzanas contiguas) a radio censal.
---	---

- **abstract class "Panel":**

A diferencia del 2do TP que fuimos encargados de realizar, se nos ocurrió que 'chupar' o abstraer cierta información que se repite en todos los paneles era una muy buena idea para implementar.

Bajo esa idea, nace esta clase abstracta Panel (también barajamos la idea de implementar este mismo concepto con una interfaz, pero preferimos una clase abstracta). Su función principal es guardar operaciones e información extra para aprovechar la herencia, simplificar y reciclar código.

Sus variables principales son:

- VentanaPrincipal ventana - JButton btnVolverInicio - información acerca de:
- colores fijos para botones y paneles.
- fuentes de texto fijos para etiquetas y botones.
- valores fijos para el tamaño y la posición de los botones y paneles.

Sus operaciones principales son:

iniciarBtnInicio()	Se setean las características del botón para volver al panel de Inicio.
ignorarLetras(JTextField jtf)	Dado un JTextField, cualquier caracter igual a una letra que sea tipeado será 'consumida' para dar la idea de que sólo se pueden escribir números.
abstract mostrarPanel();	Método abstracto para ser implementado por las subclases: 'CensoSinArchivo', 'Extras', 'Asignaciones'. La idea es mostrar el panel que se ha llamado.
abstract ocultarPanel();	Método abstracto para ser implementado por las subclases: 'CensoSinArchivo', 'Extras', 'Asignaciones'. La idea es ocultar el panel que se ha llamado.

- **class Inicio extends Panel:**

Esta clase es una subclase de Panel y se encarga de mostrar la primera ventana que el usuario tiene para interactuar. En esencia teníamos pensado poner tres botones, pero como la idea de leer y escribir un archivo no se implementó, nos quedamos con dos botones solamente.

Dichos botones son los dos siguientes:

- JButton btnIniciarCensoSinArchivo
- JButton btnIniciarExtras

Una vez presionado algún botón, se crea el siguiente panel a ser mostrado.

- ***class CensoSinArchivo extends Panel:***

Esta clase es una subclase de Panel y se encarga de mostrar la segunda ventana interactiva. Esta ventana es la que más cosas tiene para interactuar a diferencia del resto de los paneles.

Por ello mismo, separamos cada cosa y las juntamos en un “paquete de inicialización” siguiendo un criterio específico: hay tres paquetes distintos referenciando a las variables que permiten al usuario hacer una cosa en concreto.

Primer paquete: el “paquete ingresar cantidad de manzanas” hace referencia al conjunto de variables funcionales al ingreso de la cantidad de manzanas (vértices) que va a tener la ciudad (grafo), es decir, lo que el usuario ve en pantalla:

- JLabel lblCantManzas
- JTextField textCantManzanas
- JButton btnAceptarCantidad;

Segundo paquete: el “paquete ingresar censistas” hace referencia al conjunto de variables funcionales al ingreso del nombre y DNI de un censista, es decir, lo que el usuario ve en pantalla:

- JLabel lblIngresarCensista
- JLabel lblCensistaNombre
- JLabel lblCensistaDNI
- JTextField tfCensistaNombre
- JTextField tfCensistaDNI
- JButton btnIngresarCensista

Tercer paquete: el “paquete Ingresar manzanas contiguas” hace referencia al conjunto de variables funcionales al ingreso de las aristas (manzanas contiguas), es decir, lo que el usuario ve en pantalla:

- JLabel lblIngresarContiguas
- JLabel lblManzanaID1
- JLabel lblManzanaID2

- JTextField tfManzanaID1
- JTextField tfManzanaID2
- JButton btnIngresarContiguas

Cuarto y último paquete: “cosas extras” es el conjunto de variables funcionales a ciertas características que no encajaban muy bien en los “paquetes” anteriores. Tenemos por ejemplo:

- un recordatorio de cuantas manzanas el usuario ingresó
- un aviso importante en cursiva
- una etiqueta para capturar y mostrar las distintas excepciones
- tres variables con fuentes fijas para las etiquetas, campos de texto y botones, respectivamente.
- un botón “btnAsignarCensistas” para avanzar hacia la siguiente ventana interactiva (Panel Asignaciones).
- ***class Asignaciones extends Panel:***

Esta clase es una subclase de Panel y se encarga de mostrar la tercera ventana interactiva. Esta ventana es la que le permite al usuario visualizar, mediante texto, qué censistas fueron asignados a qué cantidad de manzanas.

Así como en el panel “CensoSinArchivo”, este panel también tiene varios “paquetes” de variables agrupados por funcionalidad.

Primer paquete: el “paquete de censistas registrados” es el conjunto de variables funcionales a mostrar quienes son los censistas registrados y que manzanas se les fueron asignadas.

- JLabel lblCensistasRegistrados
- JTextPane txpCensistasRegistrados
- JScrollPane spCensistasRegistrados

Segundo paquete: el “paquete de manzanas registradas” es el conjunto de variables funcionales a mostrar la referencia de todas las manzanas registradas junto con sus respectivas manzanas contiguas.

- JLabel lblManzanasRegistradas
- JTextPane txpManzanasRegistradas
- JScrollPane spManzanasRegistradas

Tercer paquete: el “paquete censistas asignados a cierta cantidad de manzanas” es el conjunto de variables funcionales a mostrar mediante texto, qué censista fue asignado a qué cantidad de manzanas.

- JLabel lblAsignaciones
- JTextPane txpAsignaciones
- JScrollPane spAsignaciones

Cuarto y último paquete: “*cosas extras*” es el conjunto de variables funcionales a ciertas características que no encajaban muy bien en los “paquetes” anteriores. Tenemos por ejemplo:

- un botón “btnReiniciarInformacion”
- tres variables con fuentes fijas para las etiquetas, campos de texto y el botón, respectivamente.
- ***class Extras extends Panel:***

Esta clase es una subclase de Panel y se encarga de brindarle al usuario la oportunidad de testear los dos algoritmos desarrollados para el TP: el algoritmo goloso y el de fuerza bruta.

Para ello, necesitamos una cantidad de vértices y una cantidad de censistas; ambas cantidades son libres de ser propuestas por el usuario.

Así como lo hicimos en las dos clases previamente mencionadas, aquí también usamos un criterio de “paquetes de variables” para organizar la lectura del código.

Primer paquete: el “*paquete iniciar Stress Test*” es el conjunto de variables funcionales a ingresar la cantidad de vértices y de censistas a testear.

- JLabel lblCantVertices
- JLabel lblCantCensistas
- JTextField txpCantVertices
- JTextField txpCantCensistas
- JButton btnComparar

Segundo paquete: el “*paquete solver goloso*” es el conjunto de variables funcionales a mostrar, mediante texto, el tiempo de ejecución del algoritmo goloso y el tipo de solución que se haya resuelto.

- JLabel lblSolverGoloso
- JTextPane txpSolverGoloso
- JScrollPane spSolverGoloso

Tercer paquete: el “*paquete solver fuerza bruta*” es el conjunto de variables funcionales a mostrar, mediante texto, el tiempo de ejecución del algoritmo de fuerza bruta y el tipo de solución que se haya resuelto.

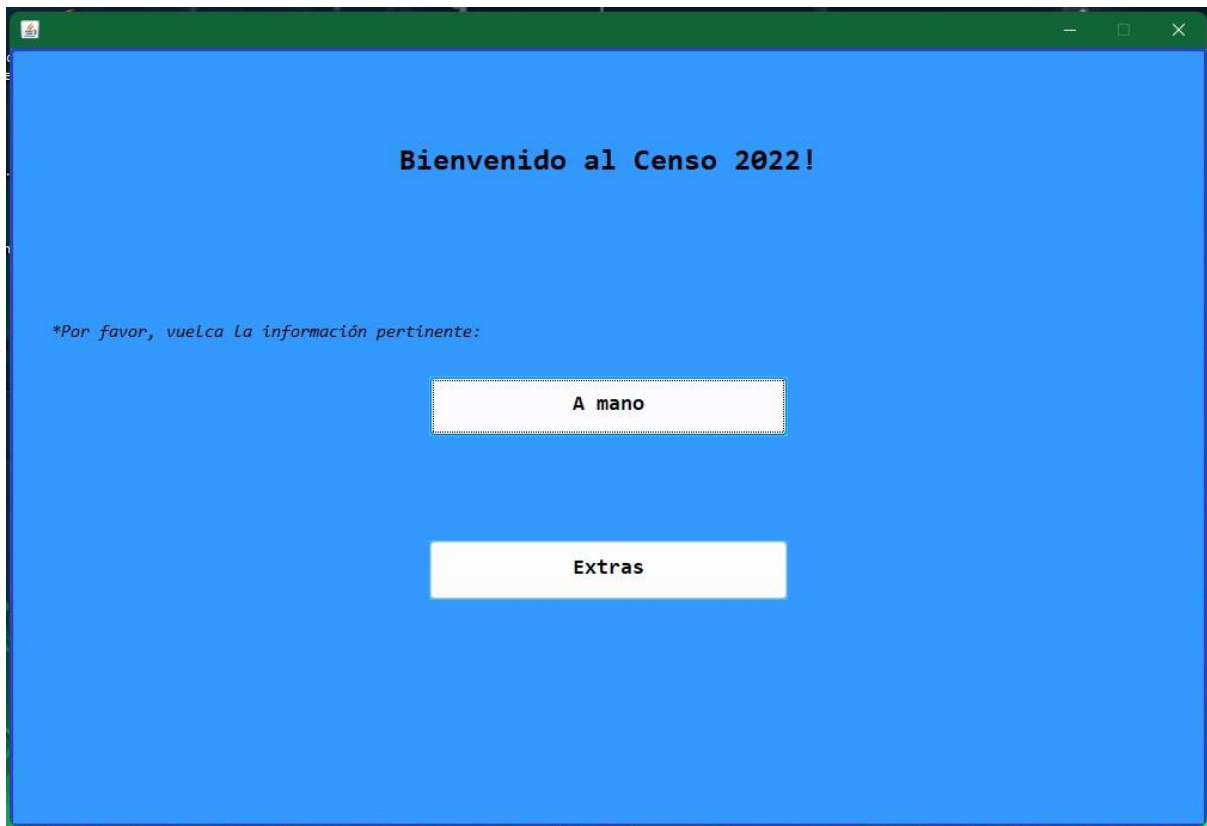
- JLabel lblSolverFuerzaBruta

- JTextPane txpSolverFuerzaBruta
- JScrollPane spSolverFuerzaBruta

Cuarto y último paquete: “cosas extras” es el conjunto de variables funcionales a ciertas características que no encajaban muy bien en los “paquetes” anteriores. Tenemos por ejemplo:

- una etiqueta de excepciones, para mostrar los posibles errores al ingresar cantidades inválidas
- dos variables con fuentes fijas para las etiquetas, campos de texto.

Panel inicio:



Panel CensoSinArchivo:

—

□

×

**Por favor, brinde la informacion pertinente...*

Manzanas propuestas: 2

Ingresar Censista:

Nombre:

DNI:

ingresar

Ingresar manzanas contiguas:

ID 1:

ID 2:

ingresar

Volver al inicio

asignar censistas

Panel Asignaciones:

—

□

×

Registro de Censistas:

Nombre: pepe
DNI: 12345

Registro de Manzanas:

ID: 0
Manzanas vecinas: | 1 |
La manzana fue censada

ID: 1
Manzanas vecinas: | 0 |
La manzana fue censada

Asignaciones:

Nombre: pepe
DNI: 12345
Manzanas a censar:
Manzana numero: 0
Manzana numero: 1

Volver al inicio

Reiniciar info

Panel Extras:

Solver Goloso:

Tiempo Solver Goloso: 0,001
 Solucion:
 ID: 0
 Manzanas vecinas: | 1 || 10 || 11 || 12 || 13 || 16 |
 La manzana fue censada

ID: 1
 Manzanas vecinas: | 17 || 18 || 19 || 15 || 14 || 9
 || 8 || 2 || 3 || 4 || 5 || 6 || 7 || 10 || 11 || 12
 || 13 || 16 || 0 |
 La manzana fue censada

ID: 2
 Manzanas vecinas: | 10 || 11 || 12 || 13 || 16 || 1 |
 La manzana fue censada

ID: 3
 Manzanas vecinas: | 11 || 12 || 13 || 16 || 1 || 10 |
 La manzana fue censada

Solver Fuerza Bruta:

Tiempo Solver Fuerza Bruta: 0,482
 Solucion:
 ID: 0
 Manzanas vecinas: | 1 || 10 || 11 || 12 || 13 || 16 |
 La manzana fue censada

ID: 1
 Manzanas vecinas: | 17 || 18 || 19 || 15 || 14 || 9
 || 8 || 2 || 3 || 4 || 5 || 6 || 7 || 10 || 11 || 12
 || 13 || 16 || 0 |
 La manzana fue censada

ID: 2
 Manzanas vecinas: | 10 || 11 || 12 || 13 || 16 || 1 |
 La manzana fue censada

ID: 3
 Manzanas vecinas: | 11 || 12 || 13 || 16 || 1 || 10 |
 La manzana fue censada

Cantidad de vertices: 20 **Cantidad de censistas:** 10

Volver al inicio **Iniciar Comparacion**