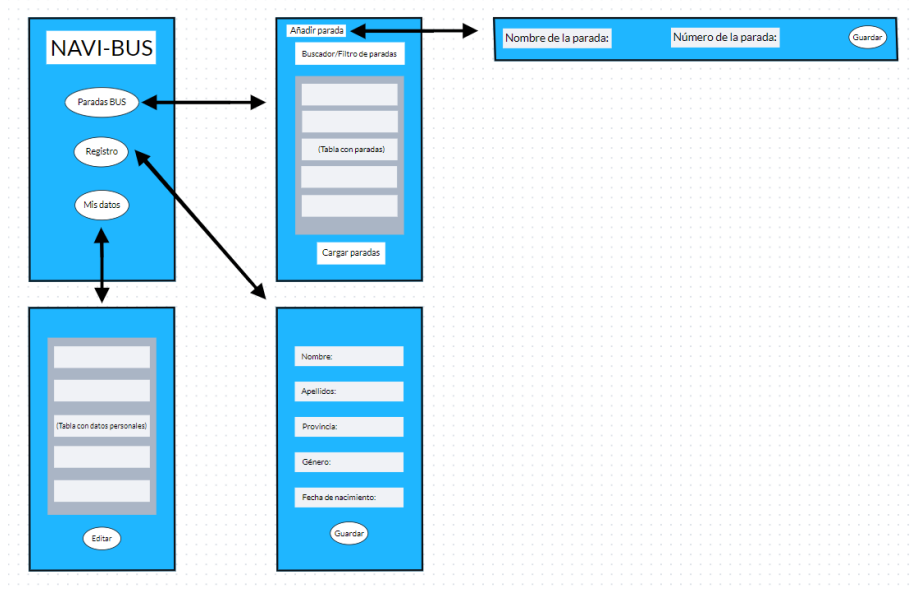


OBJETIVO Y ALCANCE DE LA APLICACIÓN:

La aplicación Navi-Bus es una aplicación de transporte cuyo objetivo es buscar, filtrar y añadir paradas de bus que te muestran su información correspondiente. También cuenta con la opción de registrar un usuario y poder modificar los datos que éste haya introducido. Es una aplicación pensada para expandir su funcionalidad en un futuro con la integración de una base de datos en la que poder consultar diversos tipos de transporte e información en tiempo real.

SKETCHING:



WIREFRAMMING:



PANTALLA PRINCIPAL

Imagen NAVI-BUS: logotipo de la aplicación con función estética e informativa.

Botón Paradas Bus: dirige hacia la pantalla de las paradas de bus.

Botón Registro: dirige hacia la pantalla del formulario para registrar un usuario.

Botón Mis Datos: dirige hacia la pantalla de la tabla con los datos registrados del usuario.



PANTALLA BUSCADOR DE PARADAS

Botón Añadir parada: dirige hacia la pantalla del formulario para añadir una nueva parada.

Elemento Buscador de paradas: busca y filtra la tabla de paradas.

Tabla con paradas: contiene todos los registros de las paradas.

Botón Cargar paradas: carga automáticamente unas paradas preestablecidas en la tabla.

PANTALLA REGISTRO

Elemento Nombre: guarda el nombre que introduzca el usuario.

Elemento Apellidos: guarda los apellidos que introduzca el usuario.

Elemento Fecha de nacimiento: guarda la fecha que seleccione el usuario.

Elemento Género: guarda el género que seleccione el usuario.

Elemento Provincia: guarda la provincia que seleccione el usuario.

Botón Guardar: guarda todos los datos introducidos y crea un usuario con ellos.

PANTALLA MIS DATOS:

Elemento Tabla con Datos: muestra todos los datos del usuario guardado.

Elemento Editar: permite editar los campos de datos para cambiarlos.

PANTALLA FORMULARIO PARADAS:

Elemento Nombre de la parada: guarda el nombre que introduzca el usuario.

Elemento Número de la parada: guarda el número que introduzca el usuario.

Botón Guardar: guarda todos los datos introducidos y crea una nueva parada con ellos.

OBJETIVOS QUE CUMPLIR EN EL PROYECTO:

- Especifica claramente el objetivo y alcance de la aplicación: **página 1 de este documento.**
- Realizado sketching; Tiene en papel las pantallas por las que se navegará indicando su función y relaciones entre ellas: **página 1 de este documento.**
- Realizado sketching; Todas las funcionalidades de la aplicación que indicó que se iban a realizar tienen su reflejo en algún camino, pantalla del sketching: **página 1 de este documento.**
- Wireframing; Especifica para cada pantalla los distintos elementos, funcionamiento y objetivo de cada uno de ellos (Ej:Añade un usuario, comprueba dni válido antes de añadirlo...): **página 1-2 de este documento.**
- Comentarios en el código indicando el objetivo y características de las clases, métodos, propiedades...
- Se empieza con JFrame y se abren Jdialog, manteniendo comunicación entre ellos.
- Crea métodos públicos para acceder a propiedades privadas: **ejemplo en líneas 29-38 de PantallaDatos.java.**
- Utiliza paquetes para gestionar el software, siguiendo las indicaciones dadas en clase; gui, dto, lógica...
- Separa código de control, gestión de información, comprobaciones...(lógica de negocio) del código gui o de datos.
- Usa tablas gestionando el modelo y añadiendo entradas: **ejemplo en líneas 157-175 en PantallaParadas.java.**
- La tabla se puede ordenar por el valor de las columnas teniendo en cuenta la conversión del índice de la fila seleccionada en la tabla con su correspondencia en la lista original: **en líneas 206-209 en PantallaParadas.java.**
- Permite realizar un filtrado de filas en la tabla: **en líneas 170-173 en PantallaParadas.java.**
- Utiliza elementos básicos de formulario: **ejemplo en líneas 291/298 en PantallaDatos.java.**
- Utiliza elementos avanzados de formulario (spinner, combobox, radioButton...) usando sus valores: **ejemplo en líneas 169/171 en PantallaRegistro.java.**
- Maneja eventos tanto con las herramientas facilitadas por netBeans como implementando EventListener creados por el alumno.
- Usa ArrayList y bucles para gestionarlo: **ejemplo en línea 13 en logicaParadas.java.**
- Utiliza SimpleDateFormat para mostrar el formato de fecha: **en línea 17 en Usuario.java.**
- Ha utilizado la propiedad "resizable" en algún Gap y en algún componente: **utilizado en propiedades de Horizontal Gap en PantallaDatos.java y en propiedades de la tabla de PantallaParadas.java.**
- Ha usado 2 tipos de Layout: **utilizado FlowLayaout en PantallaParadas.java y GridLayaout en PantallaFormParadas.java.**
- Usa al menos 2 tipos de JOptionPane capturando y utilizando la información que devuelve el objeto: **en línea 31/104 en PantallaformParadas.java.**
- Realiza validación de formularios, tanto de forma manual como usando ValidationPanel: **validador manual en líneas 22-35 en PantallaFormParadas.java y validador automático en líneas 31-44 en PantallaRegistro.java.**
- Realiza esa validación con al menos 2 eventos diferentes (al pulsar enviar, al perder foco...).
- Ha implementado los mensajes de validación de librería en español: **en paquete org.netbeans.validation.api.builtin.stringvalidation.**

-Utiliza imágenes en algún elemento de la aplicación: **en JLabelNaviBus en PantallaPrincipal.java.**

-Ha asociado un icono a la app: **guardado en gui.imagenes.**