

**MANUAL TECNICO**  
**[IPC1]P1Fase1\_Dic2020**

**SERGIE DANIEL ARIZANDIETA YOL**  
**CARNET 202000119**

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**ESCUELA DE CIENCIAS Y SISTEMAS**  
**Guatemala, 19 de Diciembre del 2020**

## INTRODUCCIÓN

- EL objetivo de este manual es presentar el software de [IPC1]P1Fase1\_Dic2020.
- La fase uno del proyecto busca la realización y organización de los datos por medio de una interfaz especializada a los administradores.
- Enfocado a la parte de edición de los datos y carga de todo tipo de administración que sea orientado a un usuario o aplicación específica.
- Permite la evaluación de condiciones para creación, eliminación y edición de los aspectos considerados usuarios y aplicaciones.
- El manual se encontrará la descripción de los métodos y clases utilizados para el diseño de la [IPC1]P1Fase1\_Dic2020.

## Clases por paquete

### Default

- **Main:** su único objetivo es inicializar el programa y guardar variables y valores iniciales.

### Interfaz

- **Ingreso:** es una clase que posee distinta cantidad de métodos con el objetivo de poder ingresar validando usuario y contraseña y en caso de olvido proporcionan la recuperación de la contraseña.
- **Pantallas:** clase con todos los métodos de JFrame para poder desplegar en todos los casos que hay ayudando a la organización del software dentro de ella.
- **Principal:** es la pantalla principal de un usuario de tipo administrados que únicamente y exclusivamente solo ellos tendrán acceso para poder hacer las modificaciones y manejo de datos correspondientes.

### Save

- **Almacenar datos:** clase que se encarga de obtener todos los datos del programa, pero únicamente los datos iniciales de ellos mismos que nos ayuda de pivote para resguardar la inicialización de la cantidad de usuarios administradores y operaciones básicas.
- **DatosInterraciones:** clase que se encarga de todos los procedimientos de actualización búsqueda, cambio, eliminación y actualización de los JFrame para que la aplicación siempre recurra a la misma para las operaciones.
- **Reporte:** es una clase que se encarga de organizar los documentos para crear y guardar un reporte de extensión HTML para lo que son las aplicaciones y usuarios del programa.

## Métodos por clase

### Main

- **Main:** solo se le hace el llamado al método login(); para la inicialización del programa e inicialización de variables iniciales de reporte y administradores.

### Ingreso

- **Ingreso:** despliega la ventana de login para proceder ingresar a la ventana principal con la opción de ingresar haciendo llamado al método abrirventana.
- **abrirventana:** método que realiza con validación de los datos de usuario y contraseña para validar el acceso o denegarlo.
- **Principal:** método que permite interactuar con la primera página de los administradores con distintas opciones a realizar que coaduna llama a diferente ventana de la app para poder seguir con la aplicación.

### Pantallas

- **PerfilAdmin:** despliega la ventana para los el perfil de administrador solo sirve de pivote para todas las acciones ya que solo posee contenido gráfico y hace llamadas a distintas operaciones de si tantos lados para poder completar las operaciones así mismo para el demás método.
- **Recuperar:** despliega ventana a par poder recuperar el contraste con lo que es el usuario y la frase secreta que se desplegara en mensaje con la contraseña.
- **Usuario amín:** abre ventana para las agregar nuevos administradores a la lista de estos hacía referencia a otros métodos auxiliares como a addAdmib.
- **Aplicaciones:** ventana específica para las aplicaciones y sus aplicaciones al programa haciendo referencia a otros para las ediciones, búsquedas hasta eliminaciones además de generar el reporte con métodos como ReporteJuntar y ReporteApp, y también con CargaAplicaciones.
- **Editar:** Ventana para poder realizar la edición de los usuarios ya ingresado haciendo llamadas a distintos métodos auxiliares como EliminarUsuarioAdmin, EditarAdmin, ReportesUsuario y ReporteJuntar.

### Principal

- **Principal:** método con la finalidad de mostrar la parte grafica de la primera pagina de los usuarios administradores, desplegando una tabla inicial, teniendo diferentes funciones para llamados a otras metodos como: PerfilAdmin, PerfilAdmin, Editar y login.

### Almacenar datos

- **addAdmib:** su propósito es tener un procedimiento de validaciones para poder crear un nuevo usuario administrador con respecto a los datos establecidos en los textbox o mejor dicho JFields.
- **Tabla:** se encarga de poder establecer los valores a una tabla vacía.

## Cargas masivas

- **CARGAaplicaciones:** método encargado de la recolección de los datos de un documento para la obtención de todos los datos.

## Datosinteraccioneo

- **actializacionAdmin:** método enfocado a recolectar los datos de las actualizaciones de los datos de un administrador anteriormente discriminado.
- **Recuperar:** permite la búsqueda de coherencia de usuario y frase secreta para poder desplegar en pantalla la contraseña.
- **MOstrarApp:** muestra en una ventana flotante la búsqueda específica de una aplicación.
- **ActualizarCargas:** método enfocado a recolectar los datos para las actualizaciones de las aplicaciones, buscando errores de repitencia así procediendo a que modificarlos
- **EliminarCarga:** Elimina una aplicación según la id de una app.
- **EliminarUsuarioAdmin:** Elimina un administrador según la id de una app.
- **EditarAdmin:** método enfocado a recolectar los datos para las actualizaciones de los usuarios, buscando errores de repitencia así procediendo a que modificarlos.

## Reporte

- **ReporteJuntar:** permite juntas lo que es el cuerpo del reporte HTML para una mejor, fácil optimizada realización de métodos.
- **ReporteUsuarios:** permite crear la parte individual de los usuarios, es decir su propio reporte.
- **ReporteApps:** permite crear la parte individual de las aplicaciones, es decir su propio reporte.
- **ReporteMostrar:** Permite mostrar y sobre escribir el documento par poder visualizar el resultado final del reporte.
- **ReporteStart:** Reporte inicial y final para poder llevar un buen uso de HTML, ejecutándola desde el main con la finalidad de tener de manera más sencilla la manera de implantar código.

## Importaciones usadas

```
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
```