**Resumen**

En el transcurso de esta práctica laboral, se abordó el desafío de desarrollar un sistema integral para automatizar las operaciones de una central telefónica, facilitando la gestión de llamadas nacionales e internacionales. Para resolver esta problemática, se diseñó una solución que incluía clases bien estructuradas y una interfaz de usuario intuitiva, alineada con los requisitos establecidos. Este enfoque no solo optimizó el procesamiento de datos relacionados con abonados y tarifas, también permitió calcular costos y generar informes detallados sobre la actividad telefónica. Las pruebas realizadas confirmaron la eficacia del sistema, que se estableció como herramienta valiosa para la central telefónica.

Introducción

En un mundo donde la comunicación es un pilar esencial de la interacción humana y empresarial, las centrales telefónicas se encuentran ante el imperativo de modernizar sus operaciones. La creciente demanda de servicios de llamadas nacionales e internacionales plantea un desafío significativo: la necesidad de gestionar eficientemente un volumen cada vez mayor de tráfico telefónico. Este contexto exige la implementación de sistemas que no solo faciliten la conexión entre abonados, sino que también optimicen la administración de tarifas y permitan la generación de informes detallados sobre la actividad telefónica.

La problemática central radica en la automatización de procesos que, hasta ahora, han sido realizados manualmente. Esta dependencia de métodos tradicionales puede dar lugar a errores humanos y retrasos en la gestión, lo que afecta la calidad del servicio ofrecido a los usuarios. Además, es crucial contar con un sistema que proporcione un seguimiento preciso de las llamadas y un control efectivo sobre los costos asociados a cada tipo de comunicación. En este sentido, se hace evidente la necesidad de diseñar una solución que contemple tanto la gestión integral de datos de los abonados como el cálculo dinámico de tarifas basadas en el tipo y duración de las llamadas.

Para abordar esta complejidad, se requiere un diseño robusto y flexible que permita procesar información en tiempo real. Este enfoque debe garantizar no solo una experiencia fluida para los usuarios, sino también una administración eficiente por parte del personal de la central telefónica. El presente informe detalla el enfoque adoptado para desarrollar un sistema integral que responda a estas necesidades, así como los resultados obtenidos durante las pruebas realizadas, destacando su eficacia y potencial para transformar las operaciones de la central telefónica en cuestión.

***Excepciones***

En el desarrollo de nuestro programa, implementamos un sistema de manejo de errores para gestionar situaciones inesperadas de manera efectiva. Este enfoque nos permite anticipar y controlar problemas que podrían surgir durante la ejecución, garantizando que el programa se mantenga estable. Al ofrecer mensajes claros y comprensibles al usuario en caso de fallos, evitamos confusiones y mejoramos la experiencia general. A continuación, se presentarán las clases que responden a las excepciones, proporcionando una visión detallada de su implementación y propósito en el sistema.

**PhoneNumberAlreadyExistsE:** es lanzada por el método **addPhoneNumber()** en la clase **Client**. Su función es evitar que se agregue un número de teléfono que ya existe en el listado de números del cliente.

**PhoneNumberDoesNotExistE**: es lanzada por el método **removePhoneNumber()** en la clase **Client**. Su función es prevenir la eliminación de un número de teléfono que no se encuentra en el listado de números del cliente.

**ClientAlreadyExistsE:** es lanzada por el método **addClient()** en la clase **Central**. Su función es evitar que se intente añadir un cliente que ya está presente en el listado de clientes de la central telefónica.

**TaxAlreadyExistsE**: es lanzada por el método **addTax()** en la clase **Central**. Su función es evitar que se agregue una tarifa que ya existe en el listado de tarifas de la central telefónica.

**InvalidMonthE**: es lanzada por el método **getMonthEarning()** en la clase **Central**. Su función es verificar que el mes proporcionado como parámetro, del que se desea calcular su ganancia, sea un entero válido, es decir, que esté entre 1 y 12.

**InvalidOperationE**: es lanzada por el método **getMonthEarning()** en la clase **Central**. Su función es verificar que la operación proporcionada como parámetro sea un entero válido, es decir, que esté entre 1 y 3. Este rango responde al tipo de llamada del que se desea calcular la ganancia mensual: 1 para llamadas nacionales, 2 para internacionales y 3 para el total de llamadas del mes.

**WrongPeriodFormatE**: es lanzada por los métodos **getHotCountries()** ,**getHotProvince()** y **getClientOwe()** en la clase Central. Su función es verificar que el período proporcionado como parámetro tenga el formato correcto de (mes-mes). Para realizar esta validación, se apoya en el método auxiliar **checkPeriod()**, que retorna verdadero si el período no es válido. \*\*\*

**InvalidCredentials**: es lanzada por el método **verifyClient** en la clase **Central**. Esta excepción fue creada con el propósito de utilizarse en la clase de la interfaz de usuario **ClientLoginPanel** para verificar que el cliente esté registrado en la central. Si el cliente no está registrado, se lanza la excepción y se solicita que cree una cuenta. Al gestionar esta situación, garantizamos que el usuario reciba un mensaje claro y pertinente, facilitando así un proceso de inicio de sesión más intuitivo y eficiente.

**Clases de la Interfaz Gráfica de Usuario**

A continuación, se explicarán las clases de la interfaz gráfica de usuario que son fundamentales, ya que son el punto de interacción directa entre el usuario y el sistema. Estas clases están diseñadas para ofrecer una experiencia intuitiva y accesible, permitiendo a los usuarios realizar diversas operaciones de manera eficiente. La correcta implementación de estas clases no solo mejora la usabilidad del programa, sino que también facilita la comprensión y gestión de las funcionalidades disponibles.

**ClientLoginPanel:** da la bienvenida al usuario en la central telefónica y le brinda la opción de registrarse con su cuenta para acceder a sus datos. Si el usuario no está registrado, se lanza la excepción **InvalidCredentials**, lo que da paso al **ClientRegisterPanel** para que el usuario pueda crear una cuenta en la central telefónica. Además, si el usuario desea acceder a todos los datos de la central como administrador, este panel facilita la transición al **AdminPanel**. De esta manera, **ClientLoginPanel** actúa como un punto de entrada versátil, adaptándose a las necesidades del usuario y garantizando un acceso adecuado a las funcionalidades del sistema.

**ClientRegisterPanel:** está diseñada para aquellos usuarios que no están registrados en la central telefónica y desean crear una cuenta. Este panel solicita todos los datos necesarios, incluyendo el código del usuario y su tipo de cliente, y verifica que esta información sea correcta. Una vez validados los datos, se agrega al cliente a la central, permitiéndole disfrutar de todas las opciones y funcionalidades disponibles como cliente. De esta manera, **ClientRegisterPanel** facilita el proceso de registro, asegurando que los nuevos usuarios puedan acceder rápidamente a los servicios ofrecidos por la central telefónica.

**AdminLoginPanel**: se encarga de gestionar el proceso de inicio de sesión del administrador de la central telefónica. Ofrece al administrador la posibilidad de ingresar su código, el cual está establecido como "admin". En caso de que se introduzca un código diferente, el acceso será denegado. Este panel es fundamental para asegurar que solo personas autorizadas puedan acceder a toda la información sensible de la central telefónica, protegiendo así la integridad y seguridad de los datos.

**ClienPanel:** muestra al cliente, una vez registrado, sus números de teléfono y su historial de llamadas recientes. Además, ofrece las opciones de agregar o eliminar un número de teléfono, manejando correctamente las excepciones **PhoneNumberAlreadyExistsE** y **PhoneNumberDoesNotExistE**, asegurando así que las operaciones se realicen sin errores.

Una de las funcionalidades más importantes que proporciona este panel es la capacidad de realizar una llamada. Para ello, se presenta una ventana que solicita toda la información necesaria para la llamada ,según el paquete DAYOS, y se verifica que todos los datos ingresados sean correctos antes de proceder. Una vez validados, la llamada se guarda en el listado de llamadas de la central telefónica, garantizando un registro preciso y eficiente de las interacciones del cliente.

**AdminPanel**: se erige como la más crucial de todas, ya que proporciona una visión completa de toda la información relacionada con la central telefónica. Este panel otorga al administrador la capacidad de analizar todos los clientes, sus números de teléfono y el historial de llamadas, incluyendo detalles relevantes sobre cada interacción.

Además, cuenta con un panel titulado "Ganancias en un mes", que se conecta con el método **getMonthEarning()** de la clase Central. En este espacio, el administrador puede ingresar el mes del año que desea examinar y el tipo de llamadas para obtener la ganancia mensual correspondiente. También incluye otro panel que permite al administrador introducir un número de teléfono deseado y utilizar el método **getClientOwe** de la clase Central para conocer lo que debe pagar ese cliente.

Por otro lado, dispone de un botón titulado "Ver estadísticas", que dirige al **StatsPanel**, donde se pueden explorar otras funcionalidades importantes de la central telefónica. De esta manera, el **AdminPanel** actúa como un centro integral para la gestión y análisis de datos dentro de la central, facilitando a los administradores el acceso a información crítica para la toma de decisiones.