**INFORME**

1. **INTRODUCCIÓN**

El presente proyecto consiste en desarrollar una aplicación con la funcionalidad de contratación de personal para la empresa “MiPrimerContrato.co”, el cual tendrá un cliente y un servidor. El cliente realizará solicitudes hacia un servidor para consultar e ingresar de datos, y subir un documento PDF para el proceso de contratación. El servidor estará escuchando las peticiones de los clientes, y consultará los datos necesarios desde una base de datos, para posteriormente enviar una respuesta al cliente de acuerdo a la solicitud requerida.

1. **DISEÑO**

Para el desarrollo de este proyecto, se utilizó el lenguaje de programación C# empleando el IDE Visual Studio. Dentro de este entorno, se definieron tres proyectos dentro de la solución: uno para el cliente, uno para el servidor y una biblioteca de clases. La comunicación se la realizará empleando el protocolo TCP, junto con la Serialización de Objetos.

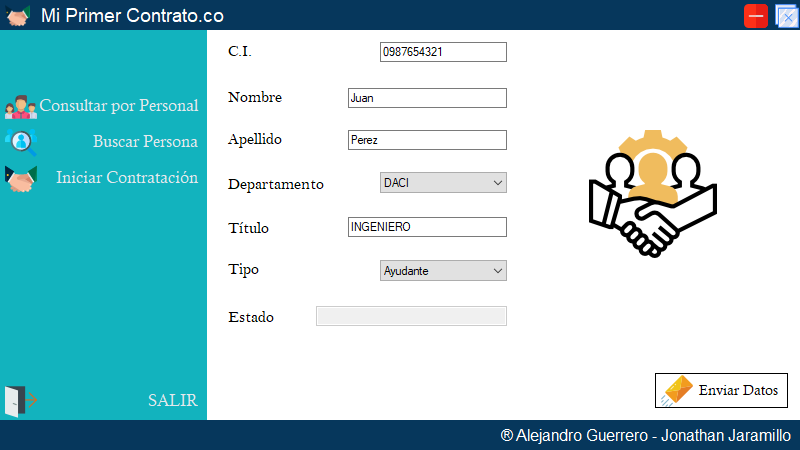
* 1. **Diseño de la Biblioteca de Clases**

La biblioteca de clases abarcará a un par de clases que serán utilizadas para el desarrollo de la aplicación. Estas clases son:

* **Persona:** esta clase será utilizada para contener todos los datos necesarios para el proceso de contratación. Esta clase será Serializable con el objetivo de enviar un objeto de dicha clase hacia el servidor, y que este pueda procesarla en base a su atributo “*comando*”, el cual especificará el tipo de solicitud que requiera el cliente.
* **ConexionBaseDeDato:** esta clase permite al servidor poder interactuar con la base de datos para agregar y consultar a una persona específica, guardar el documento PDF que el cliente envíe y obtener la cantidad de personas contratadas de acuerdo al Tipo de Personal.
  1. **Diseño del Cliente**

El cliente dispondrá de un conjunto de formularios con los cuales podrá interactuar para enviar las distintas solicitudes al servidor, y obtener una respuesta por parte de este.

En la Figura 1, se muestra el formulario donde el cliente iniciará el proceso de contratación ingresando sus datos personales y enviándolos al servidor. Posteriormente obtendrá una respuesta si los datos fueron ingresados con éxito o no.



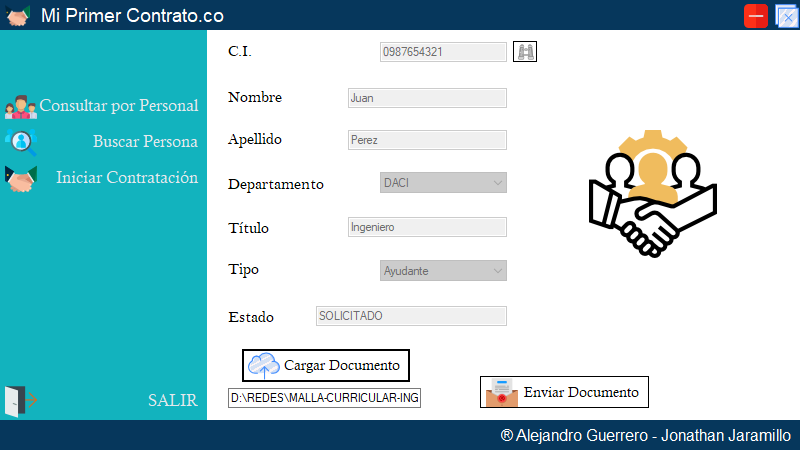
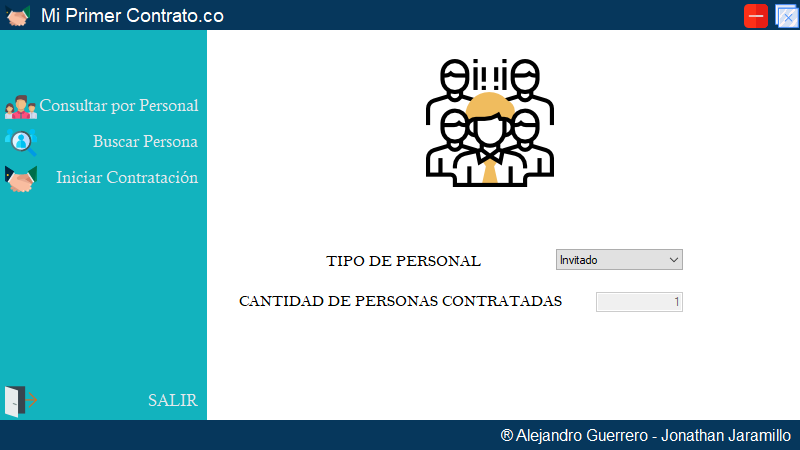
**Fig. 1.** Ingreso de datos.

El formulario mostrado en la Figura 2 le permite al cliente consultar su estado dentro del proceso con su número de cédula. Si su estado es solicitado, este podrá cargar un documento PDF para enviarlo al servidor y a su vez ingresará a la base de datos. El servidor responderá si desea ser contratado para lo cual el cliente enviará su respuesta, actualizando su estado.

Finalmente en la Figura 3, se encuentra el formulario donde el cliente podrá consultar la cantidad de personas contratadas de acuerdo al tipo de personal.

* 1. **Diseño del Servidor**

En el lado del servidor, se realiza la Deserialización del objeto Persona enviado por el cliente para procesar su solicitud.

**Fig. 2.** Formulario para enviar un documento PDF al servidor. **Fig. 3.** Formulario de consulta de personas contratadas.

El desafío más grande que se tuvo fue el manejo de las diferentes peticiones del cliente en el lado del servidor, por ello, la solución que encontramos fue emplear un atributo adicional en la clase Persona que defina la solicitud del cliente, y para poder procesar todo el conjunto de datos de la clase en el lado del servidor, se procedió a Serializar la clase con la finalidad de enviar el objeto y realizar la Deserialización del mismo para leer cada uno de sus campos y procesar dicha información.

En nuestra opinión, la nota que nos merecemos es 10/10, debido a que el sistema desarrollado cumple con todos los requisitos planteados; desde el envío y almacenamiento de datos, hasta las respuestas generadas por parte del servidor.

Al implementar este programa se adquirió nueva experiencia en el manejo de comunicaciones, sobre todo el envío de objetos Serializados los cuales pueden contener en sus atributos distintos tipos de datos, sobre todo *bytes* referentes a un documento (en este caso PDF) y enviarlos en un solo conjunto al servidor para que este pueda procesarlos correctamente.

1. **CONCLUSIONES**

**Alejandro Guerrero**

* Al emplear la Serialización de objetos podemos representar, transportar y almacenar los datos de manera fácil, debido a que este mecanismo permite representar al objeto como un arreglo de bytes, para posteriormente realizar el proceso inverso y tener al objeto tal como fue creado.

**Jonathan Jaramillo**

1. **RECOMENDACIONES**

**Alejandro Guerrero**

* Al ser un programa que requiere el envío de documentos, se recomienda utilizar el protocolo TCP para que no se pierda información.

1. **BIBLIOGRAFÍA**

**[1]** Ceballos, J. “Enciclopedia de Microsoft *Visual C#. Interfaces Gráficas y Aplicaciones para Internet con Windows Forms y ASP.NET*”. 4th ed. RA-MA Editorial. Madrid: España (Octubre, 2013)

**[2]** Albahari, J. & Albahari, B. “*C# 6.0 in a Nutshell*”. 6th ed. O’Reilly Media, Inc. United States of America (Diciembre, 2015)

**[3]** Cosio, N. “*C# Avanzado*”. 1st ed. Fox Andina. Buenos Aires: Argentina (2011)