



**UNIVERSIDAD DE LA FRONTERA**

# Gestor PhoneBook VLB

## PhoneBook VLB Manager

### Avance 02

Integrantes:

Bayron Muñoz

Victor Mardones

Luis Burgos

Profesor: Samuel Sepúlveda

Asignatura: Programación

Fecha: 28/08/2020

# **Índice**

<b>Índice</b>	<b>2</b>
<b>Introducción</b>	<b>3</b>
<b>Retroalimentación</b>	<b>4</b>
<b>Diferencias entre la planificación inicial y la real</b>	<b>5</b>
<b>Sección de solución del software</b>	<b>6</b>
<b>Diseño de la interfaz gráfica</b>	<b>9</b>
<b>Diagrama estructural del programa</b>	<b>15</b>
<b>Datos que gestiona la solución</b>	<b>15</b>
<b>Conclusión</b>	<b>16</b>
<b>Anexos</b>	<b>17</b>
<b>Autoevaluación de Victor Mardones:</b>	<b>18</b>
<b>Autoevaluación de Luis Burgos:</b>	<b>19</b>
<b>Autoevaluación de Bayron Muñoz:</b>	<b>20</b>
<b>Coevaluación:</b>	<b>21</b>

## Introducción

En un mundo globalizado como el actual, la comunicación es un medio muy utilizado por la mayoría de las personas en el mundo, haciendo posible la transmisión de información desde cualquier lugar del planeta Tierra. Existen varias herramientas de comunicación, pero frecuentemente se usan los teléfonos fijos o móviles. Estos objetos utilizan una secuencia de dígitos llamada comúnmente números telefónicos que sirve para identificar la línea telefónica dentro de una red conmutada. Para administrar estos números se necesita de un programa que los pueda almacenar de manera ordenada y ante esta necesidad se crea el proyecto Gestor PhoneBook VLB.

El Gestor PhoneBook VLB es un proyecto en desarrollo que tiene como objetivo el administrar números telefónicos para el fácil acceso del usuario a esta información de contacto cuando sea requerida.

Este documento se encargará de explicar los avances hechos en el proyecto hasta ahora, con fecha del segundo avance. Se incorpora la retroalimentación del profesor, aplicación de POO, descripción de las clases que conforman el proyecto (atributos y métodos), modelo UML de las clases considerando las relaciones y multiplicidad, diseño de las GUIs con su mapa de navegación y comentar la función que hace cada una, definición de la gestión de datos a partir del avance 01, carta gantt que se encuentra en formato PDF, conclusiones en base a lo aprendido en clase y anexos. Se optó por no añadir bibliografía en este informe, ya que no parece que corresponda dado el proyecto que estamos realizando. No obstante, la mayoría de los anexos pueden encontrarse en los documentos dentro del GitHub del proyecto.

## **Retroalimentación**

### **Primer Avance oficial:**

En síntesis, el primer avance que se ha entregado tenía como objetivo explicar a profundidad el proceso de creación del programa, creando el repositorio en github para tener un respaldo de cómo se ha ido avanzando en el código, prototipo del diseño gráfico, el diagrama para representar las clases y sus funciones, las pruebas unitarias, datos que gestiona la solución para el almacenamiento de los contactos, etc. Además, se han terminado los avances propuestos del avance 01, estos incluían apodo, dirección, fecha de cumpleaños, posibilidad de guardar múltiples teléfonos, apodos, emails y notas dentro de cada contacto, y un gestor de archivos que permite guardar y editar archivos en formato JSON, quedando pendiente el manual del programa escrito en PDF y la implementación de la interfaz gráfica.

Como sugerencia se ha mencionado que el diseño de la interfaz gráfica debe estar listo a estas alturas, detallar sobre los datos y archivos, para tener claro qué datos, formatos y archivos serán usados, también se indicó que los enlaces de la Carta Gantt y de la interfaz gráfica en Moqups habían caducado o no se podían acceder.

Las sugerencias mencionadas anteriormente se han tomado en cuenta y se han mejorado de acuerdo con los aspectos del informe, los cuales han sido incluidos en el mismo.

## **Diferencias entre la planificación inicial y la real**

El proyecto Gestor PhoneBook VLB no ha cambiado demasiado con respecto a la planificación inicial que se tuvo al formular el proyecto, cabe mencionar algunos de los cambios de planificación que se hicieron :

- En un principio se planificó que se pudieran hacer llamadas directamente desde el gestor. El proceso para implementar esta función resulta ser bastante complicado dado el conocimiento actual sobre programación del grupo de proyecto, por lo cual se decidió no implementarse en el proyecto final.
- Se discutió sobre la posibilidad de añadir fotos de perfil para cada contacto, para que el usuario los pueda identificar fácilmente cuando se implemente la interfaz gráfica con el uso de JFrame. Dada la baja prioridad que se ha dado a esta tarea y la poca importancia que tiene para el funcionamiento del programa, se ha decidido dejarla como una funcionalidad adicional que podría ser implementada cuando el resto del proyecto esté prácticamente listo. Esta tarea puede encontrarse al final de nuestra Carta Gantt.
- Otra funcionalidad que se descartó debido a su poca importancia con respecto a la ejecución de las funciones esenciales del programa es la posibilidad de exportar e importar la información sobre la agencia (que se guarda en un archivo JSON) directamente a los sistemas operativos de teléfonos inteligentes iOS y Android. Esta tarea puede encontrarse al final de la Carta Gantt, ya que se implementará sólo si el resto del proyecto está prácticamente listo y se tiene el tiempo suficiente.
- Además de los tres puntos mencionados en esta sección, el resto de los cambios han sido mejoras para la implementación original del proyecto en el avance 01. Estas mejoras serán explicadas con mayor detalle dentro de las presentaciones sobre la defensa y el código junto a su ejecución del avance 02, las cuales vienen incluidas en la entrega, junto a este informe.

## Sección de solución del software

Para satisfacer las expectativas de un buen proyecto se implementarán las herramientas entregadas en clase, conocimientos sobre la POO, herencia, nivel de encapsulamiento, modelado de clases con UML disponible el acceso web en los anexos del informe y herramientas que se usaron para el desarrollo del avance 01.

La solución presentada para el avance 02 consta de 3 paquetes (packages): lanzador y phonebook (clase modelo) y json (clase que gestiona los archivos).

1. lanzador: este paquete sólo contiene una clase:
  - **Principal:** Se encarga de hacer todo lo necesario para lanzar el programa de manera correcta. Toda su funcionalidad depende de las clases dentro de los otros paquetes.
  
2. phonebook: este paquete tiene las clases necesarias para que el programa se ejecute correctamente, las cuales se muestran a continuación:
  - **Agenda:** Almacena los contactos y hace uso de sus funciones para gestionarlos.
  - **Contacto:** Contiene atributos necesarios para almacenar los datos de contacto de una persona.
  - **Direccion:** Contiene atributos de la dirección de la persona con su respectivo tipo (Ciudad, Calle, Número).
  - **FechaCumple:** Contiene atributos de la fecha de cumpleaños de la persona.
  - **Telefono:** Contiene atributos del numero telefónico de la personas con su respectivo tipo (Celular, Fijo, Trabajo).
  - **Menu:** Contiene atributos y métodos para los menús en general. Su uso principal es para crear una base que el resto de menús dentro del código puede heredar.

- **MenuPrincipal:** Contiene los datos del menú al que se accede desde la clase Lanzador cuando el programa comienza su ejecución. Extiende la clase Menu.
- **MenuEditor:** Funciona como menú de edición de los contactos, permitiendo al usuario elegir los datos que desea editar. Al salir de este menú se pide confirmación para guardar los cambios hechos. Extiende la clase Menu.
- **Editor:** Contiene los atributos necesarios para editar un atributo de una instancia de la clase Contacto. Editor extiende la clase Menu. No se usa directamente, pero todas las clases editoras heredan de esta clase.
- **EditorNombre:** Permite editar el nombre del contacto seleccionado. Extiende la clase Editor.
- **EditorTelefono:** Permite editar los números telefónicos del contacto seleccionado. Extiende la clase Editor.
- **EditorDireccion:** Permite editar la dirección del contacto seleccionado. Extiende la clase Editor.
- **EditorFechaCumple:** Permite editar la fecha de cumpleaños del contacto seleccionado. Extiende la clase Editor.
- **EditorEmail:** Permite editar los emails del contacto seleccionado. Extiende la clase Editor.
- **EditorApodo:** Permite editar el apodo del contacto seleccionado. Extiende la clase Editor.
- **EditorNota:** Permite editar las notas del contacto seleccionado. Extiende la clase Editor.
- **Validador:** Contiene los métodos que se usan en otras clases para la validación de Strings, emails o números enteros. Es posible que para el próximo avance esta clase sea movida a su propio paquete de utilidades.

3. json: este package se encarga de administrar los datos que ofrece el package phonebook para dejarlos en un archivo JSON.
- **GestorArchivo:** Clase que permite leer, borrar y crear archivos. La funcionalidad de la clase GestorJson depende de la utilidad que proporciona esta clase.
  - **GestorJSON:** Se encarga de convertir los datos dentro de la agenda a un conjunto ordenado de JSONArrays y JSONObject, y después los guarda en un archivo JSON, mediante el uso de la clase GestorArchivo. Se usa cuando se hacen cambios a la agenda y al iniciar el programa, para cargar los datos de un archivo “agenda.json”, sólo si es que existe.

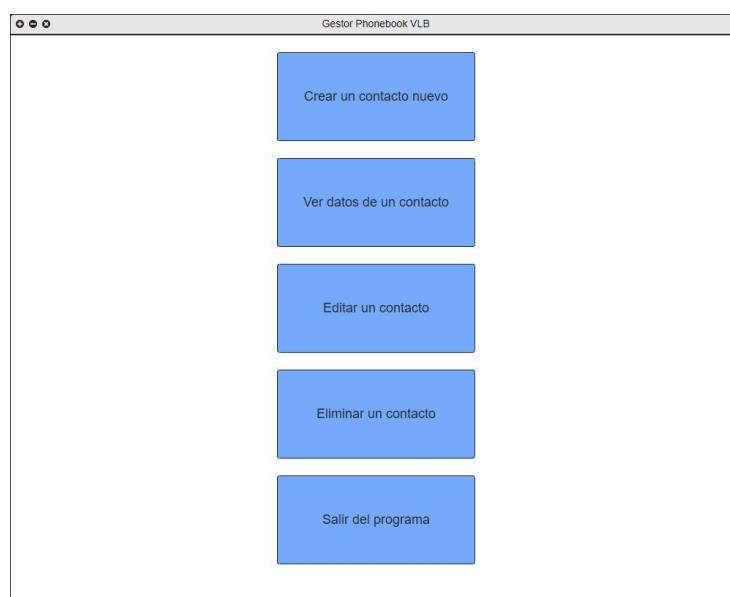


## Diseño de la interfaz gráfica

La interfaz gráfica está definida y lista para llevarla a código y ser implementada oficialmente para el avance 03 del proyecto. El diseño se puede encontrar en la plataforma Moqups a través del siguiente link (si no funciona, por favor hágase saber lo antes posible mediante Slack y se enviará un enlace nuevo):

<https://app.moqups.com/xblfzqRdYi/view/page/a11d4eb35>

Se explicará en detalle el diseño a continuación:



Este es el menú principal del programa, funciona con botones que proporcionan una navegación por las distintas páginas creadas, similar a lo que hace la clase MenuPrincipal.

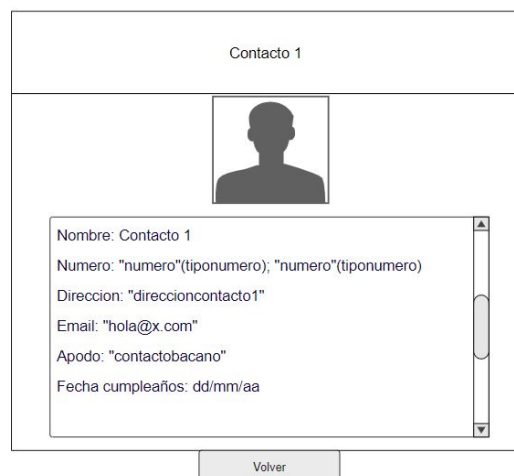


Para simplificar la explicación se toman capturas del contenido principal que tiene la ventana.



Al presionar los botones del menú principal, se mostrará una ventana similar a esta en las paginas verDatosContacto, editarContacto y eliminarcontacto. Esta interfaz lista todos los contactos creados con el botón de crear contacto o cargados del archivo que almacena los contactos en formato JSON, tiene un buscador de contactos por nombre y un icono, cada contacto tendra un boton que depende la pagina que se encuentre tendra una funcion especifica.

Cuando se presiona el botón de un contacto de la lista (previamente presionando el botón de ver datos de un contacto en el menú principal) se muestra la siguiente interfaz con los datos que la agenda proporciona.



Al presionar el botón de editar un contacto, luego seleccionar un contacto se mostrará la interfaz siguiente que tiene varias opciones a escoger por el usuario.

"Contacto 1"

Cambiar Nombre	Editar números de telefono	Editar direccion
Editar fecha de cumpleaños	Editar emails	Editar apodos
	Editar notas	

Volver

A continuación se mostrarán imágenes con cada opción.

Esta página se encarga de modificar el nombre de un contacto ya ingresado.

Nombre a editar de "Contacto 1"

Ingrese nuevo nombre

Ingrese texto

Guardar

Cancelar

Volver

El botón para editar número de teléfono abre esta página y permite ver los números registrados y luego tiene un botón agregar un nuevo número de teléfono con su tipo disponible.

Numero a editar de "Contacto 1"

1- "99999"; "fijo"

2- "5225232"; "celular"

3- "242425"; "casa"

4- "533553"; "celular"

Editor de Dirección

Volver

Nuevo numero

Ingrese nuevo numero

☒ Celular

☐ Fijo

☒ casa

Ingrese texto

Guardar

Cancelar

El botón para editar dirección abre esta lámina y permite el ingreso de una nueva dirección que contiene ciudad, calle y número.

Direccion a editar de "Contacto 1"

Ingrese nueva direccion

☒ Ciudad

Ingrese texto

☒ calle

Ingrese texto

☒ numero

Ingrese texto

Guardar

Cancelar

Volver

Nueva direccion

Al presionar el botón para editar la fecha de cumpleaños se despliega la página siguiente, que contiene un botón para agregar la nueva fecha de cumpleaños con un calendario.

Fecha de cumpleaños a editar de "Contacto 1"

Volver

Nueva fecha

Ingresa una nueva fecha

August 25, 2020

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Guardar

Cancelar

Para editar emails u apodos las interfaces gráficas son las siguientes:

Donde muestran su contenido y el botón para agregar nueva información.

Emails a editar de "Contacto 1"

Lista de Emails

1- "jasjash@gmail.com"  
2- "jashjas@ufromail.cl"  
3- "prueba@hotmail.com"  
4- "universidas@hola.com"

Volver

Nuevo Email

Apodos a editar de "Contacto 1"

Lista de apodos

1- "jasjash"  
2- "jashjas"  
3- "prueba"  
4- "universidas"

Volver

Nuevo apodo

ingrese nuevo Email

Guardar

Cancelar

Ingresa un nuevo apodo

Guardar

Cancelar

Y el botón de eliminar contacto abre la información que tiene el contacto que se quiere eliminar y al presionar el botón manda una señal si quiere eliminar o no.

Contacto 1



Nombre: Contacto 1  
Numero: "numero"(tiponumero); "numero"(tiponumero)  
Direccion: "direccioncontacto1"  
Email: "hola@x.com"  
Apodo: "contactobacano"  
Fecha cumpleaños: dd/mm/aa

Volver

Eliminar

¿Desea eliminar el contacto?

Eliminar

Cancelar

Al diseño de la interfaz le faltan pequeñas modificaciones que no pudieron ser implementadas ya que el programa ofrece una cantidad limitada de objetos que se le pueden añadir a las páginas. Para este caso solo es capaz de añadir datos a cada atributo del contacto, para el tercer avance será posible ver las opciones de borrar y editar datos de un atributo del contacto.

## Diagrama estructural del programa

Para modelar mejor la estructura y función que realiza el programa se utiliza el Lenguaje Unificado de Modelado (UML) en la plataforma Visual Paradigm.

El proyecto consta actualmente de tres paquetes, de los cuales uno de ellos guarda las clases que administran la agenda (phonebook), otro guarda las clases que permiten guardar los datos de la agenda (json) y el último se encarga de lanzar el programa (lanzador). Estos paquetes tienen relaciones. Asimismo estos tienen clases que se relacionan entre sí.

(enlace del modelado de clases en Visual Paradigm Online en el anexo).

## Datos que gestiona la solución

Los datos que gestiona el programa PhoneBook VLB:

Los contactos y sus datos estarán escritos en el archivo llamado “agenda.json”, es un archivo JSON que contiene los contactos dentro de la agenda y sus datos en formato JSON como se muestra a continuación:

```
[
  {
    "contacto": {
      "emails": ["doym@ufromail.cl"],
      "fechacumple": {
        "numeroMes": 11,
        "mes": "noviembre",
        "dia": 11
      },
      "notas": [
        "Saca malas notas",
        "No tiene amigos",
        "Tiene 2 perros"
      ],
      "direccion": {
        "numero": 8,
        "ciudad": "temuco",
        "calle": "Francisco salazar"
      },
      "telefonos": [
        {
          "tipo": "Celular",
          "numero": 973637287
        },
        {
          "tipo": "Fijo",
          "numero": 666666
        }
      ],
      "apodos": ["java"],
      "nombre": "Javier"
    },
    {"contacto": {
      "telefonos": [{
        "tipo": "Celular",
        "numero": 9999999
      }],
      "nombre": "ba1"
    }},
    {"contacto": {"nombre": "hola"}},
    {"contacto": {"nombre": "m"}}
  ]
```

Este formato de archivo permite el almacenamiento de datos, donde cada contacto puede manejarse como un objeto desde Java, permitiendo exportar o importar los contactos solo traspasando este archivo a la raíz del programa, o verlos en una página web, almacenarlos en una base de datos, etc.

Como se muestra en la imagen de la izquierda del informe, cada contacto tiene sus datos almacenados dentro del archivo JSON, los cuales se irán actualizando durante la ejecución del programa (al momento de crear, editar o eliminar un contacto) y se cargan durante su inicio.

## Conclusión

Como se ha mencionado en el informe del avance 01 la retroalimentación es una parte fundamental brindada por el profesor para guiar el avance del proyecto por buen camino. Además, la materia vista en clase cumple un rol muy importante y necesario para el siguiente avance.

La creación del diagrama UML del proyecto nos permitió usar una herramienta que agiliza en gran cantidad el desarrollo del proyecto, mejorando la planificación y control que deben llevar las futuras implementaciones de código, y que ayuda a la comprensión del código por parte de nuevos desarrolladores.

Como la carga laboral es bastante elevada en estos tiempos, precisar una herramienta para manejar las tareas (como Ganttproject) es necesario para el desarrollo pausado y eficiente. El sitio web GitHub fue el que hizo posible la conexión remota del trabajo utilizando el sistema de control de versiones git, facilitando el traspaso de código base ahorrando tiempo. El uso de el formato JSON es una ventaja para poder almacenar los datos en un archivo que después se puede exportar a una página web o a otra computadora o dispositivo tecnológico compatible con este formato.

Gestor PhoneBook se creó con el fin de ayudar a las personas a administrar sus contactos de manera sencilla. Se sabe que aún faltan mejoras a implementar y se tomarán en cuenta para el tercer avance del proyecto. Para dicho avance se va a retomar la idea de crear el manual para el uso del programa, tener una interfaz gráfica funcional y analizar la posibilidad de agregar imágenes para los contactos.

La realización de este proyecto ayuda a formar profesionales capacitados para resolver cualquier problema relacionado con el ámbito de la programación, aplicando conceptos que se han visto a lo largo de las clases y permitiendo analizar los problemas desde otra perspectiva (la perspectiva de un programador). Esto ha generado en nosotros una actitud de esfuerzo y dedicación constante para prepararnos para el futuro profesional.



## Anexos

### Enlaces de utilidad sobre el trabajo:

Repositorio del proyecto en GitHub:

[https://github.com/VictorMardones/Gestor\\_PhoneBook\\_VLB](https://github.com/VictorMardones/Gestor_PhoneBook_VLB)

Carta Gantt:

[https://drive.google.com/file/d/1BvpC6a\\_10Ra5qB5gEKPj0eXb01FkisYf/view?usp=sharing](https://drive.google.com/file/d/1BvpC6a_10Ra5qB5gEKPj0eXb01FkisYf/view?usp=sharing)

Diagrama UML del programa (archivo en el drive del avance 02, favor de abrir con Visual Paradigm Online):

[https://drive.google.com/file/d/1\\_YfIYZNmhl12iphlD8SVLC-OCG4lrwWd/view?usp=sharing](https://drive.google.com/file/d/1_YfIYZNmhl12iphlD8SVLC-OCG4lrwWd/view?usp=sharing)

Diseño de Interfaz Gráfica:

<https://app.moqups.com/xblfzqRdYi/view/page/a11d4eb35>

Carpeta con Defensa y PPT:

[https://drive.google.com/drive/folders/19L2yYKkcsW8UGU9I46\\_coZ6FgXiqPY95?usp=sharing](https://drive.google.com/drive/folders/19L2yYKkcsW8UGU9I46_coZ6FgXiqPY95?usp=sharing)

La mayoría de los documentos también se puede encontrar en el repositorio de GitHub:

[https://github.com/VictorMardones/Gestor\\_PhoneBook\\_VLB/tree/master/documentos](https://github.com/VictorMardones/Gestor_PhoneBook_VLB/tree/master/documentos)

## Autoevaluación de Victor Mardones:

Criterios	Indicadores	Puntaje
Organización	Cumplí con las actividades de trabajo comprometidas en los tiempos asignados.	5
	Cumplí con las actividades de trabajo comprometidas respetando y ajustándome a pautas y formatos de entrega.	5
	Participé en todas las reuniones de trabajo previo a la presentación de la propuesta y durante la construcción de esta	3
	Cumplí con las actividades de trabajo comprometidas	5
Trabajo colaborativo	Compartí ideas y opiniones con el grupo respecto de los desafíos planteados en la tarea	4
	Demostre respeto por las opiniones e ideas de mis compañeros respecto de los desafíos planteados en la tarea.	5
	Mantuve una actitud de colaboración y apoyo a mis compañeros de grupo con el propósito de construir una solución a la problemática entregada.	5
	Busqué y use materiales de información en servicio del trabajo solicitado.	5
Total:		37

## Autoevaluación de Luis Burgos:

Criterios	Indicadores	Puntaje
Organización	Cumplí con las actividades de trabajo comprometidas en los tiempos asignados.	2
	Cumplí con las actividades de trabajo comprometidas respetando y ajustándome a pautas y formatos de entrega.	2
	Participé en todas las reuniones de trabajo previo a la presentación de la propuesta y durante la construcción de esta	3
	Cumplí con las actividades de trabajo comprometidas	2
Trabajo colaborativo	Compartí ideas y opiniones con el grupo respecto de los desafíos planteados en la tarea	5
	Demostré respeto por las opiniones e ideas de mis compañeros respecto de los desafíos planteados en la tarea.	5
	Mantuve una actitud de colaboración y apoyo a mis compañeros de grupo con el propósito de construir una solución a la problemática entregada.	5
	Busqué y use materiales de información en servicio del trabajo solicitado.	3
Total:		27

## Autoevaluación de Bayron Muñoz:

Criterios	Indicadores	Puntaje
Organización	Cumplí con las actividades de trabajo comprometidas en los tiempos asignados.	3
	Cumplí con las actividades de trabajo comprometidas respetando y ajustándome a pautas y formatos de entrega.	5
	Participé en todas las reuniones de trabajo previo a la presentación de la propuesta y durante la construcción de esta	4
	Cumplí con las actividades de trabajo comprometidas	4
Trabajo colaborativo	Compartí ideas y opiniones con el grupo respecto de los desafíos planteados en la tarea	5
	Demostre respeto por las opiniones e ideas de mis compañeros respecto de los desafíos planteados en la tarea.	5
	Mantuve una actitud de colaboración y apoyo a mis compañeros de grupo con el propósito de construir una solución a la problemática entregada.	3
	Busqué y use materiales de información en servicio del trabajo solicitado.	3
Total:		32

## Coevaluación:

Criterios	Indicadores	Puntajes		
		Bayron Muñoz	Luis Burgos	Victor Mardones
Organización	Cumplió con las actividades de trabajo comprometidas en los tiempos asignados.	5	4	5
	Cumplió con las actividades de trabajo comprometidas respetando y ajustándose a pautas y formatos de entrega.	5	5	5
	Participó en todas las reuniones de trabajo previo a la presentación de la propuesta y durante la construcción de esta	5	4	5
	Cumplió con las actividades de trabajo comprometidas	5	4	5
Trabajo colaborativo	Compartió ideas y opiniones con el grupo respecto de los desafíos planteados en la tarea	4	4	5
	Demostró respeto por las opiniones e ideas de mis compañeros respecto de los desafíos planteados en la tarea.	5	5	5
	Mantuvo una actitud de colaboración y apoyo a mis compañeros de grupo con el propósito de construir una solución a la problemática entregada.	5	3	5
	Buscó y usó materiales de información en servicio del trabajo solicitado.	4	5	5
<b>Total:</b>		38	34	40