

# Análisis de requerimientos COP16

**Juan Manuel Ramirez - A00381746**

**Victor Manuel Cardona - A00406968**

**Análisis de requerimientos - Primera fase COP16**

Cliente	Alcaldía
Usuario	<ul style="list-style-type: none"><li>Organizador voluntario</li></ul>
Requerimientos funcionales	<p>R0: Registrar el organizador voluntario</p> <p>R1: Posibilidad de escoger ruta programada</p> <p>R2: Registrar la cantidad de participantes y guías</p> <p>R3: Registrar datos meteorológicos</p> <p>R4: Calcular los buses necesarios para transportar a la cantidad de personas registradas</p>
Contexto del problema	<p>La COP16 realiza eventos orientados a educar a la comunidad caleña sobre la biodiversidad. Entre estos eventos se encuentran los de la Franja Multicolor, los cuales cuentan con rutas a sitios biodiversos de la ciudad de Cali.</p> <p>Una temperatura adecuada para llevar a cabo las rutas es entre 20 C° y 25 C°, con una humedad entre el 40% y el 60%.</p> <p>Para este tipo de caminatas se debe conocer la identificación del encargado del grupo, así que se requiere nombre y cédula del organizador voluntario.</p> <p>Para conocer la cantidad de buses necesarios, se necesita saber la cantidad de asistentes a las caminatas; sabiendo que cada bus tiene una capacidad de 25 personas, incluyendo participantes y guías.</p> <p>A la hora de programar la ruta que se debe llevar a cabo, se debe conocer punto de encuentro, hora de inicio y hora de finalización de la actividad</p>

	<p>Las rutas se llevan a cabo del 19 de Octubre hasta el 1 de Noviembre</p> <p>Hay 3 rutas programadas para visitar.</p> <p>El organizador voluntario debe estar al tanto de las rutas disponibles.</p>
Requerimientos no funcionales	<ul style="list-style-type: none"> <li>- Un cálculo efectivo de la cantidad de buses necesarios</li> <li>- Interfaz intuitiva</li> <li>- Facilidad para digitar la información</li> <li>- Actualización constante de los horario disponibles para las caminatas</li> <li>- La aplicación debe estar lista antes del 19 de Octubre</li> <li>- Implementación de restricciones en los datos ingresados por el usuario</li> </ul>
Requerimientos de proceso	Realizarlo en el lenguaje de programación Java

Nombre o identificador	R0: Registrar el organizador voluntario		
Resumen	El sistema debe permitir que el organizador voluntario digite la información para el registro, pidiendo nombre y cédula, y darle un mensaje de bienvenida		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	Número de documento	String	Sólo números enteros como mínimo 5 dígitos y máximo 15
	Nombre del organizador	String	Sólo letras
Resultado o Postcondición	Se actualiza la información del organizador en la base de datos, y se muestra un mensaje de error si no se cumplen las condiciones de la entrada, si se cumplen, el programa le da la bienvenida		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	Mensaje de confirmación	String	“Bienvenido [Nombre]”

	Mensaje de error	string	“Error. Los datos ingresados no son válidos”
--	------------------	--------	--

Nombre o identificador	R1: Posibilidad de escoger una ruta programada		
Resumen	El sistema debe permitir que el organizador voluntario pueda digitar el nombre de la ruta que se desea programar, para que se pueda actualizar de manera correcta la base de datos.		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	Nombre de la ruta	String	Solo letras. Digitar rutas disponibles
Resultado o Postcondición	Se despliega en la pantalla el punto de encuentro de la actividad, hora de inicio y hora de finalización de esta..		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	Punto de encuentro de la actividad	String	“Barrio[ nombre]”o “cra x # x-x”
	Hora de inicio	String	“XX:XX am/pm”
	Hora de finalización	String	“XX:XX am/pm”

Nombre o identificador	R2: Registrar la cantidad de participantes y guías		
Resumen	Permite que el organizador voluntario digite el número de participantes y guías que irán a la actividad indicada, para tener un registro de los asistentes a la actividad		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	Cantidad de participantes	Int	Sólo números, mayores que 0

	Cantidad de guías	Int	Sólo números, mayores que 0
Resultado o Postcondición	Se guardan estos datos en el sistema		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	N/A	N/A	N/A

Nombre o identificador	R3: Registrar datos meteorológicos		
Resumen	El sistema debe permitir al organizador voluntario ingresar los datos meteorológicos. En caso tal de que la temperatura esté entre 20 C° y 25 C° y la humedad relativa sea de 40% y 60%, se le debe mostrar un mensaje al usuario.		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	Temperatura	Double	Sólo números
	Humedad	Double	Sólo números
Resultado o Postcondición	Si la temperatura registrada está entre 20 y 25 grados, y la humedad relativa está entre 40 y 60 por ciento, se le mostrará un mensaje al usuario		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	Mensaje de confirmación	String	“¡Hace un buen día para caminar por Cali!”

Nombre o identificador	R4: Calcular los buses necesarios para transportar a la cantidad de personas registradas		
Resumen	Según la cantidad de guías y participantes que el usuario digitó, se hace el cálculo de los buses necesarios para la caminata; teniendo en cuenta que cada bus puede transportar a 25 personas		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	N/A	N/A	N/A
Resultado o Postcondición	Número de buses necesarios para realizar la caminata		

	Nombre salida	Tipo de dato	Formato
Salidas	Cantidad de buses	String	“Como la cantidad de participantes es X, se necesitan X buses para llevar a cabo la actividad”

### Análisis de requerimientos - Segunda fase COP16

Cliente	Alcaldía
Usuario	<ul style="list-style-type: none"> <li>● Organizador Voluntario</li> <li>● Administrador de la aplicación</li> </ul>
Requerimientos funcionales	R0: Almacenar la información de lugares biodiversos R1: Mostrar los lugares registrados R2: Mostrar el departamento con más lugares biodiversos registrados
Contexto del problema	<p>Gracias al discurso ambiental que tiene la COP16, se hace obligatorio la divulgación de lugares biodiversos los cuales debemos conocer para cuidarlos de la mejor manera.</p> <p>Para almacenar los lugares biodiversos, se necesita la información básica del lugar: Nombre del lugar, nombre del departamento en el que está ubicado y el área del lugar en cantidad de kilómetros cuadrados.</p>
Requerimientos no funcionales	<ul style="list-style-type: none"> <li>- Mostrar la información de manera ordenada y fácil de entender</li> <li>- Interfaz intuitiva</li> <li>- Facilidad para digitar la información de los lugares biodiversos</li> <li>- Personalización del programa según la COP16</li> <li>- Se necesita que el programa sea capaz de almacenar hasta 30 lugares biodiversos.</li> <li>- Con el propósito de garantizar una buena visualización de la información, se necesita que el programa sea capaz de organizar</li> </ul>

	los lugares biodiversos de menor a mayor área en kilómetros cuadrados.  - La aplicación debe estar lista antes del 19 de octubre
Requerimientos de proceso	Realizarlo en el lenguaje de programación Java

Nombre o identificador	R0: Almacenar la información de lugares biodiversos		
Resumen	El sistema debe permitir que el organizador voluntario o un administrador de la aplicación, digite la información de un lugar biodiverso, incluyendo nombre del lugar, departamento en el que se encuentra y área del lugar en kilómetros cuadrados		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	Nombre del lugar	String	Nombre único
	Nombre del departamento	String	Solo los departamentos del Valle, Chocó, Cauca y Nariño
	Área del lugar	Double	Números mayores a 0
Resultado o Postcondición	Se actualiza la información del lugar biodiverso en la base de datos, para su posterior organización. Se muestra un mensaje de error si no se cumplen las condiciones de la entrada o uno de confirmación si se cumplen		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	Mensaje de confirmación	String	“¡Lugar registrado con éxito!”
	mensaje de error	String	“Error. Los datos ingresados no son válidos”

Nombre o identificador	R1: Mostrar los lugares que se registraron, ordenados de menor a mayor área
------------------------	---

Resumen	El programa debe ser capaz de mostrar los lugares de menor a mayor área que están guardados en la base de datos.		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	N/A	N/A	N/A
Resultado o Postcondición	Se muestran los lugares biodiversos registrados de menor a mayor área, en kilómetros cuadrados.		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>
	Lugar de menor área en kilómetros cuadrados	String	“[Nombre del lugar biodiverso]”
	...	String	“[Nombre del lugar biodiverso]”
	Lugar de mayor área en kilómetros cuadrados	String	“[Nombre del lugar biodiverso]”

Nombre o identificador	R2: Mostrar el departamento con más lugares biodiversos registrados		
Resumen	Para facilitar la visualización de la información, el programa deberá mostrar el departamento con más lugares biodiversos registrados.		
Entradas	<b>Nombre entrada</b>	<b>Tipo de dato</b>	<b>Condición valores válidos</b>
	N/A	N/A	N/A
Resultado o Postcondición	Se muestra en pantalla el nombre del departamento con más lugares biodiversos registrados por el usuario y cuántos tiene.		
Salidas	<b>Nombre salida</b>	<b>Tipo de dato</b>	<b>Formato</b>

	Nombre de departamento	String	“El departamento con más lugares biodiversos es [nombre], con X lugares biodiversos”
--	------------------------	--------	--