

Criterio	EXCELENTE (4 Puntos)	ADECUADO (3 Puntos)	MINIMO (2 Puntos)	INSUFICIENTE (1 Punto)
<b>Funcionamiento</b>	No tiene errores aparentes Funcionamiento eficiente con gran volumen de información y muy buen control de errores y excepciones. Muy Alto grado de autonomía, interface intuitiva y documentación adecuada	No tiene errores aparentes Funcionamiento y controla errores y excepciones. No es facil entender su funcionamiento intuitivamente se requiere de documentacion o manuales , se incluye escenarios básico en la documentación	Existe errores en 1 o 2 opciones, Funcionamiento y se realiza el control de errores y excepciones. Se requiere de documentacion o manuales con minima informacion para realizar su funcionamiento, si detalles de otros escenarios excepto el escenario basico.	Contiene errores que no hace que funcione el programa
<b>Estructuras y Herramientas</b>	Buen Nivel de abstracción Buena organización del programa en librerías. Uso adecuado de estructuras del lenguaje que mejoran reutilización y organización del código (metodos, funciones, namespace, clases, enums, typedefs). Uso adecuado de los paradigmas, POO, Genérico y Concurrente. Uso adecuado de la librería Estándar de C++.	No tiene un buen nivel de abstraccion, se repite codigo o las estructuras son muy basicas Hace uso de al menos 2 paradigmas de programación: POO, Generica y Concurrente Hace uso de libreria estandar significativamente	No tiene un buen nivel de abstraccion, se repite codigo o las estructuras son muy basicas Hace uso de al menos 1 paradigmas de programación: POO, Generica y Concurrente Hace uso de libreria estandar muy ligeramente	Codigo muy repetitivo, escaso uso de estructuras, uso de 1 solo paradigma, uso muy escaso de librerías
<b>Algoritmo</b>	Se consideraron criterios de eficiencia de tiempo y espacio en el uso de los algoritmos Se utilizó algoritmos existentes de la librería estándar de forma intensiva y de forma adecuada Se utiliza algoritmos creativos de complejidades mayores a lineales (nlog) para gran volumen de información, algoritmos de busqueda, recorrido, ordenamiento o que exigen un buen analisis	No se consideraron criterios de eficiencia de tiempo y espacio en el uso de los algoritmos o Se utilizó algoritmos existentes de la librería estándar de forma limitada Se utiliza al menos 2 algoritmos creativos de complejidad mayor a lineal (nlog), algoritmos de busqueda, recorrido, ordenamiento o que exigen un buen analisis	No se consideraron criterios de eficiencia de tiempo o espacio en el uso de los algoritmos o Se utilizó algoritmos existentes de la librería estándar solo en 1 o 2 casos. Se utiliza al menos 1 algoritmo creativo de complejidad mayor a lineal (nlog), algoritmos de busqueda, recorrido, ordenamiento o que exigen un buen analisis	No se utiliza criterios de eficiencia, tiempo, o espacio Algoritmos muy simples y de muy baja exigencia
<b>Diseño de Software</b>	Se utiliza al menos 5 patrones de Diseño y principios de abstraction, separacion de intereses y buen acoplamente y cohesion.	Se utiliza al menos 4 patrones de Diseño y principios de abstraction, separacion de intereses y buen acoplamente y cohesion.	Se utiliza al menos 3 patrones de Diseño y principios de abstraction, separacion de intereses y buen acoplamente y cohesion.	Se utiliza al menos 2 patrones de Diseño y principios de abstraction, separacion de intereses y buen acoplamente y cohesion.
<b>Presentación y Documentación</b>	Se utilizó Github para el control de versiones, el repositorio incluye una documentación adecuada con un README.md bien elaborado con al menos 4 fuentes bibliograficas relevantes. Se observa participación de todos los miembros del team en la actualización del repositorio, buena organizacion del codigo fuente y de los recursos requeridos. Se elaboró una presentación de impacto, buen sonido, amena, bien estructurada con temas claros, con las características más sobresalientes de su implementación.Los miembros del equipo pueden explicarse claramente el funcionamiento del programa.	Se utilizó Github para el control de versiones, el repositorio incluye una documentación adecuada con un README.md bien elaborado con al menos 3 fuentes bibliograficas relevantes. Se observa participación de todos los miembros del team en la actualización del repositorio, se observa cierta organizacion y estructura en los archivos y documentos. Se elaboró una presentación, buen sonido, amena, bien estructurada con temas claros, con las características más sobresalientes de su implementación. Los miembros del equipo pueden explicarse claramente el funcionamiento del programa.	Se utilizó Github para el control de versiones, el repositorio incluye una documentación adecuada con un README.md bien elaborado con al menos 2 fuentes bibliograficas relevantes. Se observa participación de todos los miembros del team en la actualización del repositorio, se observa cierta organizacion y estructura en los archivos y documentos. Se elaboró una presentación, estructurada, con las características más sobresalientes de su implementación. Los miembros del equipo pueden explicarse claramente el funcionamiento del programa.	Se utilizó Github para el control de versiones, el repositorio incluye una documentación adecuada con un README.md bien elaborado con al menos 1 fuentes bibliograficas relevantes. Se observa participación de todos los miembros del team en la actualización del repositorio. Se elaboró una presentación, con las características más sobresalientes de su implementación. Los miembros del equipo pueden explicarse claramente el funcionamiento del programa.