

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

CÁLCULO DEL ÍNDICE DEL CALIDAD DEL AGUA CON PYTHON EN CUERPOS DE AGUA SUPERFICIAL LOTICOS, IMPACTADOS POR ACTIVIDADES HUMANAS.

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN AGUA Y SANEAMIENTO AMBIENTAL**

LISETH CAROLINA COLLAGUAZO CARRERA

DIRECTOR: DRA.MORALES CASA VERÓNICA ELIZABETH

DMQ, marzo 2023

CERTIFICACIONES

Yo, Liseth Carolina Collaguazo Carrera declaro que el trabajo de integración curricular aquí descrito es de mi Elaboración; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

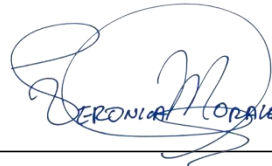


Liseth Collaguazo

liseth.collaguazo@epn.edu.ec

lisethcarolina2112@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Liseth Carolina Collaguazo Carrera, bajo mi supervisión.



DRA. MORALES CASA VERÓNICA ELIZABETH
DIRECTOR

veronica.morales@epn.edu.ec

DECLARACIÓN DE ELABORACIÓN

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Liseth Carolina Collaguazo Carrera

AGRADECIMIENTO

En primer lugar, doy gracias a Dios por encaminar mis pasos en la vida, y por permitirme tener tan buena experiencia dentro de mi universidad, gracias a mi universidad por permitirme formarme como un profesional, gracias a cada maestro que hizo parte de este proceso integral de formación.

Mi profundo agradecimiento a mis padres, quienes siempre han estado a mi lado alentándome a seguir en cada uno de mis sueños y días, con sus palabras alentadoras me ayudaron a seguir creciendo y formándome en muchos aspectos.

Agradezco mucho por la ayuda a mi tutora, por haberme guiado en la elaboración este trabajo que ha sido una gran bendición en todo sentido.

Gracias a mis amigos que estuvieron presentes en esta etapa tan importante de mi vida, siempre ofreciéndome su apoyo en todo momento.

DEDICATORIA

Mi tesis la dedico con todo mi amor y cariño a mis padres Clemencia y Carlos, por su sacrificio y esfuerzo, por darme una carrera para el futuro y por creer en mi capacidad.

A mi hermana Katherin que siempre ha estado apoyándome con sus palabras de aliento que no me dejaban decaer para que siguiera adelante y siempre sea siempre perseverante y cumpla con mis ideales.

A mis amigos pasados, quienes sin esperar nada a cambio compartieron su conocimiento, alegrías y tristezas, y a todas aquellas personas que durante la carrera estuvieron a mi lado apoyándome.

Gracias a todos.

ÍNDICE DE CONTENIDO

1	DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1	Objetivo general.....	1
1.2	Objetivos específicos	1
1.3	Alcance.....	2
1.4	Marco teórico.....	2
2.1.1	Actividades humanas que afectan a la calidad de agua en Ecuador.....	2
2.1.2	Principales actividades económicas que contaminan cuerpos de agua	2
2.1.3	Cuerpos lóticos que presentan contaminación.....	3
2.1.4	Calidad de agua	3
2.1.5	¿Qué es un índice de calidad de agua?.....	4
2.1.6	ICAS que se han aplicado en América Latina.....	5
2.1.7	Pasos para determinación de ICA	5
2.1.8	Índice más utilizado para la determinación del ICA.....	5
2.1.9	Aplicación de curvas de función para la determinación del ICA.....	6
2.1.10	Métodos de linealización de curvas	7
2.1.11	Método Delphi	7
2.1.12	Aplicaciones de las herramientas de Python.....	8
3.	METODOLOGIA	8
3.1	Investigación bibliográfica de los ICA.....	10
3.2	Matriz de comparación entre métodos de cálculo ICA	10
3.2.1	Criterios que se emplearon para la selección del ICA a implementarse en la programación.....	10
3.3	Parámetros y cálculo del método ICA-NSF	13
3.4	Insumos para el diseño del programa en Python	16
3.4.1	Linealización de curvas de función (método ICA-SNF)	16

3.4.2	Selección del método matemático para la linealización de las curvas.....	20
3.5	Creación del código de programación para la determinación del ICA	21
3.5.1	Criterios para el diseño del programa	21
3.5.2	Formas de presentación	24
3.6	Validación	25
3.6.1	Validación número de parámetros	25
3.6.2	Validación del método HPI.....	25
4.	RESULTADOS Y DISCUSIÓN.....	26
4.1	Matriz bibliográfica de aplicación de ICAs para cuerpos de agua lóticos.....	26
4.2	Matriz de comparación entre métodos de cálculo del ICA.....	29
4.3	Método seleccionado: ICA-NSF	32
4.4	Diseño del programa en Python.....	33
4.4.1	Linealización de curvas de función	33
4.5	Selección de ecuaciones de determinación del Qi para la utilización en el código de programación.....	36
4.6	Creación del código de programación.....	37
4.6.1	Criterios de cálculo del ICA global	37
4.6.2	Forma de presentación visual del programa del cálculo del ICA-NSF.....	39
4.7	Validación	41
4.7.1	Validación del ICA-NSF con 9 parámetros de calidad del agua	41
4.7.2	Validación ICA-NSF con 7 parámetros de calidad del agua (número mínimo requerido por el programa en Python propuesto).....	43
4.7.3	Validación con ICA-HPI	44
4.8	Flexibilidad del programa	44
4.9	Manual para el usuario.....	45
5.	CONCLUSIONES Y RECOMENDACIONES	45
5.1	Conclusiones	45
5.2	Recomendaciones	46

6. REFERENCIAS BIBLIOGRÁFICAS	48
7. ANEXOS.....	52
ANEXO I	52
ANEXO II	54
ANEXO III	59
ANEXO IV.....	60
ANEXO V.....	104
ANEXO VI.....	104
ANEXO VII.....	105

ÍNDICE DE TABLAS

Tabla 1: Parámetros de calidad del agua	3
Tabla 2: Ventajas y limitaciones del ICA.....	4
Tabla 3: Código de colores.....	12
Tabla 4: Calificación cuantitativa	12
Tabla 5: Ejemplo de selección del método aplicado al ICA-NSF	13
Tabla 6: Criterios específicos para la determinación del Qi en el caso de que el parámetro de calidad del agua este fuera del rango de la gráfica de función	15
Tabla 7: Datos de la gráfica de la turbidez	17
Tabla 8: Intervalos de la gráfica de oxígeno disuelto	19
Tabla 9: Puntos de la gráfica de la turbidez.....	19
Tabla 10: Escala de clasificación del ICA-NSF	23
Tabla 11: Escala de clasificación del ICA-HPI.....	23
Tabla 12: Matriz bibliografica de aplicación de ICAs en cuerpos de agua lóticos	26
Tabla 13: Tabla de comparación de los métodos de cálculo de ICA aplicado para cuerpos de agua lóticos (criterios de selección en función de codificación de color y ponderación de importancia).....	29
Tabla 14: Tabla de comparación de los métodos de cálculo de ICA aplicado para cuerpos de agua lóticos (resultados de la aplicación de la metodología de selección)	31
Tabla 15: Porcentajes finales para la selección del método a utilizar	32
Tabla 16: Parámetros de calidad el agua y valores de los pesos específicos (W) del índice de calidad de agua NSF.	32
Tabla 17: Datos obtenidos en Excel de linealización de curvas utilizando líneas de tendencia.....	33
Tabla 18: Datos de las ecuaciones obtenidas en funciones a trozos.....	34
Tabla 19: Comparación de resultados para la determinación del Qi, utilizando los dos métodos de linealización propuestos, con datos del estudio en el Río Monjas-Sector de Pomasqui (mes de junio)	36
Tabla 20: Tabla de comparación de los porcentajes de error entre los métodos de la linealización propuestos en el presente estudio.....	37
Tabla 21: Tabla de ponderaciones y determinación de Qi para el HPI	38
Tabla 22: Escala de clasificación para el ICA HPI y ICA NSF	38
Tabla 23: Comparación del ICA-NSF con el calculado en el Río Sinincay.....	41

Tabla 24: Comparación del ICA-NSF en el Río Pomasqui	42
Tabla 25: Comparación del ICA-NSF con 7 parámetros de calidad del agua	43
Tabla 26: Comparación del índice HPI (Río San Juan-Lima).....	44

ÍNDICE DE ILUSTRACIONES

Figura 1: Diagrama de flujo metodológico para la elaboración del programa	8
Figura 2: Calculo del ICA con el método NSF	14
Figura 3: Gráfica de la turbidez	17
Figura 4: Gráfica de Turbidez en Excel	18
Figura 5: Diagrama de determinación de pendiente.....	18
Figura 6: Gráfica del oxígeno disuelto.....	19
Figura 7: Captura de la forma uno del programa.....	39
Figura 8: Presentación del programa en forma de menú	40
Figura 9: Pantalla dinámica de Python.....	40

RESUMEN

El presente trabajo tiene el objetivo de determinar el índice de calidad del agua ICA en función de nueve parámetros medidos y evaluados in situ mediante un código de programación en Python. El proyecto se enfoca en la obtención de la calidad del agua mediante el índice de calidad del agua (ICA) que inicia con la identificación y caracterización de los nueve parámetros necesarios (oxígeno disuelto, coliformes fecales, pH, DBO, nitratos, fosfato, variación de la temperatura, turbidez, sólidos disueltos). Posteriormente se centra en la realización del código de programación en donde se considera el método NSF (National Sanitation Foundation) y el HPI (índice de metales pesados) para la obtención de ICA global, con parámetros de calidad del agua medidos in-situ y/o laboratorio. Una vez calculado dicho valor se programa para que el pseudocódigo indique al usuario el estado del agua (excelente, buena, media, mala y muy mala), y recomendaciones de su uso según el resultado del ICA global obtenido (consumo humano, en la agricultura, pesca y vida acuática, uso industrial y uso recreativo). El programa fue validado mediante el estudio realizado en el Río Monjas con un error entre 1 al 5%, implicando por lo tanto confiabilidad para su utilización.

PALABRAS CLAVE: ICA-NSF, Python, HPI, metales pesados, Ecuador, validación, calidad del agua, calidad del agua

ABSTRACT

The objective of this work is to determine the water quality index (WQI) based on nine parameters measured and evaluated in situ using a Python programming code. The project focuses on obtaining the water quality through the water quality index (WQI) that starts with the identification and characterization of the nine necessary parameters (dissolved oxygen, fecal coliforms, pH, BOD, nitrates, phosphate, temperature variation, turbidity, dissolved solids). Subsequently, it is focused on the realization of the programming code where the NSF (National Sanitation Foundation) method and the HPI (heavy metal index) are considered to obtain the global AQI, with water quality parameters measured in-situ and/or in the laboratory. Once this value is calculated, the pseudocode is programmed to indicate to the user the state of the water (excellent, good, average, bad, poor and very poor), and recommendations for its use according to the result of the global AQI obtained (human consumption, in agriculture, fishing and aquatic life, industrial use and recreational use). The program was validated by means of the study carried out in the Monjas River with an error between 1 and 5%, thus implying reliability for its use.

KEYWORDS: ICA-NSF,Python,HPI,Qi,Wi,Heavy metals,Ecuador,error

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El estado de los recursos naturales del Ecuador demuestra la necesidad de una adecuada protección de la naturaleza. La falta de atención del país a este tema conduce a una mala selección de metodología y parámetros al determinar el ICA. Esto conduce a resultados inexactos que afirman falsamente la calidad del cuerpo de agua. Considerando que el Estado ecuatoriano debe considerar los efectos de sus acciones sobre los recursos naturales circundantes al crear nuevas leyes. Esto se debe a que los recursos naturales en los países en desarrollo, como el Ecuador, no reciben la debida protección por parte del Estado. Por lo tanto, al redactar nuevas leyes, se deben utilizar criterios de selección adecuados para cualquier método o parámetro utilizado en la promulgación. Esto se debe a que algunas métodos y parámetros podrían causar problemas ambientales con contaminación de metales, como usar una ley con índices de la presencia de metales en el agua como la ICA (Virela et.al,2020).

En el proyecto se pretende crear un programa en Python que se base en un método conocido para determinar el Índice de Calidad del Agua, como resultado se obtendrá un programa que determine la calidad del agua en base a los datos obtenidos en laboratorio o in-situ. Además, se desea incorporar al programa una función que permita determinar en función del Índice de Calidad del Agua (ICA) y lo relacione con el uso que se podría dar en función del valor obtenido. también se anhela adicionar una subclasificación en función del ICA los cuales estarán condicionados en base al uso que se destina el agua indicándose las medidas o límites aconsejables.

1.1 Objetivo general

Crear un programa en lenguaje Python para determinar el índice de calidad del agua (ICA) para aguas superficiales lóxicas, que permitan definir su grado de contaminación y su posible uso.

1.2 Objetivos específicos

1. Recolectar la información bibliográfica para seleccionar el método y parámetros más adecuados para determinar el ICA en aguas lóxicas.
2. Desarrollar un programa en lenguaje Python con el método y parámetros seleccionados de acuerdo con la disponibilidad de información bibliográfica.

3. Realizar la validación del programa con información secundaria.

1.3 Alcance

Para determinar el índice de calidad de agua (ICA) se utilizará un lenguaje de programación de Python donde su acceso es libre sin tener la necesidad de adquirir una licencia de paga. Para realizar la creación de esta herramienta tecnológica se revisará bibliografía la cual nos permitirá seleccionar los métodos y parámetros que son necesarios para determinar el grado de contaminación en cuerpos de aguas lóaticas que tuvieron actividades humanas. Luego de la creación de este programa se realizará la validación con estudios ya realizados para corroborar la información.

1.4 Marco teórico

2.1.1 Actividades humanas que afectan a la calidad de agua en Ecuador

La mayor parte de la contaminación de las aguas lóaticas se relacionan con las actividades, agrícolas, mineras y ganaderas que se desarrollan en las extensiones del territorio. Las mismas actividades que provocan la contaminación de las aguas superficiales que por medio de la escorrentía que fluye por la superficie, los contaminantes son arrastrados y disueltos. Las sustancias que principalmente se encuentran en las aguas son: los fertilizantes y de los fitosanitarios, materia orgánica y sustancias tóxicas que están ligadas a las actividades ganaderas y mineras (Fernández et al., 2005).

Un ejemplo de actividades que afectan a la calidad del agua en Ecuador es la actividad minera, por ejemplo en Zaruma y Portovelo, provocando impactos negativos como afección en la salud de las personas y contaminación ambiental. Estos impactos negativos se producen debido a que en la explotación a cielo abierto se levantan sedimentos los cuales pueden contener metales pesados. Un efecto visible de la contaminación por minería es el color del agua, el cuál se presenta por que los elementos contaminantes se distribuyeron a lo largo de la cuenca hídrica (Guerra & Zaldumbide, 2010).

2.1.2 Principales actividades económicas que contaminan cuerpos de agua

La contaminación del agua se da cuando se presenta cualquier tipo de cambio puede ser químico, físico o biológico en la que la calidad del agua se ve afectado y por ende causa daño en cualquier ser vivo que beba esa agua.(Guadarrama et al., 2016)

La contaminación hídrica que se presenta en los ríos, lagos, arroyos, lagunas, mares y océanos se dan principalmente por actividades antrópicas del hombre como, por ejemplo: ganadería, actividades agrícolas, actividades industriales y residenciales. La contaminación de los cuerpos de agua superficial se atribuye a la actividad agrícola porque en esta actividad se utiliza fertilizantes industriales y agroquímicos o pesticidas, los mismos que se depositan en los suelos. Otras actividades que producen contaminación hídrica son la minería, la actividad industrial por ejemplo las curtiembres, la actividad hidrocarburífera, las aguas residuales domésticas, etc. Mediante la escorrentía de la lluvia, estos residuos pueden ser arrastrados a los cuerpos de agua superficial aledaños. (Pérez et al., 2018).

La contaminación de agua no solo se da por actividades con influencia del hombre, sino que también se puede presentar por la escorrentía producida por la lluvia, donde los componentes inorgánicos de los suelos pueden ser arrastrados al cuerpo de agua incrementando las concentraciones de calcio, magnesio, sulfatos, nitratos, sodio y potasio (Pérez et al., 2018).

2.1.3 Cuerpos lóticos que presentan contaminación

Los cuerpos de agua lóticos o conocidos como aguas corrientes son masas de agua que se mueven en una misma dirección como por ejemplo los ríos, arroyos y manantiales (Gómez Cerezo, 2022.).

En la actualidad los cuerpos lóticos han sido contaminados por sustancias tóxicas o vertimiento sustancias por actividades productivas y económicas de las comunidades que se encuentran aledaños. Los cuerpos lóticos que presentan contaminación son: ríos, lagos, arroyos, lagunas, mares y océanos (Lina & Mora, 2016).

2.1.4 Calidad de agua

Según Warren, (1971) citado en Tejero, (2000) define la calidad del agua como cualquier cambio en el estado del agua, basado en datos científicos y juicio de expertos, sin ninguna influencia sobre su uso por el hombre o los organismos que la habitan, generalmente en detrimento.

a) Parámetros de calidad de agua

Según Arellano & Guzmán citado en Hinojoza, (2018) dice que existen los siguientes parámetros de calidad del agua:

Tabla 1: Parámetros de calidad del agua

Parámetros físicos	Parámetros químicos	Parámetros biológicos
Se relacionan con las propiedades físicas del agua y, por lo tanto, se definen por el tacto, la vista, el olfato y el gusto, que incluyen temperatura, color, olor, sabor, turbidez y materia en suspensión.	Están relacionados con el agua debido a su capacidad para disolver diferentes sustancias, incluidos metales, alcalinidad, compuestos orgánicos, sólidos disueltos totales, dureza, nutrientes y fluoruro.	En el agua vive una gran cantidad y variedad de especies biológicas que realizan sus propios ciclos de vida. Estas especies, como parte del agua, se consideran parámetros de calidad del agua porque su presencia o ausencia puede indicar el estado del cuerpo de agua y también servir como indicador de la presencia de contaminantes.

Fuente:(Hinojoza, 2018)

2.1.5 ¿Qué es un índice de calidad de agua?

El ICA (Índice de calidad del agua) es un procedimiento mediante el cual se puede evaluar la calidad de un recurso hídrico (Quiroz et al., 2017).

El ICA es una herramienta que se utiliza para la evaluación del recurso hídrico. Puede ser expresado como una expresión simple o compleja de algunos parámetros, que puede estar representado por un número, una categoría, una descripción verbal o un símbolo (Torres et al., 2009). El ICA es un número mediante el cual se expresa cual es la calidad del recurso hídrico Torres et al., (2009). A continuación se presenta las ventajas y desventajas de los ICA.

Tabla 2: Ventajas y limitaciones del ICA

Ventajas	Limitaciones
<ul style="list-style-type: none"> • Permite mostrar una variación espacial y temporal de la calidad del agua. • Es un método de fácil uso el cual utiliza los datos que son generados en el laboratorio. • Se utiliza para evaluar la calidad del agua sin importar que uso se le puede dar. • Es de fácil uso para los usuarios. • Permite identificar cuáles son las áreas problemáticas del recurso hídrico. • Ayudan a conocer para que usos se les puede dar al agua lo que permite mejorar la comunicación con el público y aumentar la conciencia sobre la contaminación del recurso hídrico. 	<ul style="list-style-type: none"> • Se muestran los datos en forma de resumen. • No se muestra una información completa de la calidad del agua. • No se puede evaluar los riesgos que tiene la fuente de agua evaluada. • No existe un índice de calidad universal sin embargo se utiliza de acuerdo a las condiciones ambientales o en la región que se aplica. • No se basa en una generalización conceptual universal.

Fuente: (Torres et al., 2009)

2.1.6 ICAS que se han aplicado en América Latina

Entre los ICAS más utilizados destaca el ICA propuesto por Brown et al. (1970), que es una versión revisada del ICA (WQI) desarrollado por la Fundación Nacional de Saneamiento (ICA-NSF), este índice se distribuye y utiliza ampliamente. Otra aplicación de este índice se evidenció en países de Sudamérica como por ejemplo en Colombia y en otros países de América latina (Quiroz et al., 2017).

b) Índices de calidad de agua que se aplican en Ecuador

Tras realizar una investigación bibliográfica como por ejemplo Coello et al., 2015; García-González et al., 2021; Gutiérrez, 2019; Hinojoza, 2018; Quiroz et al., 2017; Ureta Valdez et al., 2019; Vidal & Carreño Mendoza, 2018 se concluyó que el índice más aplicado en Ecuador es el ICA propuesto por la Fundación Nacional de Saneamiento de los Estados Unidos (NSF). Un ejemplo de la aplicación de este índice se dio en el río Portoviejo en la provincia de Manabí (Quiroz et al., 2017).

2.1.7 Pasos para determinación de ICA

Para establecer un ICA generalmente se aplican los siguientes pasos (Aguirre et al., 2016):

- Definir cuál es el propósito de la calidad
- Selección de parámetros que se consideran para determinar el ICA
- La ponderación de los parámetros seleccionados
- Rangos de concentración para cada parámetro seleccionado

2.1.8 Índice más utilizado para la determinación del ICA

c) El ICA – NSF

En base a lo analizado se determinó que la aplicación del ICA “Water Quality Index” (WQI) es el más viable para aplicarlo en la determinación del ICA mediante un programa. Este método es utilizado por ser multiparámetro el mismo que para ser determinado fue analizado en base a tres estudios en los cuales se estudiaron los diferentes parámetros, en el tercer estudio que se realizó se desarrolló una curva de nivel de valoración para cada una de las variables las cuales tenían un rango de 0 a 100 los mismo que se localizaron en el eje horizontal el valor y en el eje de las verticales los niveles de cada variable (Pérez Alvarado, 2017).

Para determinar las relaciones funcionales o curvas de función para cada contaminante se realizó un promedio de estos.

El índice de calidad WQI o también conocido como (NSF) se utilizó para medir los cambios en la calidad del agua de una determinada fuente o cuerpo de aguas en el mismo que intervienen nueve parámetros (Chávez et al., 2016).

Para determinar el índice WQI se utiliza las ponderaciones de Brown (1970), el mismo que tiene dos formas las cuales son:

En la primera forma establece que se puede determinar el índice de calidad aditivo o conocido como el *WQIa* como se muestra en la ecuación 1, el cuál es la suma lineal de factores de escala de la variable los cuales estarán multiplicándose por su ponderación (Chávez et al., 2016).

$$WQIa = \sum_{i=1}^9 Qi * Wi \quad \text{Ecuación 1}$$

Donde

Qi = es el factor de importancia o ponderación de la variable

Wi = es el factor de escala de la variable, depende de la magnitud de la variable (Méndez et al., 2020)

La otra forma es el índice de calidad multiplicativo, *WQIm*, el cual es el producto lineal multiplicativo del índice de calidad de cada parámetro elevado a la potencia de su ponderación como se muestra en la ecuación 2.

$$WQIa = \sum_{i=1}^9 Qi^{Wi} \quad \text{Ecuación 2}$$

Donde

Qi = Sub índice de i-ésimo parámetro

Wi = peso o porcentaje asignado al i-ésimo (Torres et al., 2022.)

2.1.9 Aplicación de curvas de función para la determinación del ICA

Las curvas de función es la que representa una escala de evaluación de cada parámetro. La principal función de las curvas es transformar los valores medidos a una escala adimensional en diferentes unidades como por ejemplo mgL-1, porcentaje, unidades, etc (Quiroz et al., 2017).

Conocidas también como las curvas funcionales depende del conocimiento técnico de los mecanismos que se conoce sobre los valores de las variables, además depende de la

hidrogeología de los cauces, de todas las actividades económicas que se dan en la capacidad operativa (Pérez et al., 2018).

2.1.10 Métodos de linealización de curvas

d) Excel para obtener curvas

La regresión lineal es el nombre que se le da a las líneas que atraviesan la nube de puntos y se ajustan mejor a ellas. Suponiendo que medimos la distancia vertical de cada punto a una línea recta. La línea de tendencia buscada será aquella para la cual la suma de estas distancias sea menor. (Sánchez, 2022).

e) Método matemático con funciones a trozos

Las funciones a trozos se definen mediante un pequeño tramo o también conocido como intervalo de su dominio en el cual puede venir expresados en dos o más expresiones (Martínez, 2022). La importancia de las funciones definidas a trozos se da porque estas funciones son capaces de describir cualquier variación o cambio de una cantidad los cuales pueden ser cambios de manera rápida o lenta. Un claro ejemplo de la aplicación de estas funciones es en la rama de la economía donde se presenta la función de producción en el cual existe la relación entre el producto obtenido y la combinación de los factores (Universidad Europea de Madrid, 2022).

2.1.11 Método Delphi

El método Delphi es un método de estructuración que nace de un proceso de comunicación grupal donde se selecciona al grupo de expertos de un determinado tema, para realizar este proceso las estimaciones que se realizan los expertos lo hacen dentro de una conversación en la cual se efectúa rondas sucesivas de acuerdo con el objeto o tema que se va a tratar (Astigarraga, 2022.). Además, este método de recopilación de opiniones representativas de un grupo es crucial para esta investigación. Sin embargo, no se puede tomar decisiones sin suficiente información. Ahí es donde la investigación cualitativa es útil. Es un enfoque específico para la recopilación de datos que, idealmente, no depende del análisis estadístico (Reguant-Álvarez & Torrado-Fonseca, 2016).

Por otro lado, este método Delphi tiene una capacidad de predicción que se basa en el juicio intuitivo el mismo que es emitido por el grupo de expertos que participan.

Para poder aplicar el método anteriormente mencionado se debe seguir los siguientes pasos:

- Formulación del problema
- Elección de expertos
- Elaboración y lanzamiento de cuestionarios de los temas a tratar
- Desarrollo práctico y explotación de resultados

2.1.12 Aplicaciones de las herramientas de Python

El programa llamado Python es un programa de acceso fácil y posee licencia de código abierto. Actualmente es uno de los lenguajes de programación con mayor difusión y popularidad, y puede ser utilizado para cualquier fin, así como distribuirse y modificarse. Python tiene las facilidades imperativas y funcionales que está basada en el lenguaje ABC (Challenger-Pérez et al., 2014).

f) Aplicación al área ambiental

Python ha sido un programa muy empleado en el área ambiental debido a que es una herramienta que se utiliza para el análisis temporal y espacial de una variable ambiental y la interacción que va a tener con el medio ambiente, además se utiliza por los sistemas de investigación geográfica.

g) Análisis de la calidad de agua con Python

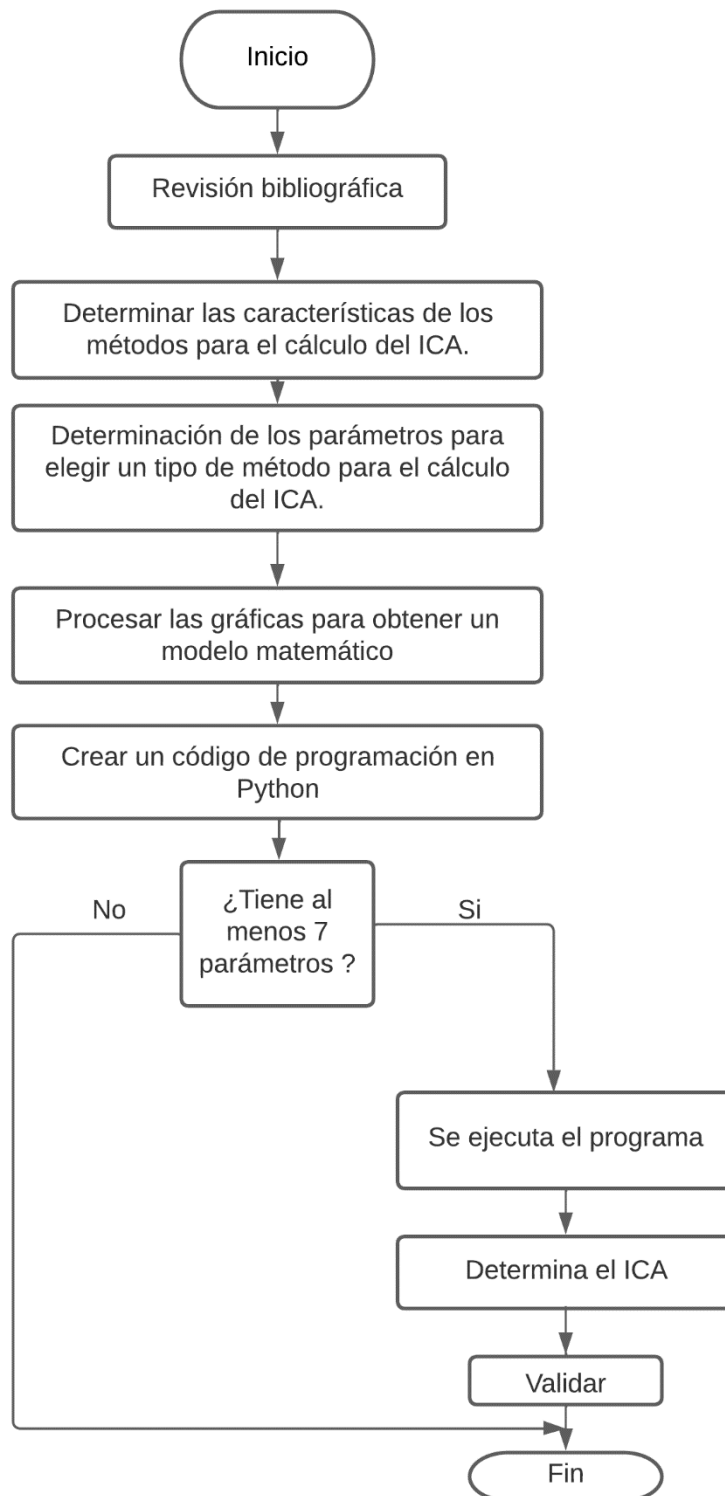
Un ejemplo de la aplicación de Python es la aplicación del programa en la determinación de caudal ecológico en el río Tulumayo en Huancayo (Perú) mediante este programa se pueden automatizar los resultados que pueden ser replicas a nivel nacional (Ortega, 2020).

3. METODOLOGIA

En este capítulo se va a detallar la metodología que se aplicó para crear un programa en Python que nos permita determinar el ICA para cuerpos lóticos a partir de las gráficas que se utilizan en el método NSF (National Sanitation Foundation).

La figura 1 se muestra el diagrama de flujo que se utilizó para el desarrollo de presente trabajo, en el cual se detalla cada etapa utilizada en la metodología.

Figura 1: Diagrama de flujo metodológico para la elaboración del programa



Fuente: Elaboración propia

Para la selección del procedimiento de cálculo del ICA se realizó mediante el método Delphi.

Este capítulo se presenta la estructura del código de programación que se utilizó para crear el programa en Python.

3.1 Investigación bibliográfica de los ICA

Los criterios para la conformación de la matriz bibliográfica de los ICAs fueron:

- Se realizó una revisión bibliográfica de los índices que fueron aplicados para la determinación del ICA en cuerpos lóticos.
- Se identificó los ICA que tienen mayor aplicación a nivel mundial, con su respectiva metodología.
- Se registró los parámetros utilizados en la determinación del ICA.
- Se evaluó si los índices utilizan ecuaciones matemáticas o curvas de función.

3.2 Matriz de comparación entre métodos de cálculo ICA

3.2.1 Criterios que se emplearon para la selección del ICA a implementarse en la programación.

Según Quiñones-Huatangari et al. (2020) para establecer una comparación entre los diferentes tipos de índices se debe seguir los siguientes pasos:

1. Determinar cuáles son las variables para evaluar una semejanza entre los índices analizados.
2. Seleccionar cuales son los índices que tienen mayor semejanza entre sí.
3. Determinar que escalas de obtención del índice (por ejemplo, ecuaciones, gráficas de función o tablas).
4. Tomar en cuenta si el ICA fue aplicado en países de América Latina y/o en el Ecuador, y que sea para la aplicación de cuerpos lóticos con afectación de actividades humanas.

Los pasos para seleccionar el método que se aplicó en este estudio fueron:

a) PASO 1: Determinar los criterios de comparación

En el siguiente trabajo se tomaron en cuenta los siguientes criterios de comparación, para la consideración:

- El número de parámetros utilizados es igual o mayor a 7.
- El índice se aplicó en América Latina.
- Si el ICA se aplicó en Ecuador (en el presente estudio con fines de validación).

- El método se aplicó en aguas lóxicas.
- El método de cálculo del índice de ICA tiene curvas de función de referencia.
- Los índices deben incluir al menos tres de los parámetros mencionados a continuación: OD (Oxígeno Disuelto), DBO (Demanda Bioquímica de Oxígeno), nitrógeno amoniacal y nitrógeno en forma de nitrato, fosfato, potencial de hidrógeno (pH) y sólidos disueltos totales (STD).
- El índice tiene dentro de sus parámetros metales y sustancias tóxicas las cuales están presentes en impacto humano.
- Dentro de sus parámetros tiene la temperatura.

Los criterios enumerados se fundamentan en lo mencionado por Digesa, (2022), que indica que la temperatura es un parámetro físico muy importante en el estudio de la calidad del agua debido a la influencia que tiene en las reacciones químicas. Además, este parámetro es un indicador de medidas secundarias, como el pH, la resistividad eléctrica y otros parámetros químicos; que se ven afectadas por variaciones de la temperatura.

De acuerdo con el sistema de gestión de calidad NTP ISO/IEC 17025 (México) establece que para determinar el índice de calidad de agua se deben medir los parámetros que por su naturaleza están cambiando como, por ejemplo: pH, Temperatura, Conductividad, Oxígeno Disuelto.

Además, en el cálculo del índice básico, los pesticidas y las sustancias tóxicas no son estudiadas. Sin embargo, en caso de que se presenten se puede evaluar, pero para ello se tomará en consideración la concentración en la que se presentan. Así en el caso de que la cantidad excediera de 0.1 mg/l, por ejemplo, el ICA HPI O IC de metales automáticamente colocará un valor de 0, la misma consideración se toma en cuenta en caso de las sustancias tóxicas esto fue establecido por el Registro federal Volumen 63 de México, número 237 (diciembre 10 de 1998).

b) PASO 2: Codificación por colores

El código de colores permite clasificar los criterios de acuerdo con su importancia, por ejemplo:

- El color rojo indica que la información tiene gran importancia o es valiosa (Gómez Rocío, 2016).
- El color verde se utiliza para indicar que la información es importante o es una información aceptada (Gómez Rocío, 2016).

- El color amarillo significa que la información puede ser no tan importante (Gómez Rocío, 2016).

En la siguiente tabla se muestra cuáles fueron los colores que se tomaron en cuenta en base a la teoría de colores para la calificación de los criterios los mismos que ayudaron a la selección del método que se usó. Adicionalmente a cada color se le asignó un valor de importancia de acuerdo con la relevancia que tenga dicha información.

Tabla 3: Código de colores

Código de colores	Valores
ROJOS	0.17
VERDES	0.1
AMARILLOS	0.06

Fuente: Elaboración propia

h) PASO 3: Ponderación de importancia

Para determinar el grado de importancia se utilizó el método Delphi antes mencionado en la sección 1.4.11 del marco teórico.

En la siguiente tabla se muestra cuáles fueron los criterios que se tomaron en cuenta para seleccionar el método en base a una calificación cuantitativa, en donde la máxima calificación fue 3 y la mínima fue 0.

Tabla 4: Calificación cuantitativa

Método de determinación del método	
Si cumple	3
Si cumple al 50% el parámetro	2
Si cumple al 25% con el parámetro	1
Si no cumple con el parámetro	0

Fuente: Elaboración propia

A continuación, se presenta un ejemplo de la aplicación de los pasos mencionados anteriormente. En este ejemplo primero se realizó una categorización por colores de acuerdo con la importancia que tendrá cada criterio, luego se aplicó una calificación la cual es el producto de la importancia del color y de la calificación cuantitativa que se aplicaba si cumplía el criterio.

Tabla 5: Ejemplo de selección del método aplicado al ICA-NSF

Criterios de selección	Índice de calidad de agua ICA - NSF
Si el número de parámetros es mayor a 7	0.51
El tipo de índice se aplica en América Latina	0.3
El tipo de índice se aplicó en Ecuador	0.3
El tipo de método se aplica en aguas lóxicas	0.51
El tipo de índice tiene curvas de función de referencia	0.18
Los índices incluyen al menos tres de los siguientes parámetros: OD (Oxígeno Disuelto), DBO (Demanda Bioquímica de Oxígeno), nitrógeno amoniacal y nitrógeno en forma de nitrato (N-NH ₄ ⁺ y N-NO ₃ ⁻), fósforo en forma de ortofosfato (P-PO ₄ ³⁻), potencial de hidrógeno (pH) y sólidos disueltos totales (STD).	0.18
El índice tiene dentro de sus parámetros metales y sustancias tóxicas las cuales están presentes en impacto humano	0.34
Dentro de sus parámetros tiene la temperatura	0.51

Fuente: Elaboración propia

En la sección 3.2 de resultados, se presenta la matriz completa en donde se encuentran los otros métodos de cálculo del ICA comparados.

Luego de aplicar este método, se calculó el porcentaje a cada uno de los métodos que fueron evaluados, el cual permitirá determinar cuál es el índice de calidad del agua que se aplicó para la elaboración de este trabajo.

3.3 Parámetros y cálculo del método ICA-NSF

Este apartado fue desarrollado en función de los resultados obtenidos en el punto 2.2 (selección el ICA).

a) Parámetros que usa el ICA-NSF

El método de NSF para la determinación de la calidad del agua, ha ido evolucionando en los últimos años, inicialmente el método utilizó 18 parámetros; y según las necesidades de evaluación del cuerpo de agua y recursos disponibles para el monitoreo de calidad del agua se ha llegado a utilizar incluso xx parámetros. En el presente estudio se trabajó con el criterio de manejar un número prudente de parámetros de calidad del agua que son los más comunes y fáciles de obtener sin incurrir en costos importantes en el monitoreo. Debido a que existen Gobiernos Autónomos Descentralizados que no poseen suficientes recursos para llevar campañas de monitoreos con gran cantidad de parámetros.

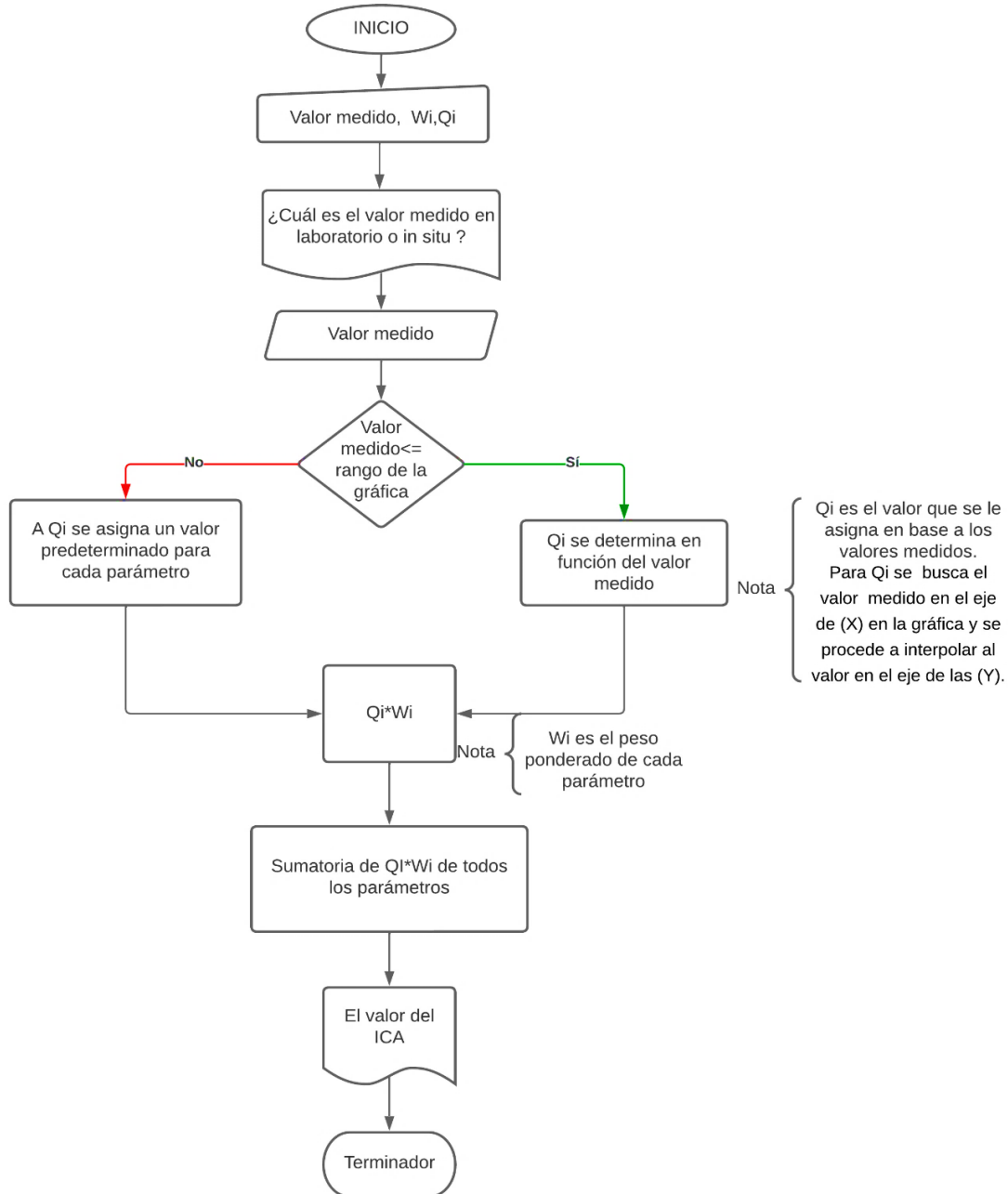
Por lo tanto, los parámetros con los que se trabajó en el diseño del programa fueron 9:fosfato,nitrato,coliformes fecales,oxigeno disuelto, pH, OD, SDT,temperatura,turbidez.De

los cuáles, el programa tiene la flexibilidad de trabajar hasta con un mínimo de 7 parámetros.

b) Pasos para calcular el ICA

En la siguiente imagen se muestra cómo se calcula el ICA en base al método ICA-NSF. Como se explicó en los puntos anteriores, este método utiliza curvas o gráficas de función las cuales transforman el valor de los parámetros de calidad del agua medidas (in-situ o laboratorio) en valores adimensionales que permiten una relación entre ellos. Las curvas de función utilizadas en este estudio fueron las propuestas por Walter Snyder, Henry Vaughan y Nathan Sinai

Figura 2: Calculo del ICA con el método NSF



Fuente: Elaboración propia

Cada parámetro del ICA-NSF tiene un rango determinado en la gráfica de función, en caso de que la medida de dicho parámetro este fuera del rango se aplicó las siguientes consideraciones:

Tabla 6: Criterios específicos para la determinación del Qi en el caso de que el parámetro de calidad del agua este fuera del rango de la gráfica de función

Parámetro	Rango	Valor asignado
-----------	-------	----------------

Coliformes fecales	Mayores de 100,000 NMP/100 mL	Qi = 3.
pH	Menor o igual a 2 unidades o es mayor o igual a 10 unidades	Qi = 2
DBO5	Mayor de 30 mg/L	Qi = 2
Nitratos	100 mg/L	Qi=2
Fosfatos	Mayor de 10 mg/L	Qi= 5
Temperatura*	es mayor de 15°C	Qi=9
Turbidez	Mayor de 100 NTU	Qi =5
Sólidos disueltos Totales	Mayores de 500 mg/L	Qi=3
Saturación de OD	Mayor de 140%	(Qi) =47

* Para el parámetro de Temperatura (Qi) primero se debe calcular la diferencia entre la T° Ambiente y la T° del agua medida, y con el valor obtenido proceder al cálculo.

Fuente: Elaboración propia

3.4 Insumos para el diseño del programa en Python

3.4.1 Linealización de curvas de función (método ICA-SNF)

Para determinar las ecuaciones de cada una de las gráficas se utilizó dos métodos matemáticos, y se calculó el error de cada uno para identificar cuál método es el más preciso:

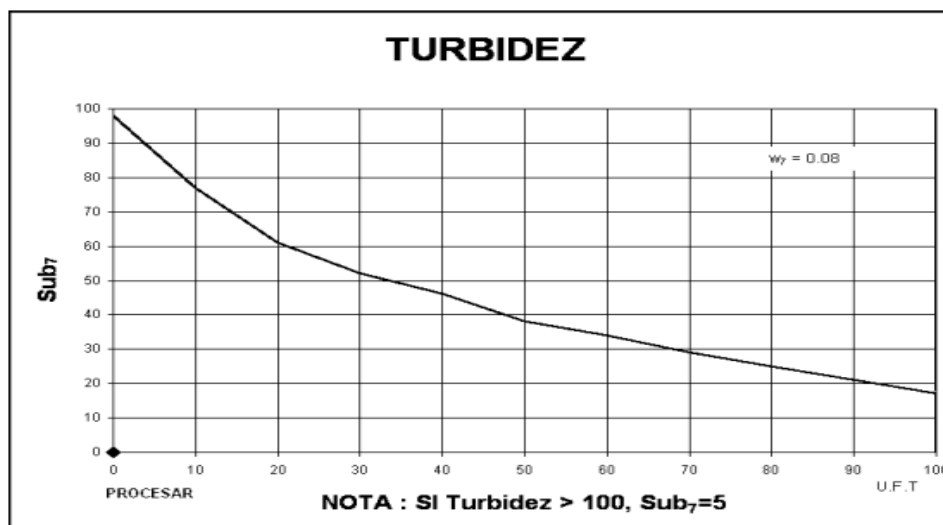
a) Línea de tendencia

En este estudio se aplicó el método matemático de líneas de tendencia en las gráficas de las curvas de función de cada parámetro del ICA-NSF.

Las líneas de tendencia en Excel son una herramienta de análisis que ofrece este programa, debido a que permite analizar datos. El uso de las líneas de tendencia se da principalmente para pronosticar el comportamiento futuro que tendrán los datos. Esta función muestra la ecuación en la gráfica en función de la tendencia que tenga por ejemplo lineal, polinómica o logarítmica. (Sánchez, 2022.)

Como ejemplo de cálculo, se utiliza la curva de función de la turbidez

Figura 3: Gráfica de la turbidez



Fuente:(Fernández et al., 2005)

A partir de la figura 3 se tiene la siguiente tabla con los puntos que forman la gráfica:

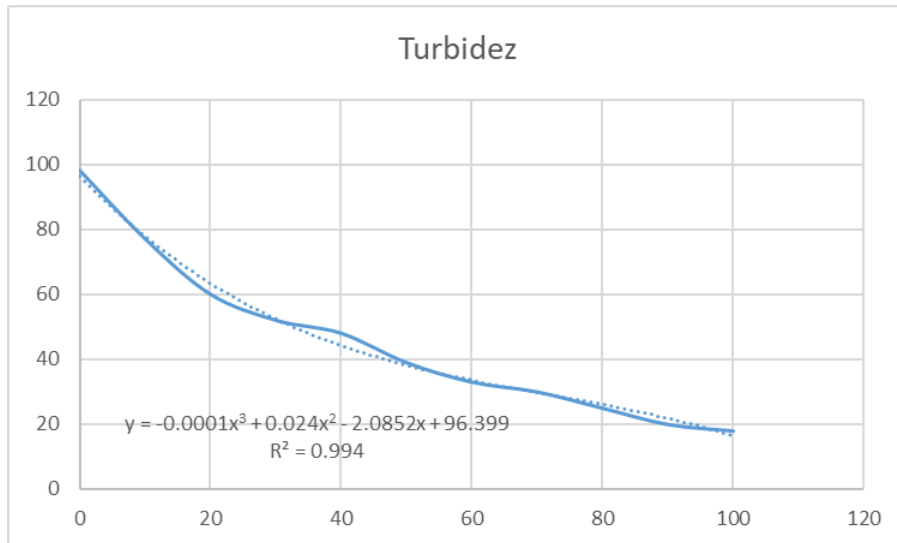
Tabla 7:Datos de la gráfica de la turbidez

x	y
0	98
10	77
20	60
30	52
40	48
50	39
60	33
70	30
80	25
90	20
100	18

Fuente: Elaboración propia

Con los valores obtenidos en la tabla 6 se graficó en Excel y se obtuvo la siguiente gráfica con su respectiva ecuación $y = -0.0001x^3 + 0.024x^2 - 2.0852x + 96.399$, en este caso se utilizó la función línea de tendencia polinómica de grado 3 ya que fue su mejor ajuste con $R^2 = 0.994$

Figura 4: Gráfica de Turbidez en Excel



Fuente: Elaboración propia

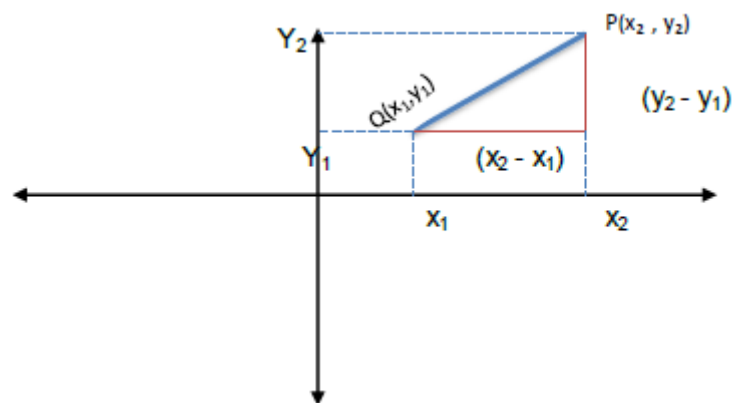
b) Ecuaciones a trozos

Para determinar las ecuaciones a trozos se utilizó el método punto-pendiente.

El método punto pendiente indica que una recta se define por su pendiente (m) y sus puntos de coordenadas (x₁, y₁). Se encuentra la ecuación X, Y que satisfaga las coordenadas (X, Y) de cualquier punto en el segmento de línea y que no satisfaga ningún otro número real. Matemáticamente, el cálculo de la ecuación se realiza de la siguiente manera

Si P(x, y) es cualquier punto en el plano xy:

Figura 5: Diagrama de determinación de pendiente



Fuente:(Muñoz, 2022.)

La pendiente de la recta que une P con el punto dado Q (x₁ y₁) es:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{Ecuación 3}$$

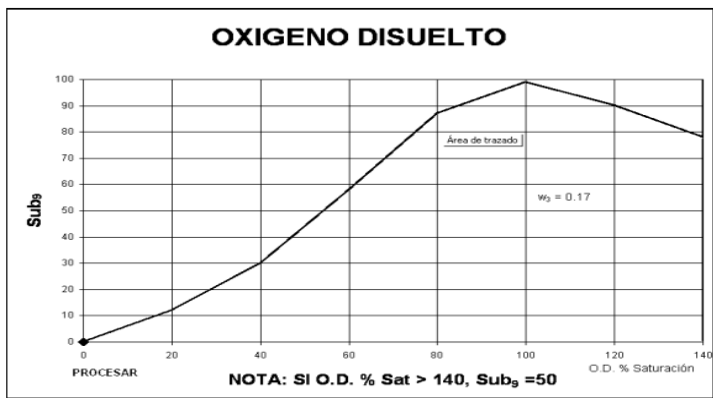
Y m es la pendiente, si $P(x, y)$ está sobre la recta específica, por lo tanto, tenemos que la ecuación de la recta es:

$$(y - y_1) = m(x - x_1) \quad \text{Ecuación 4}$$

Como ejemplo de cálculo, para la determinación de la ecuación de la curva de función de oxígeno disuelto se siguió los siguientes pasos:

1. A partir de la gráfica de función de la calidad de oxígeno (Figura 6) se debe dividir la gráfica en intervalos pequeños, como se muestra en la tabla 7.

Figura 6: Gráfica del oxígeno disuelto



Fuente: (Fernández et al., 2005)

Tabla 8: Intervalos de la gráfica de oxígeno disuelto

Oxígeno disuelto	
Número de intervalo	Intervalo
1	0 a 20
2	20 a 40
3	40 a 60
4	60 a 80
5	80 a 100
6	100 a 120
7	120 a 140

Fuente: Elaboración propia

2. A partir de los intervalos se tomó un intervalo en el cual se tomó un punto final y un punto inicial con lo cual se determinó la pendiente (m).

Tabla 9: Puntos de la gráfica de la turbidez

Intervalo	De 0 a 20 del % de saturación	
	x	y
Punto inicial	0	0
Punto final	20	10

Fuente: Elaboración propia

3. Luego se calculó la pendiente para lo cual se utilizó la ecuación (3).

$$m = \frac{10 - 0}{20 - 0}$$

$$m = \frac{1}{2}$$

4. Después de obtener la pendiente, se tomó un punto cualquiera con la única condición que el punto pertenezca a la gráfica. Para determinar la ecuación de la recta se utiliza la ecuación (4).

$$(y - 10) = \frac{1}{2}(x - 20)$$

$$y = \frac{1}{2}x - 10 + 10$$

$$y = \frac{1}{2}x$$

5. Finalmente se tiene que la función que satisface a la gráfica en el intervalo de 0-20 del % de saturación.

$$y = \frac{1}{2}x$$

Además, en las funciones a trozo se tiene que $y = f(x)$ va a representar el Qi el mismo que es peso individual de cada parámetro y (x) va a representar el valor medido del parámetro.

3.4.2 Selección del método matemático para la linealización de las curvas

Para la selección del método matemático que se utilizó para la linealización de curvas de función que se detallan en el Anexo 2 se tomó en cuenta la teoría de errores. La cuál se utiliza para cuantificar la imperfección del método seleccionado, esta se obtiene de la diferencia entre el valor medido de una magnitud y el valor referencia o conocido como el valor real (Centro Nacional de Metrología, 2017).

La veracidad de un dato se puede estimar por el medio del error relativo como se muestra en la ecuación 5, en base al error relativo se puede concluir que cuando menor sea el porcentaje de error relativo mayor será la veracidad del método (Ruis et al., 2010).

La ecuación para determinar el error relativo es:

$$\%error = \frac{valor\ medido - valor\ real}{valor\ real}$$

Ecuación 5

En base a lo mencionado anteriormente acerca del porcentaje de error para la selección del método matemático se aplicó el porcentaje relativo, en el cual se tomó como valores verdaderos los datos del estudio realizado en el río Monjas en el sector de Pomasqui por Gutiérrez, (2019). Y como los valores medidos se tomaron los valores que se obtuvieron aplicando las ecuaciones de los dos métodos que se ocuparon para linealizar las gráficas de función del método ICA-NSF. Después de aplicar la ecuación del error relativo se concluyó que el método con menor porcentaje de error es el método de funciones a trozos lo que indicó que es el método con mayor veracidad, que se explica en mayor detalle en los resultados obtenidos en la sección 3.5 de resultados.

3.5 Creación del código de programación para la determinación del ICA

3.5.1 Criterios para el diseño del programa

Los criterios que se tomó en cuenta para la creación del programa fueron los parámetros que utiliza el método ICA-NSF, y su respectiva metodología de cálculo que fue explicada en la sección 2.3 y 2.4.

Adicionalmente, se incorporó un índice de metales basado como menciona Chirinos (2022), donde indica que la determinación del Índice de Metales Pesados o conocido como HPI puede ser calculado comparando las concentraciones de los metales con el Estándar de Calidad Ambiental del Agua, en función del método explicado a continuación. En este estudio se tomó la normativa del TULSMA (Anexo VI, Tabla 1) ya que es la normativa de calidad del agua en Ecuador, y los metales considerados en el presente estudio fueron: aluminio total, cadmio total, cobre total, hierro total, manganeso total, plomo total, zinc total. La selección fue bajo la consideración de metales tóxicos para la salud humana y contaminación ambiental.

Según Abdullah (2013) mencionado en Chirinos (2022) menciona que para el cálculo del Índice de Contaminación por Metales Pesados se utiliza la ponderación de cada metal en la calidad del agua analizada. El cálculo de HPI comprende:

- a. El cálculo del coeficiente de ponderación del *i*-ésimo parámetro.
- b. El cálculo de la calificación de calidad para cada uno de los metales pesados.
- c. La suma de estos subíndices en el índice general.

Para obtener el coeficiente de ponderación del i-ésimo parámetro, se obtiene aplicando la ecuación 6 como se muestra a continuación

$$W_i = \frac{k}{S_i} \quad \text{Ecuación 6}$$

Donde:

W_i = coeficiente de ponderación (Chirinos, 2022)

S_i = norma recomendada para el i-ésimo parámetro (Chirinos, 2022)

k = constante de proporcionalidad (Chirinos, 2022)

Donde W_i es el coeficiente de ponderación y S_i es el valor permisible recomendado para el i-ésimo parámetro en el Texto Unificado de Legislación Secundaria del Ministerio del Ambiente (TULSMA), mientras que k es la constante de proporcionalidad. Además, la constante de proporcionalidad es la establecida en el método HPI. (Capparelli et al., 2021)

Para determinar la calificación del individual de cada parámetro se aplica la ecuación 7 como se muestra a continuación:

$$Q_i = 100 * \frac{V_i}{S_i} \quad \text{Ecuación 7}$$

Donde:

Q_i = subíndice i-ésimo parámetro

V_i = valor medido del parámetro

S_i = límites permisibles para el i-ésimo parámetro

Donde Q_i es el subíndice i-ésimo parámetro, V_i es el valor del parámetro o el promedio de los resultados de los análisis de agua realizados y S_i el límite estándar permisible para el i-ésimo parámetro o valor aceptado en el TULSMA. (Capparelli et al., 2021)

Para determinar el índice de metales pesados (HPI) se aplica la ecuación 8 como se muestra a continuación:

$$HPI = \frac{\sum_{i=1}^n W_i * Q_i}{\sum_{i=1}^n W_i} \quad \text{Ecuación 8}$$

Además, se propone que el usuario puede determinar en el programa el ICA-NSF y el HPI, y en este caso calcular el ICA global. Según García (2012) para determinar el ICA global se puede aplicar la ecuación 9, como se muestra a continuación.

$$ICA_{Global} = \frac{ICA_{NSF} + ICA_{HPI}}{2} \quad \text{Ecuación 9}$$

Donde

ICA_{NSF} = es el ICA obtenido con el método NSF

ICA_{HPI} = es el ICA obtenido de la presencia de metales en el cuerpo de agua

También se incorporó en el programa una función que permita al usuario verificar los usos del agua en diferentes campos, en función del ICA global determinado como se muestra en la Tabla 1 del Anexo 3. Para esto, se consideró que el ICA global puede ser calculado ya se únicamente con el ICA-SNF o el HPI, o con los dos; dependiendo de la disponibilidad de los parámetros de calidad del agua medidos.

Además, se utilizó las tablas 10 y 11 para poder unificar los criterios que se utilizaron para evaluar el resultado del ICA global donde si el ICA de metales o conocido como HPI es mayor a 100 % se consideró mala.

Tabla 10: Escala de clasificación del ICA-NSF

Calidad del agua	Valor del ICA
Excelente	91 – 100
Buena	71 – 90
Media	51 – 70
Mala	26 – 50
Muy Mala	0 – 25

Fuente: Elaboración propia

Tabla 11: Escala de clasificación del ICA-HPI

HPI	CALIDAD DEL AGUA
0- 25	Muy buena.
26- 50	Buena.
51 - 75	Pobre.
76 - 100	Muy pobre.

Fuente: (Chirinos, 2022)

Para la unificación de tablas se consideró que el sí ICA_HPI es 100% en la tabla de clasificación del ICA_NSF será cero lo que significa que el cuerpo hídrico evaluado será malo lo que significa que el agua necesita algún tipo de tratamiento. Esta consideración se adoptó porque la escala de calificación de la calidad del agua la misma que está definida dentro del rango de 1 a 100, si supera este límite se asigna el valor de cero como lo menciona en la norma emitida por el Registro Federal de México. Cuando la concentración de metales supera los límites permisibles se estima que el agua está en condiciones de contaminación importante con metales pesados.

3.5.2 Formas de presentación

Para la presentación visual del programa se trabajó con tres alternativas:

a) Opciones secuenciales

La primera forma con la que se comenzó a crear el código de programación fue que el programa preguntará de manera consecutiva al usuario el valor de los parámetros medidos, sin opción a que el usuario pueda elegir cual parámetro ingresar.

b) Opción menú

En esta forma de presentación del código se creó un menú con todos los parámetros que utiliza el método ICA-NSF para facilitar al usuario que pueda ingresar en el orden que desee los parámetros. Además, en esta forma de presentación del código permite que el usuario pueda modificar el valor del parámetro ingresado y también permite mostrar una lista con todos los valores ingresados. Después de elegir la acción se ejecuta la acción elegida por el usuario.

c) Opción con librería Tkinter

En esta forma de presentación del código se creó una ventana dinámica con la librería Tkinter de Python la cual permitió crear una interfaz gráfica dinámica al momento de la ejecución. Además, en esta forma de presentación se creó botones de control que permitan al usuario realizar acciones como, por ejemplo, calcular el ICA, guardar el cálculo o eliminar el cálculo de la base de datos.

Se escogió la mas adecuada para el usuario, en función de la familiaridad y facilidad para la utilización del usuario.

3.6 Validación

3.6.1 Validación número de parámetros

a) Con 9 parámetros

Para validar con 9 parámetros se utiliza la metodología que establece este método ICA-NSF, en el cual se debe obtener el valor Q_i de la curva, el mismo que se debe multiplicar por su factor de ponderación para obtener un subtotal, puntaje parcial o subíndice. Luego de tener el puntaje parcial de los 9 parámetros se debe sumar y ese el ICA (Universidad de Pamplona, 2022).

b) Con 7 parámetros

Para realizar la validación con 7 parámetros se tomó en cuenta lo que dice Universidad de Pamplona (2022) que se puede realizar unas variaciones en los pesos de ponderación donde se distribuye su peso entre las demás variables para luego hacerle un recálculo tomando en cuenta solo las variables que se ingresaron los valores.

c) Uso del agua en base al índice de calidad del agua

En cuanto al valor numérico del ICA global, si es consistente en el cálculo, este no representa la posibilidad de exceder la comparación. Teniendo en cuenta los estándares generales, a continuación, se muestran algunas pautas propuestas por el grupo de expertos Dinius (1987). Al combinar los valores de ICA, se determinaron seis rangos de estado de calidad del agua: (E) excelente; (A) Aceptable; (LC) Ligeramente contaminado; (C) contaminado; (FC) fuertemente contaminado y (EC) excesivamente contaminado. En base a esta clasificación y de acuerdo con el uso previsto del agua, se han desarrollado los siguientes criterios, indicando medidas recomendadas o restricciones (Anexo 3, Tabla 1). Cabe mencionar que estos criterios no deben ser considerados como dogmas, sino que deben ser analizados caso por caso.

3.6.2 Validación del método HPI

Para validar con el método HPI se utiliza la metodología que establece este método ICA-HPI, en el cual se debe obtener el valor Q_i , el mismo que se debe multiplicar por su factor de ponderación para obtener un subtotal, puntaje parcial o subíndice. Luego de tener el puntaje parcial de los 7 parámetros se debe sumar y ese el ICA HPI o ICA metales.

4. RESULTADOS Y DISCUSIÓN

4.1 Matriz bibliográfica de aplicación de ICAs para cuerpos de agua lóticos

La documentación bibliografía revisada fue en total 12 para la construcción de la matriz comparativa de ICAs, los cuales consistieron en estudios en diferentes países en el cálculo del ICA con diferentes métodos. Los criterios incorporados en la matriz son explicados en la sección 2.1.

Tras la revisión bibliográfica de los diferentes índices de calidad de agua se tiene la siguiente tabla:

Tabla 12: Matriz bibliografica de aplicación de ICAs en cuerpos de agua lóticos

Nombre del método	Descripción	Parámetros	Aplicación
Índice de calidad de agua ICA - NSF	El índice de Calidad de Agua "Water Quality Index" (WQI), fue desarrollado en 1970 por la National Sanitation Foundation (NSF) de Estados Unidos, por medio del uso de la técnica de investigación Delphi de la "Rand Corporation's" (Ball & Church, 1980). Esta técnica es utilizada comúnmente en paneles de expertos, que para la época fueron 142.	Oxígeno disuelto, Coliformes fecales, pH, DBO, Nitratos, Fosfato, Desviación de la temperatura, Turbidez, Sólidos disueltos	Ecuador, Cuenca, Portoviejo
El Índice de calidad del agua Dinius	Este método utiliza el método DELPHI de encuestas, donde como resultado realizó un Ica multiplicativo y con asignación de pesos específicos por parámetro.	Los Nitratos, los Fosfatos, el Cambio de Temperatura, el Oxígeno Disuelto, la Demanda Bioquímica de Oxígeno, la Turbidez, los Sólidos Disueltos Totales, el Potencial de Hidrógeno y los Coliformes Fecales.	México
El Índice de calidad de agua para Oregon	Este método utiliza el método DELPHI, en este índice se utiliza una transformación logarítmica para convertir los resultados de las variables en subíndices.	Temperatura, oxígeno disuelto (porcentaje de saturación y concentración), DQO, pH, Sólidos totales, Amonio y Nitratos, Fosfato total y Coliformes fecales	Perú
Índice RPI (River Physiochemical Index)	Este índice se basa en el Índice de Oregon para su desarrollo y determinó que era importante que Idaho considerara una sola ecuación de sólidos disueltos y no múltiples como se considera en el Índice de Oregon debido a sus diferencias geomorfológicas. Además, el índice se correlaciona con índices biológicos como el índice de peces (RFI), el índice de diatomeas (RDI) y el índice de macroinvertebrados (RMI), que muestran una alta correlación al medir los impactos humanos, especialmente los impactos en la agricultura y la silvicultura.	Temperatura, oxígeno disuelto (porcentaje de saturación y concentración), DQO, pH, Sólidos totales, Amonio y Nitratos, Fosfato total y Coliformes fecales	Sur de Idaho

El Índice de British Columbia (BCWQI)	El índice se puede aplicar a cualquier masa de agua que haya sido analizada para medir la calidad y es flexible en cuanto al tipo y número de parámetros analizados. La masa de agua a la que se aplicará el índice puede determinarse mediante uno o más puntos de muestreo. Si tiene suficientes datos, un punto le dará buenos resultados. El cálculo de este índice se basa en una combinación de 3 factores: El número de variables que no se ajustan a los objetivos de calidad de agua (Factor F1: Alcance) El número de veces que estos objetivos no se cumplen (Factor F2: Frecuencia) La cantidad por la cual los objetivos no se cumple (Factor F3: Amplitud)	Oxígeno disuelto, pH, fósforo total, nitrógeno total, Coliformes Fecales, arsénico, plomo, mercurio, 2-4 D y lindano	Provincia Santa fé
Índice De Calidad De Agua De Montoya	Este tipo de índice se aplica para medir sobre el impacto sobre los sistemas acuáticos. En este tipo de índice se utilizan 18 variables donde los pesos específicos están sujetos a cambiar de acuerdo a la naturaleza del cuerpo de agua, es decir, lago, bahía o río.	a) Cantidad de materia orgánica: determinadas por el porcentaje de saturación del Oxígeno Disuelto (OD) y la Demanda Bioquímica de Oxígeno (BDO ₅). b) Materia Bacteriológica presente: determinada por Coliformes totales (COT) y Coliformes fecales (COF). c) Características físicas: determinadas por el color (COL) y la Turbiedad (TUR). d) La materia orgánica es determinada por alcalinidad (ALC), dureza (DUR), Cloruros (CLO), conductividad específica (CON), concentración de iones Hidrogeno (pH), grasas y aceites (GyA), sólidos suspendidos (SST), sólidos disueltos (SDT), nutrientes: nitratos nitrógeno amoniacal, fosfatos y detergentes (SAAM).	Jalisco
Índice Dalmatia	En este tipo de índice se incluye 9 variables. En este tipo de índice se utiliza en los siguientes tipos de agua que son: aguas subterráneas y superficiales, aguas que no pueden ser utilizadas en su estado natural y aguas que no puede ser usadas.	DBO, OD, coliformes fecales, pH, Temperatura, Nitratos totales, Fósforo Total, Sólidos Totales y Turbidez.	Europa
Índice de contaminación por nutrientes	Este tipo de índice es una derivación de las curvas de calidad de las variables que constituyen el método. Además, este índice es	Amonio, Nitrógeno Total, nitritos más nitratos, fósforo total, ortofosfatos, pH, Clorofila a, conductividad y turbidez.	Holanda

Fuente: Elaboración propia

4.2 Matriz de comparación entre métodos de cálculo del ICA

Con el objetivo de seleccionar el método adecuado para el cálculo del ICA para el programa en Python, se realizó una matriz de comparación de diferentes métodos; con los diferentes criterios de selección explicados en la sección 2.2.1. En el cual se utilizó determinación de criterios de selección, codificación de colores de dichos criterios y finalmente su ponderación.

En la siguiente tabla se presenta como se aplicó las ponderaciones y codificaciones de color en base al método Delphi anteriormente mencionado.

Tabla 13: Tabla de comparación de los métodos de cálculo de ICA aplicado para cuerpos de agua lóticos (criterios de selección en función de codificación de color y ponderación de importancia)

Parámetros	Índice de calidad de agua ICA - NSF	El Índice de calidad del agua Dinius	El Índice de calidad de agua para Oregon	Índice RPI (River Physiochemical Index)	El Índice de British Columbia (BCWQI)	Índice De Calidad De Agua De Montoya	Índice Dalmatia	Índice de contaminación por nutrientes
Si el número de parámetros es mayor a 7	3	3	3	3	3	3	3	3
El tipo de índice se aplica en América Latina	3	3	3	0	0	0	0	0
El tipo de índice se aplicó en Ecuador	3	0	0	0	0	0	0	0
El tipo de método se aplica en aguas lóticas	3	3	3	3	3	3	3	3
El tipo de índice tiene curvas de función de referencia	3	2	0	1	1	1	1	3
Los índices incluyen al menos tres de los siguientes parámetros: OD (Oxígeno Disuelto), DBO (Demanda Bioquímica de Oxígeno), nitrógeno	3	3	3	3	3	3	3	3

amoniaco y nitrógeno en forma de nitrato (N-NH ₄ ⁺ y N-NO ₃ ⁻), fósforo en forma de ortofosfato (P-PO ₄ ³⁻), potencial de hidrógeno (pH) y sólidos disueltos totales (STD).								
El índice tiene dentro de sus parámetros metales y sustancias tóxicas las cuales están presentes en impacto humano	2	0	3	3	3	0	0	3
Dentro de sus parámetros tiene la temperatura	3	3	3	3	0	0	3	0

Fuente: Elaboración propia

En la tabla 14 se observa los resultados obtenidos tras aplicar la metodología planteada en la sección 2.2.1 para seleccionar el método que se aplicará en el programa en Phytón.

Tabla 14: Tabla de comparación de los métodos de cálculo de ICA aplicado para cuerpos de agua lóticos (resultados de la aplicación de la metodología de selección)

Parámetros	Índice de calidad de agua ICA - NSF	El Índice de calidad del agua Dinius	El Índice de calidad de agua para Oregon	Índice RPI (River Physiochemical Index)	El Índice de British Columbia (BCWQI)	Índice De Calidad De Agua De Montoya	Índice Dalmatia	Índice de contaminación por nutrientes
Si el número de parámetros es mayor a 7	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
El tipo de índice se aplica en América Latina	0.3	0.3	0.3	0	0	0	0	0
El tipo de índice se aplicó en Ecuador	0.3	0	0	0	0	0	0	0
El tipo de método se aplica en aguas lóticas	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
El tipo de índice tiene curvas de función de referencia	0.18	0.12	0	0.06	0.06	0.06	0.06	0.18
Los índices incluyen al menos tres de los siguientes parámetros: OD (Oxígeno Disuelto), DBO (Demanda Bioquímica de Oxígeno), nitrógeno amoniacal y nitrato, fosfato, potencial de hidrógeno (pH) y sólidos disueltos totales (STD).	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
El índice tiene dentro de sus parámetros metales y sustancias toxicas las cuales están presentes en impacto humano	0.34	0	0.51	0.51	0.51	0	0	0.51
Dentro de sus parámetros tiene la temperatura	0.51	0.51	0.51	0.51	0	0	0.51	0
TOTAL	2.83	2.13	2.52	2.28	1.77	1.26	1.77	1.89

Fuente: Elaboración propia

En la siguiente tabla se muestra cual es el método seleccionado tras la aplicación de una sumatoria del producto del valor de importancia con el valor de la calificación cuantitativa asignada a cada criterio.

Tabla 15: Porcentajes finales para la selección del método a utilizar

Índice de calidad de agua ICA - NSF	El Índice de calidad del agua Dinius	El Índice de calidad de agua para Oregon	Índice RPI (River Physiochemical Index)	El Índice de British Columbia (BCWQI)	Índice De Calidad De Agua De Montoya	Índice Dalmatia	Índice de contaminación por nutrientes
87.5	70.83	83.33	66.66	54.16	41.66	54.166	62.5

Fuente: Elaboración propia

El método que salió seleccionado en función de la calificación que se muestra en la Tabla 14 y 15 es el método ICA_NSF. Según García, (2012) dice que es el método más aplicado en la actualidad en los países de América y Europa es el ICA desarrollado por la Fundación de Saneamiento Nacional de los Estados Unidos (NSF) debido a que utiliza parámetros fisicoquímicos para su evaluación.

4.3 Método seleccionado: ICA-NSF

a) Los parámetros que usa el método NSF

En la siguiente tabla se muestra cuáles son los parámetros del método ICA-NSF seleccionados para el presente estudio y que ponderación de importancia tiene cada parámetro. Estas ponderaciones fueron establecidas bajo la metodología Delphi, para cada parámetro en función de lo propuesto originalmente por Brown, (1970).

Tabla 16: Parámetros de calidad el agua y valores de los pesos específicos (W) del índice de calidad de agua NSF.

Parámetro	Unidad	Valor de (W)
DBO	(DBO-mg/l)	0.1
SDT	(SDT-mg/l)	0.08
Turbidez	NTU	0.08
Coliformes fecales	(NMP/100 ml)	0.15
Nitratos	(NO ₃ -mg/l)	0.1

Fosfatos	(PO4-mg/l)	0.1
Temperatura	°C	0.1
PH	PH	0.12
OD	(OD-% Sat.)	0.17

Fuente: (BROWN, R.1970)

En el programa se establece el criterio de recalcu de pesos específicos en el caso de que sea menor de 9 parámetros de calidad del agua (hasta mínimo 7 parámetros). Se realiza a través de la distribución de los valores de ponderación entre los parámetros que si tienen medida. (Universidad de Pamplona,2022).

4.4 Diseño del programa en Python

4.4.1 Linealización de curvas de función

Hay que recordar que el objetivo de la linealización de las curvas de función es obtener el valor de Q_i para el cálculo del ICA-SNF.

a) Línea de tendencia

En la siguiente tabla se muestran los datos obtenidos en el programa Excel con la función de la línea de tendencia.

Tabla 17: Datos obtenidos en Excel de linealización de curvas utilizando líneas de tendencia

Parámetro calidad agua	Ecuación
DBO (mg/l)	$y = 94.795e^{-0.099x}$
STD (mg/l)	$y = -5E-09x^4 + 6E-06x^3 - 0.0022x^2 + 0.2111x + 80.755$
Turbidez (NTU)	$y = -0.0001x^3 + 0.024x^2 - 2.0852x + 96.399$
Coliformes fecales (NMP/100 ml) Rango (500-1000000)	$y = 289.1x^{-0.369}$
Coliformes fecales (NMP/100 ml) Rango (1-200)	$y = 103.49x^{-0.18}$
Nitratos (mg/l)	$y = 9059x^{-1.733}$
Fosfatos (mg/l)	$y = -0.0833x^3 + 2.0714x^2 - 17.988x + 61.286$
Temperatura (°C)	$y = 409.57x^{-1.023}$
pH (pH) Rango (0 – 7.5)	$y = -0.3813x^3 + 8.7437x^2 - 36.78x + 42.865$
pH (pH) Rango (7.5 – 12)	$y = 3.816x^2 - 96.345x + 605.84$
OD (mg/l) Rango (0-100)	$y = -0.0002x^3 + 0.0288x^2 - 0.0894x + 0.9127$

OD (mg/l) Rango (100 – 140)	$y = -0.525x + 152.67$
---------------------------------------	------------------------

Fuente: Elaboración propia

4.4.1.1 Ecuaciones a trozos

Para determinar las ecuaciones, se realizó por el método de dos puntos para encontrar la ecuación como se explica en la sección 2.4.1, para este caso se utilizó el punto inicial y el punto final lo cual nos da como resultado la siguiente tabla:

Tabla 18: Datos de las ecuaciones obtenidas en funciones a trozos.

Parámetros calidad agua	Intervalos	Ecuaciones
DBO (mg/l)	0 a 5	$y = -9x + 100$
	5 a 10	$y = -4.4x + 77$
	10 a 15	$y = -2.6x + 59$
	15 a 20	$y = -1.6x + 44$
	20 a 25	$y = -0.8x + 28$
	25 a 30	$y = -0.6x + 23$
Sólidos Disueltos Totales (mg/l)	0 a 50	$y = 0.16x + 80$
	50 a 100	$y = -0.04x + 90$
	100 a 300	$y = -0.13x + 99$
	300 a 500	$y = -0.15x + 105$
Turbidez (NTU)	0 a 10	$y = -2.1x + 98$
	10 a 20	$y = -1.7x + 94$
	20 a 40	$y = -0.6x + 72$
	40 a 50	$y = -0.9x + 84$
	50 a 70	$y = -0.45x + 61.5$
	70 a 100	$y = -0.4x + 58$
Coliformes fecales (NMP/100 ml)	1 a 2	$y = -9x + 108$
	2 a 20	$y = -14/9 x + 838/9$
	20 a 200	$y = -2/15x + 194/3$
	200 a 500	$y = -1/30x + 134/3$
	500 a 2000	$y = -0.006x + 31$
	2000 a 10000	$y = -0.001125x + 21.25$
	10000 a 20000	$y = -0.0002x + 12$
	20000 a 50000	$y = -0.0001x + 10$

	50000 a 100000	$y = -0.00002x + 6$
Nitratos (mg/l)	0 a 10	$y = -4.8x + 98$
	10 a 30	$y = -1.1x + 61$
	30 a 50	$y = -0.9x + 55$
	50 a 60	$y = -0.2x + 20$
	60 a 70	$y = -0.1x + 14$
	70 a 80	$y = -0.2x + 21$
	80 a 90	$y = -0.1x + 13$
	90 a 100	$y = -0.2x + 22$
Fosfatos (mg/l)	0 a 1	$y = -60x + 100$
	1 a 2	$y = -12x + 52$
	2 a 3	$y = -8x + 44$
	3 a 5	$y = -3.5x + 30.5$
	5 a 7	$y = -2.5x + 25.5$
	7 a 8	$y = -1x + 15$
	8 a 9	$y = 7$
	9 a 10	$y = -2x + 25$
Temperatura (°C)	10 a 0	$y = 3.4x + 91$
	0 a 5	$y = -4.2x + 91$
	5 a 10	$y = -5.4x + 97$
	10 a 15	$y = -3x + 73$
	15 a 20	$y = -1.6x + 52$
	20 a 25	$y = -0.6x + 32$
	25 a 30	$y = -1.4x + 52$
pH (pH)	2 a 4	$y = 5x - 10$
	4 a 5	$y = 19x - 66$
	5 a 6	$y = 23x - 86$
	6 a 7	$y = 38x - 176$
	7 a 7.5	$y = 8x + 34$
	7.5 a 8	$y = -20x + 244$
	8 a 9	$y = -34x + 356$
	9 a 10	$y = -30x + 320$

	10 a 11	$y = -12x + 140$
	11 a 12	$y = -8x + 96$
Oxígeno disuelto (mg/l)	0 a 20	$y = 0.6x$
	20 a 40	$y = 0.9x - 6$
	40 a 60	$y = 1.45x - 28$
	60 a 80	$y = 1.5x - 31$
	80 a 100	$y = 0.55x + 45$
	100 a 120	$y = -0.5x + 150$
	120 a 140	$y = -0.55x + 156$

Fuente: Elaboración propia

4.5 Selección de ecuaciones de determinación del Qi para la utilización en el código de programación

En la tabla 19 se muestra el valor Qi obtenido con los dos métodos de linealización de las curvas de función. En la tabla 20 se muestra el porcentaje de error, que se obtuvo con los datos del estudio realizado en el río Monjas en el sector de Pomasqui por Gutiérrez (2019), donde se aplicó las ecuaciones obtenidas en excel y las ecuaciones obtenidas por funciones a trozos. Como fin de esta comparación, es identificar el método de linealización con menor error para la determinación del Qi.

Tabla 19: Comparación de resultados para la determinación del Qi, utilizando los dos métodos de linealización propuestos, con datos del estudio en el Río Monjas-Sector de Pomasqui (mes de junio)

Parámetros	Unidades	valor medido	Qi Determinado en el estudio	Qi Funciones a trozos	Qi Funciones de Excel
pH	U pH	7,21	90	91,68	94,00
Oxígeno disuelto	%	17	10	10,20	10,00
Turbidez	NTU	43,1	44	45,21	43,10
Fosfato	mg/L	272	2	2,00	2,00
Nitratos	mg/L	0,7	90	94,64	100,00
Sólidos disueltos totales	mg/L	423	45	41,55	70,45
DBO5	mg/L	50	2	2,00	2,00

Coliformes fecales	NMP/100mL	10000	10	10,00	10,00
Variación de temperatura	°C	-0,6	88	88,96	90,00

Fuente: Elaboración propia

Tabla 20:Tabla de comparación de los porcentajes de error entre los métodos de la linealización propuestos en el presente estudio

Parámetros	Unidades	% de error de ecuaciones a trozos	% de error de ecuaciones obtenidas en excel
pH	U pH	2%	3%
Oxígeno disuelto	%	2%	2%
Turbidez	NTU	3%	5%
Fosfato	mg/L	0%	0%
Nitratos	mg/L	5%	6%
Solidos disueltos totales	mg/L	8%	70%
DBO5	mg/L	0%	0%
Coliformes fecales	NMP/100mL	0%	0%
Variación de temperatura	°C	1%	1%

Fuente: Elaboración propia

Tras realizar el cálculo del error relativo se concluye que el método de funciones a trozos tiene menor porcentaje de error para la determinación de Qi, en cada parámetro lo que significa que este método es más veraz y preciso.

4.6 Creación del código de programación

El programa se desarrolló en el programa Python de libre acceso para el cual se utilizó los siguientes pasos para la conformación del código.

4.6.1 Criterios de cálculo del ICA global

Como se mencionó en la sección de metodología para la determinación del ICA global se utilizó dos métodos ICA-NSF y HPI (índice de metales pesados).

En el caso de la ICA-NSF se utiliza el procedimiento descrito en la sección 2.3 y 2.4, en donde el Qi se determina con las funciones obtenidas con el método de trozos y el cálculo del ICA con las ponderaciones de la tabla 16. En la Tabla 10 de la sección 2.5.1 se presentó la valoración de la calidad del agua, en función del ICA-NSF obtenido con una escala del 0 al 100. Mientras que, para el ICA-HPI en la siguiente tabla se muestra las ponderaciones y

las fórmulas que se utilizaron para determinar el Qi con los límites permisibles basados en el TULSMA (Anexo VI, Tabla 1) con lo cual se determinó el índice de metales pesados. La determinación del HPI se realiza según lo especificado en la sección 2.5. En la Tabla 11 de la sección 2.5.1 se presentó la valoración de la calidad del agua, en función del HPI obtenido con una escala del 0 al 100.

Tabla 21: Tabla de ponderaciones y determinación de Qi para el HPI

Parámetro	Ponderación (Wi)	Formula (Qi)
Aluminio total (mg/l)	0.2	$Q_i = 100 * \frac{\text{Aluminio}}{0.2}$
Cadmio total (mg/l)	100	$Q_i = 100 * \frac{\text{Cadmio}}{0.01}$
Cobre total (mg/l)	5	$Q_i = 100 * \frac{\text{Cobre}}{1}$
Hierro total (mg/l)	0.2	$Q_i = 100 * \frac{\text{Hierro}}{1}$
Manganeso total (mg/l)	5	$Q_i = 100 * \frac{\text{Manganeso}}{0.1}$
Plomo total (mg/l)	20	$Q_i = 100 * \frac{\text{Plomo}}{0.05}$
Zinc total (mg/l)	0.5	$Q_i = 100 * \frac{5}{0.2}$

Fuente: Elaboración propia

Como se mencionó anteriormente, dependiendo del número y parámetros que disponga el usuario, el programa podrá calcular únicamente el ICA-NSF (mínimo 7 parámetros, máximo 9 parámetros), o únicamente el HPI (mínimo 1 metal, máximo 7 metales), o ambos a través de la determinación del ICA global según lo explicado en la sección 2.5. En la siguiente tabla se muestra la escala de clasificación que se utilizó para evaluar la calidad del agua en base al porcentaje del ICA global obtenido, en el caso de calcular el ICA-NSF y HPI. Además, en el programa que se diseñara el ICA. HPI será denominado ICA metales para mejor comprensión por parte del usuario.

Tabla 22: Escala de clasificación para el ICA HPI y ICA NSF

Calidad del agua	Valor del ICA
Excelente	91 – 100
Buena	71 – 90
Media	51 – 70
Mala	26 – 50
Muy Mala	0 – 25

Fuente: Elaboración propia

Adicionalmente, como ya se ha mencionado, con los resultados de ICA global se puede obtener los usos del agua más adecuados en función de los resultados obtenidos. En la tabla 1 del Anexo III se presenta los usos en función de los porcentajes del ICA global. Incorporar la tabla que utilizas en el programa para determinar los usos del agua.

4.6.2 Forma de presentación visual del programa del cálculo del ICA-NSF

a) Ingreso continuo de los valores

En la siguiente imagen se muestra la primera prueba que se realizó para comenzar la creación del código del programa, donde al usuario se le mostraba de manera continua los parámetros que debe ingresar.

Figura 7: Captura de la forma uno del programa

```
Ingrese el valor obtenido de la DBO: 4
El Qi de DBO es 4
Ingrese el valor obtenido de la pH: 7
El Qi de pH es 7
```

Fuente: Elaboración propia

b) Uso de un menú

La otra forma que se optó es la creación de un menú donde permita que el usuario que va a ocupar el programa pueda escoger el parámetro en el cual va a ingresar el valor medido ya sea en laboratorio o insitu.

Figura 8: Presentación del programa en forma de menú

```
Bienvenido al cálculo del ICA
El numero minimo de parámetros para determinar el ICA por el metodo Nsf son 7 parámetros
Seleccione el parámetro e ingrese el valor obtenido

La lista de parámetros está vacía

Seleccione un parámetro:
1) DBO
2) SDT
3) Turbidez
4) Coliformes fecales
5) Nitratos
6) Fosfatos
7) Temperatura
8) pH
9) OD
10) Calcular
0) Salir
Opción: 1
Has elegido la opción 1:
Ingrese el valor obtenido de la DBO en mg/L: 3

Se han ingresado los siguientes parámetros:
DBO = 3.0 con un valor igual a 73.0
```

Fuente: Elaboración propia

c) Pantalla dinámica

La tercera forma de presentación que se utilizó fue la de pantallas dinámicas, esta opción es la que se utilizó para crear el programa de cálculo del ICA-NSF y ICA-HPI. Finalmente, se optó por esta forma de presentación debido a que el interfaz con el usuario es más amigable cuando se realiza la ejecución de este, como se muestra en la figura 9.

Figura 9: Pantalla dinámica de Python

Fuente: Elaboración propia

Para mayor detalle del código utilizado para la creación del programa de cálculo del ICA global (ICA NSF-HPI) se adjunta el código completo en el Anexo 4.

Además, para facilitar el uso del programa se optó por convertir el programa creado en un archivo “.exe”, debido a que este tipo de archivos tienen la facilidad de ejecutarse directamente sin la necesidad de instalar Python. También se anexa un link el cual se encuentra en el Anexo 5, donde se encontrará el código final en Python y el aplicativo del programa en extensión “.exe”. Así también, en el Anexo 5 se adjunta un manual para el uso del programa desarrollado en presente estudio.

4.7 Validación

El fin de la validación es comprobar la confiabilidad del uso del programa de cálculo del ICA global (ICA NSF-HPI), y se realiza la comparación de resultados de estudios que hayan sido efectuados con la misma metodología determinado el % de error.

4.7.1 Validación del ICA-NSF con 9 parámetros de calidad del agua

Para validar el programa se tomó valores de ICA-NSF de estudios realizados en otros lugares con las gráficas de las curvas de función originales, para comparar el porcentaje de ICA-NSF obtenido en este trabajo según la metodología explicada en secciones anteriores.

Para esto se tomó los siguientes estudios:

- Río Sinincay – Cuenca (año)

Para la validación se tomó en cuenta los resultados obtenidos del análisis de la calidad del Agua del Río Sinincay de la provincia de Azuay, los parámetros utilizados fueron obtenidos mediante el respectivo análisis de laboratorio y mediciones in-situ.

Tabla 23: Comparación del ICA-NSF con el calculado en el Río Sinincay

MES: JUNIO			
ESTACIÓN	CHOCHAS	EL CHORRO	CENTRO PARROQUIAL
PARAMETROS	M1	M2	M3
Temperatura (°C)	10.6	11.5	12.3
Sólidos disueltos totales (mg/l)	29.45	163	105.4
Turbidez (NTU)	7	182	37
pH (pH)	8.2	8.3	8.1

Fosfatos (mg/l)	0.07	0.61	0.23
Nitratos (mg/l)	0.4	3.8	2.2
Oxígeno disuelto	101.67	85.97	34.87
DBO ₅ (mg/l)	0.3	0.9	2.6
Coliformes fecales (NMP/100 ml)	100	1400	9000

ICA PYTHON	80.30	61.91	55.10
ICA CALCULADO ESTUDIO	81.00	64.00	58.00
%ERROR	1%	3%	5%

Fuente: Elaboración propia

- Evaluación espacio temporal de la calidad del agua del río Monjas, sectores Pomasqui y San Antonio de Pichincha mediante ICA-NSF en dos meses del año xx, en este estudio se determinó el ICA a través del cálculo manual del Qi mediante gráficas de función.

Tabla 24: Comparación del ICA-NSF en el Río Pomasqui

Parámetros	MESES EVALUADOS	
	MAYO	JUNIO
pH (pH)	7.23	7.21
Oxígeno disuelto (mg/l)	16.1	17
Turbidez (NTU)	65.1	43.1
Fosfatos (mg/l)	13.1	272
Nitratos (mg/l)	5.5	0.7
Sólidos disueltos totales	482	423
DBO ₅ (mg/l)	54	50
Coliformes fecales (NMP/100 ml)	12000	10000
Temperatura (°C)	-0.3	-0.6

ICA PYTHON	35.85	39.93
ICA CALCULADO ESTUDIO	36.04	40.22
% ERROR	0,527%	0.721%

Fuente: Elaboración propia

Se concluye que los resultados de Qi calculados son similares, y el cálculo del ICA-NSF tiene un error máximo del 0,721% respecto al ICA real determinado del estudio en el mes de junio.

En ambos casos de estudio utilizados para comparación, el error calculado entre ICA obtenido de Python y el ICA real del estudio es menor al 5%. Por lo tanto, se considera que el programa es apto para el uso en los lugares requeridos por el usuario según el método ICA-NSF sugerido por la Fundación Nacional de Sanidad de Estados Unidos

4.7.2 Validación ICA-NSF con 7 parámetros de calidad del agua (número mínimo requerido por el programa en Python propuesto)

Para validar con 7 parámetros se tomó un estudio realizado por la Universidad de Pamplona, (2022) donde aplicó dicha variación y redistribución de ponderaciones, y los resultados obtenidos se muestran a continuación:

Tabla 25: Comparación del ICA-NSF con 7 parámetros de calidad del agua

Parámetros	Valores medidos
pH (pH)	7.67
Oxígeno disuelto (mg/l)	82
Turbidez (NTU)	5
Fosfatos (mg/l)	0.5
Nitratos (mg/l)	5
Sólidos disueltos totales (mg/l)	150
DBO5 (mg/l)	0
Coliformes fecales (NMP/100 ml)	12
Temperatura (°C)	0
ICA PYTHON	80.56
ICA CALCULADO ESTUDIO	78.40
% ERROR	2.80%

Fuente: Elaboración propia

Se concluye que los resultados de Qi calculados son similares, y el cálculo del ICA-NSF tiene un error máximo del 2,80% respecto al ICA real determinado del estudio realizado por la Universidad de Pamplona.

En ambos casos de estudio utilizados para comparación, el error calculado entre ICA obtenido de Python y el ICA real del estudio es menor al 5%. Por lo tanto, se considera que el programa es apto para el uso en los lugares requeridos por el usuario según el método ICA-NSF sugerido por la Fundación Nacional de Sanidad de Estados Unidos

4.7.3 Validación con ICA-HPI

Para validar el ICA-HPI, se tomó los resultados obtenidos del análisis de la calidad del agua del Río San Juan de la ciudad de Lima en el año 2021, los cuales fueron obtenidos mediante el respectivo análisis de laboratorio y mediciones in-situ. En el estudio de referencia se utiliza el mismo método propuesto de cálculo del índice HPI, lo que facilita su comparación para la validación respectiva.

Tabla 26: Comparación del índice HPI (Río San Juan-Lima)

Parámetros de calidad del agua	Valores medidos
Aluminio total (mg/l)	0.4990
Cadmio total (mg/l)	0.0002
Cobre total (mg/l)	0.2094
Hierro total (mg/l)	0.6854
Manganeso total (mg/l)	0.6626
Plomo total (mg/l)	0.0082
Zinc total (mg/l)	0.2198

ICA PYTHON	20.62
ICA CALCULADO ESTUDIO	20.62
% ERROR	0%

Fuente: Elaboración propia

En el caso comparado el error existente entre ICA obtenido de Python y el ICA real del estudio es menor al 5%. Por lo tanto, se considera que el programa es apto para el uso en los lugares requeridos por el usuario según el método ICA-HPI el mismo que es sugerido por el Estándar de Calidad Ambiental.

4.8 Flexibilidad del programa

Según lo explicado en sección 2.5 de la metodología, para determinar el ICA global se consideró dos métodos de cálculo del ICA, que son:

- El método ICA-NSF
- El método ICA-HPI (índice de metales)

En el cálculo del ICA global no necesariamente se deben tener todos los valores de los parámetros tanto del método NSF como del método HPI. El ICA global se puede calcular si el usuario tiene los datos tanto de los 16 parámetros, o un número menor de parámetros

tanto para el ICA NSF o HPI; incluso el usuario puede utilizar cualquiera de los dos métodos separados.

Además, hay que tener en cuenta que en el programa diseñado el método ICA-NSF se puede calcular con un mínimo 7 y un máximo 9 parámetros. Y por otro lado, el índice HPI que se utiliza para metales no tiene un requerimiento mínimo de parámetros. Por lo tanto, el programa calculará el ICA global en función de los parámetros ingresados y adicionalmente proporcionará información de los tipos de uso del agua que se pueda realizar.

4.9 Manual para el usuario

Con el fin de proporcionar al usuario información del funcionamiento del programa en Python utilizado, se realizó un manual con la siguiente conformación:

- Introducción
- Objetivos del programa
- Como ingresar al programa
- Cálculos del ICA_NSF e ICA_HPI (metales)
- Resultados
- Función de las pestañas
- Pasos para buscar en la base de datos

El manual se adjunta en el Anexo VII.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Como resultado de la investigación bibliografía que se realizó para la elaboración de este trabajo se concluyó que el método de cálculo de índice de calidad del agua con mayor aplicación en Ecuador es el método ICA-NSF.
- Se concluye que el programa desarrollado en Python es eficaz para la evaluación de la calidad del agua porque disminuye tiempos de cálculo para la obtención del

índice de calidad del agua (ICA) en base a 9 parámetros de calidad del agua de entrada (máximo) según el método NSF o como mínimo 7 parámetros.

- Luego de varios análisis queda en evidencia que el método NSF es el método más confiable en la determinación del ICA.
- Tras realizar las validaciones se concluye que el uso del programa elaborado en Python tiene un margen de error inferior al 5% lo que conlleva a concluir que es un programa confiable para determinar el ICA. Para el ICA-NSF se obtuvo un error entre el 0,5 y 5%, y para el ICA-HPI se obtuvo un error del 0%.
- El método para linealizar las curvas de función con menor error (10%) fue el de trozos, que proporcionó un mejor ajuste al momento de calcular el Q_i para la determinación del ICA-NSF.
- Los resultados de este trabajo, en resumen, la creación del programa podría ayudar a disminuir el tiempo en el cálculo del ICA.
- Los usos del agua definidos con la obtención del ICA en este estudio fueron: recreativo, industrial, agua potable, vida acuática y agricultura.
- Como consecuencia de lo expuesto en este trabajo, se concluye que la mejor manera de crear un programa es cuando el usuario tiene un interfaz amigable y de fácil uso.

5.2 Recomendaciones

- Tras realizar el cálculo del ICA se recomienda que se realicen las actividades recomendadas de tratamiento para cada uso.
- Con la plantación del índice de calidad del agua creado, se pueden plantear mejoras para ayudar a realizar actividades de tratamiento para su debido uso.
- Se recomienda que al utilizar el programa en el parámetro de temperatura se ingrese el valor de la diferencia entre la temperatura ambiente y la temperatura medida para evitar un porcentaje de error alto.
- En caso de que el estudio analizado tenga metales se debe utilizar el ICA-HBI para determinar e ICA de los metales o en caso de que en el estudio no exista metales y existan los otros parámetros se recomienda solo utilizar el ICA-NSF.
- Para trabajos futuros se recomienda que realizar distintos análisis, estadísticos para adoptar un ICA de metales con mayor grado de confiabilidad.

- Para la utilización del programa diseñado se recomienda que antes de ingresar el valor medido el usuario debe asegurarse que tengan las mismas unidades.

6. REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, M., Vanegas, E., & García, N. (2016). Aplicación del Índice de Calidad del Agua (ICA). Caso de estudio: Lago de Izabal, Guatemala. *Revista Ciencias Técnicas Agropecuarias*, 25(2).
- Astigarraga, E. (n.d.). *EL MÉTODO DELPHI*.
- Capparelli, M. V., Cabrera, M., Moulatlet, G. M., Pinos Vélez, V., Pérez González, A., Solis, O. L., Galarza, E., Alvear, D., Vasco, S., Guamangallo, J., Cevallos, M., Shiguango, Lady, & Ramos, M. (2021). *EVALUACIÓN DEL GRADO DE AFETACIÓN DE LA ACTIVIDAD MINERA SOBRE LOS ECOSISTEMAS ACUÁTICOS EN LA PROVINCIA DEL NAPO* Autores del informe Docentes e investigadores: Estudiantes.
- Centro Nacional de Metrología. (2017). *Guía para la validación y la verificación de los procedimientos de examen cuantitativos empleados por el laboratorio clínico*. 22–47.
- Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García. (2014). El lenguaje de programación Python/The programming language Python. *Ciencias Holguín*, XX, 4–12. <http://www.linuxjournal.com/article/2959>
- Chávez, H., Leiva, D., Rascón, J., Hoyos, I., & Corroto, F. (2016). Índice de calidad del agua según NSF del humedal laguna Los Milagros (Tingo María, Perú) Water quality index-NSF of wetland lagoon Los Milagros (Tingo Maria, Peru) Benazir. *Indes*, 2(2).
- Chirinos, C. (2022). *ÍNDICE DE CALIDAD DE AGUA Y CONTENIDO DEMETALES PESADOS EN EL RÍO SAN JUAN, CERRO DE PASCO*. Universidad Nacional Agraria La Molina.
- Coello, J. R., Ormaza, R. M., Recalde, C. G., & Rios, A. C. (2015). Aplicación del ICA-NSF para determinar la calidad del agua de los ríos Ozogoché, Pichahuiña y Pomacocho-Parque Nacional Sangay-Ecuador. *Revista Del Instituto de Investigación de La Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica*, 16(31).
- Digesa. (2022). *PARÁMETROS ORGANOLÉPTICOS*.
- Fernández, N., Ramos, G., & Solano, F. (2005). Índices de Calidad de Agua (ICAs) e Índices de Contaminación (ICOs) de Importancia Mundial. In *Índices de Calidad y de Contaminación del Agua* (pp. 42–116).

- García, T. (2012). *PROPUESTA DE ÍNDICES DE CALIDAD DE AGUA PARA ECOSISTEMAS HÍDRICOS DE CHILE* [PROPUESTA DE ÍNDICES DE CALIDAD DE AGUA PARA ECOSISTEMAS HÍDRICOS DE CHILE]. Universidad de Chile.
- García-González, J., Osorio-Ortega, M. A., Saquicela-Rojas, R. A., & Cadme, M. L. (2021). Determinación del índice de calidad del agua en ríos de Santo Domingo de los Tsáchilas, Ecuador. *Ingeniería Del Agua*, 25(2).
<https://doi.org/10.4995/ia.2021.13921>
- Gómez Cerezo, R. (n.d.). Estructura y funcionamiento de los ecosistemas acuáticos continentales: un análisis comparativo. Escalas y procesos. In *Universidad de Murcia*. Universidad de Murcia.
- Gómez Rocío. (2016). *El color en la Psicología*.
https://tauja.ujaen.es/bitstream/10953.1/3474/1/Gmez_Caizares_Roco_TFG_Psicologia.pdf
- Guadarrama, R., Kido, J., Roldan, G., & Salas, M. (2016). Contaminación del agua. *Revista de Ciencias Ambientales y Recursos Naturales* , 2, 1–10.
www.ecorfan.org/spain
- Guerra, M., & Zaldumbide, D. (2010). *La agonía del Puyango: agua, minería y contaminación*. www.digesa.sld.pe/pw_deepa/pdf/
- Gutiérrez, S. (2019). *Evaluación espaciotemporal de la calidad del agua del río Monjas, sectores Pomasqui y San Antonio de Pichincha mediante ICA-NSF*. Universidad Central del Ecuador.
- Hinojoza, N. (2018). *EVALUACIÓN DE LA CALIDAD DEL AGUA DEL RÍO SAN PEDRO, SECTOR VALLE DE LOS CHILLOS, MEDIANTE EL ÍNDICE DE CALIDAD DE AGUA (ICA-NSF)*. Universidad Central del Ecuador .
- Lina, P., & Mora, S. (2016). LA BIORREMEDIACION COMO ALTERNATIVA DE RECUPERACIÓN PARA CUERPOS DE AGUA LÉNTICOS EN LA CIUDAD DE BOGOTÁ. *Boletín Semillas Ambientales*, 10, 6–11.
- Martínez, M. (n.d.). *FUNCIONES DEFINIDAS A TROZOS*. Math.0.
www.matematicasjmmm.com

- Méndez, P., Arcos, J., & Cazorla, X. (2020). DeterminacionDelIndiceDeCalidadDelAguaNSFDeIRioCop. *DOMINIO DE LAS CIENCIAS*, 6, 734–746. <https://doi.org/https://dx.doi.org/10.23857/dc.v6i3.1245>
- Muñoz, O. (n.d.). Ecuación de la recta. In *Escuela de Ciencias Aplicadas*.
- Ortega, K. (2020). *Determinación del caudal ecológico superficial mediante el método Q95% y Python en la sub cuencadel río Tulumayo – Chanchamayo 2020*. Universidad Continental.
- Perez Alvarado, J. K. (2017). Determinacion Del Indice De Calidad Del Agua Del Rio Moquegus Por Influenca Del Vertimiento De La Planta De Tratamiento De Aguas Residuales -Omo,Durante El Periodo 2014-2015. In *Universidad José Carlos Mariategui*.
- Pérez, J. I., Nardini, A. G., & Galindo, A. A. (2018). Comparative analysis of water quality indices applied to the ranchería river, la guajira-Colombia. *Informacion Tecnologica*, 29(3), 47–58. <https://doi.org/10.4067/S0718-07642018000300047>
- Quiñones-Huatangari, L., Ochoa, L., Milla-Pino, M. E., Bazán, J. F., Gamarra, O. A., & Rascón, J. (2020). Water quality index using fuzzy logic Utcubamba River, Peru. *Revista de Ciencias Agrícolas*, 37(1), 6–18. <https://doi.org/10.22267/rcia.203701.124>
- Quiroz, L., Izquierdo, E., & Menéndez, C. (2017). *Aplicación del índice de calidad de agua en el río Portoviejo, Ecuador*. XXXVIII, 41–51.
- Quiroz, S., Izquierdo, E., & Menéndez, C. (2017). Aplicación del índice de calidad de agua en el río Portoviejo, Ecuador. *INGENIERÍA HIDRÁULICA Y AMBIENTAL*, XXXVIII, 41–51. <https://www.researchgate.net/publication/331015238>
- Reguant-Álvarez, M., & Torrado-Fonseca, M. (2016). El método Delphi. *REIRE*, 9(1), 87–102. <https://doi.org/10.1344/reire2016.9.1916>
- Ruis, A., García, J., & Mesa, J. (2010). ERROR, INCERTIDUMBRE, PRECISIÓN Y EXACTITUD, TÉRMINOS ASOCIADOS A LA CALIDAD ESPACIAL DEL DATO GEOGRÁFICO. *UNIVERSIDAD DE JAÉN*, 1–8.
- Sánchez, J. (n.d.). *Correlación lineal y regresión*. Departamento. Geología . <http://web.usal.es/javisan/hidroPág.1>

- Tejero, I. (2000). *Problemas de ingeniería sanitaria y ambiental: Vol. I*. Escuela Técnica Superior Ingenieros de Caminos, Canales y Puertos de Santander.
- Torres, P., Camilo, C., & Paola, P. (2009). WATER QUALITY INDEX IN SURFACE SOURCES USED IN WATER PRODUCTION FOR HUMAN CONSUMPTION. A CRITICAL REVIEW. *Ingenierías Universidad de Medellín*, 81–93.
- Universidad Europea de Madrid. (n.d.). *FUNCIONES REALES DE UNA VARIABLE LAS FUNCIONES ELEMENTALES Y EL DOMINIO*.
- Ureta Valdez, R., Méndez Zambrano, P., & Cazar Rivera, E. (2019). Influencia de la zona urbana de Macas en el Índice Calidad de Agua del río Jurumbaino. *Ciencia Digital*, 3(3.1). <https://doi.org/10.33262/cienciadigital.v3i3.1.679>
- Vidal, L. R. L., & Carreño Mendoza, Á. L. (2018). Calidad de agua de consumo humano en las comunidades balsa en medio, Julián y Severino de la microcuenca Carrizal, Ecuador. *Revista Del Instituto de Investigación de La Facultad de Minas, Metalurgia y Ciencias Geográficas*, 21(42).

7. ANEXOS

ANEXO I

Tesis-Liseth Collaguazo

ORIGINALITY REPORT

12%

SIMILARITY INDEX

11%

INTERNET SOURCES

2%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1	www2.slideshare.net Internet Source	1 %
2	sinat.semarnat.gob.mx Internet Source	1 %
3	repositorio.uniautonoma.edu.co Internet Source	1 %
4	Submitted to Aliat Universidades Student Paper	1 %
5	www.villamaria.gob.ar Internet Source	1 %
6	Submitted to Universidad Nacional de Itapúa Student Paper	<1 %
7	repository.udistrital.edu.co Internet Source	<1 %
8	fr.slideshare.net Internet Source	<1 %
9	prezi.com Internet Source	<1 %



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 3 de marzo de 2023

De mi consideración:

Yo, DRA. VERÓNICA ELIZABETH MORALES CASA, en calidad de Directora del Trabajo de Integración Curricular titulado **DESARROLLO DE UN PROGRAMA EN PHYTON PARA DETERMINAR EL ICA** asociado al CÁLCULO DEL ÍNDICE DEL CALIDAD DEL AGUA CON PHYTON EN CUERPOS DE AGUA SUPERFICIAL LOTICOS, IMPACTADOS POR ACTIVIDADES HUMANAS elaborado por la estudiante **LISETH CAROLINA COLLAGUAZO CARRERA** de la carrera en AGUA Y SANEAMIENTO AMBIENTAL, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito de las secciones: Resumen, Introducción, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



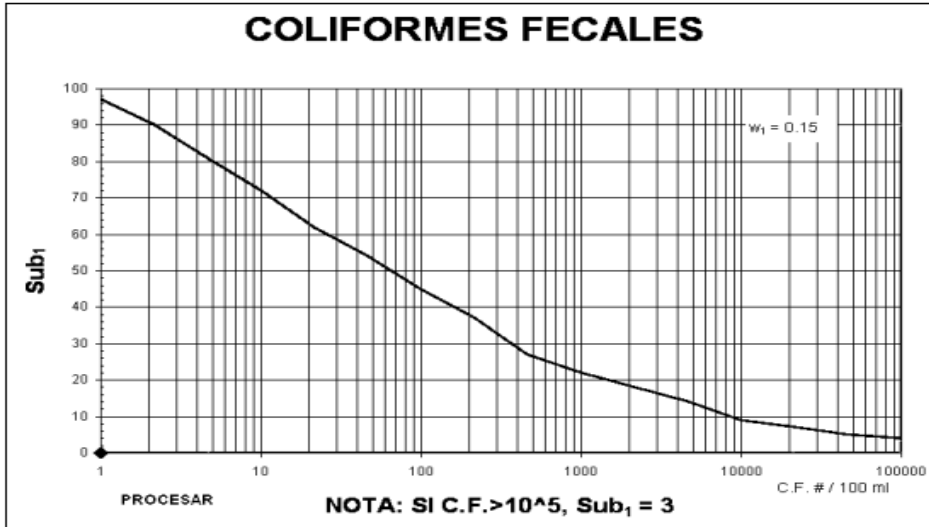
Dra. Verónica Morales C.
Profesora a tiempo completo
Carrera de Agua y Saneamiento - ESFOT

ANEXO II

Las gráficas de función de los parámetros del método NSF.

Figura 1.

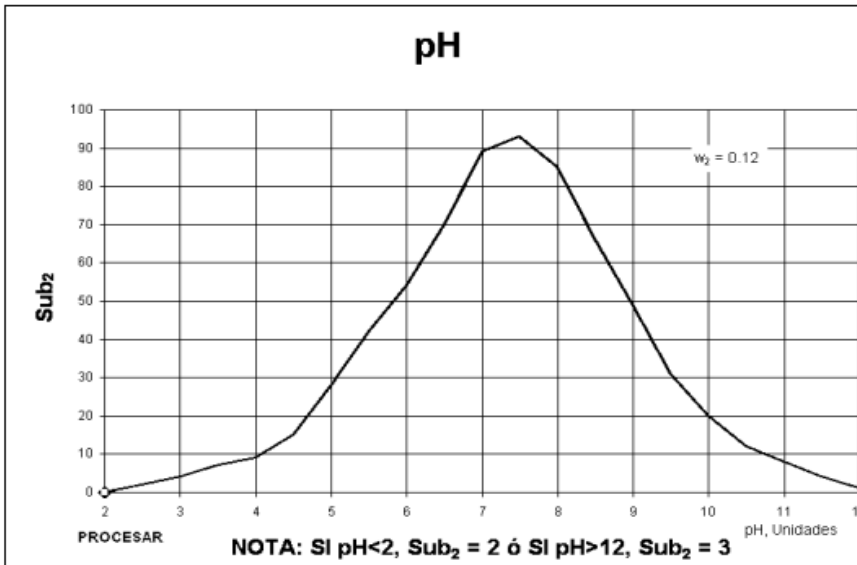
Valoración de la calidad de agua en función de Coliformes Fecales



Fuente: (SNET,2021)

Figura 2.

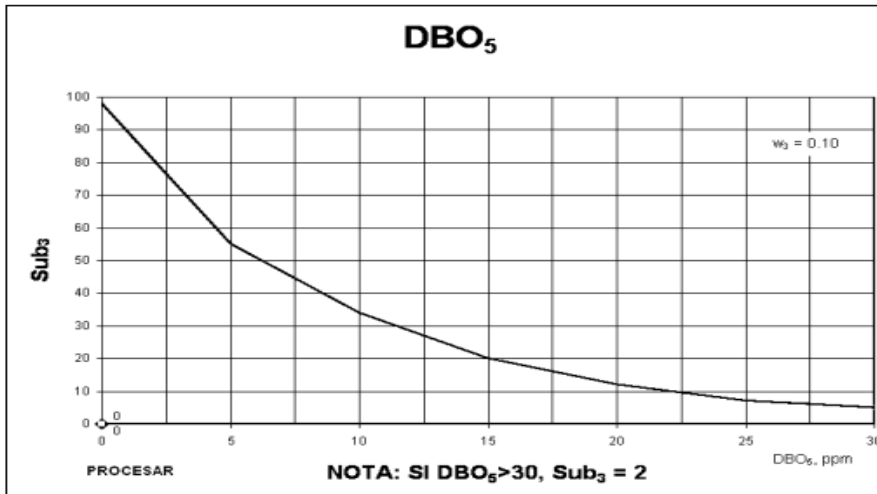
Valoración de la calidad de agua en función del pH



Fuente: (SNET,2021)

Figura 3.

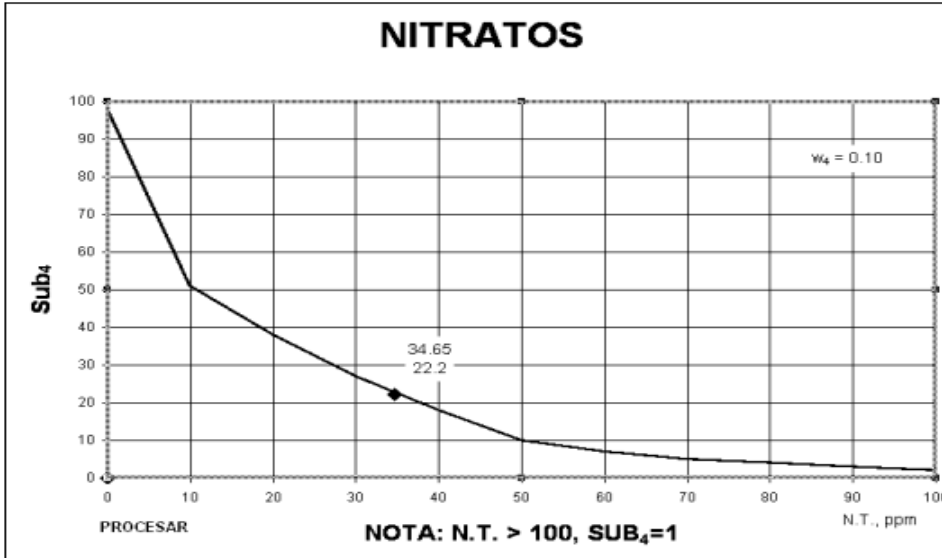
Valoración de la calidad de agua en función de la DBO5



Fuente: (SNET,2021)

Figura 4.

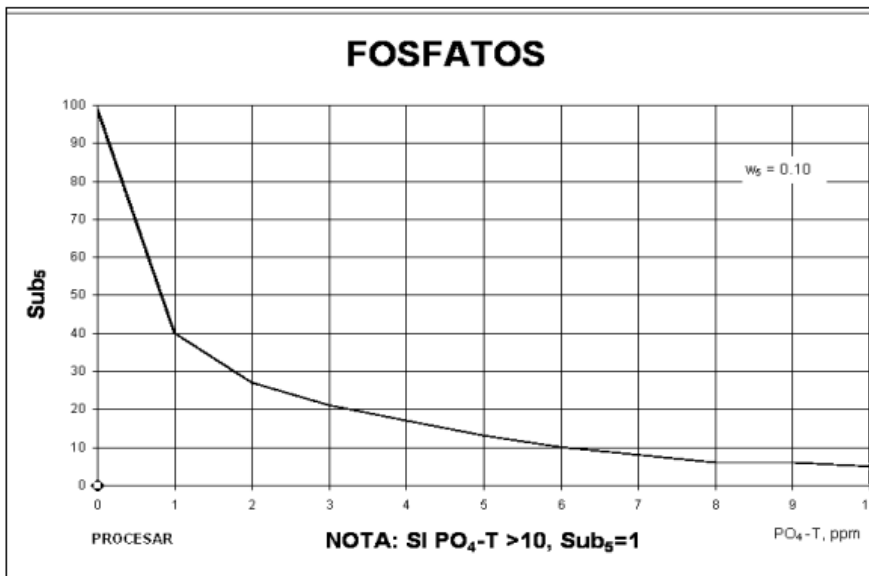
Valoración de la calidad de agua en función del Nitrógeno



Fuente: (SNET,2021)

Figura 5.

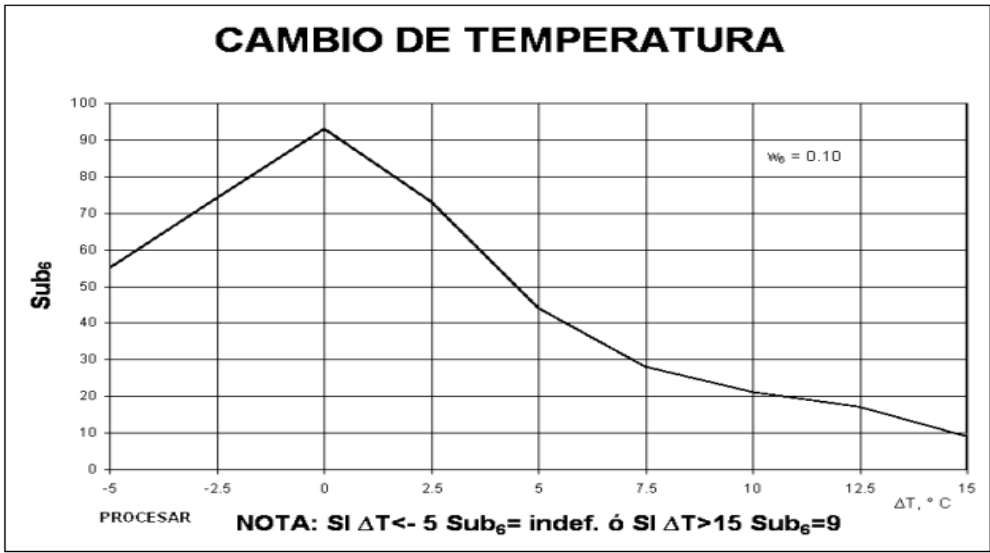
Valoración de la calidad de agua en función del Fósforo



Fuente: (SNET,2021)

Figura 6.

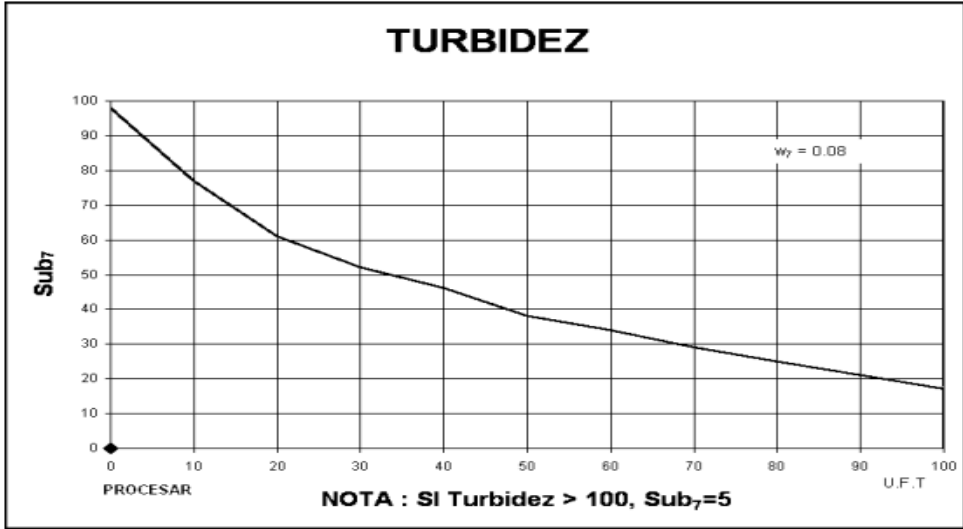
Valoración de la calidad de agua en función de la Temperatura



Fuente: (SNET,2021)

Figura 7.

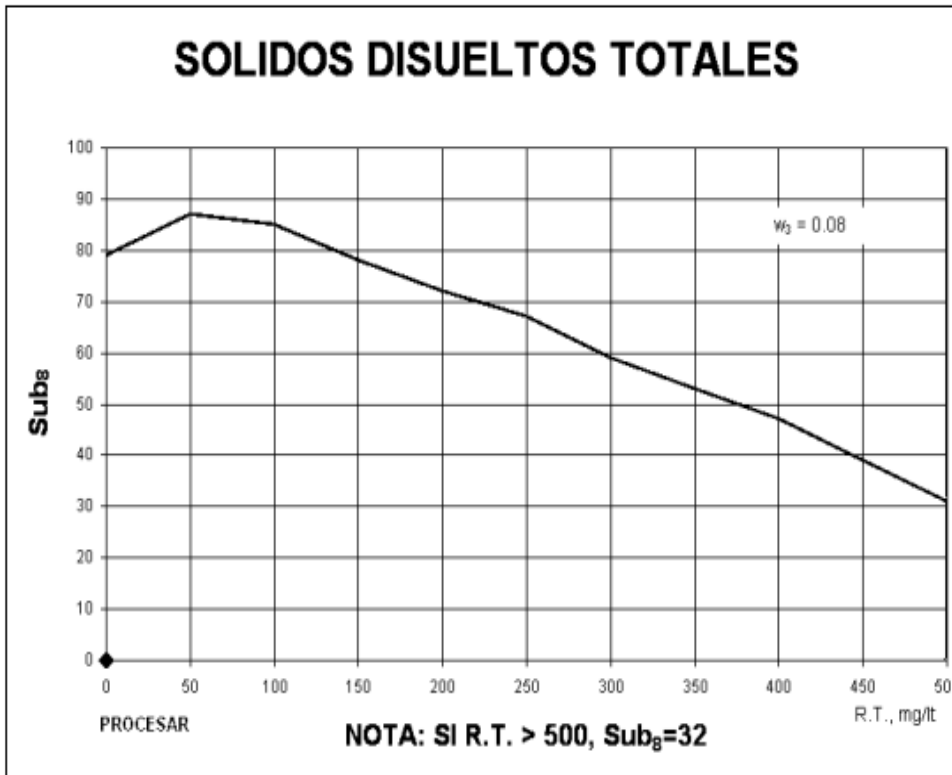
Valoración de la calidad de agua en función de la Turbidez



Fuente: (SNET,2021)

Figura 8.

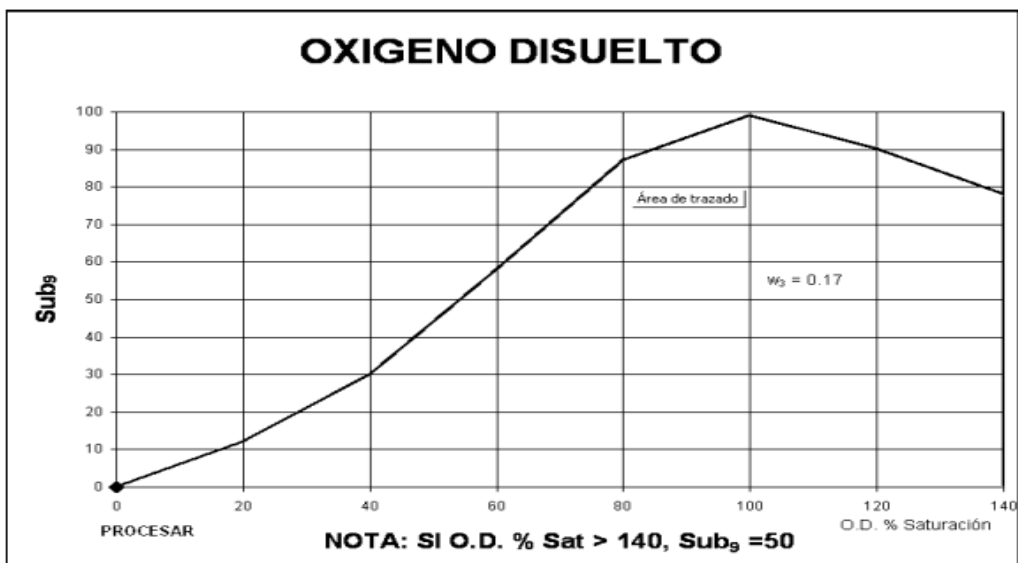
Valoración de la calidad de agua en función del Residuo Total



Fuente: (SNET,2021)

Figura 9.

Valoración de la calidad de agua en función del Residuo Total



Fuente: (SNET,2021)

ANEXO III

En el Anexo 3 se muestran los criterios de uso del agua según el estado de ella.

Tabla 1: Criterios de Uso

Para uso de agua potable	
Parámetros	Criterios específicos
90-100	No requiere purificación para consumo.
80-90	Purificación menor requerida.
70-80	Dudoso su consumo sin purificación.
50-70	Tratamiento potabilizador necesario.
40-50	Dudosa para consumo.
0-40	Inaceptable para consumo.
Para uso en Agricultura	
Parámetros	Criterios específicos
90-100	No requiere purificación para riego.
70-90	Purificación menor para cultivos que requieran de alta calidad de agua.
50-70	Utilizable en mayoría de cultivos
30-50	Tratamiento requerido para la mayoría de los cultivos.
20-30	Uso solo en cultivos muy resistentes.
0-20	Inaceptable para riego.
Para uso en Pesca y Vida Acuática	
Parámetros	Criterios específicos
70-100	Pesca y vida acuática abundante.
60-70	Límite para peces muy sensitivos.
50-60	Dudosa la pesca sin riesgos de salud.
40-50	Vida acuática limitada a especies muy resistentes.
30-40	Inaceptable para actividad pesquera.
0-30	Inaceptable para vida acuática.
Para uso industrial	
Parámetros	Criterios específicos
90-100	No se requiere purificación.
70-90	Purificación menor para industrias que requieran alta calidad de agua para operación.

50-70	No requiere tratamiento para mayoría de industrias de operación normal.
30-50	Tratamiento para mayoría de usos.
20-30	Uso restringido en actividades burdas.
0-20	Inaceptable para cualquier industria.
Para uso recreativo	
Parámetros	Criterios específicos
70-100	Cualquier tipo de deporte acuático.
50-70	Restringir los deportes de inmersión, precaución si se ingiere dada la posibilidad de presencia de bacterias
40-50	Dudosa para contacto con el agua.
30-40	Evitar contacto, sólo con lanchas.
20-30	Contaminación visible, evitar cercanía.
0-20	Inaceptable para recreación.

Fuente: (LIEBMAN, H.1969)

ANEXO IV

En el Anexo IV se muestra el código de programación que se utilizó para crear el programa:

```
import tkinter as tk
import sqlite3 as sql

## CONSTANTES GLOBALES
epsilon = 1e-15
cuadro_Agua_Potable = {
    'No requiere purificación para consumo.': [90, 100],
    'Purificación menor requerida': [80, 90-epsilon],
    'Dudoso su consumo sin purificación': [70, 80-epsilon],
    'Tratamiento potabilizador necesario.': [50, 70-epsilon],
    'Dudosa para consumo.': [40, 50-epsilon],
    'Inaceptable para consumo.': [0, 40-epsilon]
}
cuadro_Agricultura = {
    'No requiere purificación para riego.': [90, 100],
    'Purificación menor para cultivos de alta calidad': [70, 90-epsilon],
    'Utilizable en la mayoría de cultivos': [50, 70-epsilon],
    'Tratamiento requerido para la mayoría de cultivos.': [30, 50-epsilon],
    'Uso solo en cultivos muy resistentes.': [20, 30-epsilon],
    'Inaceptable para riego.': [0, 20-epsilon]
}
cuadro_Pesca_Vida = {
    'Pesca y vida acuática abundante.': [70, 100],
    'Límite para peces muy sensitivos': [60, 70-epsilon],
    'Dudosa la pesca sin riesgos de salud': [50, 60-epsilon],
    'Vida acuática limitada a especies muy resistentes.': [40, 50-epsilon],
    'Inaceptable para actividad pesquera.': [30, 40-epsilon],
    'Inaceptable para vida acuática.': [0, 30-epsilon]
```

```

}
cuadro_Industrial = {
  'No requiere purificación.': [90, 100],
  'Purificación menor para industrias de alta calidad.': [70, 90-epsilon],
  'No requiere tratamiento para la mayoría de industrias.': [50, 70-epsilon],
  'Tratamiento requerido para la mayoría de usos.': [30, 50-epsilon],
  'Uso restringido en actividades burdas.': [20, 30-epsilon],
  'Inaceptable para cualquier industria.': [0, 20-epsilon]
}
cuadro_Recreativo = {
  'Cualquier tipo de deporte acuático.': [70, 100],
  'Restricción para deportes de inmersión.': [50, 70-epsilon],
  'Dudosa para contacto con el agua.': [40, 50-epsilon],
  'Evitar contacto, solo con lanchas.': [30, 40-epsilon],
  'Contaminación visible, evitar cercanía.': [20, 30-epsilon],
  'Inaceptable para recreación.': [0, 20-epsilon]
}

```

```

ponderaciones_NSF = {
  'DBO': 0.1,
  'SDT': 0.08,
  'Turbidez': 0.08,
  'Coliformes fecales': 0.15,
  'Nitratos': 0.1,
  'Fosfatos': 0.1,
  'Temperatura': 0.1,
  'pH': 0.12,
  'OD': 0.17,
}

```

```

ponderaciones_Metales = {
  'Aluminio': 0.2,
  'Cadmio': 100,
  'Cobre': 5,
  'Hierro': 0.2,
  'ManganesoTotal': 5,
  'Plomo': 20,
  'ZincTotal': 0.5
}

```

VARIABLES GLOBALES

```

calculo_parametros_NSF = {}
calculo_parametros_Metales = {}
calculo_parametros_total = {}

```

```

ponderaciones_aux = {}

```

```

parametros_NSF = {
  'DBO': False,
  'SDT': False,
  'Turbidez': False,
  'Coliformes fecales': False,
  'Nitratos': False,
  'Fosfatos': False,
  'Temperatura': False,
  'pH': False,
}

```

```

'OD': False,
}

parametros_Metales = {
'Aluminio': False,
'Cadmio': False,
'Cobre': False,
'Hierro': False,
'ManganesoTotal': False,
'Plomo': False,
'ZincTotal': False
}

ICAs_calculados = {
'ICA_NSF': False,
'ICA_Metales': False
}

## DEFINICION DE FUNCIONES EXTRAS
def ponderaciones_nuevas():
    aux = {}
    for key in ponderaciones_NSF:
        if key in parametros_NSF:
            ponderaciones_aux[key]=ponderaciones_NSF[key]
        else:
            aux[key]=ponderaciones_NSF[key]
    pond_aux = 0
    for key in aux:
        pond_aux = pond_aux + aux[key]
    pond_aux = pond_aux/len(ponderaciones_aux)
    for key in ponderaciones_aux:
        ponderaciones_aux[key] = ponderaciones_aux[key] + pond_aux

## DEFINICION DE FUNCIONES PARA VALIDACIONES DE LOS CAMPOS
def validar_entrada_real(val):
    try:
        float(val)
        return True
    except:
        return False

def validar_valor_vacio(valor):
    return valor=="

## DEFINICION DE FUNCIONES PARA VALIDACIONES DE CALCULO
def validar_calcular_ICA_NSF():
    cont = 0
    for key in parametros_NSF:
        if parametros_NSF[key]==True:
            cont = cont + 1
    return cont>=7

def validar_calcular_ICA_Metales():
    cont = 0
    for key in parametros_Metales:
        if parametros_Metales[key]==True:

```

```

        cont = cont + 1
    return cont==7

def validar_calcular_ICA_Total():
    cont = 0
    for key in ICAs_calculados:
        if ICAs_calculados[key]==True:
            cont = cont + 1
    return cont>=1

def validar_Guardar():
    bandera_INFO = False
    bandera_NSF = False
    bandera_Metales = False
    bandera_Resultado = False
    bandera_FINAL = False
    if (not validar_valor_vacio(campo_NombreEstudio.get())
        and not validar_valor_vacio(campo_FechaEstudio.get())
        and not validar_valor_vacio(campo_LugarEstudio.get())
        and not validar_valor_vacio(campo_RioEstudio.get())
        and not validar_valor_vacio(campo_EncargadoEstudio.get())
        and not validar_valor_vacio(campo_DescripcionEstudio.get())
    ):
        bandera_INFO = True
    cont = 0
    if (not validar_valor_vacio(campo_DBO.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_SDT.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Turbidez.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_ColiformesFecales.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Nitratos.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Fosfatos.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Temperatura.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_pH.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_OD.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_ver_ICA_NSF.get())):
        cont = cont + 1
    if (cont >= 8):
        bandera_NSF = True

    if (not validar_valor_vacio(campo_Aluminio.get())
        and not validar_valor_vacio(campo_Cadmio.get())
        and not validar_valor_vacio(campo_Cobre.get())
        and not validar_valor_vacio(campo_Hierro.get())
        and not validar_valor_vacio(campo_ManganesoTotal.get())
        and not validar_valor_vacio(campo_Plomo.get())
        and not validar_valor_vacio(campo_ZincTotal.get())
        and not validar_valor_vacio(campo_ver_ICA_Metales.get())
    )

```

```

):
bandera_Metales = True

if (not validar_valor_vacio(campo_ICA_AguaPotable.get())
    and not validar_valor_vacio(campo_ICA_Agricultura.get())
    and not validar_valor_vacio(campo_ICA_Pesca.get())
    and not validar_valor_vacio(campo_ICA_Industrial.get())
    and not validar_valor_vacio(campo_ICA_Recreativo.get())
    and not validar_valor_vacio(campo_ICA_Total.get())):
    bandera_Resultado = True

if (bandera_INFO and bandera_Resultado and (bandera_NSF or bandera_Metales)):
    bandera_FINAL = True

return bandera_FINAL

def validar_Actualizar():
    bandera_INFO = False
    bandera_NSF = False
    bandera_Metales = False
    bandera_Resultado = False
    bandera_FINAL = False
    if (not validar_valor_vacio(campo_NombreEstudio.get())
        and not validar_valor_vacio(campo_FechaEstudio.get())
        and not validar_valor_vacio(campo_LugarEstudio.get())
        and not validar_valor_vacio(campo_RioEstudio.get())
        and not validar_valor_vacio(campo_EncargadoEstudio.get())
        and not validar_valor_vacio(campo_DescripcionEstudio.get())
    ):
        bandera_INFO = True
    cont = 0
    if (not validar_valor_vacio(campo_DBO.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_SDT.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Turbidez.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_ColiformesFecales.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Nitratos.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Fosfatos.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_Temperatura.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_pH.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_OD.get())):
        cont = cont + 1
    if (not validar_valor_vacio(campo_ver_ICA_NSF.get())):
        cont = cont + 1
    if (cont >= 8):
        bandera_NSF = True

    if (not validar_valor_vacio(campo_Aluminio.get())
        and not validar_valor_vacio(campo_Cadmio.get()))

```

```

and not validar_valor_vacio(campo_Cobre.get())
and not validar_valor_vacio(campo_Hierro.get())
and not validar_valor_vacio(campo_ManganesoTotal.get())
and not validar_valor_vacio(campo_Plomo.get())
and not validar_valor_vacio(campo_ZincTotal.get())
and not validar_valor_vacio(campo_ver_ICA_Metales.get())
):
bandera_Metales = True

if (not validar_valor_vacio(campo_ICA_AguaPotable.get())
and not validar_valor_vacio(campo_ICA_Agricultura.get())
and not validar_valor_vacio(campo_ICA_Pesca.get())
and not validar_valor_vacio(campo_ICA_Industrial.get())
and not validar_valor_vacio(campo_ICA_Recreativo.get())
and not validar_valor_vacio(campo_ICA_Total.get())):
bandera_Resultado = True

if (bandera_INFO and bandera_Resultado and (bandera_NSF or bandera_Metales)):
bandera_FINAL = True

return bandera_FINAL

def validar_Visualizar():
if (not validar_valor_vacio(campo_NombreEstudio.get())):
return True
else:
return False

def validar_Eliminar():
if (not validar_valor_vacio(campo_NombreEstudio.get())):
return True
else:
return False

## DEFINICION DE FUNCIONES PARA CALCULOS INTERMEDIOS
def calcular_DBO():
DBO = campo_DBO.get()
if (not validar_valor_vacio(DBO)):
if (validar_entrada_real(DBO)):
DBO=float(DBO)
if DBO>=0 and DBO<=30:
if DBO<5:
calculo_parametros_NSF['DBO']=-9*DBO+100
elif DBO>=5 and DBO<10:
calculo_parametros_NSF['DBO']=-4.4*DBO+77
elif DBO>=10 and DBO<15:
calculo_parametros_NSF['DBO']=-2.6*DBO+59
elif DBO>=15 and DBO<20:
calculo_parametros_NSF['DBO']=-1.6*DBO+44
elif DBO>=20 and DBO<25:
calculo_parametros_NSF['DBO']=-0.8*DBO+28
elif DBO>=25 and DBO<=30:
calculo_parametros_NSF['DBO']=-0.6*DBO+23
parametros_NSF['DBO']=True
elif DBO>30:

```

```

        calculo_parametros_NSF['DBO']=2
        parametros_NSF['DBO']=True
    else:
        parametros_NSF['DBO']=False
        tk.messagebox.showinfo(title="ERROR", message="DBO entre 0 y 30")
        campo_DBO.delete(0, tk.END)
        campo_DBO.insert(0, "")
    else:
        parametros_NSF['DBO']=False
        tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
        campo_DBO.delete(0, tk.END)
        campo_DBO.insert(0, "")
    else:
        parametros_NSF['DBO']=False

def calcular_SDT():
    SDT = campo_SDT.get()
    if (not validar_valor_vacio(SDT)):
        if (validar_entrada_real(SDT)):
            SDT=float(SDT)
            if SDT>=0 and SDT<=500:
                if SDT<50:
                    calculo_parametros_NSF['SDT']=0.16*SDT+80
                elif SDT>=50 and SDT<100:
                    calculo_parametros_NSF['SDT']=-0.04*SDT+90
                elif SDT>=100 and SDT<300:
                    calculo_parametros_NSF['SDT']=-0.13*SDT+99
                elif SDT>=300:
                    calculo_parametros_NSF['SDT']=-0.15*SDT+105
                elif SDT<=500:
                    calculo_parametros_NSF['SDT']=-0.15*SDT+105
            parametros_NSF['SDT']=True
        elif SDT>500:
            parametros_NSF['SDT']=True
            calculo_parametros_NSF['SDT']=20
        else:
            parametros_NSF['SDT']=False
            tk.messagebox.showinfo(title="ERROR", message="SDT entre 0 y 500")
            campo_SDT.delete(0, tk.END)
            campo_SDT.insert(0, "")
    else:
        parametros_NSF['SDT']=False
        tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
        campo_SDT.delete(0, tk.END)
        campo_SDT.insert(0, "")
    else:
        parametros_NSF['SDT']=False

def calcular_Turbidez():
    Turbidez = campo_Turbidez.get()
    if (not validar_valor_vacio(Turbidez)):
        if (validar_entrada_real(Turbidez)):
            Turbidez=float(Turbidez)
            if Turbidez>=0 and Turbidez<=100:
                if Turbidez<10:
                    calculo_parametros_NSF['Turbidez']=-2.1*Turbidez+98

```

```

elif Turbidez>=10 and Turbidez<20:
    calculo_parametros_NSF[Turbidez]=-1.7*Turbidez+94
elif Turbidez>=20 and Turbidez<40:
    calculo_parametros_NSF[Turbidez]=-0.6*Turbidez+72
elif Turbidez>=40 and Turbidez<50:
    calculo_parametros_NSF[Turbidez]=-0.9*Turbidez+84
elif Turbidez>=50 and Turbidez<70:
    calculo_parametros_NSF[Turbidez]=-0.45*Turbidez+61.5
elif Turbidez>=70 and Turbidez<=100:
    calculo_parametros_NSF[Turbidez]=-0.4*Turbidez+58
parametros_NSF[Turbidez]=True
elif Turbidez>100:
    parametros_NSF[Turbidez]=True
    calculo_parametros_NSF[Turbidez]=5
else:
    parametros_NSF[Turbidez]=False
    tk.messagebox.showinfo(title="ERROR", message="Turbidez entre 0 y 100")
    campo_Turbidez.delete(0, tk.END)
    campo_Turbidez.insert(0, "")
else:
    parametros_NSF[Turbidez]=False
    tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    campo_Turbidez.delete(0, tk.END)
    campo_Turbidez.insert(0, "")
else:
    parametros_NSF[Turbidez]=False

def calcular_ColiformesFecales():
    Coliformes_fecales = campo_ColiformesFecales.get()
    if (not validar_valor_vacio(Coliformes_fecales)):
        if (validar_entrada_real(Coliformes_fecales)):
            Coliformes_fecales=float(Coliformes_fecales)
            if Coliformes_fecales>=1 and Coliformes_fecales<=100000:
                if Coliformes_fecales<2:
                    calculo_parametros_NSF[Coliformes fecales]=-9*Coliformes_fecales+108
                elif Coliformes_fecales>=2 and Coliformes_fecales<20:
                    calculo_parametros_NSF[Coliformes fecales]=-((14/9)*Coliformes_fecales+838/9)
                elif Coliformes_fecales>=20 and Coliformes_fecales<200:
                    calculo_parametros_NSF[Coliformes fecales]=-((2/15)*Coliformes_fecales+194/3)
                elif Coliformes_fecales>=200 and Coliformes_fecales<500:
                    calculo_parametros_NSF[Coliformes fecales]=-((1/30)*Coliformes_fecales+134/3)
                elif Coliformes_fecales>=500 and Coliformes_fecales<2000:
                    calculo_parametros_NSF[Coliformes fecales]=-0.006*Coliformes_fecales+31
                elif Coliformes_fecales>=2000 and Coliformes_fecales<10000:
                    calculo_parametros_NSF[Coliformes fecales]=-0.001125*Coliformes_fecales+21.25
                elif Coliformes_fecales>=10000 and Coliformes_fecales<20000:
                    calculo_parametros_NSF[Coliformes fecales]=-0.0002*Coliformes_fecales+12
                elif Coliformes_fecales>=20000 and Coliformes_fecales<50000:
                    calculo_parametros_NSF[Coliformes fecales]=-0.0001*Coliformes_fecales+10
                elif Coliformes_fecales>=50000 and Coliformes_fecales<=100000:
                    calculo_parametros_NSF[Coliformes fecales]=-0.00002*Coliformes_fecales+6
            parametros_NSF[Coliformes fecales]=True
        elif Coliformes_fecales>100000:
            parametros_NSF[Coliformes fecales]=True
            calculo_parametros_NSF[Coliformes fecales]=2
        else:

```



```

    parametros_NSF['Coliformes fecales']=False
    tk.messagebox.showinfo(title="ERROR", message="Coliformes fecales entre 1 y
100000")
    campo_ColiformesFecales.delete(0, tk.END)
    campo_ColiformesFecales.insert(0, "")
else:
    parametros_NSF['Coliformes fecales']=False
    tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    campo_ColiformesFecales.delete(0, tk.END)
    campo_ColiformesFecales.insert(0, "")
else:
    parametros_NSF['Coliformes fecales']=False

def calcular_Nitratos():
    Nitratos = campo_Nitratos.get()
    if (not validar_valor_vacio(Nitratos)):
        if (validar_entrada_real(Nitratos)):
            Nitratos=float(Nitratos)
            if Nitratos>=0 and Nitratos<=100:
                if Nitratos<10:
                    calculo_parametros_NSF['Nitratos']=-4.8*Nitratos+98
                elif Nitratos>=10 and Nitratos<30:
                    calculo_parametros_NSF['Nitratos']=-1.1*Nitratos+61
                elif Nitratos>=30 and Nitratos<50:
                    calculo_parametros_NSF['Nitratos']=-0.9*Nitratos+55
                elif Nitratos>=50 and Nitratos<60:
                    calculo_parametros_NSF['Nitratos']=-0.2*Nitratos+20
                elif Nitratos>=60 and Nitratos<70:
                    calculo_parametros_NSF['Nitratos']=-0.1*Nitratos+14
                elif Nitratos>=70 and Nitratos<80:
                    calculo_parametros_NSF['Nitratos']=-0.2*Nitratos+21
                elif Nitratos>=80 and Nitratos<90:
                    calculo_parametros_NSF['Nitratos']=-0.1*Nitratos+13
                elif Nitratos>=90 and Nitratos<=100:
                    calculo_parametros_NSF['Nitratos']=-0.2*Nitratos+22
                parametros_NSF['Nitratos']=True
            elif Nitratos>100:
                parametros_NSF['Nitratos']=True
                calculo_parametros_NSF['Nitratos']=1
            else:
                parametros_NSF['Nitratos']=False
                tk.messagebox.showinfo(title="ERROR", message="Nitratos entre 0 y 100")
                campo_Nitratos.delete(0, tk.END)
                campo_Nitratos.insert(0, "")
        else:
            parametros_NSF['Nitratos']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
            campo_Nitratos.delete(0, tk.END)
            campo_Nitratos.insert(0, "")
    else:
        parametros_NSF['Nitratos']=False

def calcular_Fosfatos():
    Fosfatos = campo_Fosfatos.get()
    if (not validar_valor_vacio(Fosfatos)):
        if (validar_entrada_real(Fosfatos)):

```

```

Fosfatos=float(Fosfatos)
if Fosfatos>=0 and Fosfatos<=10:
    if Fosfatos<1:
        calculo_parametros_NSF['Fosfatos']=-60*Fosfatos+100
    elif Fosfatos>=1 and Fosfatos<2:
        calculo_parametros_NSF['Fosfatos']=-12*Fosfatos+52
    elif Fosfatos>=2 and Fosfatos<3:
        calculo_parametros_NSF['Fosfatos']=-8*Fosfatos+44
    elif Fosfatos>=3 and Fosfatos<5:
        calculo_parametros_NSF['Fosfatos']=-3.5*Fosfatos+30.5
    elif Fosfatos>=5 and Fosfatos<7:
        calculo_parametros_NSF['Fosfatos']=-2.5*Fosfatos+25.5
    elif Fosfatos>=7 and Fosfatos<8:
        calculo_parametros_NSF['Fosfatos']=-Fosfatos+15
    elif Fosfatos>=8 and Fosfatos<9:
        calculo_parametros_NSF['Fosfatos']=7
    elif Fosfatos>=9 and Fosfatos<=10:
        calculo_parametros_NSF['Fosfatos']=-2*Fosfatos+25
    parametros_NSF['Fosfatos']=True
elif Fosfatos>10:
    parametros_NSF['Fosfatos']=True
    calculo_parametros_NSF['Fosfatos']=2
else:
    parametros_NSF['Fosfatos']=False
    tk.messagebox.showinfo(title="ERROR", message="Fosfatos entre 0 y 10")
    campo_Fosfatos.delete(0, tk.END)
    campo_Fosfatos.insert(0, "")
else:
    parametros_NSF['Fosfatos']=False
    tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    campo_Fosfatos.delete(0, tk.END)
    campo_Fosfatos.insert(0, "")
else:
    parametros_NSF['Fosfatos']=False

def calcular_Temperatura():
    Temperatura = campo_Temperatura.get()
    if (not validar_valor_vacio(Temperatura)):
        if (validar_entrada_real(Temperatura)):
            Temperatura=float(Temperatura)
            if Temperatura>=- 10 and Temperatura<=30:
                if Temperatura<0:
                    calculo_parametros_NSF['Temperatura']=3.4*Temperatura+91
                elif Temperatura>=0 and Temperatura<5:
                    calculo_parametros_NSF['Temperatura']=-4.2*Temperatura+91
                elif Temperatura>=5 and Temperatura<10:
                    calculo_parametros_NSF['Temperatura']=-5.4*Temperatura+97
                elif Temperatura>=10 and Temperatura<15:
                    calculo_parametros_NSF['Temperatura']=-3*Temperatura+73
                elif Temperatura>=15 and Temperatura<20:
                    calculo_parametros_NSF['Temperatura']=-1.6*Temperatura+52
                elif Temperatura>=20 and Temperatura<25:
                    calculo_parametros_NSF['Temperatura']=-0.6*Temperatura+32
                elif Temperatura>=25 and Temperatura<=30:
                    calculo_parametros_NSF['Temperatura']=-1.4*Temperatura+53
            parametros_NSF['Temperatura']=True

```

```

else:
    parametros_NSF['Temperatura']=False
    tk.messagebox.showinfo(title="ERROR", message="Temperatura entre -10 y 30")
    campo_Temperatura.delete(0, tk.END)
    campo_Temperatura.insert(0, "")
else:
    parametros_NSF['Temperatura']=False
    tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    campo_Temperatura.delete(0, tk.END)
    campo_Temperatura.insert(0, "")
else:
    parametros_NSF['Temperatura']=False

def calcular_pH():
    pH = campo_pH.get()
    if (not validar_valor_vacio(pH)):
        if (validar_entrada_real(pH)):
            pH=float(pH)
            if pH>=2 and pH<=12:
                if pH<4:
                    calculo_parametros_NSF['pH']=5*pH-10
                elif pH>=4 and pH<5:
                    calculo_parametros_NSF['pH']=19*pH-66
                elif pH>=5 and pH<6:
                    calculo_parametros_NSF['pH']=23*pH-86
                elif pH>=6 and pH<7:
                    calculo_parametros_NSF['pH']=38*pH-176
                elif pH>=7 and pH<7.5:
                    calculo_parametros_NSF['pH']=8*pH+34
                elif pH>=7.5 and pH<8:
                    calculo_parametros_NSF['pH']=-20*pH+244
                elif pH>=8 and pH<9:
                    calculo_parametros_NSF['pH']=-34*pH+356
                elif pH>=9 and pH<10:
                    calculo_parametros_NSF['pH']=-30*pH+320
                elif pH>=10 and pH<11:
                    calculo_parametros_NSF['pH']=-12*pH+10
                elif pH>=11 and pH<=12:
                    calculo_parametros_NSF['pH']=-8*pH+96
                parametros_NSF['pH']=True
            elif pH<2:
                parametros_NSF['pH']=True
                calculo_parametros_NSF['pH']=0
            elif pH>12:
                parametros_NSF['pH']=True
                calculo_parametros_NSF['pH']=0
            else:
                parametros_NSF['pH']=False
                tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
                campo_pH.delete(0, tk.END)
                campo_pH.insert(0, "")
        else:
            parametros_NSF['pH']=False

def calcular_OD():
    OD = campo_OD.get()

```

```

if (not validar_valor_vacio(OD)):
    if (validar_entrada_real(OD)):
        OD=float(OD)
        if OD>=0 and OD<=140:
            if OD<20:
                calculo_parametros_NSF['OD']=0.6*OD
            elif OD>=20 and OD<40:
                calculo_parametros_NSF['OD']=0.9*OD-6
            elif OD>=40 and OD<60:
                calculo_parametros_NSF['OD']=1.45*OD-28
            elif OD>=60 and OD<80:
                calculo_parametros_NSF['OD']=1.5*OD-31
            elif OD>=80 and OD<100:
                calculo_parametros_NSF['OD']=0.55*OD+45
            elif OD>=100 and OD<120:
                calculo_parametros_NSF['OD']=-0.5*OD+150
            elif OD>=120 and OD<=140:
                calculo_parametros_NSF['OD']=-0.55*OD+156
            parametros_NSF['OD']=True
        elif OD>140:
            parametros_NSF['OD']=True
            calculo_parametros_NSF['OD']=50
        else:
            parametros_NSF['OD']=False
            tk.messagebox.showinfo(title="ERROR", message="OD entre 0 y 140")
            campo_OD.delete(0, tk.END)
            campo_OD.insert(0, "")
    else:
        parametros_NSF['OD']=False
        tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
        campo_OD.delete(0, tk.END)
        campo_OD.insert(0, "")
else:
    parametros_NSF['OD']=False

def calcular_Aluminio():
    Aluminio = campo_Aluminio.get()
    if (not validar_valor_vacio(Aluminio)):
        if (validar_entrada_real(Aluminio)):
            Aluminio = float(Aluminio)
            calculo_parametros_Metales['Aluminio']=100*Aluminio/0.2
            parametros_Metales['Aluminio']=True
        else:
            parametros_Metales['Aluminio']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['Aluminio']=False

def calcular_Cadmio():
    Cadmio = campo_Cadmio.get()
    if (not validar_valor_vacio(Cadmio)):
        if (validar_entrada_real(Cadmio)):
            Cadmio = float(Cadmio)
            calculo_parametros_Metales['Cadmio']=100*Cadmio/0.01
            parametros_Metales['Cadmio']=True
        else:

```

```

        parametros_Metales['Cadmio']=False
        tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['Cadmio']=False

def calcular_Cobre():
    Cobre = campo_Cobre.get()
    if (not validar_valor_vacio(Cobre)):
        if (validar_entrada_real(Cobre)):
            Cobre = float(Cobre)
            calculo_parametros_Metales['Cobre']=100*Cobre
            parametros_Metales['Cobre']=True
        else:
            parametros_Metales['Cobre']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['Cobre']=False

def calcular_Hierro():
    Hierro = campo_Hierro.get()
    if (not validar_valor_vacio(Hierro)):
        if (validar_entrada_real(Hierro)):
            Hierro = float(Hierro)
            calculo_parametros_Metales['Hierro']=100*Hierro
            parametros_Metales['Hierro']=True
        else:
            parametros_Metales['Hierro']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['Hierro']=False

def calcular_ManganesoTotal():
    ManganesoTotal = campo_ManganesoTotal.get()
    if (not validar_valor_vacio(ManganesoTotal)):
        if (validar_entrada_real(ManganesoTotal)):
            ManganesoTotal = float(ManganesoTotal)
            calculo_parametros_Metales['ManganesoTotal']=100*ManganesoTotal/0.1
            parametros_Metales['ManganesoTotal']=True
        else:
            parametros_Metales['ManganesoTotal']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['ManganesoTotal']=False

def calcular_Plomo():
    Plomo = campo_Plomo.get()
    if (not validar_valor_vacio(Plomo)):
        if (validar_entrada_real(Plomo)):
            Plomo = float(Plomo)
            calculo_parametros_Metales['Plomo']=100*Plomo/0.05
            parametros_Metales['Plomo']=True
        else:
            parametros_Metales['Plomo']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['Plomo']=False

```

```

def calcular_ZincTotal():
    ZincTotal = campo_ZincTotal.get()
    if (not validar_valor_vacio(ZincTotal)):
        if (validar_entrada_real(ZincTotal)):
            ZincTotal = float(ZincTotal)
            calculo_parametros_Metales['ZincTotal']=100*ZincTotal/0.2
            parametros_Metales['ZincTotal']=True
        else:
            parametros_Metales['ZincTotal']=False
            tk.messagebox.showinfo(title="ERROR", message="Ingrese un valor válido...")
    else:
        parametros_Metales['ZincTotal']=False

## DEFINICION DE FUNCIONES PARA LOS BOTONES DE CALCULO
def calcular_ICA_NSF():
    calcular_DBO()
    calcular_SDT()
    calcular_Turbidez()
    calcular_ColiformesFecales()
    calcular_Nitratos()
    calcular_Fosfatos()
    calcular_Temperatura()
    calcular_pH()
    calcular_OD()

    total = 0
    if (validar_calcular_ICA_NSF()):
        if len(parametros_NSF)==len(ponderaciones_NSF):
            for key in parametros_NSF:
                if key in calculo_parametros_NSF:
                    total = total + calculo_parametros_NSF[key]*ponderaciones_NSF[key]
        elif len(parametros_NSF)<len(ponderaciones_NSF):
            ponderaciones_nuevas()
            for key in parametros_NSF:
                if key in calculo_parametros_NSF:
                    total = total + calculo_parametros_NSF[key]*ponderaciones_aux[key]
            campo_ver_ICA_NSF.config(state='normal')
            campo_ver_ICA_NSF.delete(0, tk.END)
            campo_ver_ICA_NSF.insert(0, total)
            ICAs_calculados['ICA_NSF'] = True
            campo_ver_ICA_NSF.config(state='disabled')
            calculo_parametros_total['ICA_NSF'] = total
        else:
            tk.messagebox.showinfo(title="ERROR", message="Rellene al menos 7 campos...")

def calcular_ICA_Metales():
    calcular_Aluminio()
    calcular_Cadmio()
    calcular_Cobre()
    calcular_Hierro()
    calcular_ManganesoTotal()
    calcular_Plomo()
    calcular_ZincTotal()

    total = 0

```

```

acum = 0
if (validar_calcular_ICA_Metales()):
    for key in parametros_Metales:
        total = total + calculo_parametros_Metales[key]*ponderacions_Metales[key]
        acum = acum + ponderacions_Metales[key]
    total = total / acum
    if (total>100):
        total = 0
    else:
        total = 100 -total
    campo_ver_ICA_Metales.config(state="normal")
    campo_ver_ICA_Metales.delete(0, tk.END)
    campo_ver_ICA_Metales.insert(0, total)
    ICAs_calculados['ICA_Metales'] = True
    campo_ver_ICA_Metales.config(state="disabled")
    calculo_parametros_total['ICA_Metales'] = total
else:
    tk.messagebox.showinfo(title="ERROR", message="Rellene todos los campos...")

def calcular_ICA_Total():
    ica = 0
    if (validar_calcular_ICA_Total()):
        for key in calculo_parametros_total:
            ica = ica + calculo_parametros_total[key]
        ica = ica / len(calculo_parametros_total)
        campo_ICA_Total.config(state="normal")
        campo_ICA_Total.delete(0, tk.END)
        campo_ICA_Total.insert(0, ica)
        campo_ICA_Total.config(state="disabled")
        ICAs_calculados['ICA_Total'] = ica
        for key in cuadro_Agua_Potable:
            if (ica>cuadro_Agua_Potable[key][0] and ica<cuadro_Agua_Potable[key][1]):
                campo_ICA_AguaPotable.config(state="normal")
                campo_ICA_AguaPotable.delete(0, tk.END)
                campo_ICA_AguaPotable.insert(0, key)
                campo_ICA_AguaPotable.config(state="disabled")
        for key in cuadro_Pesca_Vida:
            if (ica>cuadro_Pesca_Vida[key][0] and ica<cuadro_Pesca_Vida[key][1]):
                campo_ICA_Pesca.config(state="normal")
                campo_ICA_Pesca.delete(0, tk.END)
                campo_ICA_Pesca.insert(0, key)
                campo_ICA_Pesca.config(state="disabled")
        for key in cuadro_Agricultura:
            if (ica>cuadro_Agricultura[key][0] and ica<cuadro_Agricultura[key][1]):
                campo_ICA_Agricultura.config(state="normal")
                campo_ICA_Agricultura.delete(0, tk.END)
                campo_ICA_Agricultura.insert(0, key)
                campo_ICA_Agricultura.config(state="disabled")
        for key in cuadro_Industrial:
            if (ica>cuadro_Industrial[key][0] and ica<cuadro_Industrial[key][1]):
                campo_ICA_Industrial.config(state="normal")
                campo_ICA_Industrial.delete(0, tk.END)
                campo_ICA_Industrial.insert(0, key)
                campo_ICA_Industrial.config(state="disabled")
        for key in cuadro_Recreativo:
            if (ica>cuadro_Recreativo[key][0] and ica<cuadro_Recreativo[key][1]):

```

```

        campo_ICA_Recreativo.config(state="normal")
        campo_ICA_Recreativo.delete(0, tk.END)
        campo_ICA_Recreativo.insert(0, key)
        campo_ICA_Recreativo.config(state="disabled")
    else:
        tk.messagebox.showinfo(title="ERROR", message="Debe calcular ICA-NSF e ICA-Metales
        previamente...")

```

DEFINICION DE FUNCIONES PARA LA BASE DE DATOS

```

def guardar_Info():
    if (validar_Guardar()
    ):
        connection = sql.connect('programa_ica')
        cursor = connection.cursor()
        try:
            cursor.execute("""INSERT INTO info_ica(
                nombre_estudio,
                fecha_estudio,
                lugar_estudio,
                rio_estudio,
                encargado_estudio,
                descripcion_estudio,
                dbo_nsf,
                sdt_nsf,
                turbidez_nsf,
                coliformes_fecales_nsf,
                nitratos_nsf,
                fosfatos_nsf,
                temperatura_nsf,
                ph_nsf,
                od_nsf,
                aluminio_metales,
                cadmio_metales,
                cobre_metales,
                hierro_metales,
                manganeso_total_metales,
                plomo_metales,
                zinc_total_metales,
                valor_ica_nsf,
                valor_ica_metales,
                valor_ica_total,
                resultado_ica_agua_potable,
                resultado_ica_agricultura,
                resultado_ica_pesca,
                resultado_ica_industria,
                resultado_ica_recreativo
            )
            VALUES (
                ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
            );
            """,
            (campo_NombreEstudio.get(),
            campo_FechaEstudio.get(),
            campo_LugarEstudio.get(),
            campo_RioEstudio.get(),
            campo_EncargadoEstudio.get(),

```



```

        campo_DescripcionEstudio.get(),
        campo_DBO.get(),
        campo_SDT.get(),
        campo_Turbidez.get(),
        campo_ColiformesFecales.get(),
        campo_Nitratos.get(),
        campo_Fosfatos.get(),
        campo_Temperatura.get(),
        campo_pH.get(),
        campo_OD.get(),
        campo_Aluminio.get(),
        campo_Cadmio.get(),
        campo_Cobre.get(),
        campo_Hierro.get(),
        campo_ManganesoTotal.get(),
        campo_Plomo.get(),
        campo_ZincTotal.get(),
        campo_ICA_AguaPotable.get(),
        campo_ICA_Agricultura.get(),
        campo_ICA_Pesca.get(),
        campo_ICA_Industrial.get(),
        campo_ICA_Recreativo.get(),
        campo_ver_ICA_NSF.get(),
        campo_ver_ICA_Metales.get(),
        campo_ICA_Total.get()
    )
)
connection.commit()
connection.close()
campo_NombreEstudio.delete(0, tk.END)
campo_FechaEstudio.delete(0, tk.END)
campo_LugarEstudio.delete(0, tk.END)
campo_RioEstudio.delete(0, tk.END)
campo_EncargadoEstudio.delete(0, tk.END)
campo_DescripcionEstudio.delete(0, tk.END)
campo_DBO.delete(0, tk.END)
campo_SDT.delete(0, tk.END)
campo_Turbidez.delete(0, tk.END)
campo_ColiformesFecales.delete(0, tk.END)
campo_Nitratos.delete(0, tk.END)
campo_Fosfatos.delete(0, tk.END)
campo_Temperatura.delete(0, tk.END)
campo_pH.delete(0, tk.END)
campo_OD.delete(0, tk.END)
campo_Aluminio.delete(0, tk.END)
campo_Cadmio.delete(0, tk.END)
campo_Cobre.delete(0, tk.END)
campo_Hierro.delete(0, tk.END)
campo_ManganesoTotal.delete(0, tk.END)
campo_Plomo.delete(0, tk.END)
campo_ZincTotal.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='normal')
campo_ICA_Agricultura.config(state='normal')
campo_ICA_Pesca.config(state='normal')
campo_ICA_Industrial.config(state='normal')
campo_ICA_Recreativo.config(state='normal')

```

```

campo_ver_ICA_NSF.config(state='normal')
campo_ver_ICA_Metales.config(state='normal')
campo_ICA_Total.config(state='normal')
campo_ICA_AguaPotable.delete(0, tk.END)
campo_ICA_Agricultura.delete(0, tk.END)
campo_ICA_Pesca.delete(0, tk.END)
campo_ICA_Industrial.delete(0, tk.END)
campo_ICA_Recreativo.delete(0, tk.END)
campo_ver_ICA_NSF.delete(0, tk.END)
campo_ver_ICA_Metales.delete(0, tk.END)
campo_ICA_Total.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='disabled')
campo_ICA_Agricultura.config(state='disabled')
campo_ICA_Pesca.config(state='disabled')
campo_ICA_Industrial.config(state='disabled')
campo_ICA_Recreativo.config(state='disabled')
campo_ver_ICA_NSF.config(state='disabled')
campo_ver_ICA_Metales.config(state='disabled')
campo_ICA_Total.config(state='disabled')
except:
    tk.messagebox.showinfo(title="ERROR", message="No se pudo conectar con la base de
datos")
else:
    tk.messagebox.showinfo(title="ERROR", message="Rellene los campos necesarios...")

def buscar_Info():
    if (validar_Visualizar()):
        connection = sql.connect('programa_ica')
        cursor = connection.cursor()
        try:
            cursor.execute(
                """
                SELECT * FROM info_ica WHERE nombre_estudio=?;
                """
                ,(campo_NombreEstudio.get(),)
            )
            val = cursor.fetchone()
            if (val is not None):
                campo_NombreEstudio.delete(0, tk.END)
                campo_FechaEstudio.delete(0, tk.END)
                campo_LugarEstudio.delete(0, tk.END)
                campo_RioEstudio.delete(0, tk.END)
                campo_EncargadoEstudio.delete(0, tk.END)
                campo_DescripcionEstudio.delete(0, tk.END)
                campo_DBO.delete(0, tk.END)
                campo_SDT.delete(0, tk.END)
                campo_Turbidez.delete(0, tk.END)
                campo_ColiformesFecales.delete(0, tk.END)
                campo_Nitratos.delete(0, tk.END)
                campo_Fosfatos.delete(0, tk.END)
                campo_Temperatura.delete(0, tk.END)
                campo_pH.delete(0, tk.END)
                campo_OD.delete(0, tk.END)
                campo_Aluminio.delete(0, tk.END)
                campo_Cadmio.delete(0, tk.END)
                campo_Cobre.delete(0, tk.END)

```

```

campo_Hierro.delete(0, tk.END)
campo_ManganesoTotal.delete(0, tk.END)
campo_Plomo.delete(0, tk.END)
campo_ZincTotal.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='normal')
campo_ICA_Agricultura.config(state='normal')
campo_ICA_Pesca.config(state='normal')
campo_ICA_Industrial.config(state='normal')
campo_ICA_Recreativo.config(state='normal')
campo_ver_ICA_NSF.config(state='normal')
campo_ver_ICA_Metales.config(state='normal')
campo_ICA_Total.config(state='normal')
campo_ICA_AguaPotable.delete(0, tk.END)
campo_ICA_Agricultura.delete(0, tk.END)
campo_ICA_Pesca.delete(0, tk.END)
campo_ICA_Industrial.delete(0, tk.END)
campo_ICA_Recreativo.delete(0, tk.END)
campo_ver_ICA_NSF.delete(0, tk.END)
campo_ver_ICA_Metales.delete(0, tk.END)
campo_ICA_Total.delete(0, tk.END)
campo_NombreEstudio.insert(0, val[0])
campo_FechaEstudio.insert(0, val[1])
campo_LugarEstudio.insert(0, val[2])
campo_RioEstudio.insert(0, val[3])
campo_EncargadoEstudio.insert(0, val[4])
campo_DescripcionEstudio.insert(0, val[5])
campo_DBO.insert(0, val[6])
campo_SDT.insert(0, val[7])
campo_Turbidez.insert(0, val[8])
campo_ColiformesFecales.insert(0, val[9])
campo_Nitratos.insert(0, val[10])
campo_Fosfatos.insert(0, val[11])
campo_Temperatura.insert(0, val[12])
campo_pH.insert(0, val[13])
campo_OD.insert(0, val[14])
campo_Aluminio.insert(0, val[15])
campo_Cadmio.insert(0, val[16])
campo_Cobre.insert(0, val[17])
campo_Hierro.insert(0, val[18])
campo_ManganesoTotal.insert(0, val[19])
campo_Plomo.insert(0, val[20])
campo_ZincTotal.insert(0, val[21])
campo_ICA_AguaPotable.insert(0, val[22])
campo_ICA_Agricultura.insert(0, val[23])
campo_ICA_Pesca.insert(0, val[24])
campo_ICA_Industrial.insert(0, val[25])
campo_ICA_Recreativo.insert(0, val[26])
campo_ver_ICA_NSF.insert(0, val[27])
campo_ver_ICA_Metales.insert(0, val[28])
campo_ICA_Total.insert(0, val[29])
campo_ICA_AguaPotable.config(state='disabled')
campo_ICA_Pesca.config(state='disabled')
campo_ICA_Agricultura.config(state='disabled')
campo_ICA_Industrial.config(state='disabled')
campo_ICA_Recreativo.config(state='disabled')
campo_ver_ICA_NSF.config(state='disabled')

```

```

        campo_ver_ICA_Metales.config(state='disabled')
        campo_ICA_Total.config(state='disabled')
        connection.close()
    else:
        tk.messagebox.showinfo(title="ERROR", message="No existe el registro en la base de
datos")
    except:
        tk.messagebox.showinfo(title="ERROR", message="No se pudo conectar con la base de
datos")
    else:
        tk.messagebox.showinfo(title="ERROR", message="Rellene el campo: Nombre del estudio")

def actualizar_Info():
    if (validar_Actualizar()):
        try:
            connection = sql.connect('programa_ica')
            cursor = connection.cursor()
            cursor.execute(
                """
                UPDATE info_ica
                SET fecha_estudio = ?,
                lugar_estudio = ?,
                rio_estudio = ?,
                encargado_estudio = ?,
                descripcion_estudio = ?,
                dbo_nsf = ?,
                sdt_nsf = ?,
                turbidez_nsf = ?,
                coliformes_fecales_nsf = ?,
                nitratos_nsf = ?,
                fosfatos_nsf = ?,
                temperatura_nsf = ?,
                ph_nsf = ?,
                od_nsf = ?,
                aluminio_metales = ?,
                cadmio_metales = ?,
                cobre_metales = ?,
                hierro_metales = ?,
                manganeso_total_metales = ?,
                plomo_metales = ?,
                zinc_total_metales = ?,
                valor_ica_nsf = ?,
                valor_ica_metales = ?,
                valor_ica_total = ?,
                resultado_ica_agua_potable = ?,
                resultado_ica_agricultura = ?,
                resultado_ica_pesca = ?,
                resultado_ica_industria = ?,
                resultado_ica_recreativo = ?
                WHERE nombre_estudio = ?;
                """
            , (campo_FechaEstudio.get(),
                campo_LugarEstudio.get(),
                campo_RioEstudio.get(),
                campo_EncargadoEstudio.get(),
                campo_DescripcionEstudio.get(),

```

```

        campo_DBO.get(),
        campo_SDT.get(),
        campo_Turbidez.get(),
        campo_ColiformesFecales.get(),
        campo_Nitratos.get(),
        campo_Fosfatos.get(),
        campo_Temperatura.get(),
        campo_pH.get(),
        campo_OD.get(),
        campo_Aluminio.get(),
        campo_Cadmio.get(),
        campo_Cobre.get(),
        campo_Hierro.get(),
        campo_ManganesoTotal.get(),
        campo_Plomo.get(),
        campo_ZincTotal.get(),
        campo_ICA_AguaPotable.get(),
        campo_ICA_Agricultura.get(),
        campo_ICA_Pesca.get(),
        campo_ICA_Industrial.get(),
        campo_ICA_Recreativo.get(),
        campo_ver_ICA_NSF.get(),
        campo_ver_ICA_Metales.get(),
        campo_ICA_Total.get(),
        campo_NombreEstudio.get()
    )
)
connection.commit()
connection.close()
campo_NombreEstudio.delete(0, tk.END)
campo_FechaEstudio.delete(0, tk.END)
campo_LugarEstudio.delete(0, tk.END)
campo_RioEstudio.delete(0, tk.END)
campo_EncargadoEstudio.delete(0, tk.END)
campo_DescripcionEstudio.delete(0, tk.END)
campo_DBO.delete(0, tk.END)
campo_SDT.delete(0, tk.END)
campo_Turbidez.delete(0, tk.END)
campo_ColiformesFecales.delete(0, tk.END)
campo_Nitratos.delete(0, tk.END)
campo_Fosfatos.delete(0, tk.END)
campo_Temperatura.delete(0, tk.END)
campo_pH.delete(0, tk.END)
campo_OD.delete(0, tk.END)
campo_Aluminio.delete(0, tk.END)
campo_Cadmio.delete(0, tk.END)
campo_Cobre.delete(0, tk.END)
campo_Hierro.delete(0, tk.END)
campo_ManganesoTotal.delete(0, tk.END)
campo_Plomo.delete(0, tk.END)
campo_ZincTotal.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='normal')
campo_ICA_Agricultura.config(state='normal')
campo_ICA_Pesca.config(state='normal')
campo_ICA_Industrial.config(state='normal')
campo_ICA_Recreativo.config(state='normal')

```

```

campo_ver_ICA_NSF.config(state='normal')
campo_ver_ICA_Metales.config(state='normal')
campo_ICA_Total.config(state='normal')
campo_ICA_AguaPotable.delete(0, tk.END)
campo_ICA_Agricultura.delete(0, tk.END)
campo_ICA_Pesca.delete(0, tk.END)
campo_ICA_Industrial.delete(0, tk.END)
campo_ICA_Recreativo.delete(0, tk.END)
campo_ver_ICA_NSF.delete(0, tk.END)
campo_ver_ICA_Metales.delete(0, tk.END)
campo_ICA_Total.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='disabled')
campo_ICA_Agricultura.config(state='disabled')
campo_ICA_Pesca.config(state='disabled')
campo_ICA_Industrial.config(state='disabled')
campo_ICA_Recreativo.config(state='disabled')
campo_ver_ICA_NSF.config(state='disabled')
campo_ver_ICA_Metales.config(state='disabled')
campo_ICA_Total.config(state='disabled')
except:
    tk.messagebox.showinfo(title="ERROR", message="No se pudo conectar con la base de
datos")
else:
    tk.messagebox.showinfo(title="ERROR", message="Rellene los campos necesarios...")

def eliminar_Info():
    if (validar_Eliminar()):
        connection = sql.connect('programa_ica')
        cursor = connection.cursor()
        try:
            cursor.execute(
                """
                DELETE FROM info_ica WHERE nombre_estudio=?;
                """
            )
            connection.commit()
            connection.close()
            campo_NombreEstudio.delete(0, tk.END)
            campo_FechaEstudio.delete(0, tk.END)
            campo_LugarEstudio.delete(0, tk.END)
            campo_RioEstudio.delete(0, tk.END)
            campo_EncargadoEstudio.delete(0, tk.END)
            campo_DescripcionEstudio.delete(0, tk.END)
            campo_DBO.delete(0, tk.END)
            campo_SDT.delete(0, tk.END)
            campo_Turbidez.delete(0, tk.END)
            campo_ColiformesFecales.delete(0, tk.END)
            campo_Nitratos.delete(0, tk.END)
            campo_Fosfatos.delete(0, tk.END)
            campo_Temperatura.delete(0, tk.END)
            campo_pH.delete(0, tk.END)
            campo_OD.delete(0, tk.END)

```

```

campo_Aluminio.delete(0, tk.END)
campo_Cadmio.delete(0, tk.END)
campo_Cobre.delete(0, tk.END)
campo_Hierro.delete(0, tk.END)
campo_ManganesoTotal.delete(0, tk.END)
campo_Plomo.delete(0, tk.END)
campo_ZincTotal.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='normal')
campo_ICA_Agricultura.config(state='normal')
campo_ICA_Pesca.config(state='normal')
campo_ICA_Industrial.config(state='normal')
campo_ICA_Recreativo.config(state='normal')
campo_ver_ICA_NSF.config(state='normal')
campo_ver_ICA_Metales.config(state='normal')
campo_ICA_Total.config(state='normal')
campo_ICA_AguaPotable.delete(0, tk.END)
campo_ICA_Agricultura.delete(0, tk.END)
campo_ICA_Pesca.delete(0, tk.END)
campo_ICA_Industrial.delete(0, tk.END)
campo_ICA_Recreativo.delete(0, tk.END)
campo_ver_ICA_NSF.delete(0, tk.END)
campo_ver_ICA_Metales.delete(0, tk.END)
campo_ICA_Total.delete(0, tk.END)
campo_ICA_AguaPotable.config(state='disabled')
campo_ICA_Agricultura.config(state='disabled')
campo_ICA_Pesca.config(state='disabled')
campo_ICA_Industrial.config(state='disabled')
campo_ICA_Recreativo.config(state='disabled')
campo_ver_ICA_NSF.config(state='disabled')
campo_ver_ICA_Metales.config(state='disabled')
campo_ICA_Total.config(state='disabled')
except:
    tk.messagebox.showinfo(title="ERROR", message="No se pudo conectar con la base de
datos")
else:
    tk.messagebox.showinfo(title="ERROR", message="Rellene el campo: Nombre del estudio")

```

```

## CREACION DE LA INTERFAZ

```

```

raiz = tk.Tk()
raiz.title("Programa ICA")
raiz.resizable(False, False)

```

```

## DEFINICION DE VARIABLES PARA MANIPULAR LA INTERFAZ

```

```

# Variable de entrada: Informacion
input_var_NombreEstudio = tk.StringVar()
input_var_FechaEstudio = tk.StringVar()
input_var_LugarEstudio = tk.StringVar()
input_var_RioEstudio = tk.StringVar()
input_var_EncargadoEstudio = tk.StringVar()
input_var_DescripcionEstudio = tk.StringVar()

```

```

# Variables de entrada: ICA NSF
input_var_DBO = tk.DoubleVar()
input_var_SDT = tk.DoubleVar()
input_var_Turbidez = tk.DoubleVar()

```

```

input_var_ColiformesFecales = tk.DoubleVar()
input_var_Nitratos = tk.DoubleVar()
input_var_Fosfatos = tk.DoubleVar()
input_var_Temperatura = tk.DoubleVar()
input_var_pH = tk.DoubleVar()
input_var_OD = tk.DoubleVar()

# Variables de entrada: ICA Metales
input_var_Aluminio = tk.DoubleVar()
input_var_Cadmio = tk.DoubleVar()
input_var_Cobre = tk.DoubleVar()
input_var_Hierro = tk.DoubleVar()
input_var_ManganesoTotal = tk.DoubleVar()
input_var_Plomo = tk.DoubleVar()
input_var_ZincTotal = tk.DoubleVar()

# Variables de salida: Calculos
see_var_ICA_NSF = tk.DoubleVar()
see_var_ICA_Metales = tk.DoubleVar()

# Variables de salida: Resultados
see_var_ICA_Total = tk.DoubleVar()
see_var_ICA_AguaPotable = tk.StringVar()
see_var_ICA_Agricultura = tk.StringVar()
see_var_ICA_Pesca = tk.StringVar()
see_var_ICA_Industrial = tk.StringVar()
see_var_ICA_Recreativo = tk.StringVar()

## DEFINICION DE LA INTERFAZ
# Creacion de Subventanas
ventana_Titulo_Principal = tk.Frame(raiz, pady=5)
ventana_Titulo_Principal.grid(row=0, column=0, columnspan=14)

etiqueta_Titulo_Principal = tk.Label(ventana_Titulo_Principal, text="ESCUELA POLITÉCNICA
NACIONAL - CÁLCULO DE ICA ", font="helvetica 20")
etiqueta_Titulo_Principal.grid(row=1, column=1)

etiqueta_Titulo_Principal = tk.Label(ventana_Titulo_Principal, text="Creado por Liseth Collaguazo",
font="helvetica 13")
etiqueta_Titulo_Principal.grid(row=2, column=2)

ventana_Etiqueta_Informacion = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Etiqueta_Informacion.config(bg="white")
ventana_Etiqueta_Informacion.grid(row=1, column=1)

etiqueta_Informacion = tk.Label(ventana_Etiqueta_Informacion, text="INFORMACION ")
etiqueta_Informacion.config(bg="white")
etiqueta_Informacion.config(justify='left')
etiqueta_Informacion.grid(row=1, column=0)

ventana_Informacion = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Informacion.config(bg="white")
ventana_Informacion.grid(row=2, column=0, rowspan=14, columnspan=3)

ventana_Etiqueta_Datos = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Etiqueta_Datos.config(bg="white")

```



```

ventana_Etiqueta_Datos.grid(row=1, column=5, columnspan=3)

etiqueta_Datos = tk.Label(ventana_Etiqueta_Datos, text="DATOS ")
etiqueta_Datos.config(bg="white")
etiqueta_Datos.config(justify='left')
etiqueta_Datos.grid(row=1, column=3, columnspan=3)

ventana_Datos = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Datos.config(bg="white")
ventana_Datos.grid(row=2, column=3, rowspan=14, columnspan=7)

ventana_Etiqueta_Resultados = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Etiqueta_Resultados.config(bg="white")
ventana_Etiqueta_Resultados.grid(row=1, column=11, columnspan=2)

etiqueta_Resultados = tk.Label(ventana_Etiqueta_Resultados, text="RESULTADOS ")
etiqueta_Resultados.config(bg="white")
etiqueta_Resultados.config(justify='left')
etiqueta_Resultados.grid(row=1, column=10, columnspan=4)

ventana_Resultados = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Resultados.config(bg="white")
ventana_Resultados.grid(row=2, column=10, rowspan=14, columnspan=4)

ventana_Botones = tk.Frame(raiz, highlightbackground='black', highlightthickness=1)
ventana_Botones.config(bg="white")
ventana_Botones.grid(row=16, column=0, rowspan=3, columnspan=14)

# Elemenos de la primera columna
blank_1_0 = tk.Frame(ventana_Informacion)
blank_1_0.config(width="50", height="30")
blank_1_0.config(bg="white")
blank_1_0.grid(row=1, column=0)

blank_2_0 = tk.Frame(ventana_Informacion)
blank_2_0.config(width="50", height="30")
blank_2_0.config(bg="white")
blank_2_0.grid(row=2, column=0)

blank_3_0 = tk.Frame(ventana_Informacion)
blank_3_0.config(width="50", height="30")
blank_3_0.config(bg="white")
blank_3_0.grid(row=3, column=0)

blank_4_0 = tk.Frame(ventana_Informacion)
blank_4_0.config(width="50", height="30")
blank_4_0.config(bg="white")
blank_4_0.grid(row=4, column=0)

blank_5_0 = tk.Frame(ventana_Informacion)
blank_5_0.config(width="50", height="30")
blank_5_0.config(bg="white")
blank_5_0.grid(row=5, column=0)

blank_6_0 = tk.Frame(ventana_Informacion)
blank_6_0.config(width="50", height="30")

```

```

blank_6_0.config(bg="white")
blank_6_0.grid(row=6, column=0)

blank_7_0 = tk.Frame(ventana_Informacion)
blank_7_0.config(width="50", height="30")
blank_7_0.config(bg="white")
blank_7_0.grid(row=7, column=0)

blank_8_0 = tk.Frame(ventana_Informacion)
blank_8_0.config(width="50", height="30")
blank_8_0.config(bg="white")
blank_8_0.grid(row=8, column=0)

blank_9_0 = tk.Frame(ventana_Informacion)
blank_9_0.config(width="50", height="30")
blank_9_0.config(bg="white")
blank_9_0.grid(row=9, column=0)

blank_10_0 = tk.Frame(ventana_Informacion)
blank_10_0.config(width="50", height="30")
blank_10_0.config(bg="white")
blank_10_0.grid(row=10, column=0)

blank_11_0 = tk.Frame(ventana_Informacion)
blank_11_0.config(width="50", height="30")
blank_11_0.config(bg="white")
blank_11_0.grid(row=11, column=0)

blank_12_0 = tk.Frame(ventana_Informacion)
blank_12_0.config(width="50", height="30")
blank_12_0.config(bg="white")
blank_12_0.grid(row=12, column=0)

blank_13_0 = tk.Frame(ventana_Informacion)
blank_13_0.config(width="50", height="30")
blank_13_0.config(bg="white")
blank_13_0.grid(row=13, column=0)

# Elementos de la segunda columna
etiqueta_NombreEstudio = tk.Label(ventana_Informacion, text="Nombre del estudio: ")
etiqueta_NombreEstudio.config(width="20")
etiqueta_NombreEstudio.config(justify='left')
etiqueta_NombreEstudio.config(bg="white")
etiqueta_NombreEstudio.grid(row=0, column=1)

campo_NombreEstudio = tk.Entry(ventana_Informacion, textvariable=input_var_NombreEstudio)
campo_NombreEstudio.config(width="25")
campo_NombreEstudio.config(bg="white")
campo_NombreEstudio.grid(row=1, column=1)

etiqueta_FechaEstudio = tk.Label(ventana_Informacion, text="Fecha del estudio: ")
etiqueta_FechaEstudio.config(width="20")
etiqueta_FechaEstudio.config(justify='left')
etiqueta_FechaEstudio.config(bg="white")
etiqueta_FechaEstudio.grid(row=2, column=1)

```

```
campo_FechaEstudio = tk.Entry(ventana_Informacion, textvariable=input_var_FechaEstudio)
campo_FechaEstudio.config(width="25")
campo_FechaEstudio.config(bg="white")
campo_FechaEstudio.grid(row=3, column=1)
```

```
etiqueta_LugarEstudio = tk.Label(ventana_Informacion, text="Lugar del estudio: ")
etiqueta_LugarEstudio.config(width="20")
etiqueta_LugarEstudio.config(justify='left')
etiqueta_LugarEstudio.config(bg="white")
etiqueta_LugarEstudio.grid(row=4, column=1)
```

```
campo_LugarEstudio = tk.Entry(ventana_Informacion, textvariable=input_var_LugarEstudio)
campo_LugarEstudio.config(width="25")
campo_LugarEstudio.config(bg="white")
campo_LugarEstudio.grid(row=5, column=1)
```

```
etiqueta_RioEstudio = tk.Label(ventana_Informacion, text="Río estudiado: ")
etiqueta_RioEstudio.config(width="20")
etiqueta_RioEstudio.config(justify='left')
etiqueta_RioEstudio.config(bg="white")
etiqueta_RioEstudio.grid(row=6, column=1)
```

```
campo_RioEstudio = tk.Entry(ventana_Informacion, textvariable=input_var_RioEstudio)
campo_RioEstudio.config(width="25")
campo_RioEstudio.config(bg="white")
campo_RioEstudio.grid(row=7, column=1)
```

```
etiqueta_EncargadoEstudio = tk.Label(ventana_Informacion, text="Encargado del estudio: ")
etiqueta_EncargadoEstudio.config(width="20")
etiqueta_EncargadoEstudio.config(justify='left')
etiqueta_EncargadoEstudio.config(bg="white")
etiqueta_EncargadoEstudio.grid(row=8, column=1)
```

```
campo_EncargadoEstudio = tk.Entry(ventana_Informacion,
textvariable=input_var_EncargadoEstudio)
campo_EncargadoEstudio.config(width="25")
campo_EncargadoEstudio.config(bg="white")
campo_EncargadoEstudio.grid(row=9, column=1)
```

```
etiqueta_DescripcionEstudio = tk.Label(ventana_Informacion, text="Descripción del estudio: ")
etiqueta_DescripcionEstudio.config(width="20")
etiqueta_DescripcionEstudio.config(justify='left')
etiqueta_DescripcionEstudio.config(bg="white")
etiqueta_DescripcionEstudio.grid(row=10, column=1)
```

```
campo_DescripcionEstudio = tk.Entry(ventana_Informacion,
textvariable=input_var_DescripcionEstudio)
campo_DescripcionEstudio.config(width="25")
campo_DescripcionEstudio.config(bg="white")
campo_DescripcionEstudio.grid(row=11, column=1)
```

```
# Elementos de la tercera columna
blank_1_3 = tk.Frame(ventana_Informacion)
blank_1_3.config(width="50", height="30")
blank_1_3.config(bg="white")
blank_1_3.grid(row=0, column=3)
```

```
blank_2_3 = tk.Frame(ventana_Informacion)
blank_2_3.config(width="50", height="30")
blank_2_3.config(bg="white")
blank_2_3.grid(row=1, column=3)

blank_3_3 = tk.Frame(ventana_Informacion)
blank_3_3.config(width="50", height="30")
blank_3_3.config(bg="white")
blank_3_3.grid(row=2, column=3)

blank_4_3 = tk.Frame(ventana_Informacion)
blank_4_3.config(width="50", height="30")
blank_4_3.config(bg="white")
blank_4_3.grid(row=3, column=3)

blank_5_3 = tk.Frame(ventana_Informacion)
blank_5_3.config(width="50", height="30")
blank_5_3.config(bg="white")
blank_5_3.grid(row=4, column=3)

blank_6_3 = tk.Frame(ventana_Informacion)
blank_6_3.config(width="50", height="30")
blank_6_3.config(bg="white")
blank_6_3.grid(row=5, column=3)

blank_7_3 = tk.Frame(ventana_Informacion)
blank_7_3.config(width="50", height="30")
blank_7_3.config(bg="white")
blank_7_3.grid(row=6, column=3)

blank_8_3 = tk.Frame(ventana_Informacion)
blank_8_3.config(width="50", height="30")
blank_8_3.config(bg="white")
blank_8_3.grid(row=7, column=3)

blank_9_3 = tk.Frame(ventana_Informacion)
blank_9_3.config(width="50", height="30")
blank_9_3.config(bg="white")
blank_9_3.grid(row=8, column=3)

blank_10_3 = tk.Frame(ventana_Informacion)
blank_10_3.config(width="50", height="30")
blank_10_3.config(bg="white")
blank_10_3.grid(row=9, column=3)

blank_11_3 = tk.Frame(ventana_Informacion)
blank_11_3.config(width="50", height="30")
blank_11_3.config(bg="white")
blank_11_3.grid(row=10, column=3)

blank_12_3 = tk.Frame(ventana_Informacion)
blank_12_3.config(width="50", height="30")
blank_12_3.config(bg="white")
blank_12_3.grid(row=11, column=3)
```

```
blank_13_3 = tk.Frame(ventana_Informacion)
blank_13_3.config(width="50", height="30")
blank_13_3.config(bg="white")
blank_13_3.grid(row=12, column=3)
```

```
# Elemenos de la cuarta columna
blank_1_4 = tk.Frame(ventana_Datos)
blank_1_4.config(width="50", height="30")
blank_1_4.config(bg="white")
blank_1_4.grid(row=0, column=0)
```

```
blank_2_4 = tk.Frame(ventana_Datos)
blank_2_4.config(width="50", height="30")
blank_2_4.config(bg="white")
blank_2_4.grid(row=1, column=0)
```

```
blank_3_4 = tk.Frame(ventana_Datos)
blank_3_4.config(width="50", height="30")
blank_3_4.config(bg="white")
blank_3_4.grid(row=2, column=0)
```

```
blank_4_4 = tk.Frame(ventana_Datos)
blank_4_4.config(width="50", height="30")
blank_4_4.config(bg="white")
blank_4_4.grid(row=3, column=0)
```

```
blank_5_4 = tk.Frame(ventana_Datos)
blank_5_4.config(width="50", height="30")
blank_5_4.config(bg="white")
blank_5_4.grid(row=4, column=0)
```

```
blank_6_4 = tk.Frame(ventana_Datos)
blank_6_4.config(width="50", height="30")
blank_6_4.config(bg="white")
blank_6_4.grid(row=5, column=0)
```

```
blank_7_4 = tk.Frame(ventana_Datos)
blank_7_4.config(width="50", height="30")
blank_7_4.config(bg="white")
blank_7_4.grid(row=6, column=0)
```

```
blank_8_4 = tk.Frame(ventana_Datos)
blank_8_4.config(width="50", height="30")
blank_8_4.config(bg="white")
blank_8_4.grid(row=7, column=0)
```

```
blank_9_4 = tk.Frame(ventana_Datos)
blank_9_4.config(width="50", height="30")
blank_9_4.config(bg="white")
blank_9_4.grid(row=8, column=0)
```

```
blank_10_4 = tk.Frame(ventana_Datos)
blank_10_4.config(width="50", height="30")
blank_10_4.config(bg="white")
blank_10_4.grid(row=9, column=0)
```

```

blank_11_4 = tk.Frame(ventana_Datos)
blank_11_4.config(width="50", height="30")
blank_11_4.config(bg="white")
blank_11_4.grid(row=10, column=0)

blank_12_4 = tk.Frame(ventana_Datos)
blank_12_4.config(width="50", height="30")
blank_12_4.config(bg="white")
blank_12_4.grid(row=11, column=0)

blank_13_4 = tk.Frame(ventana_Datos)
blank_13_4.config(width="50", height="30")
blank_13_4.config(bg="white")
blank_13_4.grid(row=12, column=0)

blank_14_4 = tk.Frame(ventana_Datos)
blank_14_4.config(width="50", height="30")
blank_14_4.config(bg="white")
blank_14_4.grid(row=13, column=0)

# Elementos de la quinta columna
etiqueta_ICA_NSF = tk.Label(ventana_Datos, text="ICA - NSF")
etiqueta_ICA_NSF.config(width="20")
etiqueta_ICA_NSF.config(bg="white")
etiqueta_ICA_NSF.grid(row=0, column=1, columnspan=2)

etiqueta_DBO = tk.Label(ventana_Datos, text="DBO (mg/L): ")
etiqueta_DBO.config(width="20")
etiqueta_DBO.config(bg="white")
etiqueta_DBO.grid(row=1, column=1)

etiqueta_SDT = tk.Label(ventana_Datos, text="SDT (mg/L): ")
etiqueta_SDT.config(width="20")
etiqueta_SDT.config(bg="white")
etiqueta_SDT.grid(row=2, column=1)

etiqueta_Turbidez = tk.Label(ventana_Datos, text="Turbidez (NTU): ")
etiqueta_Turbidez.config(width="20")
etiqueta_Turbidez.config(bg="white")
etiqueta_Turbidez.grid(row=3, column=1)

etiqueta_ColiformesFecales = tk.Label(ventana_Datos, text="C. Fecales (NMP/100 mL): ")
etiqueta_ColiformesFecales.config(width="20")
etiqueta_ColiformesFecales.config(bg="white")
etiqueta_ColiformesFecales.grid(row=4, column=1)

etiqueta_Nitratos = tk.Label(ventana_Datos, text="Nitratos ( mg/L): ")
etiqueta_Nitratos.config(width="20")
etiqueta_Nitratos.config(bg="white")
etiqueta_Nitratos.grid(row=5, column=1)

etiqueta_Fosfatos = tk.Label(ventana_Datos, text="Fosfatos (mg/ L): ")
etiqueta_Fosfatos.config(width="20")
etiqueta_Fosfatos.config(bg="white")
etiqueta_Fosfatos.grid(row=6, column=1)

```

```

etiqueta_Temperatura = tk.Label(ventana_Datos, text="Temperatura (°C): ")
etiqueta_Temperatura.config(width="20")
etiqueta_Temperatura.config(bg="white")
etiqueta_Temperatura.grid(row=7, column=1)

etiqueta_pH = tk.Label(ventana_Datos, text="pH (Unidades): ")
etiqueta_pH.config(width="20")
etiqueta_pH.config(bg="white")
etiqueta_pH.grid(row=8, column=1)

etiqueta_OD = tk.Label(ventana_Datos, text="OD (%Saturación): ")
etiqueta_OD.config(width="20")
etiqueta_OD.config(bg="white")
etiqueta_OD.grid(row=9, column=1)

boton_Calcular_ICA_NSF = tk.Button(ventana_Datos, text="CALCULAR ICA", font="Helvetica 9
bold", command=calcular_ICA_NSF)
boton_Calcular_ICA_NSF.config(width="20")
boton_Calcular_ICA_NSF.config(bg="dark turquoise")
boton_Calcular_ICA_NSF.grid(row=11, column=1, columnspan=2)

etiqueta_ICA_NSF = tk.Label(ventana_Datos, text="ICA - NSF: ", font="helvetica 11 bold")
etiqueta_ICA_NSF.config(width="20")
etiqueta_ICA_NSF.config(bg="white")
etiqueta_ICA_NSF.grid(row=12, column=1)

# Elemenos de la sexta columna
campo_DBO = tk.Entry(ventana_Datos, textvariable=input_var_DBO)
campo_DBO.config(width="10")
campo_DBO.config(bg="white")
campo_DBO.grid(row=1, column=2)

campo_SDT = tk.Entry(ventana_Datos, textvariable=input_var_SDT)
campo_SDT.config(width="10")
campo_SDT.config(bg="white")
campo_SDT.grid(row=2, column=2)

campo_Turbidez = tk.Entry(ventana_Datos, textvariable=input_var_Turbidez)
campo_Turbidez.config(width="10")
campo_Turbidez.config(bg="white")
campo_Turbidez.grid(row=3, column=2)

campo_ColiformesFecales = tk.Entry(ventana_Datos, textvariable=input_var_ColiformesFecales)
campo_ColiformesFecales.config(width="10")
campo_ColiformesFecales.config(bg="white")
campo_ColiformesFecales.grid(row=4, column=2)

campo_Nitratos = tk.Entry(ventana_Datos, textvariable=input_var_Nitratos)
campo_Nitratos.config(width="10")
campo_Nitratos.config(bg="white")
campo_Nitratos.grid(row=5, column=2)

campo_Fosfatos = tk.Entry(ventana_Datos, textvariable=input_var_Fosfatos)
campo_Fosfatos.config(width="10")
campo_Fosfatos.config(bg="white")
campo_Fosfatos.grid(row=6, column=2)

```

```

campo_Temperatura = tk.Entry(ventana_Datos, textvariable=input_var_Temperatura)
campo_Temperatura.config(width="10")
campo_Temperatura.config(bg="white")
campo_Temperatura.grid(row=7, column=2)

campo_pH = tk.Entry(ventana_Datos, textvariable=input_var_pH)
campo_pH.config(width="10")
campo_pH.config(bg="white")
campo_pH.grid(row=8, column=2)

campo_OD = tk.Entry(ventana_Datos, textvariable=input_var_OD)
campo_OD.config(width="10")
campo_OD.config(bg="white")
campo_OD.grid(row=9, column=2)

campo_ver_ICA_NSF = tk.Entry(ventana_Datos, textvariable=see_var_ICA_NSF, font="helvetica
11")
campo_ver_ICA_NSF.config(width="10")
campo_ver_ICA_NSF.config(bg="white")
campo_ver_ICA_NSF.grid(row=12, column=2)

# Elementos de la séptima columna
blank_1_6 = tk.Frame(ventana_Datos)
blank_1_6.config(width="50", height="30")
blank_1_6.config(bg="white")
blank_1_6.grid(row=0, column=3)

blank_2_6 = tk.Frame(ventana_Datos)
blank_2_6.config(width="50", height="30")
blank_2_6.config(bg="white")
blank_2_6.grid(row=1, column=3)

blank_3_6 = tk.Frame(ventana_Datos)
blank_3_6.config(width="50", height="30")
blank_3_6.config(bg="white")
blank_3_6.grid(row=2, column=3)

blank_4_6 = tk.Frame(ventana_Datos)
blank_4_6.config(width="50", height="30")
blank_4_6.config(bg="white")
blank_4_6.grid(row=3, column=3)

blank_5_6 = tk.Frame(ventana_Datos)
blank_5_6.config(width="50", height="30")
blank_5_6.config(bg="white")
blank_5_6.grid(row=4, column=3)

blank_6_6 = tk.Frame(ventana_Datos)
blank_6_6.config(width="50", height="30")
blank_6_6.config(bg="white")
blank_6_6.grid(row=5, column=3)

blank_7_6 = tk.Frame(ventana_Datos)
blank_7_6.config(width="50", height="30")
blank_7_6.config(bg="white")

```



```

blank_7_6.grid(row=6, column=3)

blank_8_6 = tk.Frame(ventana_Datos)
blank_8_6.config(width="50", height="30")
blank_8_6.config(bg="white")
blank_8_6.grid(row=7, column=3)

blank_9_6 = tk.Frame(ventana_Datos)
blank_9_6.config(width="50", height="30")
blank_9_6.config(bg="white")
blank_9_6.grid(row=8, column=3)

blank_10_6 = tk.Frame(ventana_Datos)
blank_10_6.config(width="50", height="30")
blank_10_6.config(bg="white")
blank_10_6.grid(row=9, column=3)

blank_11_6 = tk.Frame(ventana_Datos)
blank_11_6.config(width="50", height="30")
blank_11_6.config(bg="white")
blank_11_6.grid(row=10, column=3)

blank_12_6 = tk.Frame(ventana_Datos)
blank_12_6.config(width="50", height="30")
blank_12_6.config(bg="white")
blank_12_6.grid(row=11, column=3)

blank_13_6 = tk.Frame(ventana_Datos)
blank_13_6.config(width="50", height="30")
blank_13_6.config(bg="white")
blank_13_6.grid(row=12, column=3)

blank_14_6 = tk.Frame(ventana_Datos)
blank_14_6.config(width="50", height="30")
blank_14_6.config(bg="white")
blank_14_6.grid(row=13, column=3)

# Elementos de la octava columna
etiqueta_ICA_Metales = tk.Label(ventana_Datos, text="ICA - Metales")
etiqueta_ICA_Metales.config(width="20")
etiqueta_ICA_Metales.config(bg="white")
etiqueta_ICA_Metales.grid(row=0, column=4, columnspan=3)

etiqueta_Aluminio = tk.Label(ventana_Datos, text="Aluminio (mg/L): ")
etiqueta_Aluminio.config(width="20")
etiqueta_Aluminio.config(bg="white")
etiqueta_Aluminio.grid(row=1, column=4)

etiqueta_Cadmio = tk.Label(ventana_Datos, text="Cadmio (mg/L): ")
etiqueta_Cadmio.config(width="20")
etiqueta_Cadmio.config(bg="white")
etiqueta_Cadmio.grid(row=2, column=4)

etiqueta_Cobre = tk.Label(ventana_Datos, text="Cobre(mg/L): ")
etiqueta_Cobre.config(width="20")
etiqueta_Cobre.config(bg="white")

```

```

etiqueta_Cobre.grid(row=3, column=4)

etiqueta_Hierro = tk.Label(ventana_Datos, text="Hierro(mg/L): ")
etiqueta_Hierro.config(width="20")
etiqueta_Hierro.config(bg="white")
etiqueta_Hierro.grid(row=4, column=4)

etiqueta_ManganesoTotal = tk.Label(ventana_Datos, text="Manganeso(mg/L): ")
etiqueta_ManganesoTotal.config(width="20")
etiqueta_ManganesoTotal.config(bg="white")
etiqueta_ManganesoTotal.grid(row=5, column=4)

etiqueta_Plomo = tk.Label(ventana_Datos, text="Plomo(mg/L): ")
etiqueta_Plomo.config(width="20")
etiqueta_Plomo.config(bg="white")
etiqueta_Plomo.grid(row=6, column=4)

etiqueta_ZincTotal = tk.Label(ventana_Datos, text="Zinc (mg/L): ")
etiqueta_ZincTotal.config(width="20")
etiqueta_ZincTotal.config(bg="white")
etiqueta_ZincTotal.grid(row=7, column=4)

boton_Calcular_ICA_Metales = tk.Button(ventana_Datos, text="CALCULAR ICA", font="Helvetica 9
bold", command=calcular_ICA_Metales)
boton_Calcular_ICA_Metales.config(width="20")
boton_Calcular_ICA_Metales.config(bg="dark turquoise")
boton_Calcular_ICA_Metales.grid(row=11, column=4, columnspan=2)

etiqueta_ICA_Metales = tk.Label(ventana_Datos, text="ICA - Metales: ", font="helvetica 11 bold")
etiqueta_ICA_Metales.config(width="20")
etiqueta_ICA_Metales.config(bg="white")
etiqueta_ICA_Metales.grid(row=12, column=4)

# Elemenos de la novena columna
campo_Aluminio = tk.Entry(ventana_Datos, textvariable=input_var_Aluminio)
campo_Aluminio.config(width="10")
campo_Aluminio.config(bg="white")
campo_Aluminio.grid(row=1, column=5)

campo_Cadmio = tk.Entry(ventana_Datos, textvariable=input_var_Cadmio)
campo_Cadmio.config(width="10")
campo_Cadmio.config(bg="white")
campo_Cadmio.grid(row=2, column=5)

campo_Cobre = tk.Entry(ventana_Datos, textvariable=input_var_Cobre)
campo_Cobre.config(width="10")
campo_Cobre.config(bg="white")
campo_Cobre.grid(row=3, column=5)

campo_Hierro = tk.Entry(ventana_Datos, textvariable=input_var_Hierro)
campo_Hierro.config(width="10")
campo_Hierro.config(bg="white")
campo_Hierro.grid(row=4, column=5)

campo_ManganesoTotal = tk.Entry(ventana_Datos, textvariable=input_var_ManganesoTotal)
campo_ManganesoTotal.config(width="10")

```

```

campo_ManganesoTotal.config(bg="white")
campo_ManganesoTotal.grid(row=5, column=5)

campo_Plomo = tk.Entry(ventana_Datos, textvariable=input_var_Plomo)
campo_Plomo.config(width="10")
campo_Plomo.config(bg="white")
campo_Plomo.grid(row=6, column=5)

campo_ZincTotal = tk.Entry(ventana_Datos, textvariable=input_var_ZincTotal)
campo_ZincTotal.config(width="10")
campo_ZincTotal.config(bg="white")
campo_ZincTotal.grid(row=7, column=5)

campo_ver_ICA_Metales = tk.Entry(ventana_Datos, textvariable=see_var_ICA_Metales,
font="helvetica 11")
campo_ver_ICA_Metales.config(width="10")
campo_ver_ICA_Metales.config(bg="white")
campo_ver_ICA_Metales.grid(row=12, column=5)

# Elemenos de la decima columna
blank_1_9 = tk.Frame(ventana_Datos)
blank_1_9.config(width="50", height="30")
blank_1_9.config(bg="white")
blank_1_9.grid(row=0, column=6)

blank_2_9 = tk.Frame(ventana_Datos)
blank_2_9.config(width="50", height="30")
blank_2_9.config(bg="white")
blank_2_9.grid(row=1, column=6)

blank_3_9 = tk.Frame(ventana_Datos)
blank_3_9.config(width="50", height="30")
blank_3_9.config(bg="white")
blank_3_9.grid(row=2, column=6)

blank_4_9 = tk.Frame(ventana_Datos)
blank_4_9.config(width="50", height="30")
blank_4_9.config(bg="white")
blank_4_9.grid(row=3, column=6)

blank_5_9 = tk.Frame(ventana_Datos)
blank_5_9.config(width="50", height="30")
blank_5_9.config(bg="white")
blank_5_9.grid(row=4, column=6)

blank_6_9 = tk.Frame(ventana_Datos)
blank_6_9.config(width="50", height="30")
blank_6_9.config(bg="white")
blank_6_9.grid(row=5, column=6)

blank_7_9 = tk.Frame(ventana_Datos)
blank_7_9.config(width="50", height="30")
blank_7_9.config(bg="white")
blank_7_9.grid(row=6, column=6)

blank_8_9 = tk.Frame(ventana_Datos)

```

```

blank_8_9.config(width="50", height="30")
blank_8_9.config(bg="white")
blank_8_9.grid(row=7, column=6)

blank_9_9 = tk.Frame(ventana_Datos)
blank_9_9.config(width="50", height="30")
blank_9_9.config(bg="white")
blank_9_9.grid(row=8, column=6)

blank_10_9 = tk.Frame(ventana_Datos)
blank_10_9.config(width="50", height="30")
blank_10_9.config(bg="white")
blank_10_9.grid(row=9, column=6)

blank_11_9 = tk.Frame(ventana_Datos)
blank_11_9.config(width="50", height="30")
blank_11_9.config(bg="white")
blank_11_9.grid(row=10, column=6)

blank_12_9 = tk.Frame(ventana_Datos)
blank_12_9.config(width="50", height="30")
blank_12_9.config(bg="white")
blank_12_9.grid(row=11, column=6)

blank_13_9 = tk.Frame(ventana_Datos)
blank_13_9.config(width="50", height="30")
blank_13_9.config(bg="white")
blank_13_9.grid(row=12, column=6)

blank_14_9 = tk.Frame(ventana_Datos)
blank_14_9.config(width="50", height="30")
blank_14_9.config(bg="white")
blank_14_9.grid(row=13, column=6)

# Elemenos de la decima-primer columna
blank_0_10 = tk.Frame(ventana_Resultados)
blank_0_10.config(width="30", height="30")
blank_0_10.config(bg="white")
blank_0_10.grid(row=0, column=0)

blank_1_10 = tk.Frame(ventana_Resultados)
blank_1_10.config(width="30", height="30")
blank_1_10.config(bg="white")
blank_1_10.grid(row=1, column=0)

blank_2_10 = tk.Frame(ventana_Resultados)
blank_2_10.config(width="30", height="30")
blank_2_10.config(bg="white")
blank_2_10.grid(row=2, column=0)

blank_3_10 = tk.Frame(ventana_Resultados)
blank_3_10.config(width="30", height="30")
blank_3_10.config(bg="white")
blank_3_10.grid(row=3, column=0)

blank_4_10 = tk.Frame(ventana_Resultados)

```

```

blank_4_10.config(width="30", height="30")
blank_4_10.config(bg="white")
blank_4_10.grid(row=4, column=0)

blank_5_10 = tk.Frame(ventana_Resultados)
blank_5_10.config(width="30", height="30")
blank_5_10.config(bg="white")
blank_5_10.grid(row=5, column=0)

blank_6_10 = tk.Frame(ventana_Resultados)
blank_6_10.config(width="30", height="30")
blank_6_10.config(bg="white")
blank_6_10.grid(row=6, column=0)

blank_7_10 = tk.Frame(ventana_Resultados)
blank_7_10.config(width="30", height="30")
blank_7_10.config(bg="white")
blank_7_10.grid(row=7, column=0)

blank_8_10 = tk.Frame(ventana_Resultados)
blank_8_10.config(width="30", height="30")
blank_8_10.config(bg="white")
blank_8_10.grid(row=8, column=0)

blank_9_10 = tk.Frame(ventana_Resultados)
blank_9_10.config(width="30", height="30")
blank_9_10.config(bg="white")
blank_9_10.grid(row=9, column=0)

blank_10_10 = tk.Frame(ventana_Resultados)
blank_10_10.config(width="30", height="30")
blank_10_10.config(bg="white")
blank_10_10.grid(row=10, column=0)

blank_11_10 = tk.Frame(ventana_Resultados)
blank_11_10.config(width="30", height="30")
blank_11_10.config(bg="white")
blank_11_10.grid(row=11, column=0)

blank_12_10 = tk.Frame(ventana_Resultados)
blank_12_10.config(width="30", height="30")
blank_12_10.config(bg="white")
blank_12_10.grid(row=12, column=0)

blank_13_10 = tk.Frame(ventana_Resultados)
blank_13_10.config(width="30", height="30")
blank_13_10.config(bg="white")
blank_13_10.grid(row=13, column=0)

# Elemenos de la decima-segunda columna
boton_Calcular_ICA_Total = tk.Button(ventana_Resultados, text="VER RESULTADO",
font="Helvetica 9 bold", command=calcular_ICA_Total)
boton_Calcular_ICA_Total.config(bg="CadetBlue")
boton_Calcular_ICA_Total.grid(row=0, column=1, columns=2)

```

```

etiqueta_ICA_Total = tk.Label(ventana_Resultados, text="Resultado ICA: ", font="helvetica 11
bold")
etiqueta_ICA_Total.config(bg="white")
etiqueta_ICA_Total.grid(row=1, column=1)

etiqueta_ICA_AguaPotable = tk.Label(ventana_Resultados, text="Para Agua Potable: ")
etiqueta_ICA_AguaPotable.config(justify='left')
etiqueta_ICA_AguaPotable.config(bg="white")
etiqueta_ICA_AguaPotable.grid(row=3, column=1, columnspan=2)

campo_ICA_AguaPotable = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_AguaPotable,
font="Helvetica 8")
campo_ICA_AguaPotable.config(width="43")
campo_ICA_AguaPotable.config(bg="white")
campo_ICA_AguaPotable.grid(row=4, column=1, columnspan=2)

etiqueta_ICA_Agricultura = tk.Label(ventana_Resultados, text="Para la Agricultura: ")
etiqueta_ICA_Agricultura.config(justify='left')
etiqueta_ICA_Agricultura.config(bg="white")
etiqueta_ICA_Agricultura.grid(row=5, column=1, columnspan=2)

campo_ICA_Agricultura = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Agricultura,
font="Helvetica 8")
campo_ICA_Agricultura.config(width="43")
campo_ICA_Agricultura.config(bg="white")
campo_ICA_Agricultura.grid(row=6, column=1, columnspan=2)

etiqueta_ICA_Pesca = tk.Label(ventana_Resultados, text="Para la Pesca y Vida Acuatica: ")
etiqueta_ICA_Pesca.config(justify='left')
etiqueta_ICA_Pesca.config(bg="white")
etiqueta_ICA_Pesca.grid(row=7, column=1, columnspan=2)

campo_ICA_Pesca = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Pesca,
font="Helvetica 8")
campo_ICA_Pesca.config(width="43")
campo_ICA_Pesca.config(bg="white")
campo_ICA_Pesca.grid(row=8, column=1, columnspan=2)

etiqueta_ICA_Industrial = tk.Label(ventana_Resultados, text="Para la Industria: ")
etiqueta_ICA_Industrial.config(justify='left')
etiqueta_ICA_Industrial.config(bg="white")
etiqueta_ICA_Industrial.grid(row=9, column=1, columnspan=2)

campo_ICA_Industrial = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Industrial,
font="Helvetica 8")
campo_ICA_Industrial.config(width="43")
campo_ICA_Industrial.config(bg="white")
campo_ICA_Industrial.grid(row=10, column=1, columnspan=2)

etiqueta_ICA_Recreativo = tk.Label(ventana_Resultados, text="Para Uso Recreativo: ")
etiqueta_ICA_Recreativo.config(justify='left')
etiqueta_ICA_Recreativo.config(bg="white")
etiqueta_ICA_Recreativo.grid(row=11, column=1, columnspan=2)

campo_ICA_Recreativo = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Recreativo,
font="Helvetica 8")

```

```

campo_ICA_Recreativo.config(width="43")
campo_ICA_Recreativo.config(bg="white")
campo_ICA_Recreativo.grid(row=12, column=1, columnspan=2)

# Elementos de la decima-tercera columna
campo_ICA_Total = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Total, font="helvetica
11")
campo_ICA_Total.config(width="10")
campo_ICA_Total.config(bg="white")
campo_ICA_Total.grid(row=1, column=2)

# Elementos de la decima-tercera columna
campo_ICA_Total = tk.Entry(ventana_Resultados, textvariable=see_var_ICA_Total, font="helvetica
11")
campo_ICA_Total.config(width="10")
campo_ICA_Total.config(bg="white")
campo_ICA_Total.grid(row=1, column=2)

# Elementos de la decima-cuarta columna
blank_0_13 = tk.Frame(ventana_Resultados)
blank_0_13.config(width="30", height="30")
blank_0_13.config(bg="white")
blank_0_13.grid(row=0, column=3)

blank_1_13 = tk.Frame(ventana_Resultados)
blank_1_13.config(width="30", height="30")
blank_1_13.config(bg="white")
blank_1_13.grid(row=1, column=3)

blank_2_13 = tk.Frame(ventana_Resultados)
blank_2_13.config(width="30", height="30")
blank_2_13.config(bg="white")
blank_2_13.grid(row=2, column=3)

blank_3_13 = tk.Frame(ventana_Resultados)
blank_3_13.config(width="30", height="30")
blank_3_13.config(bg="white")
blank_3_13.grid(row=3, column=3)

blank_4_13 = tk.Frame(ventana_Resultados)
blank_4_13.config(width="30", height="30")
blank_4_13.config(bg="white")
blank_4_13.grid(row=4, column=3)

blank_5_13 = tk.Frame(ventana_Resultados)
blank_5_13.config(width="30", height="30")
blank_5_13.config(bg="white")
blank_5_13.grid(row=5, column=3)

blank_6_13 = tk.Frame(ventana_Resultados)
blank_6_13.config(width="30", height="30")
blank_6_13.config(bg="white")
blank_6_13.grid(row=6, column=3)

blank_7_13 = tk.Frame(ventana_Resultados)
blank_7_13.config(width="30", height="30")

```

```

blank_7_13.config(bg="white")
blank_7_13.grid(row=7, column=3)

blank_8_13 = tk.Frame(ventana_Resultados)
blank_8_13.config(width="30", height="30")
blank_8_13.config(bg="white")
blank_8_13.grid(row=8, column=3)

blank_9_13 = tk.Frame(ventana_Resultados)
blank_9_13.config(width="30", height="30")
blank_9_13.config(bg="white")
blank_9_13.grid(row=9, column=3)

blank_10_13 = tk.Frame(ventana_Resultados)
blank_10_13.config(width="30", height="30")
blank_10_13.config(bg="white")
blank_10_13.grid(row=10, column=3)

blank_11_13 = tk.Frame(ventana_Resultados)
blank_11_13.config(width="30", height="30")
blank_11_13.config(bg="white")
blank_11_13.grid(row=11, column=3)

blank_12_13 = tk.Frame(ventana_Resultados)
blank_12_13.config(width="30", height="30")
blank_12_13.config(bg="white")
blank_12_13.grid(row=12, column=3)

blank_13_13 = tk.Frame(ventana_Resultados)
blank_13_13.config(width="30", height="30")
blank_13_13.config(bg="white")
blank_13_13.grid(row=13, column=3)

# Definición de la Subventana de Botones
blank_button_0_0 = tk.Frame(ventana_Botones)
blank_button_0_0.config(width="72", height="30")
blank_button_0_0.config(bg="white")
blank_button_0_0.grid(row=0, column=0)

blank_button_0_1 = tk.Frame(ventana_Botones)
blank_button_0_1.config(width="72", height="30")
blank_button_0_1.config(bg="white")
blank_button_0_1.grid(row=0, column=1)

blank_button_0_2 = tk.Frame(ventana_Botones)
blank_button_0_2.config(width="72", height="30")
blank_button_0_2.config(bg="white")
blank_button_0_2.grid(row=0, column=2)

blank_button_0_3 = tk.Frame(ventana_Botones)
blank_button_0_3.config(width="72", height="30")
blank_button_0_3.config(bg="white")
blank_button_0_3.grid(row=0, column=3)

blank_button_0_4 = tk.Frame(ventana_Botones)
blank_button_0_4.config(width="72", height="30")

```



```
blank_button_0_4.config(bg="white")
blank_button_0_4.grid(row=0, column=4)

blank_button_0_5 = tk.Frame(ventana_Botones)
blank_button_0_5.config(width="72", height="30")
blank_button_0_5.config(bg="white")
blank_button_0_5.grid(row=0, column=5)

blank_button_0_6 = tk.Frame(ventana_Botones)
blank_button_0_6.config(width="72", height="30")
blank_button_0_6.config(bg="white")
blank_button_0_6.grid(row=0, column=6)

blank_button_0_7 = tk.Frame(ventana_Botones)
blank_button_0_7.config(width="72", height="30")
blank_button_0_7.config(bg="white")
blank_button_0_7.grid(row=0, column=7)

blank_button_0_8 = tk.Frame(ventana_Botones)
blank_button_0_8.config(width="72", height="30")
blank_button_0_8.config(bg="white")
blank_button_0_8.grid(row=0, column=8)

blank_button_0_9 = tk.Frame(ventana_Botones)
blank_button_0_9.config(width="72", height="30")
blank_button_0_9.config(bg="white")
blank_button_0_9.grid(row=0, column=9)

blank_button_0_10 = tk.Frame(ventana_Botones)
blank_button_0_10.config(width="72", height="30")
blank_button_0_10.config(bg="white")
blank_button_0_10.grid(row=0, column=10)

blank_button_0_11 = tk.Frame(ventana_Botones)
blank_button_0_11.config(width="72", height="30")
blank_button_0_11.config(bg="white")
blank_button_0_11.grid(row=0, column=11)

blank_button_0_12 = tk.Frame(ventana_Botones)
blank_button_0_12.config(width="72", height="30")
blank_button_0_12.config(bg="white")
blank_button_0_12.grid(row=0, column=12)

blank_button_0_13 = tk.Frame(ventana_Botones)
blank_button_0_13.config(width="72", height="30")
blank_button_0_13.config(bg="white")
blank_button_0_13.grid(row=0, column=13)

blank_button_1_0 = tk.Frame(ventana_Botones)
blank_button_1_0.config(width="72", height="30")
blank_button_1_0.config(bg="white")
blank_button_1_0.grid(row=1, column=0)

blank_button_1_1 = tk.Frame(ventana_Botones)
blank_button_1_1.config(width="72", height="30")
blank_button_1_1.config(bg="white")
```

```

blank_button_1_1.grid(row=1, column=1)

boton_Guardar = tk.Button(ventana_Botones, text="GUARDAR", font="Helvetica 9 bold",
command=guardar_Info)
boton_Guardar.config(width="15")
boton_Guardar.config(bg="dark sea green")
boton_Guardar.grid(row=1, column=2)

blank_button_1_3 = tk.Frame(ventana_Botones)
blank_button_1_3.config(width="72", height="30")
blank_button_1_3.config(bg="white")
blank_button_1_3.grid(row=1, column=3)

boton_Buscar = tk.Button(ventana_Botones, text="BUSCAR", font="Helvetica 9 bold",
command=buscar_Info)
boton_Buscar.config(width="15")
boton_Buscar.config(bg="deep sky blue")
boton_Buscar.grid(row=1, column=4, columnspan=2)

blank_button_1_6 = tk.Frame(ventana_Botones)
blank_button_1_6.config(width="72", height="30")
blank_button_1_6.config(bg="white")
blank_button_1_6.grid(row=1, column=6)

boton_Actualizar = tk.Button(ventana_Botones, text="ACTUALIZAR", font="Helvetica 9 bold",
command=actualizar_Info)
boton_Actualizar.config(width="15")
boton_Actualizar.config(bg="gold")
boton_Actualizar.grid(row=1, column=7, columnspan=2)

blank_button_1_9 = tk.Frame(ventana_Botones)
blank_button_1_9.config(width="72", height="30")
blank_button_1_9.config(bg="white")
blank_button_1_9.grid(row=1, column=9)

boton_Eliminar = tk.Button(ventana_Botones, text="ELIMINAR", font="Helvetica 9 bold",
command=eliminar_Info)
boton_Eliminar.config(width="15")
boton_Eliminar.config(bg="tomato")
boton_Eliminar.grid(row=1, column=10)

blank_button_1_11 = tk.Frame(ventana_Botones)
blank_button_1_11.config(width="72", height="30")
blank_button_1_11.config(bg="white")
blank_button_1_11.grid(row=1, column=11)

blank_button_1_12 = tk.Frame(ventana_Botones)
blank_button_1_12.config(width="72", height="30")
blank_button_1_12.config(bg="white")
blank_button_1_12.grid(row=1, column=12)

blank_button_1_13 = tk.Frame(ventana_Botones)
blank_button_1_13.config(width="72", height="30")
blank_button_1_13.config(bg="white")
blank_button_1_13.grid(row=1, column=13)

```

```
blank_button_2_0 = tk.Frame(ventana_Botones)
blank_button_2_0.config(width="72", height="30")
blank_button_2_0.config(bg="white")
blank_button_2_0.grid(row=2, column=0)
```

```
blank_button_2_1 = tk.Frame(ventana_Botones)
blank_button_2_1.config(width="72", height="30")
blank_button_2_1.config(bg="white")
blank_button_2_1.grid(row=2, column=1)
```

```
blank_button_2_2 = tk.Frame(ventana_Botones)
blank_button_2_2.config(width="72", height="30")
blank_button_2_2.config(bg="white")
blank_button_2_2.grid(row=2, column=2)
```

```
blank_button_2_3 = tk.Frame(ventana_Botones)
blank_button_2_3.config(width="72", height="30")
blank_button_2_3.config(bg="white")
blank_button_2_3.grid(row=2, column=3)
```

```
blank_button_2_4 = tk.Frame(ventana_Botones)
blank_button_2_4.config(width="72", height="30")
blank_button_2_4.config(bg="white")
blank_button_2_4.grid(row=2, column=4)
```

```
blank_button_2_5 = tk.Frame(ventana_Botones)
blank_button_2_5.config(width="72", height="30")
blank_button_2_5.config(bg="white")
blank_button_2_5.grid(row=2, column=5)
```

```
blank_button_2_6 = tk.Frame(ventana_Botones)
blank_button_2_6.config(width="72", height="30")
blank_button_2_6.config(bg="white")
blank_button_2_6.grid(row=2, column=6)
```

```
blank_button_2_7 = tk.Frame(ventana_Botones)
blank_button_2_7.config(width="72", height="30")
blank_button_2_7.config(bg="white")
blank_button_2_7.grid(row=2, column=7)
```

```
blank_button_2_8 = tk.Frame(ventana_Botones)
blank_button_2_8.config(width="72", height="30")
blank_button_2_8.config(bg="white")
blank_button_2_8.grid(row=2, column=8)
```

```
blank_button_2_9 = tk.Frame(ventana_Botones)
blank_button_2_9.config(width="72", height="30")
blank_button_2_9.config(bg="white")
blank_button_2_9.grid(row=2, column=9)
```

```
blank_button_2_10 = tk.Frame(ventana_Botones)
blank_button_2_10.config(width="72", height="30")
blank_button_2_10.config(bg="white")
blank_button_2_10.grid(row=2, column=10)
```

```
blank_button_2_11 = tk.Frame(ventana_Botones)
```

```
blank_button_2_11.config(width="72", height="30")
blank_button_2_11.config(bg="white")
blank_button_2_11.grid(row=2, column=11)
```

```
blank_button_2_12 = tk.Frame(ventana_Botones)
blank_button_2_12.config(width="72", height="30")
blank_button_2_12.config(bg="white")
blank_button_2_12.grid(row=2, column=12)
```

```
blank_button_2_13 = tk.Frame(ventana_Botones)
blank_button_2_13.config(width="72", height="30")
blank_button_2_13.config(bg="white")
blank_button_2_13.grid(row=2, column=13)
```

```
## CONFIGURACION INICIAL DE LA BASE DE DATOS
```

```
connection = sql.connect('programa_ica')
cursor = connection.cursor()
cursor.execute("""CREATE TABLE IF NOT EXISTS info_ica(
    nombre_estudio text PRIMARY KEY,
    fecha_estudio text,
    lugar_estudio text,
    rio_estudio text,
    encargado_estudio text,
    descripcion_estudio text,
    dbo_nsf real,
    sdt_nsf real,
    turbidez_nsf real,
    coliformes_fecales_nsf real,
    nitratos_nsf real,
    fosfatos_nsf real,
    temperatura_nsf real,
    ph_nsf real,
    od_nsf real,
    aluminio_metales real,
    cadmio_metales real,
    cobre_metales real,
    hierro_metales real,
    manganeso_total_metales real,
    plomo_metales real,
    zinc_total_metales real,
    valor_ica_nsf real,
    valor_ica_metales real,
    valor_ica_total real,
    resultado_ica_agua_potable text,
    resultado_ica_agricultura text,
    resultado_ica_pesca text,
    resultado_ica_industria text,
    resultado_ica_recreativo text);
""")
connection.close()
```

```
## EJECUCION DEL PROGRAMA
```

```
campo_ver_ICA_NSF.config(state='disabled')
campo_ver_ICA_Metales.config(state='disabled')
campo_ICA_Total.config(state='disabled')
campo_ICA_AguaPotable.config(state='disabled')
```

campo_ICA_Pesca.config(state='disabled')
 campo_ICA_Agricultura.config(state='disabled')
 campo_ICA_Industrial.config(state='disabled')
 campo_ICA_Recreativo.config(state='disabled')
 raiz.mainloop()

ANEXO V

En el siguiente anexo se adjunta el link para descargar los archivos

[PROGRAMA ICA-LISETH COLLAGUAZO](#)

ANEXO VI

TABLA 1: CRITERIOS DE CALIDAD DE AGUAS QUE PARA CONSUMO HUMANO Y DOMÉSTICO QUE REQUIEREN TRATAMIENTO CONVENCIONAL			
PARÁMETRO	EXPRESADO COMO	UNIDAD	CRITERIO DE CALIDAD
Aceltes y Grasas	Película visible		Ausencia
Aluminio total	Al	mg/l	0,2
Amoniaco	N	mg/l	0,5
Arsénico	As	mg/l	0,1
Coliformes Fecales	NMP	NMP/100 ml	2000
Coliformes Totales	NMP	NMP/100 ml	20000
Bario	Ba	mg/l	1,0
Cadmio	Cd	mg/l	0,01
Cianuro	CN ⁻	mg/l	0,2
Cinc	Zn	mg/l	5,0
Cobre	Cu	mg/l	1,0
Color	Color real	Unidades de Platino-Cobalto	75,0
Compuesto Fenólicos	Fenol	mg/l	0,001
Cromo	Cr ⁺⁶	mg/l	0,05
Demanda Bioquímica de Oxígeno (5 días)	DBO ₅	mg/l	<2mg/l
Hierro total	Fe	mg/l	1,0
Difeniles Policlorinados	Concentración de agente activo		No detectable
Materia Flotante	Visible		Ausencia
Mercurio	Hg	mg/l	0,002
Nitratos	N	mg/l	10,0
Nitritos	N	mg/l	1,0
Olor y sabor			Es permitido removible por tratamiento convencional
Oxígeno Disuelto	OD	mg/l	>60% del OD Sat.
pH	pH		6-9
Plata	Ag	mg/l	0,05
Plomo	Pb	mg/l	0,05
Selenio	Se	mg/l	0,01
Sulfatos	SO ₄ ⁻²	mg/l	250,0
Tensoactivos	Sustancias activas al azul de metileno	mg/l	0,5
Turbiedad		UTN	100,0

Nota: Podrán usarse aguas con turbiedades y coliformes fecales ocasionales superiores a los indicados en esta Tabla, siempre y las características de las aguas tratadas sean entregadas de acuerdo con la Norma INEN

ANEXO VII

En el siguiente anexo se adjunta el manual de usuario para la utilización del programa.



ESCUELA POLITÉCNICA
NACIONAL



**ÍNDICE DE
CALIDAD DEL
AGUA
CUERPOS DE
AGUA LÓTICOS**



Manual del usuario



SOFTWARE ICA NSF-HBI

CREADORA: LISETH COLLAGUAZO
REVISADO POR: DRA. VERÓNICA
MORALES





INTRODUCCIÓN



El agua es el líquido fuente de vida de mayor importancia en el planeta tierra, es un indicador de posibilidad de vida en otros planetas y ocupa las tres cuartas partes de la superficie terrestre (aproximadamente el 70%). Sin embargo, únicamente el 3,5% es agua dulce y tan sólo el 0,025% es accesible para consumo (FuncAgua, 2020). Debido a su importancia, versatilidad y escasez es necesario poner en práctica estrategias que permitan conocer su grado de contaminación, para posteriormente ser usada o tratada de acuerdo con las necesidades de uso del ser humano.

El ICA (Índice de calidad del agua) es un procedimiento mediante el cual se puede evaluar la calidad de un recurso hídrico (Quiroz et al., 2017). Y Puede ser expresado como una expresión simple o compleja de algunos parámetros (Torres et al., 2009).

En el presente programa se utiliza dos métodos de ICA: ICA-SNF y ICA-Metales (HPI). Para el caso del ICA-NSF ayuda a conocer el estado del agua de acuerdo con nueve parámetros de calidad del agua como son: oxígeno disuelto, coliformes fecales, pH, DBO, nitratos, fosfatos, desviación de temperatura, turbidez y sólidos disueltos. Y para el ICA-Metales se utiliza concentraciones de 6 metales: aluminio, cadmio, hierro, cobre, manganeso, plomo, zincs. Con los resultados de los dos métodos (ICA-SNF e ICA-Metales) se obtiene un ICA global que permitirá determinar el uso del agua que mejor se ajuste al valor obtenido.

El objetivo del presente programa realizado en el software de código abierto Python es determinar el índice de calidad del agua para cuerpos de agua lóticos en función de mínimo 7 parámetros y máximo 9 parámetros de calidad del agua medidos ya sea in-situ o en laboratorio, que permita a los usuarios obtener resultados de manera eficaz y confiable.

Adicionalmente, este programa utiliza pantallas dinámicas que genera al usuario una experiencia amigable y de facilidad en uso.

LISETH COLLAGUAZO
CREADOR



TABLA DE CONTENIDO

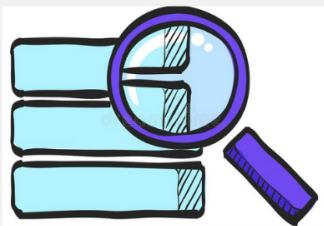
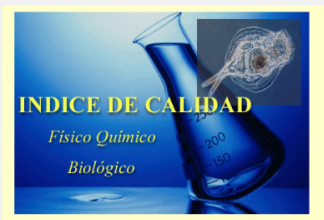


Objetivos del programa	01
Ingreso al programa.....	02
Contenido de la página del programa	03
Cálculos del ICA NSF-HBI	04
Sección resultados	05
Función de las pestañas	06
Pasos para buscar en la base de datos	07

LISETH COLLAGUAZO
CREADOR

OBJETIVOS

DEL PROGRAMA



Buena (71-90)
Media (51-70)
Mala (26-50)

Objetivo

Permitir el ingreso de valores de parámetros de calidad del agua, con los cuales se determina el Índice de Calidad del Agua Global con el método NSF y con el método HBI generando un reporte de que usos se le puede dar a este cuerpo de agua en función al ICA calculado.

Módulo de datos

En el módulo de datos el usuario debe ingresar los valores de los datos medidos de acuerdo con las unidades que se le solicita en el programa para posteriormente realizar el cálculo del ICA.

Módulo resultados

En el módulo de resultados se muestra tratamientos que se deben aplicar para los usos que se le puede dar al agua en función del ICA que calculo.



INGRESO AL PROGRAMA



Para poder acceder al programa el usuario debe ingresar al link que se muestra a continuación, el siguiente link encontrará el programa en Python o en aplicativo (.exe).

PROGRAMA ICA-LISETH COLLAGUAZO

A continuación, se presenta la pantalla de la carpeta de los archivos.

Nombre	Fecha de modificación	Tipo	Tamaño
 PROGRAMA-LISETH COLLAGUAZO	3/3/2023 4:07	Aplicación	1,240 KB
 PROGRAMA-LISETH COLLAGUAZO	3/3/2023 3:13	Python File	94 KB

Para descargar cualquiera de los archivos se debe seguir los siguientes pasos:

1. Abrir el link con el correo electrónico institucional.
2. Luego escoger que tipo de archivo se quiere utilizar.
3. Descargarlo
4. En caso de que sea el programa en extensión .exe se debe poner ejecutar caso contrario abrir el programa en Python.

IMPORTANTE !!

Si se descarga el código en formato .py se debe tener instalado Python, en cambio si se descarga el archivo en formato .exe solo se debe ejecutar sin tener la necesidad de tener instalado Python.

CONTENIDO DE LA PÁGINA DEL PROGRAMA



En esta imagen se muestra cual es la pantalla que se desplegara al ejecutar el programa en cualquiera de los dos formatos presentados.

ESCUELA POLITÉCNICA NACIONAL - CÁLCULO DE ICA

Creado por Liseth Collaguazo

INFORMACION	DATOS		RESULTADOS
Nombre del estudio: <input type="text"/> Fecha del estudio: <input type="text"/> Lugar del estudio: <input type="text"/> Río estudiado: <input type="text"/> Encargado del estudio: <input type="text"/> Descripción del estudio: <input type="text"/>	ICA - NSF DBO (mg/L): <input type="text" value="0.0"/> SDT (mg/L): <input type="text" value="0.0"/> Turbidez (NTU): <input type="text" value="0.0"/> C. Fecales (NMP/100 mL): <input type="text" value="0.0"/> Nitratos (mg/L): <input type="text" value="0.0"/> Fosfatos (mg/ L): <input type="text" value="0.0"/> Temperatura (°C): <input type="text" value="0.0"/> pH (Unidades): <input type="text" value="0.0"/> OD (%Saturación): <input type="text" value="0.0"/> <input type="button" value="CALCULAR ICA"/> ICA - NSF: <input type="text" value="0.0"/>	ICA - Metales Aluminio (mg/L): <input type="text" value="0.0"/> Cadmio (mg/L): <input type="text" value="0.0"/> Cobre(mg/L): <input type="text" value="0.0"/> Hierro(mg/L): <input type="text" value="0.0"/> Manganeso(mg/L): <input type="text" value="0.0"/> Plomo(mg/L): <input type="text" value="0.0"/> Zinc (mg/L): <input type="text" value="0.0"/> <input type="button" value="CALCULAR ICA"/> ICA - Metales: <input type="text" value="0.0"/>	<input type="button" value="VER RESULTADO"/> Resultado ICA: <input type="text" value="0.0"/> Para Agua Potable: <input type="text"/> Para la Agricultura: <input type="text"/> Para la Pesca y Vida Acuatica: <input type="text"/> Para la Industria: <input type="text"/> Para Uso Recreativo: <input type="text"/>
<input type="button" value="GUARDAR"/> <input type="button" value="BUSCAR"/> <input type="button" value="ACTUALIZAR"/> <input type="button" value="ELIMINAR"/>			

PASOS PARA EL USO DEL PROGRAMA

- El primer paso que debe hacer el usuario antes de correr el programa es llenar los datos informativos del estudio que se calculara el ICA.

Para llenar los datos informativos el usuario debe ir rellenando los casilleros con toda la información que se solicita como se muestra en la imagen.

INFORMACION
Nombre del estudio: <input type="text"/>
Fecha del estudio: <input type="text"/>
Lugar del estudio: <input type="text"/>
Río estudiado: <input type="text"/>
Encargado del estudio: <input type="text"/>
Descripción del estudio: <input type="text"/>

LISETH COLLAGUAZO
CREADOR

CÁLCULOS

ICA-NSF

Para realizar un cálculo el usuario en primer lugar debe llenar los datos informativos, luego debe rellenar todos los campos con los datos de las mediciones de los parámetros que se solicitan respetando las unidades que se utiliza para el cálculo del respectivo índice.

CONSIDERACIONES

- En el parámetro de la temperatura primero se debe calcular la diferencia entre la T° Ambiente y la T° del agua medida, con el valor obtenido proceder a ingresar en el programa para que se realice el cálculo .
- También se debe tener en cuenta que el número mínimo para el cálculo del ICA con el método es de 7 parámetros.
- Otra consideración importante que se debe tener en cuenta es que si el valor del índice de metales HBI es mayor a 100 su valor será 0 lo que significa que el agua esta contaminada.

A continuación, se presenta la pantalla en la que se debe ingresar los datos.

ICA - NSF		ICA - Metales	
DBO (mg/L):	<input type="text" value="0.0"/>	Aluminio (mg/L):	<input type="text" value="0.0"/>
SDT (mg/L):	<input type="text" value="0.0"/>	Cadmio (mg/L):	<input type="text" value="0.0"/>
Turbidez (NTU):	<input type="text" value="0.0"/>	Cobre(mg/L):	<input type="text" value="0.0"/>
C. Fecales (NMP/100 mL):	<input type="text" value="0.0"/>	Hierro(mg/L):	<input type="text" value="0.0"/>
Nitratos (mg/L):	<input type="text" value="0.0"/>	Manganeso(mg/L):	<input type="text" value="0.0"/>
Fosfatos (mg/ L):	<input type="text" value="0.0"/>	Plomo(mg/L):	<input type="text" value="0.0"/>
Temperatura (°C):	<input type="text" value="0.0"/>	Zinc (mg/L):	<input type="text" value="0.0"/>
pH (Unidades):	<input type="text" value="0.0"/>		
OD (%Saturación):	<input type="text" value="0.0"/>		
<input type="button" value="CALCULAR ICA"/>		<input type="button" value="CALCULAR ICA"/>	
ICA - NSF:	<input type="text" value="0.0"/>	ICA - Metales:	<input type="text" value="0.0"/>

Finalmente, tras llenar todos los campos solicitados para realizar el calculo se debe dar clic en CALCULAR-ICA.



SECCIÓN

RESULTADOS

Para poder ver los resultados el usuario debe dar clic en ver resultados para que el programa muestre los posibles tratamientos para los diferente usos que se le puede dar al agua en función al ICA calculado.

A continuación, se muestra un ejemplo de como se muestra la pantalla de resultados.



RESULTADOS

VER RESULTADO

Resultado ICA: 61.26293333

Para Agua Potable:
Tratamiento potabilizador necesario.

Para la Agricultura:
Utilizable en la mayoría de cultivos

Para la Pesca y Vida Acuatica:
Límite para peces muy sensitivos

Para la Industria:
No requiere tratamiento para la mayoría de industrias.

Para Uso Recreativo:
Restricción para deportes de inmersión.

¡ATENCIÓN!
En el programa no necesariamente se debe calcular el ICA NSF y HBI, sino se puede calcular el ICA NSF o HBI por separado.

NOTA

Cuando el usuario calcula los dos ICAs (NSF y HPI) el programa automáticamente calcula el ICA global y en función de esto se muestran los usos que se le puede dar al agua. Sin embargo, si solo se calcula un solo ICA también se muestran los posibles usos, implicando por lo tanto una flexibilidad del programa si no se cuenta con los 16 parámetros máximos de ingreso.

LISETH COLLAGUAZO
CREADOR

FUNCIÓN DE LAS PESTAÑAS

- GUARDAR
- BUSCAR
- ACTUALIZAR
- ELIMINAR



GUARDAR

Con la opción guardar el usuario puede guardar los cálculos que realizó en el programa en un base de datos que se generará en el programa.



BUSCAR

La opción buscar permite que el usuario al ingresar los datos informativos del estudio que anteriormente fue guardado en la base de datos pueda acceder a los cálculos que realizó.



ELIMINAR

La opción eliminar permite al usuario borrar los datos informativos y los cálculos de un estudio que estén guardados en la base de datos del programa.



ACTUALIZAR

La opción actualizar permite al usuario actualizar los datos informativos y las medidas de los parámetros de un estudio previo que este almacenado en la base de datos del programa.

LISETH COLLAGUAZO
CREADOR

PASOS PARA

BUSCAR EN LA BASE DE DATOS

1.

El usuario debe llenar los datos informativos del estudio.

2.

El usuario debe dar clic en buscar. Al momento de dar clic en buscar al usuario le aparecen llenos los campos de los parámetros que posteriormente hayan sido completados.

IMPORTANTE

EN EL CASO SE UTILICE LOS DOS MÉTODOS ICA-NSF E ICA-METALES (HPI), EL PROGRAMA REALIZARÁ EL CÁLCULO DEL ICA GLOBAL AUTOMÁTICAMENTE. Y A TRAVÉS DE LA SIGUIENTE TABLA DE VALORACIÓN DEL VALOR DEL ICA GLOBAL OBTENIDO SE DETERMINARÁ LOS DIFERENTES USOS DEL AGUA EN FUNCIÓN DE SU CALIDAD DEL AGUA.

Calidad del agua	Valor del ICA
Excelente	91 – 100
Buena	71 – 90
Media	51 – 70
Mala	26 – 50
Muy Mala	0 – 25

LISETH COLLAGUAZO
CREADOR