

# ARTÍCULO EN ESPAÑOL

**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**



**TEMA:**

**“DISEÑO Y DIMENSIONAMIENTO DE PLANTAS DE  
TRATAMIENTO CONVENCIONAL DE AGUAS RESIDUALES  
CON UTILIZACIÓN DE SOFTWARE LIBRE”**

**Autor:**

**PUENTE PONCE PABLO FRANCISCO**

**Director: Ing. Marcelo Puente**

**Ibarra, 2012**

# **“DISEÑO Y DIMENSIONAMIENTO DE PLANTAS DE TRATAMIENTO CONVENCIONAL DE AGUAS RESIDUALES CON UTILIZACIÓN DE SOFTWARE LIBRE”**

MAYO 2012

Ibarra - Ecuador

## **Problema**

Las actividades llevadas a cabo por el hombre generan residuos que producen un impacto negativo sobre el medio ambiente. Estos contaminantes pueden ser clasificados primariamente según su estado de agregación, como: gaseosos, líquidos o sólidos. Nos referiremos específicamente a los residuos líquidos que pueden provocar enfermedades infecciosas, parasitarias y relacionadas con intoxicaciones.

Si bien estos efluentes son generados durante prácticamente cualquier actividad realizada por el hombre, aquí solo consideraremos aquellos que se producen durante los procesos industriales, debido a que las descargas pueden ser colectadas con facilidad y tratadas en instalaciones denominadas plantas de tratamiento.

El diseño y dimensionamiento de estas plantas debe realizarlo el profesional en tratamiento de aguas, Este diseño puede requerir de un amplio espacio de tiempo y un alto costo para las industrias.

Debido a que existe un proceso convencional determinado que se puede cualificar y cuantificar en base a parámetros de diseño establecidos en

función de los contaminantes y caudal, se presta para desarrollar un software que permita optimizar tiempo, dinero y mejorar la gestión de empresas, profesionales, y municipios, etc.

La función de la planta de tratamiento es adecuar las características fisicoquímicas y biológicas del líquido residual a las establecidas por la normativa correspondiente, en general los límites permisibles fijados por el organismo de control están vinculados con el cuerpo receptor de las descargas, para el presente análisis, el sistema de alcantarillado.

Para realizar el diseño y dimensionamiento de las plantas de tratamiento se debe realizar la caracterización de las aguas residuales, es decir medir el caudal y determinar las características fisicoquímicas del efluente, estos datos se debe comparar con los límites de vuelco establecidos en la norma y finalmente proceder a diseñar y dimensionar la planta de tratamiento en función de los contaminantes y cantidades de los mismos a eliminar para llegar a lo exigido por el organismo de control.

## **Objetivos**

### **Objetivo General**

Desarrollar un software que permita diseñar y dimensionar plantas de tratamiento convencional de aguas en base a parámetros de diseño establecidos, optimizando tiempo y recursos para las empresas, profesionales y entidades públicas de la zona norte del país.

### **Objetivos Específicos**

- Proponer una solución adecuada de diseño en cada caso de tratamiento convencional de aguas residuales.
- Dimensionar el modelo de la planta de tratamiento de aguas.
- Facilitar el trabajo de empresas, profesionales, municipios y otras entidades.
- Optimizar recursos, tiempo y dinero.
- Satisfacer las necesidades de los clientes de la empresa “Jirón & Asociados”.

### **Alcance**

Este proyecto busca desarrollar un programa que permita a sus usuarios diseñar y dimensionar plantas de tratamiento convencional de aguas residuales para optimizar tiempo, dinero y mejorar la gestión ambiental de empresas, profesionales, municipios y otras entidades tanto públicas como privadas que estén dentro del área de influencia de la empresa “Jirón & Asociados”, empresa dedicada al desarrollo e implementación de software en la zona norte del país, y ampliar su oferta de software.

### **Justificación del Proyecto**

La Constitución de la República del Ecuador y otras normativas obligan a las

instituciones públicas y privadas a realizar el tratamiento de las aguas residuales para ser volcadas en las distintas fuentes de descarga para así disminuir los problemas que conlleva la contaminación del agua, enfermedades infecciosas, parasitarias y relacionadas con intoxicaciones

Como el agua es considerada esencial para la vida es de gran importancia que se garantice su buen estado por lo cual se hace importante que las industrias que la utilizan para sus procesos productivos busquen una manera de minimizar el impacto que los residuos de los mismos tienen sobre los efluentes producidos por su actividad.

Mediante la creación de este software se busca dar una idea general y facilitar el proceso de diseñar y dimensionar plantas de tratamiento convencional de aguas en base a parámetros de diseño establecidos con el fin de depurar las aguas residuales y garantizar a las personas y colectividades el acceso equitativo, permanente y de calidad al agua.

Programas similares pueden encontrarse en internet pero están desarrollados por empresas extranjeras dedicadas a la implementación de estas plantas, además su costo es de entre \$4995 y \$13875 por compra y de entre \$ 1995 y \$ 5550 por la licencia anual, datos de la empresa “Chempute Software”, con lo que, el tener un programa de estas características diseñado para el “ambiente” nacional y desarrollado con software libre disminuirá considerablemente el costo.

## ARQUITECTURA

### Cliente-servidor

Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes **computadoras** aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red Cliente/Servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados.

Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

### Características

En la arquitectura C/S el **remite**nte de una **solicitud** es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro** o **amo**).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- Al contratar un servicio de redes, se tiene que tener en la velocidad de conexión que le otorga al cliente y el tipo de cable que utiliza, por ejemplo: cable de cobre ronda entre 1 ms y 50 ms.

Al receptor de la solicitud enviada por cliente se conoce como servidor.

Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

### Comparación de la arquitectura C/S con otras arquitecturas de red

#### Comparación con las redes de pares

Las **redes de pares**, también conocidas como redes **par-a-par** o **peer-to-peer**

(Abreviado con las siglas **P2PM**), son otro tipo de arquitectura de red.

### Comparación con la arquitectura Cliente-Cola-Cliente

Si bien la clásica arquitectura C/S requiere uno de los puntos terminales de comunicación para actuar como un servidor, que puede ser algo más difícil de aplicar, la arquitectura Cliente-Cola-Cliente habilita a todos los nodos para actuar como clientes simples, mientras que el servidor actúa como una cola que va capturando las peticiones de los clientes.

Esta arquitectura permite simplificar en gran medida la implementación de software. La arquitectura P2P originalmente se basó en el concepto "Cliente-Cola-Cliente".

### Arquitecturas multi-capas

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o **dos capas**.

Algunas redes disponen de tres tipos de nodos:

- Clientes que interactúan con los usuarios finales.
- Servidores de aplicación que procesan los datos para los clientes.
- Servidores de la base de datos que almacenan los datos para los servidores de aplicación.

Esta configuración se llama una arquitectura de tres-capas.

### Ventajas de las arquitecturas n-capas:

La ventaja fundamental de una arquitectura **n-capas** comparado con una arquitectura

de dos niveles (o una tres-capas con una de dos niveles) es que separa hacia fuera el proceso, eso ocurre para mejorar el balance la carga en los diversos servidores; es más escalable.

### Desventajas de las arquitecturas de la n-capas:

- Pone más carga en la red, debido a una mayor cantidad de tráfico de la red.
- Es mucho más difícil programar y probar el software que en arquitectura de dos niveles porque tienen que comunicarse más dispositivos para terminar la transacción de un usuario.

### Ventajas

Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.

Esta centralización también:

- Facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio.

### Desventajas

La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos

problemas para éste (a mayor número de clientes, más problemas para el servidor).

- Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.
- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está *caído*, las peticiones de los clientes no pueden ser satisfechas.
- El software y el hardware de un servidor son generalmente muy determinantes.
- El cliente no dispone de los recursos que puedan existir en el servidor.

### **Cooperación cliente-servidor**

#### **Chained Server**

En esta clasificación de servidores se encuentran aquellos que de vez en cuando se comportan como clientes, ya sea de manera efímera o por mucho tiempo.

#### **Temporal**

Son servidores que por motivos de gestión o para ejecutar cierta parte del proceso de transacción, se comportan como clientes ya sea para solicitar la información requerida al servidor correspondiente o simplemente como un enlace hacia el "destino" de la petición. Sólo realiza la transacción solicitada y vuelve a su estado normal una vez terminada la misma.

#### **Largo Plazo**

Aquellos en los cuales realiza varias transacciones del por mucho más tiempo que la anterior, y se orienta más a la aplicación para la que fue dirigida. ej: servidor WEB.

## **HERRAMIENTAS DE SOFTWARE**

### **Servidor de aplicaciones: Jboss application server**

JBoss Application Server es la implementación de código abierto del paquete de Java EE de los servicios. Se compone de un conjunto de ofertas para los clientes empresariales que buscan perfiles preconfigurados de JBoss Enterprise Middleware componentes que han sido probados y certificados juntos para proporcionar una experiencia integrada.

El fácil usode su arquitectura de servidores y su alta flexibilidad hace a JBoss la elección ideal para los usuarios que acaban de empezar con J2EE, así como arquitectos superiores que buscan una plataforma middleware personalizable.

Debido a que es basada en Java, JBoss Application Server es multiplataforma, fácil de instalar y usar en cualquier sistema operativo que soporte Java.

También le da la flexibilidad para crear versiones personalizadas para su uso personal o comercial.

### **Lenguaje de programación: java**

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la

manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible.

En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995.

#### **Arquitectura tecnológica: jee**

Java Platform, Enterprise Edition o Java EE, es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java

EE; estandarizado por The Java Community Process / JCP.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos.

Ello permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.

#### **Sistema de gestión de base de datos relacional PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma: desinteresada, altruista, libre y/o apoyados por organizaciones comerciales.

Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

#### **Características**

Algunas de sus principales características son, entre otras:

##### **Alta concurrencia**

Mediante un sistema denominado MVCC PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

### **Amplia variedad de tipos nativos**

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas).
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

**Disparadores (triggers):** Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos.

Todos los disparadores se definen por seis características:

- El nombre del disparador o trigger.
- El momento en que el disparador debe arrancar.
- El evento del disparador deberá activarse sobre.
- La tabla donde el disparador se activará.
- La frecuencia de la ejecución.
- La función que podría ser llamada.

### **Funciones**

Bloques de código que se ejecutan en el servidor.

Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de

programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de oracle).
- C.
- C++.
- Java PL/Java web.
- PL/Perl.
- pI PHP.
- PL/Python.
- PL/Ruby.
- PL/sh.
- PL/Tcl.
- PL/Scheme.
- Lenguaje para aplicaciones estadísticas R por medio de PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores.

### **Framework de desarrollo: jsf**

Java Server Faces es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.

JSF usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL. JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de



las páginas y dar soporte para internacionalización y accesibilidad.

- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

Los objetivos de diseño que representan el foco de desarrollo de JSF son:

1. Definir un conjunto simple de clases base de Java para componentes de la interfaz de usuario, estado de los componentes y eventos de entrada.
2. Proporcionar un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML para representar un formulario.
3. Proporcionar un modelo de JavaBeans para enviar eventos desde los controles de la interfaz de usuario del lado del cliente a la aplicación del servidor.
4. Definir APIs para la validación de entrada, incluyendo soporte para la validación en el lado del cliente.
5. Especificar un modelo para la internacionalización y localización de la interfaz de usuario.

Automatizar la generación de salidas apropiadas para el objetivo del cliente, teniendo en cuenta todos los datos de configuración disponibles del cliente, como versión del navegador.

### **Cliente rico: open faces**

OpenFaces es una biblioteca de código abierto de los componentes JSF AJAX, un framework Ajax y un marco de validación en el cliente. OpenFaces se basa en el

conjunto de componentes JSF anteriormente conocido como QuipuKit.

Contiene totalmente revisado código base de QuipuKit e introduce muchos componentes nuevos y características.

OpenFaces se distribuye bajo un modelo de licencia dual. Esto significa que usted puede elegir entre el uso de la biblioteca bajo licencia GNU Lesser General Public License (LGPL) o la compra de una licencia comercial.

### **Entorno de desarrollo: netbeans**

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java.

Existe además un número importante de módulos para extender el NetBeans IDE, que es un producto libre y gratuito sin restricciones de uso.

Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación.

Entre las características de la plataforma están:

- Administración de las interfaces de usuario.

- Administración de las configuraciones del usuario.
  - Administración del almacenamiento.
  - Administración de ventanas.
- Framework basado en asistentes.

## METODOLOGÍA DE DESARROLLO:

### SCRUM

**Scrum** es una metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums. Los casos de estudio provienen de las industrias automovilísticas, así como de fabricación de máquinas fotográficas, computadoras e impresoras.

En 1991 Peter DeGrace y Leslie Stahl en su libro *Wicked Problems, Righteous Solutions*, se refirieron a esta aproximación como *scrum*, un término propio del rugby mencionado en el artículo por Takeuchi y Nonaka.

En 1995 Schwaber y Sutherland, durante el OOPSLA '95 desarrollado en Austin, presentaron en paralelo una serie de artículos describiendo *scrum*, siendo ésta la primera aparición pública de la metodología.

Durante los años siguientes, Schwaber y Sutherland, colaboraron para consolidar los artículos antes mencionados, así como sus experiencias y el conjunto de mejores prácticas de la industria que conforman a lo que ahora se le conoce como *scrum*.

En 2001, Schwaber y Mike Beedle describieron la metodología en el libro *Agile Software Development with Scrum*.

### Características

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Los roles principales en Scrum son el *ScrumMaster*, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que incluye a los desarrolladores.

Durante cada *sprint*, un periodo entre 15 y 30 días, el equipo crea un incremento de software *potencialmente entregable*.

El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del *Product Backlog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*.

Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo.

Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada.

Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de

entregar rápidamente y responder a requisitos emergentes.

Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

### **Roles en Scrum**

En Scrum se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas.

Los 'cerdos' están comprometidos a desarrollar el software de forma regular y frecuente, mientras que todos los demás son 'gallinas' -sólo interesados en el proyecto, pero en realidad indiferentes porque si falla ellos no son los cerdos-, esto es, ellos no fueron los que se comprometieron a hacerlo.

Las necesidades, deseos, ideas e influencias de los roles 'gallina' se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto Scrum.

## **TRATAMIENTO DE AGUAS RESIDUALES**

El tratamiento de aguas residuales consiste en una serie de procesos físicos, químicos y biológicos que tienen como fin eliminar los contaminantes físicos, químicos y biológicos presentes en el agua efluente del uso humano.

El objetivo del tratamiento es producir agua limpia (o efluente tratado) o reutilizable en el ambiente y un residuo sólido o fango (también llamado biosólido o lodo) convenientes para su disposición o reuso.

### **CLASIFICACIÓN DE LOS RESIDUOS**

Los residuos pueden clasificarse de acuerdo a la concentración de sólidos que tiene.

### **CARACTERÍSTICAS DE LAS AGUAS RESIDUALES**

El conocimiento de la naturaleza del agua residual es fundamental de cara al proyecto y explotación de las

infraestructuras tanto de recogida como de tratamiento y evacuación de las aguas residuales, así como para la gestión de la calidad del medio ambiente.

En las aguas residuales se pueden encontrar impurezas como materia en suspensión, material coloidal, o materia en solución; mientras que la materia en suspensión siempre se separa por medio mecánico, con intervención o no de la gravedad, la materia coloidal requiere un tratamiento fisicoquímico preliminar y la materia en solución puede tratarse en el propio estado molecular o iónico o precipitarse y separarse utilizando procesos semejantes a los empleados para la separación de los sólidos inicialmente en suspensión.

A esto es lo que se denomina tratamiento de las aguas.

### **Contaminantes de Importancia en el Tratamiento del Agua Residual.**

Las normas que regulan los tratamientos secundarios están basadas en las tasas de eliminación de la materia orgánica, sólidos en suspensión y patógenos presentes en el agua residual.

Cuando se pretende reutilizar el agua residual, las exigencias normativas incluyen también la eliminación de compuestos orgánicos y otros.

### **Contaminantes de la Industria TEXTIL.**

Los contaminantes más importantes habitualmente presentes en los líquidos residuales industriales textiles son:

- Sólidos.
  - o Suspendidos.
  - o Sedimentables.
- Materia orgánica (DBO5, DQO)
- Temperatura.
- PH.
- Aceites y/o grasas.
- Metales.
- Sulfuros.
- Color.

## CRITERIOS GENERALES PARA LA DESCARGA DE EFLUENTES

El regulado deberá mantener un registro de los efluentes generados, indicando el caudal del efluente, frecuencia de descarga, tratamiento aplicado a los efluentes, análisis de laboratorio y la disposición de los mismos, identificando el cuerpo receptor.

La Entidad Ambiental de Control deberá establecer la normativa complementaria en la cual se establezca: La frecuencia de monitoreo, el tipo de muestra (simple o compuesta), el número de muestras a tomar y la interpretación estadística de los resultados que permitan determinar si el regulado cumple o no con los límites permisibles para descargas a sistemas de alcantarillado.

## COMPONENTES DE UNA PLANTA DE TRATAMIENTO DE AGUAS RESIDUALES

### Tratamiento preliminar:

#### Rejillas

Dispositivos usados para la separación de sólidos de gran tamaño, son barras ubicadas dentro de un canal de sección rectangular y transversalmente a la dirección de la circulación del fluido.

Pueden ser:

- Manuales
- Mecánicas

#### Desarenador

Los desarenadores son dispositivos que se emplean para la separación de partículas suspendidas llamadas “arenas” que deben ser separadas antes de la etapa primaria para proteger los dispositivos de bombeo y otros elementos electromecánicos.

#### Ecuilibración

Los objetivos de las unidades de ecualización son:

- Minimizar la variabilidad de flujo
- Ecuilibrar las concentraciones

Pueden clasificarse en:

- Sistemas de corriente principal “in line”  
Este sistema puede ser:
  - De nivel constante
  - De nivel variable
- Sistemas de corriente lateral “side line”

#### Neutralización

Muchas descargas de líquidos residuales que provienen de procesos industriales presentan variaciones importantes en su acidez o alcalinidad (pH), esto puede provocar inconvenientes como:

- Corrosión en las cañerías de conducción.
- Violación en los límites establecidos para las descargas.
- Impacto sobre el proceso de tratamiento.

### Tratamiento primario:

#### Sedimentación primaria

La función de la sedimentación es remover de la fase líquida, por acción de la gravedad, las partículas orgánicas e inorgánicas, discretas o floculentas que se encuentran en suspensión; los sedimentadores pueden ser: de planta rectangular o circular.

#### Flotación

Su objetivo es remover sólidos suspendidos, grasas y aceites, además puede ser utilizado para espesar barros de origen químico o biológico.

Se clasifica en:

- Flotación natural
- Flotación inducida, esta puede ser: por aire disperso o por aire disuelto.

### Tratamiento Secundario:

#### Procesos Biológicos.

Los procesos biológicos de tratamiento de líquidos residuales son principalmente aerobios o anaerobios.

Es decir, se llevan a cabo en presencia o ausencia de oxígeno.

Cada uno de estos procesos presentan características que pueden ser ventajosas o no

### **Lagunas de estabilización**

Son estanques, excavados parcialmente en el terreno, con un área superficial y volúmenes que aseguren extensos tiempos de tratamiento para degradar la materia mediante autodepuración”.

### **Facultativas**

Las Lagunas Facultativas pueden ser de dos tipos:

- Lagunas facultativas primarias: Estos reservorios reciben las aguas residuales crudas.
- Lagunas facultativas secundarias: Estas lagunas reciben las aguas sedimentadas de la etapa primaria (usualmente el efluente de una laguna anaeróbica).

### **Aerobias**

Estanque artificial al cual se llevan las aguas residuales para sean tratadas mediante un proceso biológico de tipo aerobio de *descomposición de la materia orgánica*.

Son lagunas de baja profundidad, de 30 a 60 centímetros.

### **Sistemas de lagunas de estabilización**

Si se decide utilizar la tecnología de lagunas de estabilización, se recomienda el uso de más de una laguna de tratamiento ya que:

- El tamaño de la planta será menor.
- Es mayor la flexibilidad de la operación y el mantenimiento.

### **Barros activados**

Consiste en un tanque de aireación de forma alargada (ancho/largo = 1:5).

Esta geometría determina que el régimen de flujo hidráulico sea el de flujo pistón.

### **Barros activados mezcla completa**

En este sistema tanto el líquido afluente como el barro biológico recirculado se introducen en distintos sectores de la cámara de aireación a fin de lograr una distribución uniforme de la carga orgánica y de los microorganismos que intervienen en el proceso, como consecuencia el requerimiento de oxígeno es uniforme en todo el reactor.

### **Aireación extendida**

Emplea tiempos de residencia hidráulicos de hasta 24 horas y tiempos de residencia celular  $q_c > 14$  días.

Son procesos que generan relativamente poca biomasa en exceso, pero el requerimiento de oxígeno es mayor, para la degradación de materia orgánica y la nitrificación.

### **Procesos de película biológica**

#### **Lechos percoladores**

Es un proceso de película biológica, que a diferencia de los procesos de tratamiento biológico de las lagunas de estabilización, lagunas aireadas y lodos activados, en los que la biomasa se encuentra suspendida, los microorganismos están adheridos a un medio soporte inerte.

### **Procesos anaerobios de alta carga**

#### **• Reactor UASB**

Los reactores UASB son un tipo de biorreactor tubular que operan en régimen continuo y en flujo ascendente, es decir, el afluente entra por la parte inferior del reactor, atraviesa todo el perfil longitudinal, y sale por la parte superior.

### **Tratamiento terciario:**

El tratamiento terciario se emplea para separar la materia residual de los efluentes de procesos de tratamiento biológico, a fin de prevenir la contaminación de los cuerpos de agua receptores, o bien, obtener la calidad adecuada para el reuso, factor de importancia en la planeación de recursos hidráulicos donde el abastecimiento de agua potable es limitado.

Métodos de tratamiento terciario:

- Osmosis Inversa
- Electrodialisis
- Destilación
- Coagulación
- Adsorción
- Remoción por espuma
- Filtración
- Extracción por solvente
- Intercambio iónico
- Oxidación química

- Precipitación
- Nitrificación / Denitrificación

**Tratamiento de barros:**

- Espesamiento

Proceso que consiste en eliminar parte del agua que llevan los fangos cuando salen

de los tratamientos del agua para que cuando estos ingresen en las siguientes unidades de proceso posean menor volumen y mayor concentración.

## **"DESIGN AND DESIGN OF CONVENTIONAL TREATMENT PLANT WASTE WATER USE WITH FREE SOFTWARE"**

**Problem**

The activities carried out by the man generated wastes produce a negative impact on the environment. These contaminants can be classified primarily according to their state of aggregation, such as gaseous, liquid or solid. We refer specifically to liquid waste that can cause infectious diseases, parasitic and related poisonings.

While these effluents are generated during almost any activity by man, here we will only consider those that are produced during industrial processes, because the discharges can be easily collected and treated in facilities known treatment plants.

The design and sizing of these plants must be performed by the professional water treatment; this design may require a large space of time and higher costs for industries.

Because there is a conventional process can be determined that qualify and quantify based on established design parameters in terms of pollutants and flow, lends itself to

develop software for optimizing time, money and improve business management, professional and municipalities, etc.

The function of the treatment plant is to adapt the physicochemical and biological characteristics of the residual liquid to those established by the legislation concerned, in general the permissible limits set by the inspection body are linked to the body receiving the discharge, for the present analysis, the sewer system.

To make the design and dimensioning of treatment plants should be performed to characterize the wastewater, measure the flow and determine the physicochemical characteristics of the effluent, these data should be compared with the limits laid down in standard roll and finally proceed to design and dimension the treatment plant based on amounts of pollutants and remove them to get to that required by the inspection body.

**Objectives**

**General Objective**

Develop software that allows design and size of conventional treatment plant water parameters based on established design, optimizing time and resources for businesses, professionals and public bodies in the north of the country.

### **Specific Objectives**

- Propose design an appropriate solution in each case of conventional wastewater treatment.
- Dimension the model of the water treatment plant.
- Facilitate the work of businesses, professionals, municipalities and other entities.
- Optimize resources, time and money.
- Meet the needs of "Jirón & Associates." enterprise customers

### **Scope**

This project seeks to develop a program that allows its users to design and dimensioning of conventional treatment plant wastewater to optimize time, money and improve environmental management of businesses, professionals, municipalities and other public and private entities that are within the area influence of the company "Shred & Associates", dedicated to the development and implementation of software in the north, and extend its software offering.

### **Project Justification**

The Constitution of the Republic of Ecuador and other regulations require public and private institutions to carry out the treatment of wastewater to be dumped

at different discharge sources in order to reduce the problems associated with water pollution, infectious diseases, parasitic and related poisonings

Since water is considered essential for life is of great importance to ensure their good condition so it is important that the industries that use it for their production processes looking for a way to minimize the impact of such waste have on effluent produced by mining.

By creating this software aims to give an overview and facilitate the process of design and size of conventional treatment plant water based on design parameters established for the purpose of treating waste water and ensure people and communities access fair, permanent and water quality.

Similar programs can be found on the internet but are developed by foreign companies engaged in the implementation of these plants, plus a cost of between \$ 4995 and \$ 13,875 for purchase and between \$ 1995 and \$ 5550 for annual leave, company data "Chempute software", which, having a program of its kind designed for the "environment" national free software developed considerably reduce the cost.

## **ARCHITECTURE**

### **Client-server**

This architecture is essentially a client making requests to another program

(server) that responds. Although this idea can be applied to programs running on one computer is more advantageous in a multiuser operating system distributed across a computer network.

In this architecture the processing capacity is shared between clients and servers, but more important are the advantages of organizational due to the centralization of information management and separation of responsibilities, which facilitates and clarifies the system design.

The separation between client and server is a logical type separation, where the server does not necessarily run on a single machine and is not necessarily a single program.

Specific types of servers include web servers, file servers, mail servers, etc.

A common arrangement are multilayer systems where the server is broken down into different programs that can be executed by different computers thereby increasing the distribution system.

The client-server architecture replaces the monolithic architecture in which there is no distribution, both physical and logical level.

The network client / server communications network is one in which all customers are connected to a server, which centralizes the various resources and applications to be reckoned with and that makes available to clients whenever they requested.

This type of network can be used together in case of being used in a mixed network.

### **Features**

In architecture C / S the sender of an application is known as a client. Its features are:

- It is the initiator and requests, thus have an active role in communication (teacher or master device).
- Wait and receive responses from the server.
- Generally, you can connect to multiple servers at once.
- Typically interacts directly with end users through a GUI.
- When contracting with a network service, you have to have the connection speed that gives the customer and the type of cable used, for example, round copper wire between 1 ms and 50 ms.

The recipient of the request sent by client is known as a server.

Its features are:

- At the beginning expect to reach the customer requests, then play a passive role in communication (slave).
- Upon receipt of a request, process it and then send the response to the client.
- Generally, accept connections from a large number of customers (in some cases the maximum number of requests may be limited).
- It's not often interact directly with end users.



### **Comparison of architecture C / S with other network architectures**

#### **Comparison with peer networks**

**Peer networks**, also known as peer-to-peer

(Abridged by the acronym P2PM), are another type of network architecture.

#### **Comparison with the architecture-Cola Customer-Client**

While the classical architecture C / S requires one of the communication terminals to act as a server, which can be more difficult to implement architecture-Cola Customer-Client enables all nodes to act as clients simple, while the server acts as a tail that is capturing the customer requests.

This architecture can greatly simplify software deployment. The P2P architecture was originally based on the concept "Client-Queue-Client".

#### **Multi-layer architectures**

The client / server architecture has two generic types of network nodes: clients and servers. Consequently, these generic architectures are sometimes referred to as two-tier architectures or two layers.

Some networks have three types of nodes:

- Clients that interact with end users.
- Application servers that process data for customers.
- Server database that stores the data to application servers.

This configuration is called a three-tier architecture.

#### Advantages of n-tier architectures:

The key advantage of an n-tier architecture compared to a two-tier architecture (or a three-layers with a two-level) is that it separates out the process that happens to better balance the load on different servers, is more scalable.

#### Disadvantages of the architecture of the n-layers:

- It puts more load on the network, due to increased amount of network traffic.

It is much more difficult to program and test the software in two-tier architecture because they have to communicate more devices to complete the transaction of a user.

#### **Advantage**

Centralization of control: access, resources and data integrity are controlled by the server to a client program defective or unauthorized can't damage the system.

This centralization also:

- Facilitates the task of updating data or other resources (rather than on P2P networks).
- Scalability: You can increase the capacity of clients and servers separately.
- Easy maintenance: to be distributed the roles and responsibilities among several independent computers; you can replace, repair, upgrade, or even move a server, while clients are not affected by this change.

#### **Disadvantages**

Traffic congestion has always been a problem in the paradigm of C / S. When a

large number of simultaneous clients send requests to the same server, it might cause many problems for it (to more customers, more problems for the server).

- By contrast, in P2P networks each node on the network server also makes, the more nodes there are, the better the bandwidth you have.
- The paradigm of C / S classic is not the robustness of a P2P network.
- When a server is down, the client requests can't be met.
- The software and server hardware are usually very decisive.

The client does not have the resources that may exist on the server.

#### **Client-server cooperation**

#### **Chained Server**

In this classification of servers are those who occasionally behave as customers, whether ephemeral or so for long.

#### **Temporary**

Servers are based on management or to perform some part of the transaction, act as clients to either request the required information to the appropriate server or simply as a link to the "destination" of the request. Only perform the requested transaction and returns to normal once the same.

#### **Long-Term**

Those in which you perform multiple transactions of a lot longer than the last, and is geared more to the application for which was directed. ej: WEB server.

### **SOFTWARE TOOLS**

#### **Application server: Jboss Application Server**

JBoss Application Server is the open source implementation of Java EE package of services. It consists of a set of offerings for enterprise customers looking for preconfigured profiles of JBoss Enterprise Middleware components that have been tested and certified together to provide an integrated experience.

Easy usode server architecture and high flexibility makes JBoss the ideal choice for users just starting out with J2EE, as well as top architects looking for a customizable middleware platform.

Because it is Java based, JBoss Application Server is multiplatform, easy to install and use on any operating system that supports Java.

It also gives you the flexibility to create customized versions for your personal or commercial.

#### **Programming language: Java**

Java is an object-oriented programming, developed by Sun Microsystems in the early 90's.

The language itself takes much of its syntax from C and C + +, but has a simpler object model and eliminates low-level tools, which often lead to many errors, such as direct manipulation of pointers or memory.

Java applications are typically compiled

into bytecode, although compilation to native machine code is also possible.

At runtime, bytecode is usually interpreted or compiled to native code for execution, although direct hardware execution of bytecode by a Java processor is also possible.

The original and reference implementation of the compiler, the virtual machine and Java class libraries were developed by Sun Microsystems in 1995.

### **Technological Architecture: jee**

Java Platform, Enterprise Edition or Java EE, is a programming platform-part of the Java platform to develop and run software applications in Java programming language with distributed architecture of N layers and is based largely on modular software components running on an application server.

The Java EE platform is defined by a specification. Similar to other Java Community Process specifications, Java EE is also considered informally as a standard because providers must meet certain requirements pursuant to state that their products comply with Java EE, standardized by The Java Community Process / PCB.

Java EE includes several API specifications, such as JDBC, RMI, e-mail, JMS, Web Services, XML, etc. and defines how to coordinate.

This allows the developer to create an enterprise application portable between platforms and scalable, while integrating with legacy technologies.

### **Management system relational database postgresQL**

PostgreSQL is a management system relational database and object-oriented free, released under the BSD license.

Like many other open source projects, the development of PostgreSQL is not handled by a company and / or person, but is run by a community of developers who work: selfless, altruistic, free and / or supported by commercial organizations.

This community is called the PGDG (PostgreSQL Global Development Group).

### **Features**

Some of its main features are included:

#### **High turnout**

Through a system called MVCC allows PostgreSQL as a process writes to a table, others from accessing the same table without locks.

Each user gets a consistent view of the last thing that he did commit. This strategy is superior to the use of locks per table row or common in other bases, eliminating the need to use explicit locks.

## Wide variety of native types

PostgreSQL natively provides support for:

- Arbitrary precision numbers.
- Text of unlimited length.
- Shapes (with a variety of related functions).
- IP addresses (IPv4 and IPv6).
- CIDR Address Blocks style.
- MAC addresses.
- Arrays.

**Triggers:** A trigger or trigger is defined as a specific action is performed according to an event when it occurs within the database.

All triggers are defined by six characteristics:

- The name of the trigger or trigger.
- The time the shutter should start.
- The trigger event must be activated on.
- The table where the trigger is activated.
- The frequency of execution.
- The function could be called.

## Functions

Blocks of code running on the server. Can be written in several languages, with the power that each of them gives, from basic programming operations, such as branching and looping to the complexities of OOP or functional programming.

Triggers are functions linked to operations on data.

Some of the languages that may be used are:

- A personal language called PL / PostgreSQL (similar to PL / SQL oracle).

- C.
- C + +.
- Java PL / Java web.
- PL / Perl.
- pIPHP.
- PL / Python.
- PL / Ruby.
- PL / sh.
- PL / Tcl.
- PL / Scheme.
- Language for R statistical applications using PL / R.

PostgreSQL supports functions that return "row", where the output can be treated as a set of values.

## Development Framework: jsf

Java Server Faces is a technology and framework for Web-based Java application that simplifies the development of user interfaces in Java EE applications.

JSF uses Java Server Pages (JSP) as the technology that allows the deployment of the pages, but can also accommodate other technologies like XUL.JSF includes:

- A set of APIs for representing components of a user interface and manage its state, handle events, validate input, define a navigation scheme of the pages and support for internationalization and accessibility.
- A default set of components for user interface.
- Two custom tag libraries for JavaServer Pages that allow expressing a JavaServer Faces interface within a JSP page.
- A model of events in the server side.
- State Management.
- Managed Beans.

The design goals that represent the focus of JSF development are:

1. Define a simple set of base classes for Java interface components, user

- status of components and input events.
2. Provide a set of components for user interface, including standard HTML elements to represent a form.
  3. Provide a JavaBeans model for sending events from the controls on the user interface on the client side to server application.
  4. Define APIs for input validation, including support for validation on the client side.
  5. Specify a model for internationalization and localization of the user interface.
  6. Automating the generation of outputs appropriate for the client's objective, taking into account all the configuration data available to the client as the browser version.

#### **Rich Client: open faces**

OpenFaces is an open source library of AJAX JSF components, an Ajax framework and a framework for client-side validation.

OpenFaces is based on the set of JSF components formerly QuipuKit.

Contains fully revised codebase QuipuKit and introduces many new components and features.

OpenFaces is distributed under a dual licensing model. This means you can choose between using the library under the GNU Lesser General Public License (LGPL) or buy a commercial license.

#### **Development environment: netbeans**

NetBeans is a development environment, made primarily for the Java programming language.

There is also a significant number of modules to extend the NetBeans IDE, a

free product and free with no usage restrictions.

Sun Microsystems founded the NetBeans open source project in June 2000 and remains the main sponsor of the projects.

A module is a Java file that contains Java classes written to interact with the NetBeans APIs and a special file (manifest file) identifying it as a module. Applications built from modules can be extended by adding new modules.

The platform offers services common to desktop applications, allowing the developer to focus on application-specific logic.

Among the characteristics of the platform are:

- Administration of user interfaces.
- Managing user settings.
- Storage Management.
- Window Management.
- Framework based on attendance.

#### **METHODOLOGY DEVELOPMENT: SCRUM**

Scrum is a methodology for software development and management based on an iterative and incremental process commonly used in environments based on agile software development.

Although Scrum was focused on the management of software development processes, can be used in equipment

maintenance software, or a program management approach: Scrum of Scrums. The case studies come from the automotive and manufacturing of cameras, computers and printers.

In 1991 Peter and Leslie Stahl Degrace in his book *Wicked Problems, Righteous Solutions* referred to this approach as scrum, rugby itself a term mentioned in the article by Takeuchi and Nonaka.

Schwaber and Sutherland in 1995 during the OOPSLA '95 developed in Austin, appeared in parallel a series of articles describing scrum, marking the first public appearance of the methodology.

During the following years, Schwaber and Sutherland collaborated to consolidate the articles and their experiences and set of best practices in the industry that make up what is now known as scrum.

In 2001, Schwaber and Mike Beedle the methodology described in the book *Agile Software Development with Scrum*.

### **Features**

Scrum is a reference model that defines a set of practices and roles, and can be taken as a starting point for defining the development process to be implemented during a project.

The main roles in Scrum are the ScrumMaster who maintains the processes and works similarly to the project manager, the ProductOwner, representing stakeholders (internal or external

stakeholders) and the Team which includes the developers.

During each sprint, a period between 15 and 30 days, the team creates an increment of potentially shippable software. The set of features that go into each sprint come from the Product Backlog, which is a set of prioritized high-level requirements that define the work to be done. The Product Backlog items that are part of the sprint is determined during the **Sprint Planning meeting**.

During this meeting the Product Owner identifies the elements of product backlog that he wants completed and makes known to the team.

The team then determines the amount of this they can commit to complete during the next sprint. Durante the sprint, no one can change the sprint backlog, which means that the requirements are frozen for sprint. A key principle of Scrum is the recognition that during a project the customers can change their minds about what they want and need, and unpredictable challenges that can't easily be faced in a predictive and planned.

Therefore, Scrum adopts a pragmatic approach, accepting that the problem can't be fully understood or defined, focusing on maximizing the team's ability to deliver quickly and respond to emerging requirements.

One of the biggest advantages of Scrum is that it is very easy to learn, and requires very little effort to start using.

### **Scrum Roles**

In Scrum defines various roles, these are divided into two groups: pigs and chickens.

The 'pigs' are committed to building software regularly and frequently, while everyone else is 'chicken', only interested in the project but really indifferent because if they are not fault-pigs, that is, they which were not agreed to do so.

The needs, desires, ideas and influences of the roles 'chicken' is taken into account, but are not likely to affect, distort or hinder the Scrum project.

### **WASTEWATER TREATMENT**

The wastewater treatment consists of a series of physical processes, chemical and biological agents that are intended to remove contaminants physical, chemical and biological agents present in the effluent water from human use.

The goal of treatment is to produce clean water (or treated effluent) or reusable in the environment and solid waste or sludge (also called biosolids or sludge) suitable for disposal or reuse.

### **CLASSIFICATION OF WASTE**

Residues can be classified according to the concentration of solids that have.

### **CHARACTERISTICS OF WASTEWATER**

Knowledge of the nature of the wastewater is fundamental in order to project and operation of the infrastructure of both collecting and treatment and disposal of waste water and for the management of environmental quality.

In wastewater can be found as impurities suspended matter, colloidal material, or matter in solution, while always suspended matter is removed by mechanical means, involving or not the severity, colloidal matter requires a preliminary physicochemical treatment and matter in solution may be in the molecular or ionic state itself or precipitated and separated using processes similar to those used for the initial separation of solids in suspension.

This is what is called water treatment.

### **Importance Contaminants in Wastewater Treatment.**

The rules governing secondary treatments are based on the removal rates of organic matter, suspended solids and pathogens present in wastewater.

When trying to reuse the waste water, the regulatory requirements also include the removal of organic compounds and others.

### **Contaminants in the TEXTILE industry.**

The major pollutants are normally present in the liquid industrial waste textiles are:

- Solid.
  - Suspended.
  - Settleable.
- Organic matter (BOD5, COD)
- Temperature.
- PH.
- Oil and / or fat.
- Metals.
- Sulphides.
- Color.

### **GENERAL CRITERIA FOR DISCHARGE EFFLUENT**

The regulated must keep a record of effluent generated, indicating the flow of effluent discharge frequency, applied to the

effluent treatment, lab work and arrangement thereof, identifying the receiving body.

Environmental Control Entity shall establish complementary legislation which is set: The monitoring frequency, sample type (simple or compound), the number of samples to be taken and the statistical interpretation of the results to determine if the regulated is complying with the permissible limits for discharges to sewer systems.

## **COMPONENTS OF A TREATMENT PLANT WASTEWATER**

### **Preliminary treatment: Grids**

Devices used to separate large solids, bars are located within a channel of rectangular section and transversely to the direction of fluid flow.

They can be:

- Manuals
- Mechanical

### **Sand Trap**

The grit chambers are devices that are used for separation of suspended particles called "grit" that must be separated before the primary stage to protect the pumping devices and other electromechanical elements.

### **Equalization**

The objectives of the equalizer units are:

- Minimize flow variability
- Equalize the concentrations

Can be classified into:

- Systems mainstream "in line"

This system can be:

- In constant level

- In variable level
- Systems sides treat "side line"

### **Neutralization**

Many liquid waste discharges from industrial processes varies significantly in its acidity or alkalinity (pH), this can cause problems such as:

- Corrosion in the pipes of driving.
- Rape in the limits for discharges.
- Impact on the treatment process.

### **Primary treatment: Primary sedimentation**

The function of sedimentation is removed from the liquid phase by action of gravity, organic and inorganic particles, discrete or flocculating that are in suspension, the sedimentation may be: a rectangular or circular.

### **Flotation**

Its aim is to remove suspended solids, fats and oils can also be used to thicken sludge biological or chemical origin.

It is classified into:

- Natural Flotation.
- Flotation induced, this may be: air dispersed or dissolved air.

### **Secondary Treatment:**

#### **Biological Processes.**

The biological processes of sewage treatment are mainly aerobic or anaerobic.

That is are carried out in the presence or absence of oxygen.



Each of these processes have characteristics that could be advantageous or not.

### **Stabilization ponds**

There are ponds, partially excavated in the ground, with a surface area and volume to ensure comprehensive treatment times to degrade the art by self-purification. "

### **Facultative**

Facultative Lagoons are of two types:

- primary facultative ponds: These reservoirs are the raw sewage.
- secondary facultative ponds: These ponds are sedimented waters of the primary stage (usually the effluent from an anaerobic lagoon).

### **Aerobic**

Artificial pond which will carry the wastewater to be treated by an aerobic biological process of decomposition rate of organic matter.

They are shallow lagoons, 30 to 60 centimeters.

### **Stabilization pond systems**

If you choose to use the stabilization pond technology is recommended to use more than one treatment lagoon as:

- The size of the plant will be less.
- The greater the flexibility of operation and maintenance.

### **Activated sludge**

It consists of an aeration tank of elongated shape (width / length = 1:5).

This geometry determines the flow regime is the flow hydraulic piston.

### **Activated sludge mixed**

In this system both the liquid effluent as organic mud recirculated are introduced in various sectors of the aeration chamber to achieve a uniform distribution of organic matter and microorganisms involved in the process, because the oxygen requirement is uniformly throughout the reactor.

### **Extended aeration**

It uses hydraulic residence times of up to 24 hours and cell residence times  $q_c > 14$  days.

These processes generate relatively little biomass in excess, but the oxygen requirement is higher for organic matter degradation and nitrification.

### **Biofilm processes**

#### **Trickling beds**

Is an organic film process, unlike the biological treatment processes of the stabilization ponds, aerated lagoons and activated sludge, in which the biomass is suspended, the microorganisms are attached to half inert support.

### **High load anaerobic processes**

#### **• UASB Reactor**

UASB reactors are a type of tubular bioreactor operating in continuous and in upstream, ie, the influent enters the bottom

of the reactor, passes through all the longitudinal profile, and exits at the top.

**Tertiary treatment:**

Tertiary treatment is used to separate the remaining material from the effluent of biological treatment processes in order to prevent contamination of receiving water bodies, or get the right quality for reuse factor of importance in planning water resources where water supply is limited.

Tertiary treatment methods:

- Reverse Osmosis
- Electrodialysis
- Distillation
- Coagulation
- adsorption
- Removal of foam
- Filtration
- Solvent Extraction
- Ion exchange
- Chemical Oxidation
- Precipitation
- Nitrification / Denitrification

**Treatment of sludge:**

- Thickening

The process of removing some of the water carried by the sludge treatment when they leave the water so that when they enter into the following process units having lower volume and higher concentration.