

## **Primer Proyecto, primera parte.**

**Uno. Objetivo:** Implementar una base de datos física, así como escribir el código en capa lógica y física para el mantenimiento de entidades y el procesamiento de datos.

### **Dos. Descripción.**

#### **Facturación de servicios y recolección de impuestos en una municipalidad.**

La municipalidad ofrece servicios de proveer agua, recolectar basura, mantenimiento de parques, y además recolecta impuestos a la propiedad y a las patentes. Los servicios se pagan mensualmente. Los impuestos, aunque su valor se calcula anualmente, se pagan en 4 tractos trimestrales: en marzo, junio, setiembre y diciembre.

Hay otros servicios que se pagan de manera no recurrente, por ejemplo: reconexión de agua, cuotas de un arreglo de pago sobre una deuda con la municipalidad, y otros conceptos. Para el cobro de estos servicios, un usuario de la municipalidad ingresa estos conceptos para que sean pagados por un ciudadano específico.

Un propietario puede ser una persona física, o una persona jurídica, y puede ser dueño de una o varias propiedades en el municipio, los cuales pagan los impuestos y servicios recurrentes. Si un propietario es una persona jurídica, por ejemplo, una empresa, también existirá una persona física que es el apoderado o representante de la persona jurídica.

Cada propiedad, tiene asociado un estado de cuenta, en el cual se pueden visualizar todos los servicios e impuestos ya pagados, o aquellos que están pendientes de pago.

A través de un portal web de la municipalidad, un usuario podrá consultar los estados de cuenta de cada una de sus propiedades, aquellas de las que es dueño, o las propiedades cuyo dueño son personas jurídicas donde el usuario es apoderado.

Para pagar, el usuario visita el portal, selecciona una propiedad, y luego el estado de cuenta, que son los despliega los servicios pendientes de pago en orden cronológico del más viejo al más nuevo. Cada ítem del estado de cuenta muestra, fecha del 'recibo', concepto (nombre del servicio o impuesto), monto, fecha de vencimiento, y monto de los intereses moratorios. Los intereses moratorios serán mayores a cero, si la fecha actual es mayor a la fecha de vencimiento del ítem. Se muestra un total de todo lo pendiente, la cantidad de ítem pendientes y el monto total de los intereses moratorios.

El usuario selecciona aquellos ítemes (que también podemos llamar recibos) que quiera pagar, teniendo en cuenta que el sistema NO permitirá pagar un mes referido a un servicio, por ejemplo, agua; si algunos de los meses anteriores al mes que se desea pagar está pendiente o no está seleccionado.

Todo servicio, tiene una fecha del “recibo” y una fecha vencimiento. Para cada servicio, se tiene el día del mes que se pone al cobro, se tiene una cantidad de días para pagar antes de vencer, y la tasa anual de intereses moratorios si el propietario paga después de la fecha de vencimiento. Por ejemplo, para el servicio de agua, la fecha del recibo serán los 5 del mes, la cantidad de días para que el recibo venza son 15, entonces el recibo se emite los 5 y debe pagarse antes los 20 de cada mes. La fecha del recibo refiere al servicio del mes previo, ejemplo: el “recibo” de agua del 5 de diciembre, es el recibo por el servicio de agua de noviembre. Si el usuario pagó luego del 20 de diciembre, se cobran intereses moratorios.

Para el cobro de los intereses moratorios, se genera un nuevo recibo, ligado un concepto llamados “Intereses Moratorios”, y en la descripción debe indicar, “Intereses Moratorios de recibo por <Concepto de Cobro del recibo que generó los intereses moratorios> de fecha <fecha del recibo que genera intereses moratorios>”

El impuesto a la propiedad es de 0.5% del valor declarado de la propiedad. Los valores para patentes, recolección de basura, mantenimiento de parques y reconexión de agua, son atributos del tipo de servicio.

Hay 2 tipos de usuario, el usuario propietario, del cual ya hablamos y el usuario administrador, que ingresa a un portal en el cual puede hacer crud de las entidades principales, las cuales son:

- Propietarios

- Propiedades

- Asociación de propietarios con propiedades (Puede consultar las propiedades de un propietario, o para una propiedad su propietarios)

- Usuarios

- Asociación de usuarios con propiedades. (Puede consultar las propiedades que puede ver un usuario: o, para una propiedad, quiénes son sus usuarios)

- Asociación de propiedades con servicios.

- Servicios o impuestos.

Un usuario administrador puede ver todas las propiedades y propietarios.

### **Tres. ¿Qué funcionalidades debe implementar?**

+El portal web para el usuario administrador, desde donde podrá hacer crud de las entidades indicadas anteriormente, así como las siguientes consultas: propiedades de un propietario (se ingresa el nombre del propietario o su identificación, se listan sus propiedades), propietario de una propiedad (se ingresa el número de finca, se lista los datos de la propiedad), propiedades que puede ver un usuario (se ingresa el nombre del usuario, y se listan las propiedades) y usuario de una propiedad (se ingresa el número de finca, se listan los nombres de usuarios que pueden ver la propiedad).

#### Cuatro. ¿Qué se pide?

- a) El código en capa lógica para la creación del sitio del administrador, con la funcionalidad y la interfaz necesaria para realizar CRUD de las entidades y la consultas. Aclaración: Para esta primera entrega NO hay que hacer sitio web para el usuario no-administrador.
- b) Los SP para realizar el crud de todas las entidades indicadas en la descripción, así como la realización de consultas.
- c) Script de llenado de catálogos. Los catálogos se crean a partir del procesamiento de 3 archivos xml para la inserción de Conceptos de Cobro, Tipos de documento y usuarios administradores, se proveen ejemplos de la estructura XML:

Concepto de cobro

```
<conceptocobro id="1" Nombre="Agua" DiaCobro="25" QDiasVencimiento="10"
EsImpuesto="No" EsRecurrente="Si" EsFijo="No" TipoCC="CC Consumo" Monto=""
ValorM3="500" ValorPorcentaje="" \>
<conceptocobro id="2" Nombre="Patente Licores" DiaCobro="8"
QDiasVencimiento="12" EsImpuesto="Si" EsRecurrente="Si" EsFijo="Si" TipoCC="CC
Fijo" Monto="50000" ValorM3="" ValorPorcentaje="" \>
<conceptocobro id="3" Nombre="impuesto Propiedad" DiaCobro="8"
QDiasVencimiento="12" EsImpuesto="Si" EsRecurrente="Si" EsFijo="No" TipoCC="CC
Fijo" Monto="" ValorM3="" ValorPorcentaje="5.0" \>
```

Tipo de documento, ejemplo:

```
<TipoDoc IdcodigoDoc="1" descripcion="Cedula Nacional"/>
<TipoDocId codigoDoc="2" descripcion="Cedula Residente"/>
```

Para usuarios administradores, ejemplo:

```
<UsuarioAdmi Nombre="Jquiros" password="123hola" tipo="administrador"/>
<UsuarioAdmi Nombre="Jperez" password="123mama" tipo="administrador"/>
```

- d) Un script de simulación de operación. Este script se encarga de procesar un archivo XML, el cual tienen nodos tipo "fecha operación", para fechas que van en forma ascendente desde una fecha inicial a una fecha final, dentro del nodo de cada fecha, hay sub-nodos XML para la realización de operaciones de inserción a las diferentes entidades de la BD, que son dinámicas, en el cual para un rango de fechas, se itera en

cada fecha, en forma ascendente de la fecha menor a la fecha mayor, y en cada fecha se realiza operaciones para:

- Inserción de una propiedad
- Inserción de un propietario
- Inserción de una asociación entre propietario y propiedad.
- Inserción de responsable de propietario que es persona jurídica.
- Inserción de usuario.
- Inserción de asociación entre usuario y propiedad.

Ejemplo del nodo xml para una fecha:

```
<OperacionDia fecha="10/04/2020">
<Propiedad NumFinca="1176180" Valor="123.484.710,00" Direccion="Condominio Mar
del Plata, Filial 186"/>
<Propietario Nombre="ERICK ALONSO ALFARO HERNANDEZ" TipoDocIdentidad="1"
identificacion="304110067" />
<Propietario Nombre="LOLA PEREZ ROJAS" TipoDocIdentidad="1"
identificacion="110201818" />
<PropiedadVersusPropietario NumFinca="1176180" identificacion="304110067"/>
<PersonaJuridica docidPersonaJuridica="123456789" TipDocIdRepresentante="1"
DocIdRepresentante="110201818" Nombre="Lola Perez Rojas" />
<Usuario Nombre="PCanalla" password="mgX8Xj3xVY" tipo="normal" />
<UsuarioVersusPropiedad NumFinca="1176180" nombreUsuario="PCanalla"/>
<ConceptoCobroVersusPropiedad idcobro="Recoleccion de Basura"
NumFinca="1176180" FechaInicio="01/03/2020" />
<ConceptoCobroVersusPropiedad idcobro=2 NumFinca="1176180"
FechaInicio="01/03/2020" />
</OperacionDia>
```

Inserción de recibos: No serán considerados en la tarea 1.

Respecto de los datos, tome en cuenta los siguientes cantidades: 200 propiedades, 50 personas físicas, 10 personas jurídicas, 20 usuarios, y la asociación entre todas las entidades.

e) La documentación que es una bitácora.

### **Cinco. Prueba de los proceso de scripts que procesan los archivos XML.**

Para la revisión del proyecto, se deben correr los scripts para llenado de catálogos y la simulación de la operación del sistema, sobre una base de datos vacía, para el caso de la primera tarea, solamente consiste en la inserción de entidades. Se revisa el código de los SP y se hace prueba de la interfaz de usuario para el crud de las entidades dinámicas.

**Seis. Reglas.** Documentación: una bitácora escrita en un blog en el cual día a día, se indica la cantidad de horas trabajadas por el equipo de trabajo, se hace un relato de los avances, acerca de los problemas encontrados, como fueron resueltos; dudas, divergencias de criterio, forma en que trabajó el equipo de trabajo, problemas con la instalación del software, problemas de aprendizaje del framework, investigaciones, pruebas de concepto, experiencias, moralejas, ayuda recibida, consejos a dar, buenas prácticas descubiertas, etc.; que describa el proceso de solución de la tarea programada; una descripción sincera y detallada será bien evaluada. Incluya referencias externas a recursos utilizados en internet para solventar dudas o resolver problemas, puede incluir los mensajes de error y cómo fueron resueltos. La última entrada del blog es un resumen que indique la cantidad de horas total de trabajo, por persona, así como un análisis de resultados de los que se está entregado. Debe ser exhaustivo. Todo el código referido a base de datos, debe ser un procedimiento almacenado. Grupos de 2 personas. Motor de base de datos: MS SQL cualquier versión superior a 2014. Código en capa lógica, en el lenguaje o framework de su preferencia. **Fecha de entrega: 30 de Mayo.**