Lo aprendido en estas dos últimas clases fue sobre el lenguaje de marcas XML, la tecnología DOM, algunas cosas sobre Java, como son los namespace, clases y objetos, encapsulamiento; por otro lado, también aprendí sobre algunos conceptos relacionados a base de datos como el modelo de datos, tipos de usuarios en Oracle ,la creación de tablas en SQL y un poco sobre los servidores proxy.

Lenguaje de Marcas

Un lenguaje de marcas sirve para codificar un documento de tal manera que el ordenador o computadora pueda entenderlo y utilizarlo, este lenguaje contiene etiquetas o marcas que contienen información adicional acerca de la estructura del texto o presentación del documento. La diferencia entre un lenguaje de marcas y un lenguaje de programación es que en el lenguaje de programación se puede hacer operaciones aritméticas, mientras que en el lenguaje de marcas solo contiene etiquetas de un documento. Un ejemplo de lenguaje de marcas es el html, xml entre otros.

XML

El lenguaje XML es un lenguaje de marcas extensible, este está diseñado para ser autodescriptivo, debido a que las etiquetas puede contener información sobre los elementos por los cuales está estructurada. Una de las características importantes del XML es que no solo almacena textos, sino también puede almacenar videos, fotos, etc.

Este tipo de lenguaje está compuesta por una estructura de árbol o también llamado árbol de nodos, en el cual el nodo principal sería el nodo raíz, la cual es una especie de entidad que guarda diferentes elementos, estos elementos se dividen en nodos padres y nodos hijos, por lógica, los nodos padres son los que contienen a estos nodos hijos.

JSON

Si bien es cierto, XML es un lenguaje fácil de codificar en comparación a otros lenguajes de marca (html), existe un acrónimo de javascript, llamado JSON, el cual significa notación de objetos de javascript, esta notación es mucho más simple de comprenderla tanto para la persona que codifica como para el computador que lo lee e interpreta. El uso de esta notación ha sido generalizada como alternativa al XML, de ahí que JSON es considerado un metalenguaje de marcado.

Al comparar al JSON con el XML, podemos decir que el primero es un formato ligero de intercambio de datos, el segundo es utilizado para almacenar datos, este da soporte a base de datos, lo cual sirve de mucho cuando varias aplicaciones quieren comunicarse entre sí o integrar información.

Otra característica importante a mencionar es que JSON solo almacena textos, en cambio XML puede almacenar texto, videos y otra clase de archivos. En el caso del XML, puede incluso, almacenar ejecutable, pero esto puede ser una desventaja, ya que puede cargar ejecutables malignos y perjudicar la

información almacenada, lo cual no ocurre con JSON, debido a que este solo almacena texto.

De acuerdo a lo mencionado, se deduce que el XML pesa más que el JSON, al ser transportado mediante el http, es por ello que el JSON es más rápido.

A continuación se mostrará un ejemplo de cómo se representa una entidad, con sus respectivos atributos en XML y JSON.

JSON:

```
{"Casa": {
  "Familia": [
    {“Categoría": "Adultos", "Nombre": "Faustino", "Edad": "58 “},
    {“Categoría": "Niños", "Nombre": "Pulgarcito", "Edad": "8 “}
  ]
 }
}
```

En JSON las estructuras se consideran como arreglos, es por eso que familia, considerada como arreglo va entre corchetes.

XML:

```
<Casa >
  <Familia Categoría="Adulto" > <!--Categoría vendría a ser un atributo -->
    <Nombre>Faustino</Nombre> <!-- Adulto sería el valor de este atributo-->
    <Edad>58</Edad> <!--Nombre y Edad, serían los nodos hijos de Familia-->
  </Familia
  <Familia Categoría="Niño">
    <Nombre>Pulgarcito</Nombre>
    <Edad>8</Edad>
  </Familia
</Casa>
```

MODELO DOM

El modelo DOM, es un modelo en objetos para la representación del documento, específicamente es una interfaz de programación de aplicaciones (API), esta proporciona un conjunto estándar de objetos para representar documentos HTML y XML. Además de esto, DOM se creó con el objetivo de

manipular de forma sencilla documentos XML, de ahí que DOM también manipula documentos XHTML como si fueran XML.

La jerarquía de DOM es similar a la jerarquía de HTML, es en árbol de nodos, por ejemplo si se quiere referir a todo el documento, entonces ingreso al nodo document y de inmediato logro acceder a todo el documento; además de eso de repente solo quiero acceder a una parte del documento, por ello existen las siguientes jerarquías(nodos padres e hijos) en el DOM, en 1er orden, el nodo document, en 2do orden, nodo html, en 3er orden, los nodos head y body, entre otros.
Si nos percatamos, estos nodos vienen a ser una representación del contenido de la página web.

El DOM no me permite modificar exclusivamente una especificación html, sino que mediante el javascript, generalmente usado por el DOM en el html, puedo modificar, accediendo a los nodos de los documentos xml, con casi la misma forma que en el html.

CLASES Y OBJETOS EN JAVA

En el java, el namespace es un package o paquete, este namespace o paquete es un contenedor. El java te crea un directorio o carpeta, pero en otros lenguajes no necesariamente te va a crear una carpeta.

Para la declaración del package en el JAVA siempre se va a colocar en la primera línea, siempre y cuando esa primera línea no sea comentario, si así fuese entonces iría inmediatamente después de este comentario. Para el nombre del package, si ponemos como ejemplo al nombre de una página web, el package lo nombra al revés (de derecha a izquierda), por ejemplo si usamos de ejemplo a “http:// es.wikipedia.org/wiki/Wikipedia:Portada” en el package iría lo siguiente:
package Wikipedia:Portada.wiki.org.wikipedia.es , la cual indica que el elemento que se encuentra a la izquierda del punto está contenido por la carpeta del elemento que está inmediatamente a la derecha, y este a su vez por el que le sigue, así sucesivamente.

Respecto a las clases, para nosotros saber el nombre calificado de una clase sería el siguiente "nombre_clase.nombre_package". Ahora bien, las clases pueden tener el mismo nombre y con este nombre calificado se diferencian de las demás; además, las clases tienen dos modificadores de acceso que son el "private" y el "public", lo cual por conocimientos básicos de inglés, se entiende que la clase que tenga el modificador "private", no podrá ser accedida por otra clase; lo contrario sucede con el modificador "public", a la clase que contenga este modificador sí se podrá acceder. El nombre de una clase en java cumple la regla de camelcase, o sea la primera letra va con mayúscula

Pero una clase no está vacía, esta tiene atributos y métodos, los cuales también poseen modificadores de acceso, a comparación de las clases, los atributos tienen 4 modificadores de acceso, public, default, protected, private, ordenados en accesibilidad. En el UML los modificadores de acceso se

representan de la siguiente manera: public: "+" , default: " " o "~" , protected: "#", private:"-" .

Cada modificador del atributo tiene un significado:

-public.- todas las clases pueden acceder a ese atributo.

-default.- solo pueden acceder a ese atributo las clases que pertenezcan a ese mismo paquete.

-protected.- solo tienen acceso a este atributo, los que estén dentro de la clase y los elementos heredados.

-private.- solo la misma clase tiene acceso a ese atributo.

Encapsulamiento:

Este es un concepto que consiste en la ocultación de ciertos atributos de una clase, de manera que solo es posible acceder a estos atributos mediante los métodos definidos para dicho objeto, generalmente estos métodos son el get y set, los cuales son métodos para obtener y asignar, respectivamente, un valor al atributo relacionado con este método. Estos atributos son declarados con el modificador de acceso private y los métodos set y get son declarados generalmente con el modificador de acceso public.

Para poder acceder a los valores de estos atributos, lo único que tenemos que hacer es instanciar la clase anterior y luego, mediante esta, llamar al método get (obtener un valor) o set (asignar un valor), dependiendo de lo que se quiera.

MODELO DE DATOS

Un modelo de datos es un lenguaje orientado a hablar de una base de datos. Generalmente un modelo de datos permite describir la estructura, las restricciones de integración y las operaciones de manipulación de los datos de dicha base. Otro modo de definir al modelo de datos sería que este permite describir los elementos de la realidad que intervienen en un problema y la forma en que se relacionan.

Los modelos de datos, como bien ya han sido definidos, se clasifican en modelo conceptual, modelo lógico y modelo físico.

El modelo conceptual es el modelo de datos de alto nivel usado para entender los requerimientos de los datos, o sea con esto se logra entender mejor la realidad. Están orientados a representar a los elementos que intervienen y sus relaciones.

El modelo lógico es el modelo de datos de nivel intermedio que se usa para la representación lógica de los datos. Algunos ejemplos de modelos son el modelo relacional, jerárquico y red. Están orientados a las operaciones más que a la descripción de una realidad. Están implementados usualmente en algún manejador de base de datos.

El modelo físico es el modelo de datos de bajo nivel que se usan para de describir como se implementa la base de datos, o sea describen el formato de registro y estructura.

MODELO E-R

El modelo Entidad-Relación es un tipo de modelo de datos conceptual de alto nivel que se emplea en el diseño de las bases de datos relacionales. El modelo entidad-relación muestra la estructura de la base de datos, enfocándose en las entidades relevantes de un sistema de información así como sus interrelaciones y atributos.

Para construir el modelo E-R primero debemos de tener en cuenta los requerimientos de los usuarios y analizarlos, luego de esto se crea un esquema conceptual de la base de datos, el cual debe contener la descripción de las entidades, relaciones y restricciones; el último paso es implementarlo en algún software de gestión de base de datos adaptado al modelo E-R.

Los elementos del modelo E-R son:

Entidad: Representa a un objeto con existencia física o conceptual (entidad abstracta). Estas son independientes, no dependen de nada para existir.

Se representa con un rectángulo.

Relación: Defina la afiliación entre una o más entidades.

Se representa con un rombo.

Atributos: Es la información o características que identifican a una entidad. Estos no pueden ser vectores, solo puede almacenar un solo valor.

Se representa con óvalos.

TIPOS DE USUARIOS EN EL ORACLE

Existen dos grupos de usuarios, cuando nos conectamos al Oracle, están el grupo de usuarios que solo se puede conectar a la base de datos cuando está abierta y el otro grupo de usuarios que se pueden conectar a la base de datos estando, esta última, abierta o cerrada.

En este último grupo se encuentran los usuarios administradores que son el SYS y el SYSTEM, a estos se les otorga el rol DBA, el cual contiene la mayoría de privilegios del sistema de base de datos.

Los privilegios de administrador se caracterizan porque se puede hacer operaciones de seguridad como arranque y parada de base de datos. Estos privilegios son el SYSDBA y el SYSOPER.

Una de las características más interesantes del SYSDBA, por ejemplo, es que con este se puede usar el comando GRANT y REVOKE, los cuales sirven para

dar acceso a un usuario, el caso del GRANT, y quitarle el derecho a uno o más usuarios, el caso del REVOKE.

Ejemplo:

User: sys as sysdba

Password:

GRANT select on Personal_Information to Liz with GRANT option;
/* esto significa que le estoy otorgando el derecho “select” en la tabla Personal_Information al usuario “Liz” y además le doy opción a que ella le otorgue el derecho a otros usuarios*/

GRANT select on Personal_Information to Juan;

/* Esto significa que le estoy otorgando el derecho “select” en la tabla Personal_Information al usuario “Juan” */

REVOKE select on Personal_Information from Juan;

/*Esto significa que le estoy quitando el derecho “select” en la tabla Personal_Information al usuario “Juan” */

TABLAS EN ORACLE

Las tablas son una especie de entidades, es la unidad básica de almacenamiento, ahí vas a almacenar todo lo que tiene tu modelo E-R

Reglas para crear una tabla:

Debe empezar con una letra, tener 30 caracteres como máximo y uno como mínimo, debe contener solo letras del abecedario (mayúsculas y/o minúsculas, números y los caracteres especiales como "_", "$", "#".

Los nombres nunca pueden ser duplicados por el mismo usuario, no se debe usar palabras reservadas propias del Oracle, como por ejemplo el "Select".

Ahora bien para crear una tabla en el SQL, primero debemos ingresar ya sea mediante la consola o por el intelij idea, el comando “sqlplus” (en el caso de la consola), luego el nombre del usuario y su password, luego de esto procedemos a crear la tabla con la siguiente estructura:

```
create table name_of_table
( name_of_col1 datatype( value1 ),
  name_of_col2 datatype( value2 ),
  name_of_col3 datatype( value3 ),
);
```

De esa forma se crea una tabla; por otro lado, para poder consultar utilizamos el comando "select":

select * from name_of_table;

Para insertar un valor en una tabla se utiliza el comando “insert into”:

insert into name_of_table values(value1,value2,value3)

Para guardar las tablas en la base de datos, se usa el comando "commit", el cual refresca (guarda) la base de datos:

commit; /*aquí indica que se está grabando la base de datos*/

Ejemplo de creación de tablas en Oracle:

SQL> create table personal_information (full_name varchar2(10), age int(3), salary number(10,3)); /*el 10 indica el total de cifras que puede llegar a tener y 3 indica el número de cifras decimales que puede llegar a tener el elemento salary */

SQL>insert into personal_information (‘Junior Huamani Rojas’, 20, 8500);

/*Aquí se está insertando una fila de acuerdo al orden de columnas como fue creada la tabla; es decir primero va el nombre, luego la edad y por último el sueldo*/

SQL>commit; /*se está guardando*/

SQL>exit /*con esto se sale del SQL(base de datos)*/

Servidores Proxy

Un servidor proxy es un programa o sistema informático, que sirve de intermediario para las peticiones de un cliente A hacia otro servidor, llamado C, o sea la petición se hará de A hacia B(servidor proxy) y este último transmitirá la petición hacia C.

Este servidor proxy nos ayuda a mantener el anonimato con respecto a la petición que hagamos hacia C, por ejemplo si yo, usuario, le quiero pedir información, desde mi ordenador (cliente A), sobre una página web al servidor C, entonces este último no sabrá la procedencia de esta petición, gracias al servidor intermediario “Proxy”.

Su finalidad es permitir el acceso a internet a todos los equipos de un sistema cuando solo se puede disponer de un único equipo conectado; es decir, una única dirección IP.

Otros beneficios de los servidores Proxy, son:

- Optimizan los recursos, ya que se reduce la carga de trabajo de los servidores.

- Aumentan el rendimiento de algunas operaciones sirviendo como caché.

- Pueden crear una barrera de seguridad, evitando solicitudes no autorizadas o filtrado de datos.

Existen diferentes tipos de servidores proxy; los más conocidos son:

- Servicio Proxy o Proxy Web.- Su funcionamiento se basa en el del Proxy HTTP y HTTPs, este es el que intercepta la navegación de los clientes por páginas web, este puede proporcionar una caché compartida para las páginas web y contenidos descargados, actuando como un proxy-caché, con la cual libera de carga a los enlaces de acceso a internet.

- Proxy-caché.- Conserva el contenido solicitado por el usuario para acelerar su respuesta en futuras peticiones que contengan la misma información de la misma máquina u otras.