

Documento de entrega final

Entrega de documento final



7 de junio de 2016

Universidad VEracruzana

Ingeniería de Software

# Introducción

# Desarollo

## Análisis

## Requerimientos

## Diseño

## Estándar

|  |  |
| --- | --- |
| Propósito | Guiar el desarrollo de software con estándares |
| Cabecera de programa | La cabecera contendrá el nombre del proyecto, autores, fecha de la última edición y una descripción corta del objetivo de la clase |
| Formato de Cabecera | /\*\*  \* Clase tipo biblioteca, a través de ésta clase se pueden realizar préstamos, reservaciones,  \* búsqueda de ítems y sus correspondientes validaciones.  \*  \* @author Luis Fernando Gomez Alejandre  \* @author Francisco Gerardo Mares Solano  \* @since 06/06/2016  \*/ |
| Tabla de contenido | /\*\*  \* **Descripcion del método**  \*  \* @param parametro1  \* @param parametro2  \* @param parametro3  \* @param parametroN…  \* @return Variable de retorno  \* @throws Si lanza alguna excepción  \*/ |
| Tabla de contenido  Ejemplo | /\*\*  \* **Busca cualquier ítem que esté registrado en la base de datos con el**  \* **identificador pasado.**  \*  \* @param identificador se realiza una validación de 15 caracteres y una  \* letra inicial 'I' mayúscula o minúscula.  \* @return Si el parámetro fue pasado de manera erronea o no existe en  \* la base de datos, regresa una lista vacia.  \* Nunca regresará una lista nula y siempre será un ArrayList  \* @throws SQLException Si no existe conexión con la base de datos o algo  \* sale mal durante la conexión con la misma,  \* se lanzará una SQLException con el correspondiente mensaje de error.  \*/ |
| Identificadores | **Variables:**  Se ocupara Lower Camel Case, dónde la primera letra del identificador será minúscula, a partir de ahí, las palabras se escribirán seguidas y la primera letra de cada palabra será mayúscula. En cada caso, serán nombre distintivos y se evitarán caracteres especiales excepto para bucles for o while, dónde podrán ser iterador i,j,k… en ese orden.  **Constantes:**  Se ocupará una escritura totalmente en mayúsculas, dónde la separación entre palabras será constituido por un guion bajo ( \_ ) y de igual manera, serán nombres descriptivos y palabras completas siempre declaradas como *public static final*.  **Métodos:**  Se ocupara Lower Camel Case, dónde la primera letra del identificador será minúscula, a partir de ahí, las palabras se escribirán seguidas y la primera letra de cada palabra será mayúscula, Si este supera los 7 argumentos se debera pasar por un objeto que haga referencia, a los atributos de ese objeto |
| Ejemplos de identificadores | private ArrayList<String> telefonos;  private ArrayList<String> redesSocial;  private ArrayList<String> correosElectronicos;  public static final int RESULTADO\_DE\_SALIDA = 0; |
| Comentarios | Los comentarios de una sola línea serán precedidos por dos diagonales (//) y no ocuparán más de un renglón. Su objetivo será ayudar a entender el motivo de un código de programación.  Los comentarios de mayor extensión serán escritos con diagonales y asteriscos para su delimitación (/\*\* … \*/) y en cada renglón intermedio habrá un asterisco (\*) como marcador de inicio y tendrá un espacio de identación incluyendo el último renglón.  Para ambos casos su identación será la misma que tenga el método, sentencia o bucle al que preceda. Siempre deben agregar información de utilidad y que extienden la información común. En el caso de *edit fold* son permitidos dos renglones con doble diagonal (//). |
| Buen comentario | //Busca contacto(s) con las funciones lambda contactosEncontrados=this.listaContactos  .stream()  .filter(contacto->contacto.getNombreContacto()  .equals(nombre))  .collect(Collectors.toList()); |
| Mal comentario | // Variable para Contador  // Es para contar números.  // Es privado para que solo la clase acceda  // Entero por que no necesito flotantes  Private int contadorNumeros; |
| Secciones principales | /\*\*  \* **Descripción**  \*  \* @author Nombre1  \* @author Nombre2  \* @since Fecha  \*/ |
| Ejemplo | /\*\*  \* **Clase tipo biblioteca, a través de ésta clase se pueden realizar préstamos,**  \* **reservaciones, búsqueda de ítems y sus correspondientes validaciones.**  \*  \* @author Luis Fernando Gomez Alejandre  \* @author Francisco Gerardo Mares Solano  \* @since 06/06/2016  \*/ |
| Espacios en blanco | * Dentro de las clases, después de la declaración de variables y constantes, se dejará un espacio en blanco para seguir con los métodos. * Entre cada método de la clase debe quedar un espacio en blanco. * Los paquetes importados son consecutivos y en el primer renglón, y después de ellos, debe haber un espacio en blanco. * Los *import* deben ser consecutivos y después de ellos debe ir un espacio en blanco, van después de la declaración del paquete. * Los comentarios nunca tendrán un espacio en blanco hacia abajo. * Cada operador lógico debera de ir separado ente espacios (“ = ”, “ + ”). * Todos los parámetros llevarán un espacio después de la coma para agregar el siguiente. |
| Identación | * 4 espacios para la identación, cada anidación cumple la misma norma. * Las llaves inician junto al nombre de la función y terminan en un renglón extra al nivel del nombre de la función o método. * Solo los comentarios con múltiples comentarios pueden tener un espacio como identación. * En los comentarios despues de la descripción debera tener un espacio en blanco y despues seran las declaraciones de parametros y valores de retorno |
| Ejemplo de Identación | public class Contacto{  private int identificador;  private ArrayList<String> correosElectronicos;  private String paginaWeb;  public String getNombreContacto() {  return nombreContacto;  }  } |
| Capitalización | * Variables y métodos: Ocuparán Lower Camel Case. * Constantes: Completamente en mayúsculas * Clases: Ocuparán Upper Camel Case. * Paquetes: Siempre en minúsculas. |
| Ejemplo de capitalización | package dataaccess;  public class ItemDAOImpl implements ItemDAO{  private Statement consulta;  private ResultSet resultados;  } |

## Modulo Préstamo

## Modulo Reservación

# Conclusiones

# Referencias